Securing WiFi Fingerprint-based Indoor Localization Systems from Malicious Access Points

Fariha Tanjim Shifat, Sayma Sarwar Ela, Mosarrat Jahan
Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh
Email:2018-025-345@student.cse.du.ac.bd,2018-325-351@student.cse.du.ac.bd, mosarratjahan@cse.du.ac.bd

***Abstract*—WiFi fingerprint-based indoor localization schemes deliver highly accurate location data by matching the received signal strength indicator (RSSI) with an offline database using machine learning (ML) or deep learning (DL) models. However, over time, RSSI values degrade due to the malicious behavior of access points (APs), causing low positional accuracy due to RSSI value mismatch with the offline database. Existing literature lacks detection of malicious APs in the online phase and mitigating their effects. This research addresses these limitations and proposes a long-term reliable indoor localization scheme by incorporating malicious AP detection and their effect mitigation techniques. The proposed scheme uses a Light Gradient-Boosting Machine (LGBM) classifier to estimate locations and integrates simple yet efficient techniques to detect malicious APs based on online query data. Subsequently, a mitigation technique is incorporated that updates the offline database and online queries by imputing stable values for malicious APs using LGBM Regressors. Additionally, we introduce a noise addition mechanism in the offline database to capture the dynamic environmental effects. Extensive experimental evaluation shows that the proposed scheme attains a detection accuracy above 95% for each attack type. The mitigation strategy effectively restores the system's performance nearly to its original state when no malicious AP is present. The noise addition module reduces localization errors by nearly 16%. Furthermore, the proposed solution is lightweight, reducing the execution time by approximately 94% compared to the existing methods.**

***Index Terms*—WiFi fingerprint, indoor localization, security, malicious AP**

I. Introduction

With the rapid advancement of wireless communication technologies and the Internet of Things (IoT), location-based services have become indispensable in many application domains. Various localization mechanisms that operate either indoors or outdoors are employed to determine location. In outdoor environments, positions can be accurately determined using the Global Navigation Satellite System (GNSS). However, indoor positioning is challenging due to the attenuation of the Global Positioning System (GPS) signal caused by building structures and their materials [1]. Accurate indoor location tracking is critical for several purposes such as firefighter tracking under hazardous conditions and in emergencies, location-based smart resource management, robotics, shopping mall navigation, healthcare, and virtual reality [1], [2], [3]. Consequently, indoor localization has become a major focus for both academia and industry.

One of the most widely used and preferred techniques for indoor localization is WiFi-based fingerprinting. This method uses existing WiFi networks and does not require additional hardware installation [4]. Existing research focuses mainly on designing a highly accurate WiFi fingerprint-based indoor localization system that minimizes localization errors [5]–[7]. However, the security and privacy issues of indoor localization systems have not been thoroughly explored. The absence of adequate security mechanisms can result in security attacks, such as jamming, spoofing, and sniffing [6], [8]. These attacks ultimately lead to inaccurate location estimation, raising significant concerns about the reliability of localization schemes.

A significant security issue is the presence of malicious access points that generate false RSSI signals [8]. Over time, APs may behave maliciously due to component malfunctions [9], [10] and security attacks such as signal jamming [8], spoofing [6], distance fraud [11], [12], and physical attacks [5]. These malicious APs mislead localization systems and reduce location accuracy. Unreliable location accuracy can have serious consequences in critical situations. For example, if a person needs immediate medical attention in a hospital, inaccurate location prediction of healthcare professionals and facilities can elevate health risks. Similarly, robots navigating in complex indoor environments with unreliable localization methods may cause damage to assets and potentially lead to injuries. Existing literature deals with malicious APs during the initial database construction phase (offline phase) by selecting a subset of feasible APs [5], [6]. However, filtering out malicious APs only in the offline phase does not guarantee the long-term robustness of the indoor localization systems as a reliable AP may become malicious over time. Current literature lacks research on detecting the long-term malicious behavior of APs [10], [13]–[15].

