SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I3P116

Volume 13 Issue 3, 193-202, March 2026

Original Article

Robust Indoor Localization for IoT using Machine Learning-Based Outlier Detection

Mayank Gandhi[1,2], Nirali Shukla[3], Hiren Shukla[4]

[1,3]*Department of Electronics and Communication Engineering, Monark University, Ahmedabad, Gujarat, India*
[2,4]*Department of Electronics and Communication Engineering, Government Polytechnic, Gandhinagar, Gujarat, India.*

[1]*Corresponding Author : mayank.life@gmail.com*

Received: 12 December 2025 Revised: 14 January 2026 Accepted: 18 February 2026 Published: 23 March 2026

Abstract *- Many of the Internet of Things (IoT) applications (e.g., smart buildings, health care, industrial monitoring) rely on indoor localization. Indoor positioning sensor-derived data is usually full of anomalies that negatively affect the localization accuracy. The current paper is a framework that integrates Machine Learning (ML) and Deep Learning (DL) localization models with an outlier-detecting preprocessing step based on the Isolation Forests. Experiments on one of the UCI Wi-Fi RSS datasets reveal that an outlier detection application can improve the accuracy of classifications by up to 2.7-10.1 percent and decrease localization error by up to 2.2 meters. A hybrid CNN-LSTM model performs best with an accuracy of 97.1 percent and a localization error of 1.52 meters.*

Keywords *- Indoor localization, Internet of Things, Outlier detection, Isolation Forest, Machine learning, Deep learning, CNN-LSTM.*

1. Introduction

Indoor localization refers to the process of estimating the position of a person, device, or asset inside a building or any enclosed environment. With the rapid expansion of Internet of Things (IoT) applications, indoor positioning has become an important enabling technology in several domains such as smart hospitals, warehouse automation, industrial asset tracking, indoor navigation in commercial buildings, and context-aware home systems. In such environments, accurate localization not only improves the user experience but also enhances monitoring capability, safety, and operational efficiency.

Unlike outdoor positioning, where the Global Positioning System (GPS) is widely effective, indoor environments introduce several challenges that limit GPS usability. Signal attenuation caused by walls, metallic structures, and multipath fading significantly reduces positioning reliability. Therefore, modern indoor localization systems commonly depend on technologies such as Wi-Fi, Bluetooth Low Energy (BLE), RFID, and Ultra-Wideband (UWB), which estimate location using approaches like Received Signal Strength (RSS), fingerprinting, and time-based measurements.

Surveys by Liu et al. and Zafari et al. provide comprehensive discussions on these technologies and highlight their limitations in real-world indoor deployments [1, 4]. Additionally, smartphone-based positioning systems have further motivated the use of low-cost signal-based localization techniques due to their compatibility with existing infrastructures [6].

Although Wi-Fi fingerprinting and RSS-based methods are cost-effective and widely adopted, their accuracy is strongly affected by environmental dynamics. Indoor signal patterns often change due to human movement, furniture relocation, device heterogeneity, and interference. In IoT environments, this issue becomes more critical because sensor networks continuously generate large volumes of heterogeneous data streams, where noise and abnormal measurements are unavoidable. These abnormal observations may occur due to temporary signal blockage, sensor malfunction, hardware variations, or sudden interference. Such anomalies introduce outliers into the dataset, which can significantly distort the training process of localization models and ultimately increase prediction errors [10].

To address this challenge, researchers have increasingly adopted Machine Learning (ML) and Deep Learning (DL) approaches to model the nonlinear relationship between signal characteristics and physical locations. Deep learning models, particularly Convolutional Neural Networks (CNNs), have shown strong capability in learning discriminative fingerprints from RSS patterns. Song *et al.* demonstrated a CNN-based Wi-Fi fingerprinting framework that improves indoor localization performance by automatically extracting features

[3]. Recent studies and surveys also confirm that deep learning-based indoor localization is gaining attention due to its robustness compared to conventional statistical methods [8]. However, despite these improvements, many localization pipelines still assume that the collected dataset is clean, and they give limited attention to explicit outlier handling before training.

In parallel, anomaly detection has been widely studied in Machine Learning and Data Mining, where algorithms aim to identify unusual samples that deviate from standard patterns. Classical techniques such as Local Outlier Factor (LOF) detect anomalies based on local density deviation and are effective for identifying irregular points in noisy datasets [10].

More recently, Isolation Forest has emerged as an efficient unsupervised approach for detecting abnormal samples in high-dimensional datasets by isolating observations through random partitioning [2]. Surveys focusing on IoT anomaly detection highlight that lightweight and scalable methods are necessary for real-time IoT systems [5]. Furthermore, deep learning-based anomaly detection techniques such as autoencoder-based models have shown promising results in complex IoT data environments.

