





A Novel Framework for LiDAR Data Classification Using PCA and Ensemble Learning on Mobile Robots

Moteb Alghamdi^{1*}, Muhannad Almohaimeed²

¹ Department of Computer Engineering, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia

² Department of Information Systems, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia

Corresponding Author Email: mgamdi@taibahu.edu.sa

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.590213>

ABSTRACT

Received: 1 October 2025

Revised: 4 December 2025

Accepted: 15 December 2025

Available online: 28 February 2026

Keywords:

LiDAR, machine learning, ensemble methods, dimensionality reduction, indoor localization, mobile robots

Mobile robot navigation relies on sensing methods to enhance its localization capabilities. One crucial component in mobile robots' autonomous indoor localization systems is the LiDAR sensor. LiDAR sensors enable the mobile robots to have efficient navigation functionality and precise environmental understanding. This paper proposes a novel framework that integrates ensemble learning methods with principal component analysis (PCA) for dimensionality reduction to enhance the mobile robot localization accuracy rate and reduce the runtime overhead of processing the LiDAR data classification. The proposed model achieved a classification accuracy of 99.20% and a runtime reduction from 84.72 seconds to 6.48 seconds. These numbers demonstrate the superiority of the proposed model over existing methods. The framework addresses the LiDAR data curse of dimensionality, enhancing scalability and real-time applicability for resource-constrained systems. We evaluated the framework against state-of-the-art methods, highlighting its versatility and robustness. Additionally, it demonstrates its versatility as a practical solution for various indoor applications, including activity monitoring, navigation, and obstacle detection.

1. INTRODUCTION

The emergence of autonomous indoor navigation for robot operations has enhanced its capabilities in navigating areas such as hospitals, factories, smart buildings, and complex environments [1]. The essence of the robot's autonomy is the ability to classify its environment, such as corridors, doorways, rooms, and halls, based on sensory data [2]. For such tasks, LiDAR sensor is utilized (light detection and ranging) to provide the robot with a spatial, 3D/2D illustration-rich dataset of the target environment [3]. However, the complex and large datasets produced by LiDAR pose significant challenges, particularly when scaling up the target application [4]. LiDAR sensors generate a significant amount of data each frame; for instance, 2D LiDAR scans, which cover a 360° space, can yield up to 360 data points per frame [2]. On the other hand, 3D LiDAR can generate even more data, reaching thousands of points per frame [4]. Therefore, in applications that use LiDAR in large environments with complex details, the size of the dataset can exponentially grow as the number of features and objects increases [3].

Mapping large indoor environments, such as hospitals, can result in an abundance of features in datasets [5]. This behavior can exacerbate the issue of dimensionality, increase the computational costs associated with data processing, and complicate real-time classification [6]. The integration of

machine learning with LiDAR technology has demonstrated its effectiveness in surmounting the challenges presented by high-dimensional LiDAR datasets and presents promising opportunities for improving localization capabilities [7]. Machine learning methods can powerfully process the massive amounts of data generated by LiDAR sensors, allowing for precise and real-time classification of different environments [8]. This ability is essential for applications such as indoor navigation, where accurate detection of elements is critical [9].

By merging LiDAR's rich datasets with various machine learning methods, robots can attain superior performance in complicated tasks, promoting scalability, decreased computational costs, and improved adaptability through different environments [10, 11]. We propose a novel approach to address these challenges in LiDAR technology by leveraging principal component analysis PCA in conjunction with ensemble machine learning methods. PCA greatly reduces the number of dimensions in LiDAR data by taking out the most important features and getting rid of unnecessary data. This process makes the dataset smaller and more representative. This approach effectively addresses the issue of dimensionality, leading to a significant reduction in runtime. On the other hand, the ensemble methods maintain and enhance the classification accuracy.

The main contributions of this paper are summarised as follows:

1. Dimensionality Reduction for Scalability:

We present a PCA-based preprocessing pipeline that significantly reduces the dimensionality of LiDAR data while preserving essential structural features.

2. Enhanced Classification Accuracy:

The proposed model delivered a high classification accuracy of 99.20% across various indoor environments by integrating PCA with ensemble machine learning methods.

3. Real-Time Efficiency:

The proposed approach offers the reduction of computational cost and a real-time deployment solution on resource-constrained platforms.

This work's significance extends beyond individual applications to diverse tasks such as robotic surgery and autonomous driving that require an efficient and scalable real-time processing solution. The proposed solution can manage a vast LiDAR dataset with rapid and accurate classification, which can pave the way for real-time, efficient, and responsive robotic systems.

We organise the paper as follows: Section 2 discusses the related work on LiDAR-based classification accuracy and runtime overhead management. Section 3 illustrates the proposed methodology we implement. Section 4 details the experimental results with a focus on the accuracy improvements and computational reduction. Section 5 delves into the implications of the results for real-world applications. Section 6 brings the work to a close by exploring potential avenues for future research.

2. RELATED WORK

2.1 Traditional approaches

Due to its ability to provide a high-resolution spatial data representation of the environment in real time, LiDAR technology is popular for localization solutions in mobile robot applications. Nevertheless, LiDAR produces a large amount of data, which makes it a difficult computational problem to handle its accuracy. To solve this problem, traditionally, authors employed approaches such as Kalman filtering and Simultaneous Localization and Mapping (SLAM) to process the LiDAR data for localization tasks. A recent study suggested using an Extended Kalman Filter-based LiDAR-inertial odometry method to make tracking paths better when using a LiDAR sensor along with an IMU [12]. They were able to get a very accurate path by using a local point cloud map of the LiDAR data and the Kalman filter to combine the LiDAR data using a recursive Bayesian framework. Another study shows how the Adaptive Monte Carlo Localization (AMCL) algorithm was used with 3D LiDAR and other sensors to help the mobile robot be more stable and accurate in its location [13]. They proved that by reducing the maximum localization error for a real-world experiment from 9.4 cm to 6.1 cm and the angle error from 9.18 to 6.09 degrees, they helped achieve 34% better linear accuracy. A similar approach employed weighted contribution sampling approach to improve single-line LiDAR localization accuracy and robustness [14]. They have improved the localization precision and minimized the drift angle compared to the AMCL algorithm. Their proposal approach is an efficient method for real-world applications such as service robots and warehouse robots. A comprehensive study of the different SLAM techniques for processing LiDAR data was

presented in previous work [15]. They showed the various techniques, such as Iterative Closest Point (ICP), Normal Distributions Transform (NDT) and feature-based methods like LOAM (LiDAR Odometry and Mapping). This work demonstrates the application of various SLAM approaches to improve LiDAR processing. Another study presents a hybrid approach that enhances real-time robot localization using LiDAR data [16]. The authors of this paper use Monte Carlo localization, A-LeGO-LOAM SLAM, and NDT scan matching techniques together to make their contribution and show that it works on a Velodyne HDL-32E LiDAR and an NVIDIA Jetson Xavier platform. Their method provides high accuracy, with initial pose errors of up to 3 meters and an angular error (APE_{rot}) of 10 degrees. Although these methods show promising solutions to handle the LiDAR's huge data processing, they often struggle to provide high efficiency and robustness in real-time applications, especially in dynamic environments.

2.2 Machine learning-based approaches

With the advancement of machine learning (ML) algorithms, the opportunity for efficient LiDAR data processing approaches has increased. The utility of LiDAR data has improved significantly. Studies have used various ML algorithms, such as convolutional neural networks (CNN), support vector machines (SVM), deep learning (DL), and support vector machines (SVM). For example, one study proposed a three-layer CNN architecture that converts 3D LiDAR data point clouds into 2D to estimate a vehicle's position and orientation [17]. The CNN model made the LiDAR processing more accurate, with a cross-track positioning error of less than 30 mm. It also had a higher accuracy rate and was more stable than EKF-based feature extraction methods. Another example presented a lightweight CNN model for object detection in autonomous driving based on a LiDAR dataset [18]. The model achieved high accuracy with a significantly smaller model size (8MB) that reduced the computational cost (57 FPS) and power consumption, which makes it an efficient design for embedded devices. In a related study, researchers utilized convolutional neural networks (CNNs), PointNet-based feature extraction, and recurrent neural network (RNN) methods to propose a centimeter-level accuracy LiDAR processing model [19]. Their method achieved 92.42% accuracy within a 10 cm error, which is compared to classical LiDAR localization methods. A very useful, map-free LiDAR localization system was proposed in a recent work for quick deployment in robotics applications [20]. It used feature mixing with MLP-Mixer, frozen feature extractor, and contrastive learning. This system achieved a 3.80 m mean position error in terms of accuracy and 5× faster processing time compared to the state-of-the-art methods. Although these proposed solutions showed high accuracy figures, they required high computational resources, which limited their robustness in resource-constrained systems such as the mobile robot.