After detecting malicious APs, it is crucial to mitigate their effect to ensure the accurate functioning of the indoor localization system. One approach to achieving this goal is to reconstruct the database. However, each time a malicious AP is detected, conducting a site survey to build the database in a dynamic environment is highly expensive and labor-intensive [13]. Existing literature has explored the use of crowdsourcing techniques [15] for creating and updating the fingerprint database, which significantly reduces human labor. However, this scheme is vulnerable to various security issues [16]. Additionally, robots are utilized for autonomous database construction, which can be expensive and may not perform effectively in a challenging environments [17]. Instead, it would be more beneficial to develop a mechanism that enables

an indoor localization system to function accurately despite the presence of a reasonable percentage of malicious APs, eliminating the need for frequent database reconstruction. This would enhance the system's accuracy while reducing the costs associated with repeated database creation. Although several existing research works mitigate the effect of RSSI value alteration due to environmental factors [18], [15], [13], none of them consider the malicious behavior of APs causing abrupt and dynamic changes to RSSI values.

This paper addresses the above-mentioned shortcomings and proposes lightweight mechanisms for detecting malicious APs over the long term during their operation. Additionally, it introduces a mitigation strategy to manage the impact of malicious APs while maintaining the accuracy of location prediction over an extended period, without requiring frequent database reconstruction.

II. Related works

WiFi RSSI-based indoor localization schemes provide highly accurate location data at low additional cost. However, the location predicted by these models often becomes erroneous due to the alteration of RSSI values caused by malicious AP behaviour [5], [11], [8], changes in layout [15], removal of AP [14], position change of AP [14], [15], and environmental factors [4], [14], [18]. Existing literary works [5], [6], [19], [20] detect and filter malicious APs during the model training phase (offline phase) of indoor localization systems. Here, a subset of optimal AP is selected, and using these selected nonmalicious APs, a fingerprint database is constructed to train the machine learning model. The selection of correlated APs increases localization accuracy and reduces computational complexity. Wang et al. [5] utilized interclass dispersion to calculate AP confidence and selected APs with high confidence. Ye et al. [6] used the Pearson Correlation Coefficient (PCC), where a lower correlation value indicates greater susceptibility of an AP to being malicious. Chen et al. [19] considered the resolution capability to select a set of APs, where APs are chosen based on information gain. Panja et al. [20] used the Binary Particle Swarm Optimization (BPSO) to select a set of suitable APs. In contrast, our proposed scheme uses Spearman's Correlation Coefficient (SRCC) to select correlated APs in the offline phase. We use normalization, correlated AP selection, and noise addition as data preprocessing, where normalization speeds up the computational time, and noise addition makes the system robust to long-term signal variation due to dynamic environmental change.

Detecting malicious APs only in the offline phase is not sufficient for long-term model reliability. Several works explore long-term model robustness challenges in the online phase. For example, Yan et al. [18] proposed a crowdsourcing-based mechanism to handle RSSI signal value alteration in the online phase. This scheme utilizes a denoising autoencoder to address RSSI alteration caused by environmental dynamics. It requires extensive high-quality crowdsourced data for training, and the model is updated frequently, which can be resource-intensive. Li et al. [10] addressed signal variation caused by dynamic environmental change and temporal effects using disagreement-based semi-supervised learning. This scheme struggles to handle sudden and significant alterations in the environment, which can impact localization accuracy. Tiku and Pasricha [4] utilized Siamese neural encoders to address RSSI signal variation without the need for model retraining. This scheme handles signal alteration caused by environmental changes, human movement, and removal or replacement of AP. Significant changes in the environment can affect localization accuracy. Huang et al. [13] addressed location accuracy degradation caused by environmental changes. This scheme utilizes a Gaussian Process Regression to create an offline database and incorporates the Marginalized Particle Extended Gaussian process that uses crowdsourced data to recursively update the database. However, this scheme is susceptible to localization errors due to rapid fluctuations in signal strength, and its effectiveness depends on the quality of the crowdsourced data. Jiang et al. [14] addressed RSSI signal alteration caused by the change in the number of AP and environmental factors. The authors proposed a feature adaptive extreme learning machine-based approach that can adapt to a changing number of features. This scheme faces difficulties to cope with sudden environmental changes and runs the risk of overfitting to recent signal variations. Yang et al. [15] proposed the Altered AP Identification and Fingerprint Updating (AAIFU) scheme, which detects altered APs in the online phase by considering changes in RSSI values due to positional changes of APs. This scheme uses a Gradient Boosting Decision Tree (GBDT) regression model to identify the positional changes of APs. It then updates the database by replacing the outdated RSSI values with fresh RSSI values obtained through GBDT regression. However, the effectiveness of this scheme depends on a large volume of crowdsourced data. Moreover, positional changes of an AP only create persistent changes in RSSI values. Among these works, Li et al. [10], Huang et al. [13], Jiang et al. [14], and Yang et al. [15] update database to keep the fingerprints fresh. On the other hand, none of the schemes except Yang et al. [15], identify the causes of RSSI signal alteration. In contrast to the existing works, our proposed scheme addresses the RSSI value alteration caused by malicious AP behavior, characterized by abrupt and random changes in RSSI values. Our proposed scheme detects malicious APs in real time by analyzing data from various location queries. Additionally, it utilizes the LGBM regression model [21] to update the existing database, which mitigates the effect of malicious APs on the performance of the indoor localization scheme. Here to note that the proposed scheme handles both static and dynamic changes in RSSI values.