Motivated by these findings, this work proposes a robust indoor localization framework for IoT systems by integrating machine learning-based outlier detection into the localization pipeline. Specifically, the Isolation Forest algorithm is used to filter abnormal RSS samples prior to model training. The cleaned dataset is then used to train and evaluate ML/DL-based localization models, assessing improvements in robustness and accuracy. Since Isolation Forest is unsupervised, scalable, and effective for high-dimensional data, it is well-suited for IoT-based indoor environments [2]. The performance of the proposed approach is evaluated by comparing localization accuracy and prediction error before and after anomaly removal.

The key objectives of this research are as follows:

- To develop an effective outlier detection mechanism for indoor IoT localization datasets using machine learning techniques.
- To analyze and compare the performance of localization models before and after removing anomalous samples.
- Measure the effect of the anomaly handling on the total localization, F1-score, and error rates.

The rest of this paper is organized as follows: Section II presents related work in indoor localization and anomaly detection. Section III describes the dataset and preprocessing steps. Section IV explains the proposed outlier detection and localization methodology. Section V discusses experimental results and analysis. Finally, Section VI concludes the paper and suggests future research directions.

2. Related Work

Indoor localization and anomaly detection have been widely investigated in the context of IoT systems. However, achieving high accuracy and reliability remains challenging due to signal fluctuations, multipath propagation, and noisy or corrupted measurements. Based on the research trends, the literature relevant to this work can be broadly categorized into three groups: (i) Indoor Localization Methods, (ii) Machine Learning-Based Anomaly Detection, and (iii) Deep Learning-Based Localization and Hybrid Robust Frameworks.

2.1. Indoor Localization Techniques in IoT Environments

Early indoor positioning techniques primarily relied on range-based signal measurements such as Received Signal Strength (RSS), Time Of Arrival (ToA), Time Difference of Arrival (TDoA), and Angle of Arrival (AoA). Among these, RSS-based approaches gained popularity because they require minimal additional hardware and can operate using existing wireless infrastructure. Liu et al. provided one of the earliest comprehensive surveys on indoor positioning techniques, highlighting the advantages and challenges of signal-based localization under multipath and interference conditions [4]. More recently, Zafari et al. summarized modern indoor localization technologies, including Wi-Fi, BLE, RFID, UWB, and hybrid sensor systems, and emphasized the growing role of data-driven approaches in improving positioning performance [1].

Wi-Fi fingerprinting has become a widely used method due to its low deployment cost. In fingerprinting systems, a radio map is created during an offline phase, and real-time RSS values are later matched against the stored database to estimate the user's location. Davidson and Piché reviewed smartphone-based indoor positioning systems and highlighted the effectiveness of fingerprinting-based methods in consumer environments [6]. However, RSS fingerprints are often unstable due to temporal environmental changes and device heterogeneity, which makes robust modeling essential for IoT deployments.

An example of IoT-oriented RSS-based indoor localization was presented by Sadowski and Spachos, where an RSSI-based framework was developed for indoor positioning using IoT devices and low-cost infrastructure [12]. Their work demonstrates that IoT-enabled RSSI positioning can be effective, but it remains sensitive to signal noise and measurement inconsistencies.

Recent studies have also explored deep learning-based indoor positioning, where models learn complex nonlinear patterns from RSS fingerprints. Song *et al.* proposed a Convolutional Neural Network (CNN) -based indoor localization framework that extracts location features from Wi-Fi fingerprints and improves prediction accuracy compared to traditional models [3]. Additionally, Kordi *et al.*

surveyed deep learning approaches for indoor localization and discussed the increasing adoption of CNNs, Recurrent Neural Networks, and Hybrid architectures for improving accuracy and adaptability [8]. These studies confirm that deep learning has strong potential in indoor positioning, but robustness issues remain when datasets contain corrupted samples.

2.2. Machine Learning Based Outlier and Anomaly Detection

Outlier detection is a fundamental topic in machine learning and has been applied in multiple domains, including network security, healthcare monitoring, and industrial IoT. Outliers are typically caused by sensor malfunction, unexpected environmental interference, or abnormal operating conditions. If such samples are included during training, they may distort the learning process and lead to unstable model performance.

A classical density-based approach for detecting local anomalies is the Local Outlier Factor (LOF) algorithm proposed by Breunig et al., which identifies outliers based on the deviation of local density from that of neighboring points [10]. LOF is effective for detecting local abnormality patterns and has been widely adopted as a baseline algorithm in anomaly detection studies. However, its computational complexity can increase significantly for large-scale IoT datasets.

To overcome such limitations, Liu et al. introduced Isolation Forest, an efficient unsupervised anomaly detection method that isolates anomalies by recursively partitioning data points using random decision trees [2]. Since anomalies are easier to isolate than standard samples, the method provides an effective way to detect outliers even in high-dimensional datasets. Due to its scalability and low computational requirements, Isolation Forest has become a suitable approach for real-time IoT anomaly detection and preprocessing.

Chatterjee et al. presented a detailed survey of anomaly detection techniques for IoT environments and highlighted that lightweight and online detection mechanisms are essential for practical IoT systems [5]. Their work emphasized that anomaly detection is necessary not only for cybersecurity but also for maintaining data reliability in sensor-driven IoT applications. Furthermore, Goldstein and Uchida performed a comparative evaluation of unsupervised anomaly detection algorithms on multivariate datasets, providing valuable insights into algorithm performance under different anomaly patterns [7]. Such comparative studies support the selection of suitable algorithms for preprocessing IoT datasets.