Multiple studies have suggested various approaches to reduce the computational overhead problem. Many of these studies have proposed reducing data dimensionality to decrease the need for LiDAR data processing power. PCA is a widely used approach to reduce the computational overhead problem. The PCA approach depends on extracting the most significant features of the LiDAR data while discarding the redundant information. This method is powerful in improving

the efficiency of processing the LiDAR dataset. For example, the authors in this literature [21] used the PCA method to lower the LiDAR dataset complexity. They performed noise reduction in a 2D-transformed space, which achieved a 50% complexity reduction in the LiDAR point cloud while maintaining a high F-score of 0.92, precision of 97.27%, and recall of 86.00%. In reference [22], another PCA-based study, they used K-Nearest Neighbours (KNN) segmentation along with the PCA method to get rid of noise in outdoor LiDAR point clouds very effectively, and they were able to achieve 98% noise elimination accuracy. Authors in this study [23] proposed a lightweight CNN model that employed spherical projection, a recurrent CRF layer, and firemodule-based feature extraction to achieve a high-accuracy and fast-runtime object segmentation method for autonomous driving. This method converts 3D LiDAR data to 2D grid representation to allow 2D CNN processing, which significantly reduces the runtime. The authors of the study [24] employed a novel deep learning architecture to classify mobile robot locations as doorways, rooms, and corridors. They used ordered 2D LiDAR scans, which significantly improved the classification accuracy rate of the doorway over previous methods, with 77.44% recall, 21.04% precision, a 34.64% F1 score, and 97.12% overall accuracy. In a similar approach, authors in this study [25] proposed a novel semantic LiDAR SLAM framework, which combines optimized particle swarm optimization (PSO), semantic segmentation, and Kalman filtering for dynamic environment segmentation. Their method generates high-quality static semantic maps and improves localization accuracy.

Focusing on indoor environments, a study incorporated a 2D LiDAR into an unmanned robot to detect movement within the space and then utilized convolutional LSTM neural networks to categorize the LiDAR data into groups based on the activities of the individuals [26]. This method achieved 99% accuracy in detecting lying human bodies. Other studies, used supervised machine learning to classify 2D LiDAR-generated data into corridors, rooms, and doors, achieving 97.21% accuracy while maintaining a low computational cost [27]. Other studies classified LiDAR data using machine learning methods for the robot's pathfinding and obstacle detection. For example, the authors [28] used 2D LiDAR to collect data suitable for classifying carried objects through a multimodal data fusion algorithm, achieving classification rates of 86% and 91% using various evolution metrics. In another study, the authors [29] used the YOLO-based detection framework and Robot Operating System to classify 2D LiDAR data into objects and poses for obstacle avoidance. Their solution has achieved an efficient computational cost with qualitative results. In other studies, authors have employed hybrid sensors alongside the LiDAR to achieve better results. For instance [30], the authors combined 3D LiDAR, a camera, and Wi-Fi to achieve a localization accuracy of 0.62 m. They also achieved 89.22% floor estimation accuracy and a mean squared error (MSE) of 1.24 m for 2D tracking trajectories. They have used a hybrid visual and wireless-enhanced algorithm to achieve such accuracy with a runtime of just 0.25 seconds. In a similar study [31], the authors used panoptic segmentation and descriptor networks on 3D LiDAR data to achieve a high localization score with a mean error of 1.23 meters. These examples showcase high accuracy improvements; however, they introduce scalability challenges for wider deployment. In other studies, authors combine the use of machine learning and LiDAR data for a

robot's navigation and path planning. For instance, one study used an artificial potential field algorithm and 3D LiDAR statistics to get a 17% increase in the success rate they were after when they were trying to sort data into categories like doors, windows, and stairs in indoor settings [32]. Balancing high LiDAR data classification accuracy and runtime expenses remains an issue. Some studies sacrifice run-time efficiency to provide a high accuracy rate [26, 27]. On the other hand, studies [16, 29] enhanced the computational cost by utilizing lightweight machine learning methods like YOLO-based detection, but this was done at the cost of robustness. Our work contributes to this line of work by addressing the balance between achieving a high accuracy rate and keeping the runtime cost low. We achieved this by employing dimensionality reduction techniques using PCA and combined it with ensemble learning methods. Our approach, however, optimizes LiDAR data processing for real-time classification while maintaining high accuracy. Unlike the other related work, we managed to provide a framework that offers a scalable and efficient solution.

3. METHODOLOGY

3.1 Methods and materials

The dataset used in this research is the LidarDataFrames dataset, which is publicly available on Kaggle [33]. The dataset is introduced as a wide collection of LiDAR data derived from a mobile robot equipped with an RPLIDAR-A1 Laser Range Scanner. The dataset contains signal scans from four different environments: rooms, doorways, corridors, and halls, with each environment providing 360 different LiDAR signals. 360 input features describe each instance in the dataset, indicating the diverse LiDAR signal measurements, while a target feature reveals the type of environment. The LidarDataFrames dataset offers a complete view of the complex nature of LiDAR data, making it a valuable resource for developing robust prediction solutions because it provides broad information on factors associated with LiDAR-based localization [27].

The dataset is well-structured with a near-balanced class distribution, confirming that each environmental type is represented by a similar number of instances and minimizing the risk of classification bias. According to the dataset's paper [27], it has already undergone essential data cleaning and preparation, involving the elimination of missing or noise values. Hence, no further data preprocessing phases were necessary before starting model development. This cleaned and balanced dataset serves as the foundation for feature extraction and model training proposed in this paper.

3.2 Experimental design and workflow

In this study, we propose a predictive model for detecting the type of environment based on LiDAR data. We deployed three different experimental setups to evaluate the performance of several ensemble classifiers and dimensionality reduction techniques. Each experiment employed different predictive model strategies to evaluate the impact of these methods on the classification accuracy. The overall experimental design and workflow of the proposed LiDAR classification system is illustrated in Figure 1, which summarizes the main processing stages used in the proposed method.

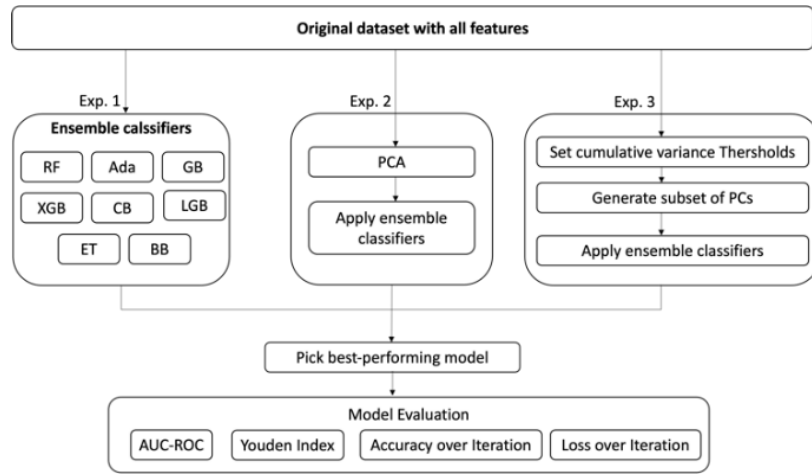


Figure 1. The detailed experimental design and workflow of the proposed methodology

3.2.1 Experiment 1: Baseline ensemble classifier deployment

In the first experiment, we applied eight different ensemble classifiers to the dataset in its original form (360 features) without using any dimensionality reduction techniques. These classifiers include:

- **Random Forest:** Is a powerful bagging-based ensemble algorithm that builds multiple decision trees and aggregates their predictions to enhance accuracy and decrease overfitting. People commonly use it due to its ability to handle high-dimensional data and noise.

- **AdaBoost:** A boosting algorithm that integrates weak classifiers in a sequential manner, iteratively adjusting the weights of misclassified cases.