III. System Model

As shown in Fig. 1, the proposed scheme augments the existing WiFi fingerprint-based indoor localization system [22] by including a *malicious effect mitigation module* and *online database*. The proposed system model comprises q access points and r reference points (RPs). The *offline database* com-

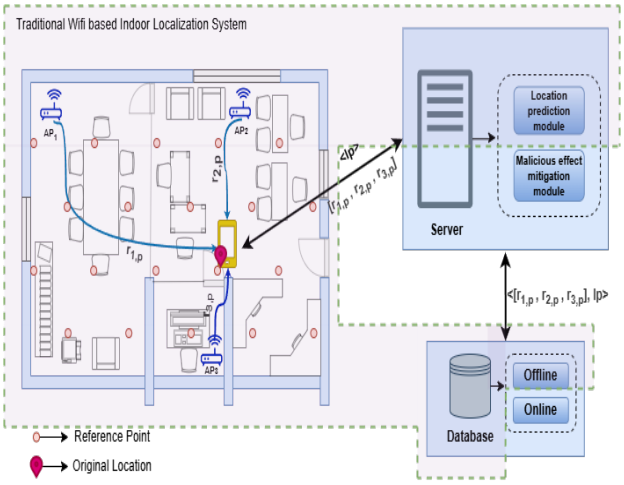


Fig. 1. System model.

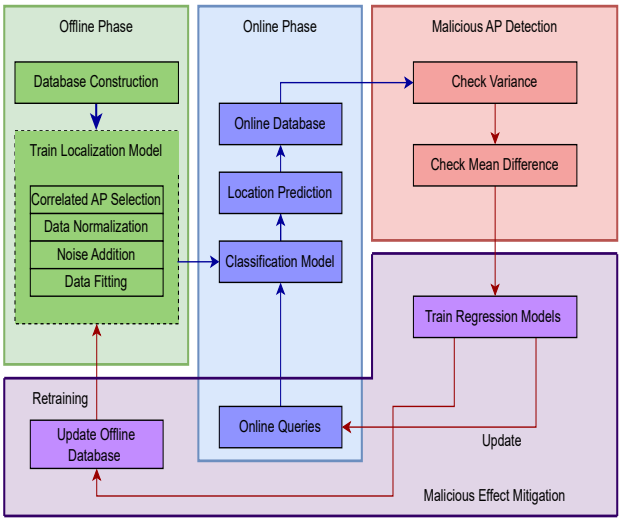


Fig. 2. Overview of the proposed scheme.

prises a set of records of form $[RSSI_j, l_j]$, where $RSSI_j = [r_{i,j}]$ is the RSSI vector and $r_{i,j}$ is the RSSI value of ith AP measured at jth RP, and l_j is the location of the jth RP, $i \in 1, 2, \ldots, q$, and $j \in 1, 2, \ldots, r$. In the *online* phase, a device measures a fingerprint vector $[RSSI]$ with respect to its location p and sends it to the *location prediction module*. The *location prediction module* matches $[RSSI]$ with the records stored in the *offline database* using either a ML or DL model and returns the estimated location l_p to the mobile device. The *online database* is constructed by saving the results of online queries along with associated RSSI vectors and used to detect malicious APs in the *online* phase. The *malicious effect mitigation module* detects malicious APs and takes measures to diminish their effect. The proposed system model subjects to false fingerprints both in *online* and *offline* phases due to various environmental factors [4], [14], temporal effects [6], malicious behavior of APs [5], [6]. We assume that the *offline and online databases*, the *location prediction module*, and the *malicious effect mitigation module* reside in a trusted server.