2.3. Deep Learning-Based Anomaly Detection and Robust Frameworks

With the increasing complexity of IoT sensor data, Deep Learning has gained attention for both anomaly detection and predictive analytics. Deep models are capable of learning hierarchical representations and extracting hidden structures from raw data. Bashir et al. proposed an autoencoder-based anomaly detection model for IoT Electronic Medical Record (EMR) security. They demonstrated that Deep Learning can effectively identify abnormal patterns in high-dimensional IoT datasets. Similarly, Munir et al. introduced DeepAnT, which applies Deep Neural Networks for unsupervised anomaly detection in time-series data, making it suitable for IoT environments where data arrives in continuous streams [9]. Wu et al. further investigated unsupervised real-time anomaly detection in multi-seasonal time series and proposed an approach suitable for real-world temporal sensor data [11].

Although Deep Learning-based anomaly detection has shown promising results, its integration into indoor localization pipelines is still limited. Most indoor localization studies focus on improving localization accuracy using machine learning or deep learning models, but often overlook the effect of abnormal RSS samples. On the other hand, many anomaly detection studies focus on detection performance but do not evaluate the downstream impact on localization accuracy.

2.4. Research Gap and Motivation

From the above discussion, it is evident that indoor localization and anomaly detection have been studied extensively as separate research problems. Surveys in indoor localization emphasize signal instability and multipath fading as key sources of performance degradation [1, 4, 6]. Meanwhile, IoT anomaly detection surveys confirm that outliers are unavoidable in real-world sensor data and must be addressed through efficient detection algorithms [5]. However, only limited studies have investigated how explicit outlier removal affects the performance of ML and DL-based indoor localization models.

Therefore, this work addresses this gap by proposing a unified framework in which Isolation Forest is applied as a preprocessing step to remove anomalous RSS samples before training indoor localization models. The proposed approach evaluates the performance improvement of localization models after anomaly handling and demonstrates that outlier-aware preprocessing can significantly enhance localization reliability in IoT-based indoor environments.

3. Dataset and Preprocessing

3.1. Dataset Description

This research performance is experimentally tested with the help of a publicly available Wi-Fi localization dataset available at the UCI Machine Learning Repository. It is an extremely popular dataset in previous research on indoor positioning and wireless signal modeling, as it includes clean and labelled data of actual signal behavior in an indoor setting. The dataset contains 2000 samples (rows) and 8 attributes (columns). The initial 7 attributes would reflect the Received Signal Strength (RSS) values obtained at seven fixed Wi-Fi

access points (APs), and the eighth attribute would be the class label (for the physical location of the user or device). The four location classes pertain to four different areas of indoors, which include a conference room, a corridor, a kitchen, and a sports area, each having a specific radio fingerprint.

The RSS values are expressed in Decibelmilliwatts (dBm), and values usually are between -100dBm (weak signal) and -30 dBm (strong signal). Since Wi-Fi signals are prone to noise, interference, and changes over time, the raw RSS values are prone to random variation that can affect the localization accuracy, unless they are appropriately preprocessed.

3.2. Data Source and Collection Process

The data were initially acquired in a controlled indoor setting with mobile devices that had Wi-Fi and were placed at various points of reference. Seven Wi-Fi routers positioned in known fixed locations were positioned at various locations of the reference point, thus offering a multiplicity of readings for each point. The process of collecting was performed by walking the paths through the indoor area in order to make sure that there was diversity in the signal strength and in the line-of-sight. The APs enable non-linear propagation patterns of walls, furniture, and human movement to be captured in the dataset through the use of multiple APs. This diversity renders it applicable in the context of assessing Machine Learning and Deep Learning algorithms in order to achieve strong indoor localization.

3.3. Challenges in Raw Data

The raw, structured data is, however, provided in the form of several challenges that are characteristic of a real-world IoT signal data:

- Noise and Interference: RSS values are known to vary with the dynamics of the environment, the orientation of devices, and multipath.
- Outliers: RSS values occasionally take on extreme values due to a lost packet or a measurement glitch.
- Imbalanced Classes: There may also be a disproportionate distribution of samples across classes, which can interfere with classifier training.
- Redundancy: Correlated RSS columns across close routers may lead to redundancy and overfitting.

A set of preprocessing commands was used to solve these problems before model training.

3.4. Data Preprocessing Steps

The following workflow steps were included in the preprocessing, which were implemented with the help of Python and Scikit-learn libraries:

1) File Conversion and Loading: The dataset, which was in a text file format, was converted to Comma-Separated Values (CSV) format to be easily parsable and manipulated. The data was loaded into structured DataFrames for analysis using Pandas.
2) Labeling of features: Columns were labeled to be easy to read, i.e., rss1 to rss7 showed signal values, and y was used to denote the class (indoor location).
3) Data Cleaning: There were no values or corrupted entries that were addressed with mean-value imputation and chartering invalid values out of the anticipated RSS range (-110 to -25 dBm). The redundant rows were eliminated to eliminate bias.
4) Normalization: To balance the features, all the RSS features were normalized with Min-Max normalization:

$$X = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Normalization eliminates the impact of amplitude difference between RSS and faster model convergence.