- **Gradient Boosting:** Another boosting method that constructs decision trees sequentially, where each decision tree aims to resolve the errors of the previous decision tree. It reduces an indicated loss function through gradient descent, making it greatly efficient for classification tasks.

- **Extreme Gradient Boosting (XGBoost):** A highly efficient and scalable extension of the Gradient Boosting technique, implemented for optimized speed and performance. It presents advanced techniques such as regularization and tree pruning, which help avoid overfitting and enhance model accuracy.

- **Categorical Boosting (CatBoost):** A gradient boosting method specially implemented for categorical data, applying effective handling of categorical features without involving broad preprocessing steps.

- **Extra Trees:** An ensemble learning algorithm that builds multiple decision trees, randomly selecting both the data objects and splits for each tree. Extra Trees presents more randomness by selecting split thresholds randomly, which can enhance computational efficiency and avoid overfitting in some situations.

- **Light Gradient-Boosting Machine (LightGBM):** A gradient boosting algorithm implemented for speed and efficiency, especially on big datasets. It attains its superior performance by using methods such as histogram-based learning to decrease computational costs and improve scalability.

- **The Balanced Bagging Classifier:** Is an ensemble learning technique that utilizes bagging and balanced resampling to tackle class imbalance issues in datasets, ensuring equal representation of each class in the training set. This technique trains multiple base estimators on the balanced training set and combines their predictions, enhancing the

performance of minority classes and decreasing bias to majority classes.

In this experiment, each of these techniques was trained and evaluated on the original dataset to compare their performance in classifying the environment type based on the LiDAR data.

Hyperparameter tuning was performed to optimize each model’s performance. The training hyperparameters for the implemented classifiers are presented in Table 1. We relied completely on the default hyperparameters specified by each classifier’s official implementation. Our objective was to assess the effect of PCA-based dimensionality reduction across a diverse set of ensemble algorithms under fair and consistent conditions, rather than to perform extensive hyperparameter optimization. Using default settings ensures a widely accepted, reproducible, and unbiased baseline across all classifiers.

Table 1. Default hyperparameters used for each classifier

Classifier	Hyperparameter	Value
Random forest	Number of estimators	100
	Criterion	Gini
	Minimum sample split	2
	Minimum sample leaf	1
AdaBoost	Number of estimators	50
	Learning rate	0.1
	Maximum depth	3
XGBoost	Learning rate	0.3
	Number of estimators	100
	Maximum depth	6
CatBoost	Iterations	1000
	Learning rate	0.03
	Depth	6
Extra trees	Number of estimators	100
	Criterion	Gini
	Minimum sample split	2
LightGBM	Number of leaves	31
	Learning rate	0.1
	Number of estimators	100
Balanced bagging	Number of estimators	10
	Sampling strategy	Auto
	Replacement	False

3.2.2 Experiment 2: Deployment of ensemble classifier and PCA – full component retention

The second experiment presented dimensionality reduction through PCA. PCA was preferred because LiDAR signals present high dimensionality and strong correlations across

neighboring beams, which lead to redundancy and increase computational cost. PCA transforms the original feature space into a set of orthogonal components sorted by variance, allowing the model to retain the most informative structure of the LiDAR data while removing noise-dominated directions.

PCA was applied to the dataset to reduce input variable redundancy, retaining the most significant variance in the dataset. PCA was implemented on the 360 variables, and all principal components were preserved during the transformation. After performing PCA, the obtained principal component matrix was used to train the same set of ensemble classifiers used in the first experiment. This step was added to see what effect orthogonal feature space transformation had on the performance of the classifier and to see if using PCA would make it work better or be more accurate.

3.2.3 Experiment 3: Deployment of ensemble classifier and PCA (reduced component retention)

The third experiment extended the second experiment by reducing the number of principal components used for model training. Several component sets were tested to find the best number of dimensions that would balance the need for fast computing with keeping all the original data's variance. We trained and tested the same set of ensemble classifiers on the reduced PCA-transformed datasets. This experiment aimed to explore how the decrease in the number of principal components affects both classification accuracy and computational efficiency. Here's how we achieved the integration of PCA and ensemble learning: PCA acted as the feature extraction phase, while the ensemble classifiers were implemented as the predictive models on the PCA-transformed feature space. This integration is intended to enhance classifier generalization by decreasing the risk of overfitting associated with high-dimensional datasets.

3.3 Data splitting and evaluation

For all experiments, the dataset was divided into a training set (70%) and a test set (30%). The training set consisted of 287 objects, while the test set contained 124 objects. Reproducibility was guaranteed by choosing random seeds during data splitting and model training.

The proposed model performance was tested and evaluated using several standard classification metrics, including:

- **Accuracy:** The percentage of correct predictions made out of all instances. The accuracy formula is illustrated below:

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

where, TP , TN , FN , and FP refer to the number of true positive cases, false positive cases, false negative cases, and false positive cases, respectively.

- **Recall:** The proportion of true positive instances identified by the model among all actual positive instances. The recall formula is presented below:

$$Recall = \frac{TP}{TP + FN}$$

- **Precision:** The percentage of genuine positives detected by the model out of all anticipated positives. The precision formula is shown below:

$$Precision = \frac{TP}{TP + FP}$$

- **F1 Score:** is the harmonic mean of precision and recall, which provides a balanced measure of model performance. Below, we present the formula for the F1 score.

$$F1 - Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

We selected these metrics to provide a comprehensive evaluation of the model's performance, taking into account its ability to accurately classify positive instances from each environment while maintaining a balance between false positives and false negatives.

After all the predictive models were tested, the one that did the best (based on accuracy and performance across all metrics) was put through more tests to find out what it learnt and what its analytical features were. To establish the optimal cut-off values, the Youden combined value of sensitivity and specificity. We also used a confusion matrix to clearly illustrate prediction performance and highlight potential misclassification areas. The performance metrics used for model evaluation included sensitivity, specificity, and AUC-ROC. In addition, the accuracy progression was tracked over training iterations, providing valuable insights into model convergence and overall stability. The loss function was also monitored throughout the iterations to evaluate the learning dynamics of the model, assess convergence behaviours, and detect signs of overfitting.

3.4 Implementation and computational setup

We conducted the entire study using a Jupyter Notebook (Anaconda3), running Python (version 3.9) on a MacBook Big Sur (version 11.7.10), equipped with 16GB of RAM and an Intel® Core i9 CPU @ 2.3 GHz. The following Python packages were utilised in this study:

- `imbalanced-learn` (version 0.12.3): Used for implementing the balanced bidding classifier.
- `xgboost` (version 2.1.2): implemented for the XGBoost classifier.
- `lightgbm` (version 4.5.0): Used to implement the LightGBM classifier.
- `catboost` (version 1.2.7): Utilized for the CatBoost classifier.
- `scikit-learn` (version 1.4.2): Used for implementing other ensemble classifiers, PCA, evaluation metrics, and data splitting.

4. RESULTS

This work uses an open-source LidarDataFrames dataset, which collects LiDAR data from a mobile robot, to classify 411 instances into four different environments. Of these, 109 (26.5%) were for room, 103 (25.1%) for hall, 100 (24.3%) for corridor, and 99 (24.1%) for doorway. The purpose of this work is to analyze the data collected based on the correct signals used to complete object identification tasks. We randomly stratified the dataset into a training set and a test set, following a 7: 3 distribution, to perform the experiments. In this study, we assessed the performance of 48 predictive models. These models consist of eight ensemble methods and

PCA. As shown in Figure 2, we compare the results of these models using ensemble methods only, PCA with the classifiers, and reduced features generated from PCA along with the classifiers. We evaluated the predictive models using various metrics, such as accuracy, precision, recall, and F1-score, all expressed as percentages.

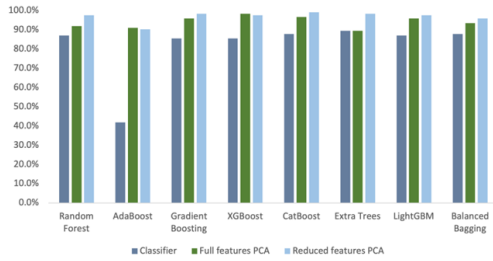


Figure 2. Evaluation of the classification accuracy for all experimental models

4.1 Experiments using ensemble methods

Table 2 presents the accuracy and several performance metrics for the eight ensemble classifiers used before applying PCA results reveals that different classifiers perform differently. In terms of accuracy, Extra Trees achieved the best classification accuracy of 89.5%, whereas AdaBoost showed the poorest accuracy score (41.9%). The rest of the classifiers achieved similar results, ranging between 85.5 and 87.9%.