IV. Proposed Scheme

As shown in Fig. 2, the proposed scheme consists of four modules: *offline phase*, *online phase*, *malicious AP detection*, and *malicious effect mitigation*. The notations used to describe the proposed scheme are listed in Table I.

TABLE I
List of Notations

Notation	Description
$r_{i,j}$	RSSI value of ith AP at jth RP
l_j	Location of the jth RP
$RSSI_j$	A vector of form $[r_{i,j}]$ for jth RP and $i \in 1, 2, \ldots, q$
$RSSI'_j$	An offline data sample of form $[r'_{i,j}]$ for jth RP and $i \in 1, 2, \ldots, n$ and $n \leq q$
$[RSSI'_j, l_j]$	A labeled sample representing location l_j for $RSSI'_j$ in offline database
$RSSI''_p$	Online query of form $[r''_{i,p}]$ for pth RP where p is unknown
$[RSSI''_p, l_p]$	A labled sample representing location l_p for $RSSI''_p$ in online database

A. Offline Phase

1) Database Construction: At first, a reliable database is created using site survey [23]. This mechanism involves a person with multiple reliable mobile devices visiting every RP. It collects several labeled samples in the form $[RSSI_j, l_j]$, where $RSSI_j$ is a RSSI fingerprint vector computed at jth RP and l_j is the location of the jth RP. The raw database created by site survey is a collection of records $[RSSI_j, l_j]$.

2) Correlated AP Selection: The proposed scheme selects a set of reliable APs from the available ones in the environment using the Spearman's Rank Correlation Coefficient (SRCC) [24] method with a threshold value set to 0.1 empirically, which is optimal for the considered UJI's dataset [25]. Suppose the number of selected correlated APs is n. The proposed scheme constructs an offline database $\{[RSSI'_j, l_j]\}$ from the raw database constructed via site survey, where $RSSI'_j$ contains RSSI entries only for the selected n APs.

3) Data Normalization: Training data in the offline database is converted in [0,1] using Eq. 1 [26].

$$r'_{i,j} = \begin{cases} 0, & \text{if } r'_{i,j} = no\_signal \\ \left(\frac{r'_{i,j} - \min}{\max - \min}\right) \times (1 - 0.25) + 0.25, & \text{if } r'_{i,j} \neq no\_signal \end{cases} \quad (1)$$

where, $r'_{i,j} \in RSSI'_j$ is the RSSI value of ith AP measured at the jth AP. Eq. 1 replaces no signal RSSI values to 0 and all other RSSI values in the range [0.25, 1]. Here, min = -100 and max = 0 [25].

4) Noise Addition: Our scheme randomly selects 10% of the samples per RP and creates a Gaussian distribution curve with a standard deviation of 0.5. Random values taken from this distribution are added to the actual RSSI values as noise.

5) Data Fitting/Model Training: Data is fitted to the *LGBM* classifier model [21] built with tuned hyperparameters.

B. Online Phase

The proposed scheme receives online queries of form $RSSI''_p = [r''_{1,p}, r''_{2,p}, r''_{3,p}, \ldots, r''_{n-1,p}, r''_{n,p}]$ from a user who

wants to know their location, where $r''_{i,p}$ is the RSSI value of the ith AP from the user's position p, and p is unknown. The localization model predicts the location l''_p (the position of the closest RP to the user's position p) for incoming queries and saves them to create an online database, a collection of labeled records $[RSSI''_p, l''_p]$ corresponding to the online queries.