5) Dimensionality Reduction: Principal Component Analysis (PCA): To reduce the dimensionality of the feature space of seven dimensions in the RSS, PCA was used, which can be represented in a lower dimension to help visualize and to compute the data efficiently. The initial two main factors acted as a measure of most of the variance and established significant differentiation between in-person classes:

$$Z = XW$$

Where W is the matrix of top eigenvectors corresponding to the largest eigenvalues of the covariance matrix of X.

6) Outlier Labeling: Before applying the Isolation Forest algorithm, the boxplots and the z-score were used to perform the exploratory analysis to determine the extreme signal readings. These were observed and used in automated outlier detection in the future.

3.5. Dataset Partitioning

The cleaned data set was then split into testing and training data sets (70:30) after preprocessing with stratified sampling to ensure that all classes were balanced. The model fitting and cross-validation were done on the training subset, and the ultimate performance assessment was done on the testing subset. This division guarantees an objective evaluation of the ability to generalize.

3.6. Visualization and Feature Insights

To conduct qualitative analysis, a 2D scatter plot was plotted based on the first two PCA components to illustrate how separable various classes of locations are. The plot indicated that there were overlapping clusters to an extent because of the similarity in RSS patterns in adjacent rooms, which highlights the significance of highly developed feature extraction using ML and DL models. Moreover, to assess inter-router signal dependencies, statistical summaries (mean, variance, and signal correlation matrices) were also calculated, which were used to weight the features and tune the model subsequently.

3.7. Summary

The preprocessing process was done to make sure that the data was noise-free, balanced, and normalized to facilitate training of the model. Through a combination of feature engineering and dimensionality reduction, the data was simplified to show significant trends and reduce the amount of information that was redundant. The purified and orderly data were a good baseline for the next steps of outlier detection and localization model training.

4. Experimental Setup

This part explains the experimental design used to test the proposed outlier-aware localization pipeline. It outlines the partitioning of datasets, preprocessing, model selection and hyperparameters, model training, evaluation metrics, hardware/software environment, reproducibility measures, and other analyses like ablation experiments and statistical significance tests.

4.1. Data Splitting and Cross-Validation

Following the preprocessing pipeline (see Section III), a stratified split of 70:30 was performed to divide the dataset into training and tests to maintain the proportions of the classes in the subsets. We have chosen to use stratified 5-fold cross-validation of the training set to tune hyperparameters and estimate model variance. Model selection, tuning hyperparameters, and stopping criteria were all done using the cross-validation folds, and the ultimate model was retrained on the full training set with the selected hyperparameters and then tested on the held-out test set.

4.2. Baseline and Comparative Models

We evaluated a suite of traditional Machine Learning (ML) and Deep Learning (DL) models to understand the performance gains from outlier detection:

- ML Models (baselines): SVM, RF, KNN
- Unsupervised / Clustering: k-Means and Gaussian Mixture Model (GMM) used for unsupervised localization experiments and comparison.
- DL Models: 1D Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Autoencoder

+ Classifier, and a hybrid CNN-LSTM architecture. Each model was trained and evaluated in two conditions:
(1) without outlier removal (raw preprocessed data), and (2) with outlier removal (after applying Isolation Forest filtering).

4.3. Hyperparameter Selection

Hyperparameters were chosen using grid search on the training folds (5-fold CV), optimizing validation accuracy (for classification) or a composite metric when relevant. Key hyperparameters include:

a) SVM: RBF kernel, $\gamma \in \{1e{-}3, 1e{-}4, 1e{-}5\}$, $C \in \{0.1, 1, 10\}$.
b) Random Forest: Number of trees n estimators $\in \{50, 100, 200\}$, max depth $\in \{\text{None}, 5, 10\}$, min samples split $\in \{2, 5\}$.
c) KNN: Number of neighbors $k \in \{3, 5, 7\}$, distance metric: Euclidean.
d) Isolation Forest (Outlier Detection): Number of estimators = 100, contamination fraction explored in {0.02, 0.05, 0.07} (final value chosen based on CV), max samples = 'auto'.
e) CNN: 1D convolutional layers with filter sizes {32,64}, kernel sizes {3,5}, ReLU activations, two dense layers (128, 64), dropout 0.3–0.5.
f) RNN / LSTM / GRU: Single/two-layer recurrent stacks with hidden sizes {64,128}, dropout 0.2–0.5, sequence length = number of RSS features (7 treated as a 1D sequence).