Table 2. Performance of using ensemble classification methods

ML Model	Accuracy	Precision	Recall	F1-Score
Random forest	87.1	89.7	87.1	86.5
AdaBoost	41.9	77.5	41.9	41.5
Gradient boosting	85.5	86.1	85.5	84.8
XGBoost	85.5	85.8	85.5	84.9
CatBoost	87.9	89.2	87.9	87.3
Extra trees	89.5	90.6	89.5	89.1
LightGBM	87.1	87.3	87.1	86.6
Balanced bagging	87.9	88.6	87.9	87.4

4.2 Experiments using PCA with ensemble methods – full component retention

First, we performed PCA on the dataset, retaining all principal components to guarantee that the full variance and information from the original dataset were used. We then deployed the PCA matrix using ensemble classifiers, generating a total of eight models. Table 3 presents the performance for these models. Significant observations revealed that the majority of classifiers achieved remarkable enhancements. The best performance was observed using PCA along with the XGBoost classifier, achieving 98.4%, 98.4%, 98.4%, and 98.4% for accuracy, precision, recall, and F1-score, respectively. In addition, the results showed that the accuracy of the AdaBoost model enhanced to 91.1% after deploying PCA, indicating an improvement of +49.2% compared to its performance without applying PCA.

Table 3. Performance of using ensemble classifier and PCA (full component retention)

ML Model	Accuracy	Precision	Recall	F1-Score
Random forest	91.9	92.1	91.9	91.8
AdaBoost	91.1	91.1	91.1	91.1
Gradient boosting	96.0	96.0	96.0	96.0
XGBoost	98.4	98.4	98.4	98.4
CatBoost	96.8	96.8	96.8	96.8
Extra trees	89.5	90.3	89.5	89.4
LightGBM	96.0	96.0	96.0	95.9
Balanced bagging	93.5	93.6	93.5	93.6

4.3 Experiments using PCA and ensemble methods (reduced component retention)

Here, we reduced the number of features using PCA to further reduce data dimensionality and improve computational efficiency. Figure 3 shows the cumulative variance explained by all principal components, as well as the data projected onto the first two components for visualization purposes. To determine the optimal number of principal components, we experimented with different thresholds for variance retention. As a result, we tested multiple subsets of features generated by PCA corresponding to different cumulative variance levels (90%, 95%, 99% and 99.9%). The evaluated subsets included 17, 28, 75, and 156 principal components, respectively. Referring to the experiments’ outcomes across eight distinct ensemble algorithms, as visualized in Figure 4, the best subset of features was identified as having 28 components, which provided the optimal balance between dimension reduction and classification performance, reaching an average accuracy of 96.9%. Specifically, the cumulative variance explained by the first 28 principal components was 95%, meaning that these components retained the majority of the original data’s variance. The PCA transformation represented each record in a low-dimensional vector space, which contributed to improving the classification process and reducing the runtime of the models. Figure 5 shows the cumulative variance explained by 28 principal components, as well as the data projected onto the first two components for visualization purposes.

We then deployed the reduced dataset (28 features) generated from PCA using the same set of classifiers. Although the accuracy of AdaBoost with PCA using a reduced set of features performed slightly lower (90.3%) than AdaBoost with PCA using a full set of features, indicating an increased number of misclassification cases, other classifiers performed well with a reduced set of features generated from PCA. Table 4 presents the performance results for these models. The results indicated that the best-performing classifier was CatBoost, along with PCA using a reduced set of features, achieving 99.2%, 99.2%, 99.2%, and 99.2% for accuracy, precision, recall, and F1-score, respectively. This model had a high precision and F1-score, which meant it could correctly classify positive cases and find a balance between precision and recall. Extra trees and gradient boosting associated with PCA based on reduced sets of features had similar performance, with an accuracy of 98.4%.

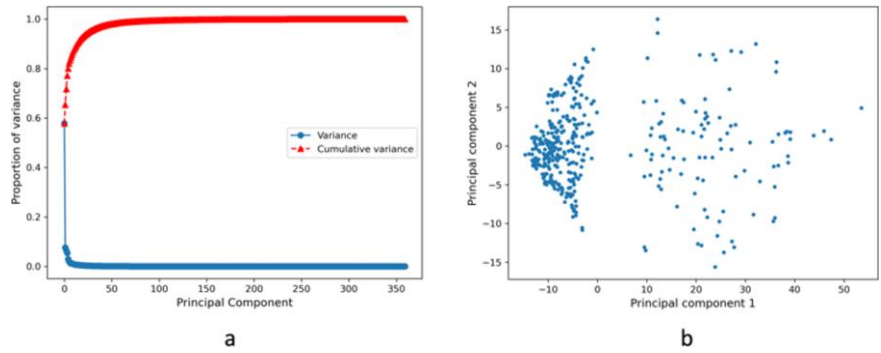


Figure 3. A low-dimensional representation of LiDAR records using PCA. (a) The cumulative variance of all principal components obtained. (b) The data projected onto the first two principal components

Using the CatBoost classifier with the smaller set of features generated from PCA gave the best accuracy, precision, recall, and F1-score of all the models that were tested. We conducted several performance metrics to further evaluate the robustness and effectiveness of the best-performing predictive model. The confusion matrix, as shown in Figure 6, further highlights the model's predictive ability. The confusion matrix provided a complete view of the classification outcomes and identified likely areas of misclassification.

We further validated the best-performing model using various assessment methods, in addition to standard performance metrics. We assessed the performance of the model using the area under the ROC curve (AUC), as shown in Figure 7, to achieve their reliability. This model achieved an average AUC of 1.0, indicating remarkable predictive performance. The model offers the best classification accuracy for the room, doorway, and hall environments at 100%. For the corridor environment, the model achieved a great identification rate with an accuracy of 99%.

We further validated the best-performing model using

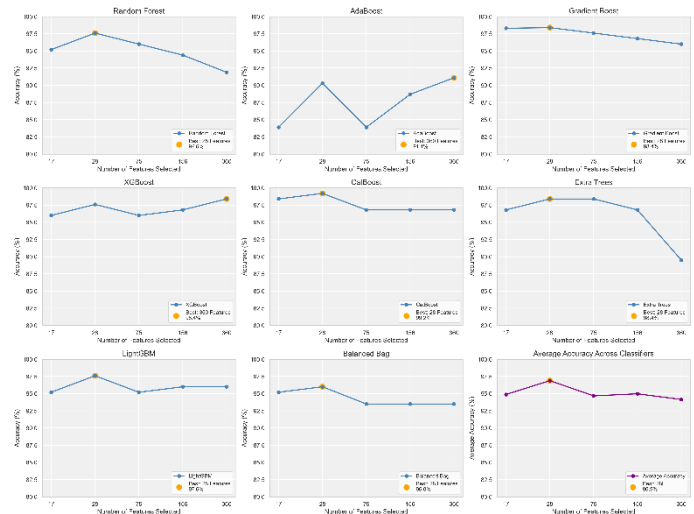


Figure 4. Classifier accuracy across different numbers of features selected using PCA. The figure illustrates the classification accuracy of eight different ensemble algorithms. Random Forest, AdaBoost, Gradient Boosting, XGBoost, CatBoost, Extra Trees, LightGBM, and Balanced Bagging evaluated on feature subsets comprising 17, 28, 75, 156 and 360 principal components

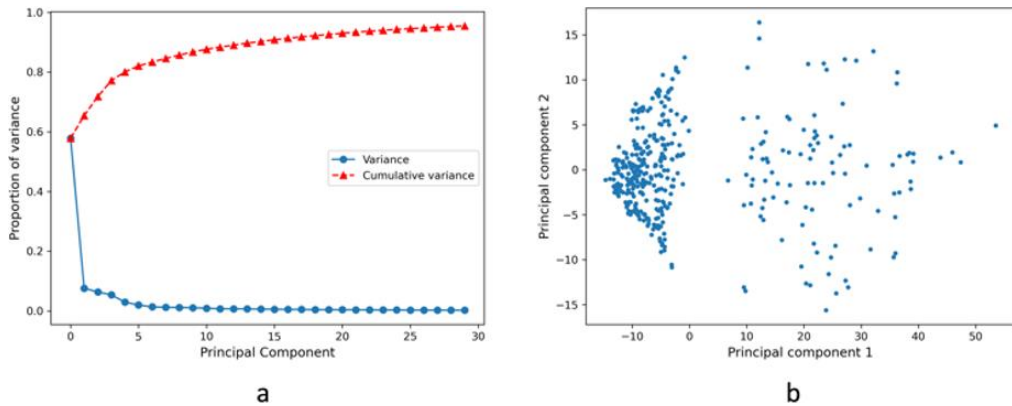


Figure 5. A low-dimensional representation of LiDAR records using PCA. (a) The cumulative variance of the first 30 principal components obtained. (b) The data projected onto the first two principal components