C. Malicious AP Detection

The proposed scheme separates online and offline data samples based on reference points and conducts two tests on the RSSI values of every AP. Let,

A_j = data samples $RSSI''_j$ for RP j in the online database

$A_{i,j}$ = collection of $r''_{i,j}$ for an AP i where $r''_{i,j} \in RSSI''_j$

B_j = data samples $RSSI'_j$ for RP j in the offline database

$B_{i,j}$ = collection of $r'_{i,j}$ for an AP i where $r'_{i,j} \in RSSI'_j$

1) Check Variance: It checks the variance of an AP for every RP in the online database. If the maximum variance value of an AP equals or exceeds the variance threshold, the proposed scheme marks that AP as malicious. The variance threshold TH_1 is set to 0.05, as nearly 99% of the variance for each AP in the offline database is within 0.05 for the UJI dataset [25]. Let,

$t = |A_{i,j}|$ = number of entries in $A_{i,j}$

$\mu = \frac{1}{t}\sum r''_{i,j}$ = mean of RSSI values of an AP i where $r''_{i,j} \in A_{i,j}$

$\sigma^2_{i,j} = \frac{1}{t}\sum(r''_{i,j} - \mu)^2$ = variance of an AP i for RP j

An AP i is malicious if $\max(\sigma^2_{i,j}) \geq TH_1$, where, $j \in 1 \ldots r$

2) Check Mean Difference: It checks the difference in the mean value of offline and online data for a specific AP for every RP. If the minimum value of the differences exceeds a defined threshold value TH_2, the AP is marked as malicious. We empirically set TH_2 to 0.005 as it detects nearly 98% malicious APs for the UJI dataset [25]. Let,

$u = |B_{i,j}|$ = number of entries in $B_{i,j}$

$v = |A_{i,j}|$ = number of entries in $A_{i,j}$

$\frac{1}{u}\sum r'_{i,j}$ = mean of RSSI values of an AP i where $r'_{i,j} \in B_{i,j}$

$\frac{1}{v}\sum r''_{i,j}$ = mean of RSSI values of an AP i where $r''_{i,j} \in A_{i,j}$

$\delta_{i,j} = (\frac{1}{u}\sum r'_{i,j} - \frac{1}{v}\sum r''_{i,j})$ = difference of mean values.

An AP i is malicious if $min(\delta_{i,j}) > TH_2$, where $j \in 1 \ldots r$

We considered 1560 data samples from the first month of the UJI dataset [25] to determine TH_1 and TH_2.

D. Malicious Effect Mitigation

1) Train Regression Model: Suppose the malicious AP detection module identifies p malicious APs, leading to m = n - p APs behaving honestly. An online query for ith RP can be rewritten as $RSSI''_i = [r''_{1,i}, r''_{2,i}, \ldots, r''_{m-1,i}, r''_{m,i}, r''_{m+1,i}, \ldots, r''_{m+p-1,i}, r''_{m+p,i}]$ where $r''_{m+1,i} \sim r''_{m+p,i}$ are RSSI entries for malicious APs. The proposed scheme deploys p LGBMRegressors [21] and trains them with offline database's reliable data $RSSI'_i$ of form $[r'_{1,i}, r'_{2,i}, \ldots, r'_{m-1,i}, r'_{m,i}]$ (contains the RSSI entries for m honest APs), where $i \in 1, 2, \ldots r$.

TABLE II
PARAMETER USED IN PERFORMANCE METRICS

Parameter	Description
True Positive (TP)	Correctly predicted positive/malicious instances
True Negative (TN)	Correctly predicted negative/nonmalicious instances
False Positive (FP)	Incorrectly predicted positive/malicious instances
False Negative (FN)	Incorrectly predicted negative/nonmalicious instances

TABLE III
EVALUATION METRICS

False Positive Rate (FPR)	Ratio of legitimate APs falsely identified as malicious	$FPR = \frac{FP}{FP+TN}$
False Negative Rate (FNR)	Ratio of malicious APs falsely identified as legitimate	$FNR = \frac{FN}{FN+TP}$
Precision	Ratio of correctly detected malicious APs	$Precision = \frac{TP}{TP+FP}$
Recall	Ratio of correctly identified malicious APs to the total actual malicious APs	$Recall = \frac{TP}{TP+FN}$
Accuracy	Ratio of correctly identified malicious APs and legal APs	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
F1 Score	Relationship between precision and recall	$F1 = 2 \times \frac{precision \times recall}{precision+recall}$
Mean Localization Error (MLE)	Deviation of the predicted location from the actual location	$MLE = \sqrt{(x-\bar{x})^2 + (y-\bar{y})^2}$