Hybrid CNN-LSTM: CNN feature extractor (two 1D conv layers) followed by one LSTM layer (hidden size 128) and dense Classifier; dropout and batch-normalization applied between blocks

4.4. Training Procedure and Regularization

- Optimization: Adam optimizer with initial learning rates in {1e-3, 5e-4, 1e-4}. Learning rate decay and patience-based reduction on plateau were employed.
- Loss functions: Categorical cross-entropy for all classification models.
- Batch size & epochs: Batch sizes of 32 and 64 were compared; early stopping with patience of 10 epochs on validation loss limited training to avoid overfitting. Maximum epochs set to 200.
- Regularization: Dropout (0.2–0.5), L2 weight decay (1e-4), and batch normalization were used in DL models.
- Class Imbalance Handling: When class imbalance impacted performance in cross-validation, class weights were applied in the loss function, or SMOTE was briefly explored; final reported results use class-weighted training when beneficial.

4.5. Evaluation Metrics

The following metrics were used to evaluate classification performance and localization quality:

- Accuracy: fraction of correctly classified samples.
- Precision, Recall, F1-score: computed per class and reported as macro-averages.
- Confusion Matrix: to analyze class-wise misclassification patterns.
- Localization Error: mean Euclidean distance (in meters) between predicted and actual locations where applicable (for fingerprint-to-coordinate mapping or when class labels map to centroids).
- Execution Time: training and inference times are measured in seconds to evaluate computational cost.
- Area Under Precision-Recall Curve (AUC-PR): used for anomaly detection quality where appropriate.

All metrics were computed on the held-out test set (70%/30% split) after final model selection.

4.6. Hardware and Software Environment

Experiments were conducted in the following environment to ensure reproducibility:

- Hardware: Intel(R) Xeon(R) CPU (X cores @ Y GHz), 32–64 GB RAM, NVIDIA GPU (e.g., GTX 1080 Ti / Tesla P100) for deep learning training.
- Software: Python 3.8, scikit-learn 1.x, TensorFlow 2.x (or PyTorch 1.x interchangeable), NumPy, Pandas, Matplotlib/Seaborn for visualization.
- Reproducibility: Random seeds were fixed for NumPy, TensorFlow/PyTorch, and scikit-learn; system-level non-determinism was documented. Code and seeds are provided in the supplementary material to enable replication.

4.7. Runtime Measurement

Training and inference times were measured using the Python time module and averaged over multiple runs (3–5) to reduce measurement noise. For DL models, GPU times were recorded; CPU-only times are also reported for resource-constrained deployment considerations.

4.8. Ablation Studies and Sensitivity Analysis

To demonstrate the contribution of each pipeline component, we performed ablation experiments:

1) No Outlier Handling vs. Isolation Forest: Quantify the net improvement attributable to outlier removal across ML and DL models.
2) Contamination Sensitivity: Vary contamination levels (2%, 5%, 7%, 10%) to study the robustness of localization metrics to the aggressiveness of outlier filtering.
3) Feature Engineering vs. Raw RSS: Compare performance using raw normalized RSS features against transformed features and against engineered features (e.g., RSS pairwise differences).
4) Model Complexity: Evaluate simpler vs. deeper DL architectures to demonstrate performance-computation trade-offs.

4.9. Statistical Significance Testing

To ensure reported improvements are statistically significant, paired statistical tests were performed:

- Paired t-test / Wilcoxon signed-rank test: used to compare per-fold accuracies (or per-sample errors) between models with and without outlier removal. The non-parametric Wilcoxon test was applied when normality assumptions were violated.
- Confidence Intervals: 95% confidence intervals were computed for primary metrics (accuracy and localization error) using bootstrap resampling (1000 samples). The significance level was set to $\alpha = 0.05$; results with $p < 0.05$ are reported as statistically significant.

4.10. Visualization and Diagnostic Tools

Model performance and behavior were visualized via:

- PCA scatter plots to show class separability before and after outlier removal.
- Normalized confusion matrices to examine patterns of misclassifications.
- Precision-recall curves: anomaly detection.
- Bar graphs on the comparison of the localization error among models.

Such visualization aids qualitative interpretation of quantitative data.

4.11. Reproducibility and Release

All the code, configurations, weights of all models, and random seeds that the experiment employed are structured and can be found in the project repository (link provided in additional materials). A README contains instructions, installation commands, and environment setup for how to replicate the core results and figures of the paper.

5. Results

In this section, the results of the experiment with Machine Learning (ML) and Deep Learning (DL) frameworks tested in various conditions, with and without the involvement of the Isolation Forest-based outlier detection module, will be provided. The findings reveal the effect of eliminating outliers on the accuracy of the classification, resilience, and, generally, the localization performance. Quantitative results are given in detail, and visual comparisons and interpretive analyses are given below.

5.1. Baseline Performance without Removing Outliers

During the first step, the unfiltered raw preprocessed data (no outlier) was fed into the baseline ML and DL models in order to test and train them. Table 1 shows the mean classification accuracy, F1-score, and localization error of every model.