Table 4. Performance of the reduced-set of features using PCA along with ensemble classification methods

ML Model	Accuracy	Precision	Recall	F1-Score
Random forest	97.6	97.8	97.6	97.6
AdaBoost	90.3	90.3	90.3	90.2
Gradient boosting	98.4	98.4	98.4	98.4
XGBoost	97.6	97.6	97.6	97.6
CatBoost	99.2	99.2	99.2	99.2
Extra trees	98.4	98.5	98.4	98.4
LightGBM	97.6	97.7	97.6	97.6
Balanced bagging	96.0	96.2	96.0	96.0

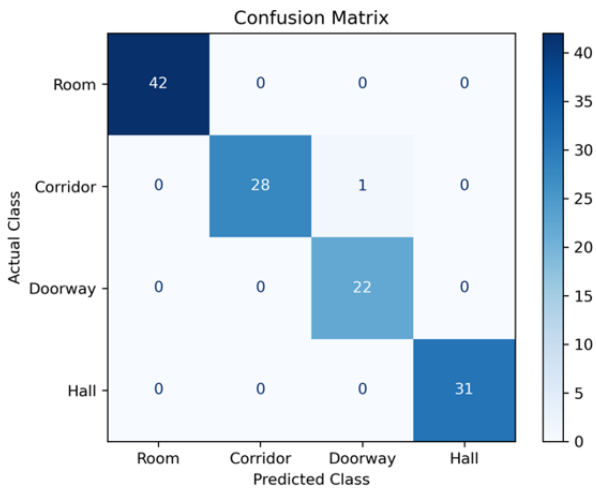


Figure 6. The confusion matrix of the best-performing model in the test set

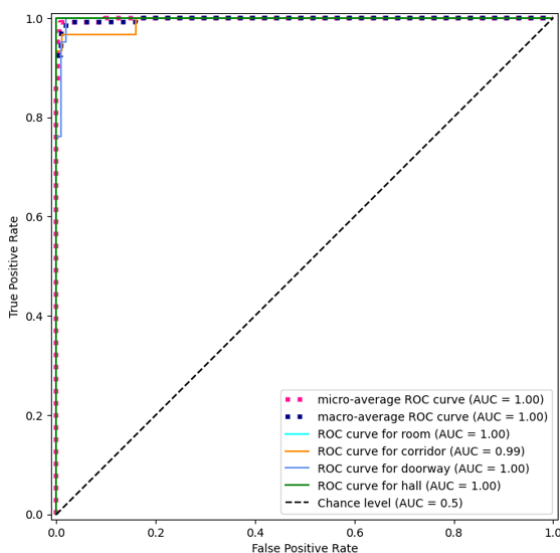


Figure 7. AUC plot for the best-performing predictive model

Moreover, we examined the learning dynamics of the model through accuracy and loss over iterations, providing valuable insights into its training behaviour and generalisation capability. As presented in Figure 8, the model's training accuracy gradually increased, while its testing accuracy similarly improved, indicating strong generalisation without overfitting. Similarly, the loss curves depicted in Figure 9 showed a continuous decrease in both training and testing

losses, further validating effective learning in the model. The learning curve also created 100 iterations with different shuffles of the training data, which supported these results and showed that the model could improve its ability to predict and stay stable across different data splits.

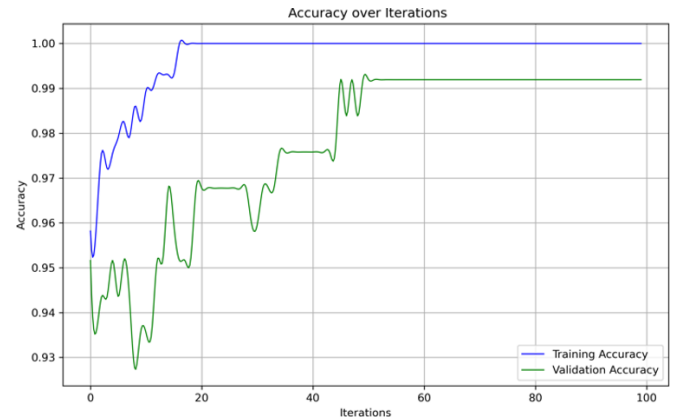


Figure 8. Accuracy over iterations for the CatBoost classifier trained on the reduced feature set (28 principal components)

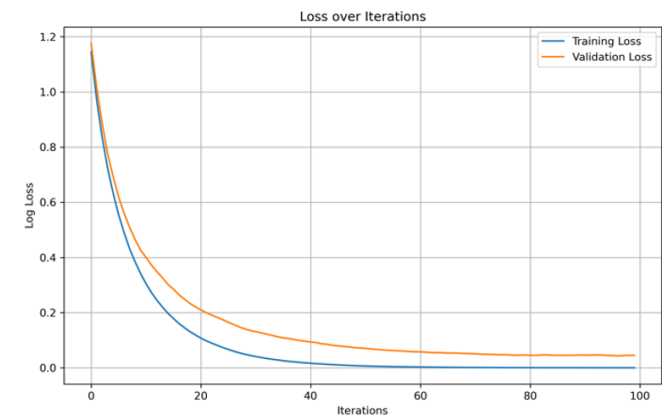


Figure 9. Loss over iterations for the CatBoost classifier trained on the reduced feature set (28 principal components)

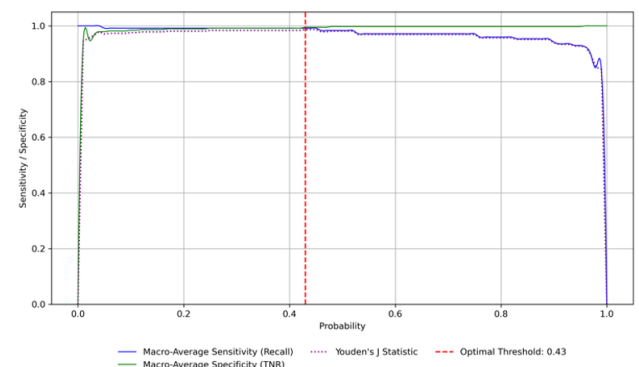


Figure 10. Sensitivity and specificity versus cut-off probability plot of the CatBoost classifier trained on the reduced feature set (28 principal components)

As depicted in Figure 10, with the increasing probability threshold, there is a drop in sensitivity and an improvement in specificity. Applying the Youden index, an optimal threshold probability of 0.43 was determined for the best-performing model, yielding macro-average sensitivity and specificity of 99.2% and 99.5%, respectively. This procedure resulted in an excellent Youden's J statistic of 0.9866, demonstrating a

balanced and robust classification performance.

To sum up, the CatBoost classifier, when combined with PCA and a smaller set of 28 features, showed better prediction performance and strong generalisation abilities. This made it a good choice for using LiDAR data to locate mobile robots.

4.3 Computational costs of the proposed experiments

The goal of this work is to reduce processing time and improve method performance. Dimensionality reduction

methods use feature extraction to transform and simplify data. This method provides a comparative evaluation of run-time across various types of experiments; see Table 5. Meanwhile the execution times vary across various orders of magnitude, a logarithmic scale was performed to present the runtime comparison in Figure 11. This visualization emphasizes the efficiency enhancements expanded through dimensionality reduction, particularly when maintaining only 28 principal components.