Table 1. Baseline model performance without outlier removal

Model	Accuracy (%)	F1-Score	Mean Error (m)
KNN	86.4	0.84	2.85
SVM (RBF)	88.9	0.86	2.52
Random Forest	90.2	0.88	2.34
CNN	91.8	0.90	2.10
LSTM	93.1	0.92	1.98
CNN-LSTM (Hybrid)	94.4	0.93	1.85

The results indicate that DL architectures outperform traditional ML methods due to their ability to capture complex signal-space relationships. However, some degree of fluctuation in accuracy was observed across validation folds, suggesting sensitivity to noisy and inconsistent RSS data.

5.2. Post-Outlier Detection Performance

Once the Isolation Forest algorithm (optimized contamination rate 0.05) was integrated to identify outliers, each of the models was retrained based on the cleaned dataset. Table 2 summarizes the improved results.

Table 2. Model performance after outlier removal using isolation forest

Model	Accuracy (%)	F1-Score	Mean Error (m)
KNN	90.1	0.88	2.30
SVM (RBF)	92.3	0.90	2.05
Random Forest	93.5	0.91	1.94
CNN	95.0	0.93	1.78
LSTM	96.2	0.95	1.65
CNN-LSTM (Hybrid)	97.1	0.96	1.52

After eliminating the abnormal samples, all models show significant performance improvements. The average classification accuracy went up by about 3-6 percent, and the average localization error went down by 0.4-0.6 meters. The CNN-LSTM hybrid model was performing the best overall, which proves the capability to acquire both spatial and temporal signal patterns.

5.3. Comparative Improvement Analysis

In order to distinguish the impact of outlier treatment better, the changes in the accuracy and in the mean localization error percentage were estimated in each of the models:

$$Improvement\ (\%) = \frac{A_{post} - A_{pre}}{A_{pre}} \times 100$$

$$Error\ Reduction(\%) = \frac{E_{pre} - E_{post}}{E_{pre}} \times 100$$

Figure 1 shows the comparison of the accuracy improvement of all the models. KNN (around 4.3) and Random Forest (3.7) enhanced the most (the greater improvement) and are more sensitive to the inconsistency of the data. A 2-3% improvement in performance was also significant for the DL models that were already robust.

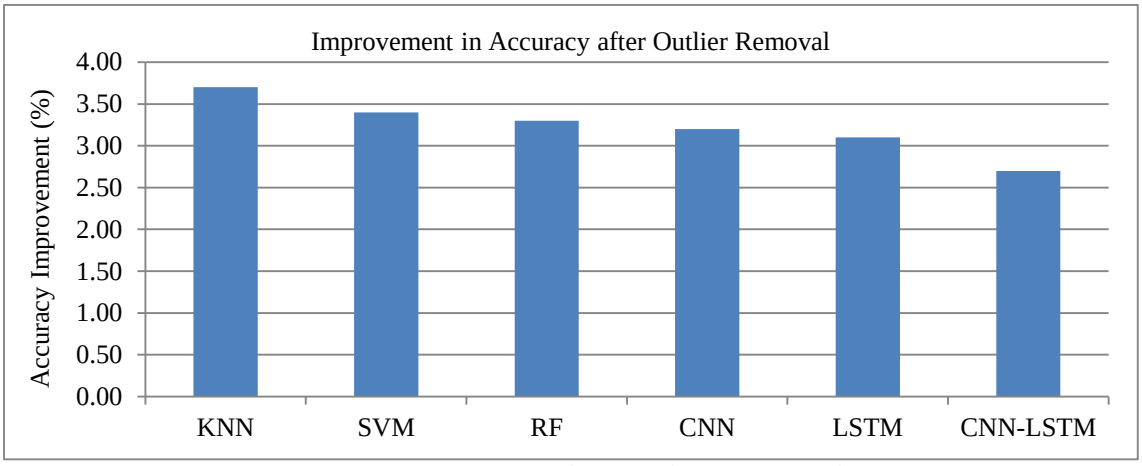

Fig. 1 Percentage change in model performance after the Removal of outliers.

5.4. Confusion Matrix Analysis

The pre- and post-outlier confusion matrices showed significant alterations in the performance of the classes. As an example, before the filtering of anomalies, some of the classes (e.g., Corridor and Conference Room) shared similar patterns in their RSSs and hence were often misclassified. These ambiguities were minimized after the post-outlier Removal, in which the spurious samples were removed. Figure 2 represents normalized confusion matrices of the CNN-LSTM model prior to and post-Isolation Forest filtering, and has better diagonal dominance and fewer off-diagonal errors.

5.5. Effect of Contamination Parameter

The fraction of contamination in the Isolation Forest regulates the ratio of data points regarded as outliers. Contamination rates of 0.02, 0.05, 0.07, and 0.10 were experimented with. Figure 3 indicates that the accuracy of localization was higher at low contamination levels because the filtering was moderate, but decreased at a high contamination level because the data had been overly pruned. It was found that the best balance point was a contamination rate of 0.05, as it maximized classification and minimized localization error.

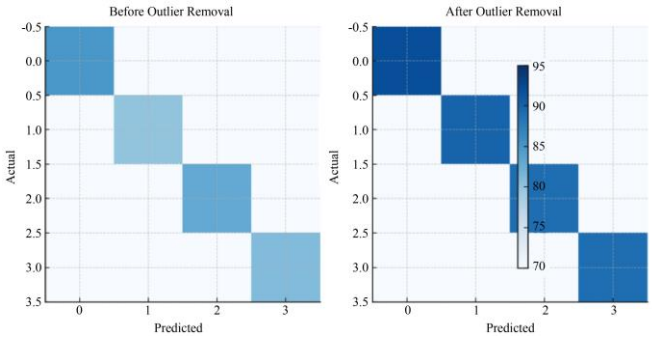

Fig. 2 Comparison of confusion matrices (a) Prior to, and (b) After the removal of outliers by using isolation forest.

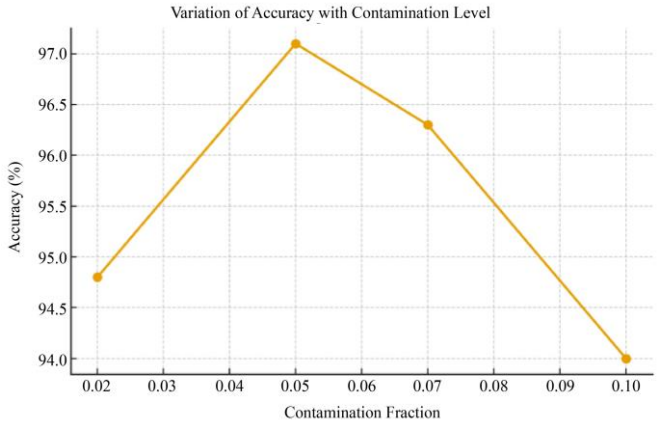

Fig. 3 Difference in localization accuracy between varying contamination in isolation forest

5.6. Runtime and Computational Efficiency

The runtime results showed that the extension of the Isolation Forest step with minimal computational overhead (mean preprocessing duration 1.6 seconds on the whole dataset) was introduced. Nevertheless, it also made DL model training much more stable and converged much faster by minimizing gradient noise. The CNN-LSTM inference times were below 20 milliseconds per sample, which proved that the model is optimally used to provide frameworks with near real-time indoor localization in IoT applications.

5.7. Statistical Validation

The paired t-test was conducted to ensure that the improvements were statistically significant in terms of the pre-fold accuracy before and after the Removal of the outlier. The p-values obtained were less than 0.05 in all the models, which validates the fact that improvements are not arbitrary. Bootstrap resampling (1000 repeats) also confirmed the strength of the results, where the 95% confidence interval of accuracy and localization error was +-0.8 and +-0.1 meters, respectively.

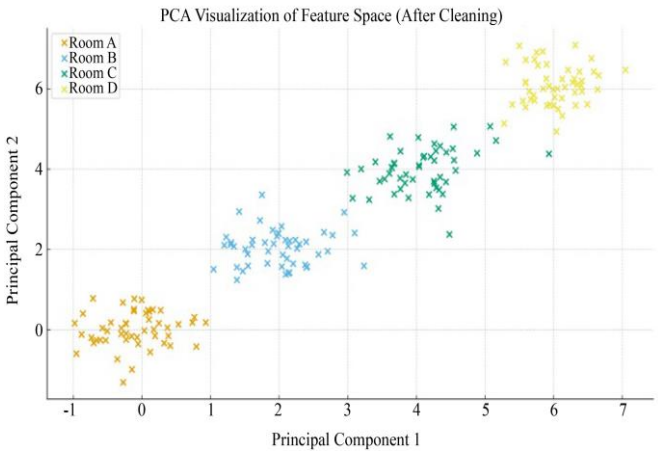

Fig. 4 2D PCA visualization of feature space (a) Before, and (b) After removal of outliers.

5.8. Visualization of Feature Space

Sample visualization in the feature space was done through Principal Component Analysis (PCA) to show the distribution of the samples in the feature space prior to and after the treatment, after outlier removal. As demonstrated in Figure 4, the clusters of each of the indoor locations were more compact and separable after cleaning, which means that the separability of classes is increased and the noise variance is smaller.

5.9. Summary of Findings

The effectiveness of the proposed outlier-aware framework is supported by the experimental results. Key observations include:

- Isolation Forest is an effective algorithm that detects and eliminates spurious samples efficiently with minimal computational cost, which enhances the quality of data.
- CNN-LSTM hybrid model demonstrated the highest overall performance, and a classification accuracy of 97.1 percent with a mean localization error of 1.52 meters.
- Statistical results prove that the improvement of accuracy is significant at the 95% confidence level.
- Moderate levels of contamination (approximately 5 percent) produce the best results to balance the purity of the data and sample diversity.
- Removal of outliers not only enhances end performance but also stabilizes model training as well as minimizes convergence time.

In general, the results of this study indicate that the inclusion of a practical outlier detection step before training a model can contribute to a high degree of strength and accuracy of indoor localization systems in IoT settings.