Table 5. Comparative analysis of the run-time across multiple experiments

ML Model	Classifier (Seconds)	Full Features PCA (Seconds)	Reduced Features PCA (Seconds)	Time Reduction (%)
Random forest	0.39	0.39	0.21	46.2%
AdaBoost	0.69	0.72	0.13	81.2%
Gradient boosting	13.37	13.12	1.34	90.0%
XGBoost	1.42	0.87	0.41	71.1%
CatBoost	84.72	81.52	6.48	92.4%
Extra trees	0.16	0.16	0.12	25.0%
LightGBM	0.73	0.44	0.11	84.9%
Balanced bagging	0.56	0.42	0.1	82.1%

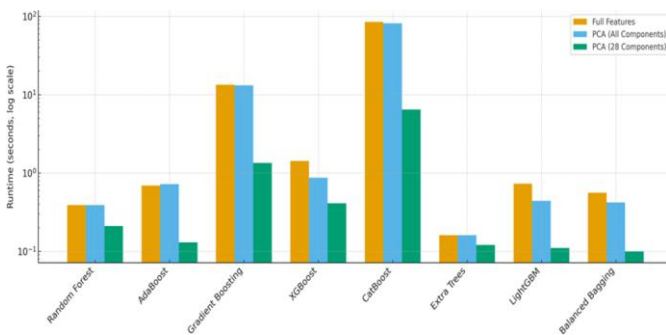


Figure 11. Runtime comparison of all ensemble classifiers across the three experimental configurations (full features, PCA with full components, and PCA with 28 components). A logarithmic scale is used to accommodate the wide range of runtime values

Our experimental results demonstrated a significant reduction in the computation cost in terms of time when using the reduced set of features generated by PCA. Moreover, the models accomplished the classification tasks within 0.1 and 6.48 seconds, including the time for applying PCA. Considering the importance of computation cost in a resource-constrained device such as a PC with ordinary specifications, the run-time of the CatBoost algorithm was drastically reduced from 84.72 to 6.48 seconds, which reduced the model's runtime by 92.4%. In other words, if we divided the 411 frames by the achieved run time, our model can process 63.43 frames per second (FPS) in such processing elements and would be faster in much more advanced processing machines. In addition, the resource consumption analysis confirmed that the experiment used only a small portion of the available CPU, peaking at 3.2%.

5. DISCUSSION

Reducing the classification computational overhead is a vital factor for supporting the real-time processing in the mobile robot's autonomous navigation using LiDAR sensors.

Furthermore, it plays a crucial role in improving the application's scalability. The capability of the classifier to classify LiDAR datasets quickly and accurately makes it an appropriate solution that can operate in dynamic indoor settings, where rapid decision-making is essential. These findings of this study demonstrate the significant benefits of combining ensemble learning and PCA for accurately sorting 2D LiDAR data into rooms, doorways, hallways, and other indoor environments and objects.

Advances in machine learning is not always about utilizing the most complicated methods or extensive feature engineering. Sometimes, simplifying model development to improve generalizability and usability represents an important innovation. Simplified models are easier for researchers to replicate and validate and are more realistic to deploy in practical, real-world environments. Our study validates this principle by attaining notable classification performance with an ensemble learning approach and dimensionality reduction via PCA, emphasizing simplicity and efficiency. The choice of methods in this study was essential in ensuring robust prediction performance. Ensemble algorithms, in general, have consistently showcased their reliability in enhancing prediction accuracy by integrating the strengths of multiple learners and mitigating individual model limitations [27]. This study highlights the effectiveness of this approach in the context of mobile robot localization using LiDAR data.

This work presents a promising contribution for scalable mobile robot indoor navigation tasks, as it provides significant accuracy with low runtime overhead and real-time applicability solutions. Specifically, integrating PCA-related features with the CatBoost classifier provided 99.2% accuracy, in addition to F1-scores, precision, and recall of 99.2% each. The performance results show that the PCA features can reduce the number of dimensions in the 2D LiDAR dataset while keeping its main features.

Previous research has highlighted the significance of AUC as an indicative metric for evaluating a model's ability to distinguish between positive and negative instances, ensuring reliable and effective classification performance [34]. The proposed solution achieved a perfect 1 score in AUC, ensuring the predictive model's robustness and reliability. In addition,

this score illustrates how specific and sensitive the model is in indoor environment classification. Most of the environment features scored a perfect classification rate under the AUC scope; however, the corridor class AUC is 0.99, which is slightly lower than the perfect score of 1.0. This minor reduction can be a result of overlapping structural features with other classes, like doorways, which impose a margin of classification ambiguity. Despite this, the corridor classification achieved an impressive score of 0.99 AUC, demonstrating the model's practicality for such practical applications. Furthermore, the optimal threshold probability of 0.43 guided by the Youden index highlights the balanced of sensitivity and specificity in the study. This guarantees not only the accurate classification of indoor environments but also minimizes false alarms, which is critical in real-time robotic applications to avoid navigation errors and unnecessary corrective actions. The achieved AUC and Youden Index demonstrated the model's significant predictive ability.

CatBoost steadily outperformed the other ensemble methods in this experiment, and this performance aligns with its design characteristics. Unlike conventional gradient boosting methods, CatBoost uses a well-ordered boosting scheme that reduces prediction shift and decreases overfitting when dealing with high-dimensional or partially sparse datasets. The LiDAR feature space in this study consists of correlated and redundant measurements, which CatBoost handles effectively through its symmetric tree structure and built-in regularization. Moreover, CatBoost is recognized for its robustness in handling noisy data and does not require extensive preprocessing or hyperparameter tuning to attain stable performance. These properties make it specifically appropriate for LiDAR-based classification tasks, where data distributions are complicated and feature importance varies across environments.

In addition, the use of PCA with the CatBoost classifier to reduce the 2D LiDAR dataset features has significantly reduced CatBoost's runtime by 92.4%, from 84.72 seconds to 6.48 seconds. This improvement is largely due to PCA reducing the number of features, which decreases the computational complexity and speeds up both training and inference. While the runtime reduction is impressive, it is vital

to note that this work was attained using standard computational resources (an Intel® Core™ i9 processor with 16GB RAM), making the approach feasible for resource-constrained mobile robots. However, the trade-off of dimensionality reduction is the potential loss of fine-grained data, which could impact performance in more complex environments. In this study, careful selection of principal components helped maintain high accuracy while improving efficiency.

Moreover, it is worth noting that AdaBoost showed the lowest classification accuracy among the ensemble classifiers evaluated in this study, with an accuracy of 41.9%, when used without applying dimensionality reduction. After applying PCA with 28 principal components, the performance of AdaBoost improved to 91.1%. However, even with this enhancement, AdaBoost's performance remained significantly lower compared to other models. This performance gap may be attributed to AdaBoost's sensitivity to noisy data and its sequential learning nature, which can exacerbate the impact of misclassified instances. These findings highlight the challenges of using AdaBoost in complex, high-dimensional datasets like LiDAR signals, where more robust models are needed to ensure reliable and generalizable performance.

Table 6 presents a comparison of the most pertinent studies that align with our work. We compared our work to thirteen studies using six criteria: the type of Lidar used, the number of objects classified in each study, the accuracy rate, the runtime, the machine learning algorithms used, and the processing element that ran the proposed methods. This helped us figure out where our work stands in comparison to these other works. The comparative analysis illustrates that there are performance trade-offs between computational efficiency and the LiDAR-based ML classification methods. Our work has a high classification accuracy (99.2%) using the suggested methods, which shows that the PCA with the ensemble learning hybrid approach works well. Leveraging the strengths of multiple machine learning classifiers (SVM, Random Forest, CatBoost, Decision Tree, LightGBM, and Naïve Bayes), our model maximises classification reliability compared to other approaches, such as deep CNNs or transformer-based architectures.

Table 6. Comparison of key studies relevant to the evaluation case, highlighting LiDAR types, target objects, achieved accuracy, runtime performance, applied machine learning algorithms, and the processing elements used

Study	LiDAR Type	Classified Objects	Accuracy	Runtime	ML Techniques	Processing Platform
[30]	2D LiDAR	Rooms, Floors	93.6%	Real-time (~50ms)	CNN + LSTM	NVIDIA Jetson TX2
[24]	2D LiDAR	Room, Corridor, Doorway	96.4%	25 FPS	PointNet, PointNet++	NVIDIA GTX 1080
[35]	3D LiDAR	Objects in road scenes	94.8%	100ms/frame	DeepLabV3+	Intel Xeon + NVIDIA V100
[28]	2D LiDAR	Obstacles	90.3%	N/A	KNN-based Clustering	Embedded ARM Processor
[36]	3D LiDAR	Vehicles, Pedestrians, Roads	97.1%	32 FPS	3D CNN	NVIDIA Titan X
[6]	3D LiDAR	Rooms, Hallways	95.2%	120ms/scan	Self-Supervised CNN	Google TPU
[37]	2D + 3D LiDAR	Objects in urban environment	98.3%	70 FPS	Transformer-based Model	NVIDIA A100
[21]	3D LiDAR	Denosing of point clouds	92.7%	15 FPS	PCA-based ML	Intel i7
[38]	2D LiDAR + RGB-D	Furniture, Doors, Walls	91.5%	Real-time (40 FPS)	CNN + Bayesian Filtering	NVIDIA Jetson Xavier
[22]	3D LiDAR	Outdoor objects	94.2%	Real-time	PCA-based Algorithm	Embedded GPU

				(10ms/frame)		
[32]	2D LiDAR	Path objects (walls, furniture)	93.2%	20 FPS	YOLO-based CNN	Raspberry Pi 4
[39]	3D LiDAR	Urban road objects	96.9%	28 FPS	Graph Neural Networks	NVIDIA RTX 3090
[27]	2D LiDAR	Rooms, doorways, corridors, halls	97.21%	Real-time (Raspberry Pi 4)	SVM, Random Forest, CatBoost, Decision Tree, Light GBM, Naïve Bayes	Raspberry Pi 4
Our Work	2D LiDAR	Rooms, doorways, corridors, halls	99.20%	63.43 FPS	CatBoost with PCA	Intel Core i9

While some methods achieve high-speed runtimes (e.g., 70 FPS, 40 FPS, 32 FPS) using deep learning and high-performance GPUs (e.g., RTX 3090, NVIDIA Titan X, A100), our method achieves a high processing speed of 63.43 FPS on an Intel Core i9 processor, making it a competitive real-time model with a superior accuracy rate. The results achieved in this work demonstrate that our model has significant potential for real-time applications, including mobile robotics and autonomous navigation. The data comparison shows that no optimal proposed method that can balance the computational efficiency and the accuracy rate across all applications.

Our work provides a new benchmark in LiDAR data classification with an accuracy of 99.2% and a competitive real-time processing rate of 63.43 FPS. By incorporating a hardware acceleration design and transitioning to an embedded AI accelerator such as FPGAs or NVIDIA Jetson. In addition, by applying model optimisation techniques such as quantisation, pruning, and knowledge distillation, further improvements in runtime, efficiency, and scalability can be achieved. We should know that the PCA method might lose some fine-grained features in environments that change a lot. For example, the corridors class's lower AUC compared to the other classes suggests that the system needs further refinement, including the integration of more contextual features that aid in distinguishing overlapping structure settings. In addition, using techniques like hybrid sensor fusion and deep learning-based feature extraction can help the model work better in different situations. Furthermore, the model currently utilises 2D LiDAR data, necessitating further enhancements to incorporate 3D LiDAR data analysis.

LiDAR technology has proven its vital role in different real-world application domains, such as logistics robots, industrial automation, and autonomous cars. In logistics robots, LiDAR is a vital piece that provides path planning and item tracking functionality, whereas in autonomous cars, LiDAR is an important component for navigation and avoiding obstacles. In addition, industrial automation and LiDAR play a vital role in real-time monitoring. These different tasks all rely on high LiDAR data classification accuracy; however, it struggles with runtime overhead with the 3D point cloud value of data for autonomous cars, while the computational efficiency might interrupt factory operations in industrial automation. Therefore, our contribution is vital and well-suited to this research-related context. PCA employment to reduce dimensionality in such applications that produce large-scale LiDAR data is vital to manage the runtime overhead.

In theory, if our proposed model is applied to LiDAR data in such large-scale applications that have a vast number of classes, our model would provide efficient computational cost with high classification accuracy. For instance, our model would score above a 95% accuracy rate classifying a LiDAR dataset that has a tenfold increase in size and scale the runtime overhead logarithmically rather than linearly. Although a validation study for such a promising theoretical prediction is

necessary, our model's efficiency and scalability show promising figures for real-world implementation.

6. CONCLUSION

In this paper, we try to solve the problem of how to quickly and accurately classify LiDAR-based datasets in indoor robotics by introducing a model that is scalable, accurate, and quick to run. With the help of combining ensemble methods with PCA, the model is able to run 13.07 times faster than the original model and achieve a high level of classification accuracy of 99.20%.

The proposed model showed potential to be a suitable design for real-time applications due to its ability to alleviate the curse of dimensionality in the LiDAR data. Through comparative evaluations, the model demonstrates its adaptability for tasks that require robust performance, such as navigation, activity monitoring, and obstacle detection.

This work contributes significantly to the field of LiDAR-driven autonomous systems, laying a solid foundation for future research in unstructured indoor environments. While the proposed methodology showed robust performance in both accuracy and computational efficiency, numerous limitations should be recognized. First, the evaluation was conducted on a single LiDAR dataset collected under controlled indoor conditions, which may limit the generalizability of the results to other environments, sensors, or robot platforms. Second, the use of PCA assumes linear correlations in the feature space, meaning that some task-specific nonlinear relationships may not be fully preserved. Third, all models were trained using default hyperparameters, which provides fairness and reproducibility but may not reflect peak performance for every classifier.

Future work will address these limitations by testing the framework on more diverse datasets, exploring nonlinear dimensionality reduction techniques, and applying systematic hyperparameter optimization to further enhance model performance and robustness. An integration of multiple data sources (sensor fusion), such as IMUs or cameras with LiDAR, is an ideal research direction in tuning such a model for complex scenarios. While this research focuses on 2D LiDAR dataset, the framework can be expanded to include 3D LiDAR datasets with further considerations. 3D point clouds are much denser and more variable, rising the computational loads of preprocessing, feature extraction, and dimensionality reduction. PCA may additionally require more components to preserve spatial structure, making real-time processing more challenging. Moreover, the proposed framework could greatly benefit from the integration of active learning techniques, which would enable the model to adapt to various new environments and reduce its reliance on human support.

In another research direction, the proposed model can benefit more from deploying it on a hardware platform like

edge computing devices or FPGA. This deployment would enhance the real-time capability of the model or its host, the robot, and address power efficiency issues. Another promising research direction would be to investigate the applicability and scalability of the model for diverse datasets to ensure the robustness of the work in real-world applications. Investigating these future trends can extend the benefits of this work to more advanced autonomous robotics applications, paving the way for systems that are both adaptable and intelligent.

ACKNOWLEDGMENT

This work is supported by Taibah University.