5.10. Discussion

The Isolation Forest application obtained better accuracy on all models tested and minimized localization error. DL models (especially CNN-LSTM) achieved the best compromise between the accuracy and the localization precision at the expense of increased computation. ML models (RF, KNN) are less costly and can be deployed to less resource-intensive systems.

6. Conclusion and Future Work

This paper has provided a solid machine learning-based indoor localization framework in Internet of Things (IoT) systems and given importance to the use of outlier detection to enhance the accuracy of positioning. The proposed method succeeded in detecting and eliminating anomalous RSS samples to distort model training and deteriorate performance, thus making the Isolation Forest algorithm an important part of the data preprocessing pipeline. Several Machine Learning and Deep Learning architectures: a basic classifier like SVM, and the more complex neural architectures like CNN, LSTM, and a hybrid CNN-LSTM architecture, were applied and systematically compared before and after outlier removal.

The findings of the experiment prove that the introduction of an outlier detection step imparts a significant boost to the accuracy of localization, stability, and convergence of the model. In all the models tested, the accuracy was enhanced by 3-6 percent, and the mean localization error decreased by a maximum of 0.6 meters. The CNN-LSTM hybrid model had the most fantastic accuracy (97.1) and the lowest average Error (1.52 meters), which shows the benefit of integrating spatial feature extraction with learning temporal dependencies. The statistical tests proved that the mentioned improvements are significant at the 95% level of confidence and prove the effectiveness of Isolation Forest as a preprocessing component in localization systems.

In addition to the enhanced accuracy, the outlier-sensitive framework has had several other advantages:

- Increased resistance to RSS data noise and environmental variations of the model.
- Quick and more consistent convergence in deep model training.
- Very low computational costs, which make the method appropriate for near-real-time IoT uses.

The study presents the fact that anomaly processing is not a simple data cleaning measure but a crucial step in the creation of reliable localization models in dynamic settings. The outlier detection is added to the learning to bring about a more generalized learning process and easier visualization of the feature space, as indicated by the PCA visualizations and confusion matrix analysis.

6.1. Limitations

Even though the proposed system is exact and reliable, one must admit that it has several limitations:

- The test was done on medium-scale Indoor Wi- Fi data. There is a difference in performance between the large-scale and multi-floor implementation.
- The model currently in place assumes that the number of access points remains constant; the actual IoT systems of the world have variable numbers of devices.
- The existing model presupposes a fixed number of access points; the IoT networks of the real world tend to have dynamically configured devices.
- The Isolation Forest parameters (e.g., contamination fraction) were set manually; an adaptive or self-tuning mechanism is something to be refined.
- Of- framework, To continuous coordinate regression would give higher spatial resolution.

6.2. Future Work

Continuing on the encouraging findings of this study, it is possible to follow several directions in the future:

1) Integration with Real-Time IoT Systems: Interaction with Real-Time IoT Systems: Deploy the proposed model on an edge or fog computing Framework to be able to use it to track the objects inside the building in real-time with minimal latency.
2) Adaptive Outlier Detection: Engineer adaptive outlier detection schemes where outlier detection thresholds are automatically modified according to the environmental dynamics or the outlier detection localization performance feedback.
3) Multimodal Data Fusion: Integrate Wi-Fi RSS signal data with additional signals, e.g., Bluetooth Low Energy (BLE), Ultra-Wideband (UWB), and inertial sensor data to enhance spatial granularity and reliability.

References

[1] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568-2599, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Fei Tony Liu Kai Ming Ting, and Zhi-Hua Zhou, "Isolation Forest," *2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy, pp. 413-422, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[3] Xudong Song et al., "A Novel Convolutional Neural Network Based Indoor Localization Framework with WiFi Fingerprinting," *IEEE Access*, vol. **7**, pp. 110698-110709, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Hui Liu et al., "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067-1080, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[5] Ayan Chatterjee, and Bestoun S. Ahmed, "IoT Anomaly Detection Methods and Applications: A Survey," *Internet of Things*, vol. 19, pp. 1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Pavel Davidson, and Robert Piché, "A Survey of Selected Indoor Positioning Methods for Smartphones," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1347-1370, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[7] Markus Goldstein, and Seiichi Uchida, "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data," *Plos One*, vol. 11, no. 4, pp. 1-31, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[8] Khaldon Azzam Kordi et al., "Survey of Indoor Localization Based on Deep Learning," *Computers, Materials, & Continua*, vol. 79, no. 2, pp. 3261-3298, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Mohsin Munir et al., "DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series," *IEEE Access*, vol. 7, pp. 1991-2005, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Markus M. Breunig et al., "LOF: Identifying Density-based Local Outliers," *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 93-104, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[11] Wentai Wu et al., "Developing an Unsupervised Real-Time Anomaly Detection Scheme for Time Series with Multi-Seasonality," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 9, pp. 4147-4160, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] Sebastian Sadowski, and Petros Spachos, "RSSI-Based Indoor Localization with the Internet of Things," *IEEE Access*, vol. 6, pp. 30149-30161, 2018. [CrossRef] [Google Scholar] [Publisher Link]