REFERENCES

- [1] Messbah, H., Emharraf, M., Saber, M. (2024). Robot indoor navigation: Comparative analysis of LiDAR 2D and visual SLAM. *IAES International Journal of Robotics and Automation*, 13: 41. <https://doi.org/10.11591/ijra.v13i1.pp41-49>
- [2] Liu, Y., Wang, S., Xie, Y., Xiong, T., Wu, M. (2024). A review of sensing technologies for indoor autonomous mobile robots. *Sensors*, 24(4): 1222. <https://doi.org/10.3390/s24041222>
- [3] Tao, Y., Popović, M., Wang, Y., Digumarti, S.T., Chebrolu, N., Fallon, M. (2022). 3D lidar reconstruction with probabilistic depth completion for robotic navigation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, pp. 5339-5346. <https://doi.org/10.1109/IROS47612.2022.9981531>
- [4] Lee, D., Jung, M., Yang, W., Kim, A. (2024). Lidar odometry survey: Recent advancements and remaining challenges. *Intelligent Service Robotics*, 17(2): 95-118. <https://doi.org/10.1007/s11370-024-00515-8>
- [5] Holmberg, M., Karlsson, O., Tulldahl, M. (2022). LiDAR positioning for indoor precision navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 359-368. <https://doi.org/10.1109/CVPRW56347.2022.00051>
- [6] Thomas, H., Agro, B., Gridseth, M., Zhang, J., Barfoot, T.D. (2021). Self-supervised learning of lidar segmentation for autonomous indoor navigation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, pp. 14047-14053. <https://doi.org/10.1109/ICRA48506.2021.9561701>
- [7] Li, J., Stevenson, R.L. (2020). Indoor layout estimation by 2D lidar and camera fusion. *arXiv preprint arXiv:2001.05422*. <https://doi.org/10.48550/arXiv.2001.05422>
- [8] Xie, F., Schwertfeger, S. (2023). Robust lifelong indoor lidar localization using the area graph. *IEEE Robotics and Automation Letters*, 9(1): 531-538. <https://doi.org/10.1109/LRA.2023.3334158>
- [9] Pasricha, S. (2024). AI and machine learning driven indoor localization and navigation with mobile embedded systems. *arXiv preprint arXiv:2408.04797*.
- [10] Tsai, C.Y., Nisar, H., Hu, Y.C. (2021). Mapless lidar navigation control of wheeled mobile robots based on deep imitation learning. *IEEE Access*, 9: 117527-117541. <https://doi.org/10.1109/ACCESS.2021.3107041>
- [11] Cherubin, S., Kaczmarek, W., Siwek, M. (2024). YOLO object detection and classification using low-cost mobile robot. *Przegląd Elektrotechniczny*, 100(9): 29-33. <http://doi.org/10.15199/48.2024.09.04>
- [12] Akai, N., Nakao, T. (2025). LiDAR-inertial Odometry based on extended Kalman filter. *Advanced Robotics*, 39(7): 357-367. <https://doi.org/10.1080/01691864.2025.2483216>
- [13] Liu, Y., Wang, C., Wu, H., Wei, Y., Ren, M., Zhao, C. (2022). Improved LiDAR localization method for mobile robots based on multi-sensing. *Remote Sensing*, 14(23): 6133. <https://doi.org/10.3390/rs14236133>
- [14] Jiang, X., Yang, D.K., Tian, Z., Liu, G., Lu, M. (2024). Single-line LiDAR localization via contribution sampling and map update technology. *Sensors*, 24(12): 3927. <https://doi.org/10.3390/s24123927>
- [15] Yue, X., Zhang, Y., Chen, J., Chen, J., Zhou, X., He, M. (2024). LiDAR-based SLAM for robotic mapping: State of the art and new frontiers. *Industrial Robot: The International Journal of Robotics Research and Application*, 51(2): 196-205. <https://doi.org/10.1108/IR-09-2023-0225>
- [16] Belkin, I., Abramenko, A., Yudin, D. (2021). Real-time lidar-based localization of mobile ground robot. *Procedia Computer Science*, 186: 440-448. <https://doi.org/10.1016/j.procs.2021.04.164>
- [17] Joerger, M., Wang, J., Hassani, A. (2022). On uncertainty quantification for convolutional neural network LiDAR localization. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, Aachen, Germany, pp. 1789-1794. <https://doi.org/10.1109/IV51971.2022.9827445>
- [18] Wu, B., Iandola, F., Jin, P.H., Keutzer, K. (2017). Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Honolulu, HI, USA, pp. 129-137. <https://doi.org/10.1109/CVPRW.2017.60>
- [19] Lu, W., Zhou, Y., Wan, G., Hou, S., Song, S. (2019). L3-net: Towards learning based lidar localization for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6389-6398. <https://doi.org/10.1109/CVPR.2019.00655>
- [20] Goswami, R.G., Patel, N., Krishnamurthy, P., Khorrami, F. (2025). FlashMix: Fast map-free lidar localization via feature mixing and contrastive-constrained accelerated training. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Tucson, AZ, USA, pp. 2011-2020. <https://doi.org/10.1109/WACV61041.2025.00202>
- [21] Duan, Y., Yang, C., Chen, H., Yan, W., Li, H. (2021). Low-complexity point cloud denoising for LiDAR by PCA-based dimension reduction. *Optics Communications*, 482: 126567. <https://doi.org/10.1016/j.optcom.2020.126567>
- [22] Cheng, D., Zhao, D., Zhang, J., Wei, C., Tian, D. (2021). PCA-based denoising algorithm for outdoor lidar point cloud data. *Sensors*, 21(11): 3703. <https://doi.org/10.3390/s21113703>
- [23] Wu, B., Wan, A., Yue, X., Keutzer, K. (2018). Squeezeseg: Convolutional neural nets with recurrent

- CRF for real-time road-object segmentation from 3D lidar point cloud. In 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, pp. 1887-1893. <https://doi.org/10.1109/ICRA.2018.8462926>
- [24] Kaleci, B., Turgut, K., Dutagaci, H. (2022). 2DLaserNet: A deep learning architecture on 2D laser scans for semantic classification of mobile robot locations. *Engineering Science and Technology, an International Journal*, 28: 101027. <https://doi.org/10.1016/j.jestch.2021.06.007>
- [25] Li, F., Fu, C., Sun, D., Li, J., Wang, J. (2024). SD-SLAM: A semantic SLAM approach for dynamic scenes based on LiDAR point clouds. *Big Data Research*, 36: 100463. <https://doi.org/10.1016/j.bdr.2024.100463>
- [26] Bouazizi, M., Lorite Mora, A., Ohtsuki, T. (2023). A 2D-Lidar-equipped unmanned robot-based approach for indoor human activity detection. *Sensors*, 23(5): 2534. <https://doi.org/10.3390/s23052534>
- [27] Alenzi, Z., Alenzi, E., Alqasir, M., Alruwaili, M., Alhmiedat, T., Alia, O.M.D. (2022). A semantic classification approach for indoor robot navigation. *Electronics*, 11(13): 2063. <https://doi.org/10.3390/electronics11132063>
- [28] Mochurad, L., Hladun, Y., Tkachenko, R. (2023). An obstacle-finding approach for autonomous mobile robots using 2D LiDAR data. *Big Data and Cognitive Computing*, 7(1): 43. <https://doi.org/10.3390/bdcc7010043>
- [29] He, F., Zhang, L. (2023). Design of indoor security robot based on robot operating system. *Journal of Computer and Communications*, 11(5): 93-107.
- [30] Zhou, G., Xu, S., Zhang, S., Wang, Y., Xiang, C. (2022). Multi-floor indoor localization based on multi-modal sensors. *Sensors*, 22(11): 4162. <https://doi.org/10.3390/s22114162>
- [31] Zhang, L., Digumarti, T., Tinchev, G., Fallon, M. (2023). Instaloc: One-shot global lidar localisation in indoor environments through instance learning. *arXiv preprint arXiv:2305.09552*. <https://doi.org/10.48550/arXiv.2305.09552>
- [32] Alparslan, O., Cetin, O. (2023). Real-time indoor path planning using object detection for autonomous flying robots. *Intelligent Automation & Soft Computing*, 36(3): 3355-3370. <https://doi.org/10.32604/iasc.2023.035689>
- [33] Alhmiedat, T. (2024). LidarDataFrames. <https://www.kaggle.com/datasets/tareqalhmiedat/lidardaframes>.
- [34] Tu, J.B., Liao, W.J., Liu, W.C., Gao, X.H. (2024). Using machine learning techniques to predict the risk of osteoporosis based on nationwide chronic disease data. *Scientific Reports*, 14(1): 5245. <https://doi.org/10.1038/s41598-024-56114-1>
- [35] Miyamoto, R., Adachi, M., Nakamura, Y., Nakajima, T., Ishida, H., Kobayashi, S. (2019). Accuracy improvement of semantic segmentation using appropriate datasets for robot navigation. In 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, pp. 1610-1615. <https://doi.org/10.1109/CoDIT.2019.8820616>
- [36] Dewan, A., Oliveira, G.L., Burgard, W. (2017). Deep semantic classification for 3D lidar data. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, pp. 3544-3549. <https://doi.org/10.1109/IROS.2017.8206198>
- [37] Pronobis, A., Jensfelt, P. (2012). Large-scale semantic mapping and reasoning with heterogeneous modalities. In 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, pp. 3515-3522. <https://doi.org/10.1109/ICRA.2012.6224637>
- [38] Qi, X., Wang, W., Liao, Z., Zhang, X., Yang, D., Wei, R. (2020). Object semantic grid mapping with 2D LiDAR and RGB-D camera for domestic robot navigation. *Applied Sciences*, 10(17): 5782. <https://doi.org/10.3390/app10175782>
- [39] Thomas, H., Goulette, F., Deschaud, J.E., Marcotegui, B., LeGall, Y. (2018). Semantic classification of 3D point clouds with multiscale spherical neighborhoods. In 2018 International Conference on 3D Vision (3DV), Verona, Italy, pp. 390-398. <https://doi.org/10.1109/3DV.2018.00052>