2) Update Online Queries and Offline Database: The LGBMRegressors predicts p RSSI values when input data is $RSSI'_i$ of form $[r'_{1,i}, r'_{2,i}, \ldots, r'_{m-1,i}, r'_{m,i}]$ and use these data to replace the values of malicious APs $r'_{m+1,i}, \ldots, r'_{m+p-1,i}$, $r'_{m+p,i}$ in the offline $RSSI'_i$ record. Similarly, when the input data is $RSSI''_i$ of form $[r''_{1,i}, r''_{2,i}, \ldots, r''_{m-1,i}, r''_{m,i}]$, the predicted p RSSI values replace $r''_{m+1,i}, \ldots, r''_{m+p-1,i}$, $r''_{m+p,i}$ in the online query $RSSI''_i$. The localization model is retrained with the updated offline database. The updated query $RSSI''_i$ is used by the classification model to predict the location. The process of predicting p RSSI values is inspired by Montoliu et al. [27].

V. EXPERIMENTAL EVALUATION

We implemented the proposed scheme using Python and evaluated its performance using the UJI's dataset [25]. The UJI's dataset [25] is the longest public dataset with 103584 WiFi fingerprints. We used only the data of floor 3 for brevity, as also done in [4]. This data set contains 4320 data samples for training purposes for floor 3 [28]. We incorporated noise-added samples in this set and extended its size to 4800. Initially, there were 620 APs in the environment. However, we took only 40 APs after correlated AP selection. We trained our localization model which is a LGBMClassifier [21], with the offline dataset. We used seven performance metrics shown in Table III to evaluate the performance. The parameters used in defining performance metrics are shown in Table II.

A. Impact of Noise Addition

Figure 3 shows mean localization errors over the data of 25 months from UJI's dataset [25]. The curve labeled *SRCC (0.1)* shows the effect of correlated AP selection only, the curve labeled *SRCC (0.1) + Normalized* shows the impact of AP selection and normalization, and the curve labeled

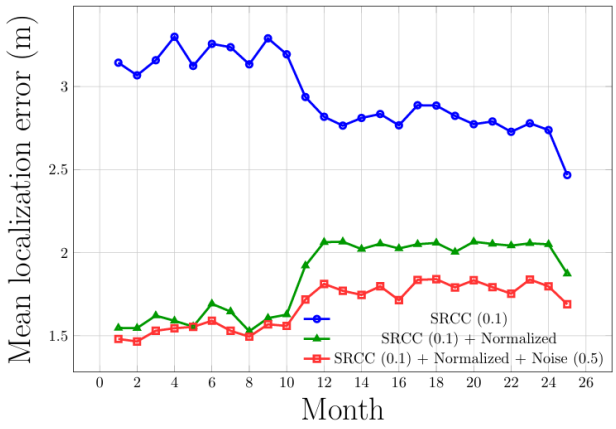


Fig. 3. Impact of noise addition.

TABLE IV
ATTACK DESCRIPTION AND ACRONYMS

Attack Type	Acronym	Description
Constant Value	**CV**	A constant RSSI value is set for the malicious AP. Here, the constant value is 100, interpreted as no signal from the AP. It also represents that an AP is removed.
Random Value	**RV**	Random RSSI values are assigned for APs.
Actual RSSI Plus Constant Offset	**ARCO**	A fixed offset ([-100,0]) is added to the actual RSSI value for malicious APs. This also represents the movement of AP from its original position.
Actual RSSI Plus Random Offset	**ARRO**	A random offset ([-100,0]) is taken and added to the actual RSSI value.

SRCC (0.1) + Normalized + Noise (0.5) shows the effect of AP selection, normalization, and noise addition. Figure 3 shows a notable reduction in localization error for the curve *SRCC (0.1) + Normalized + Noise (0.5)* with respect to *SRCC (0.1) + Normalized* after the 10th month because the model becomes robust to RSSI fluctuations in the long term. The MLE is reduced by nearly 16% in the 16th month, which is the maximum change of the red curve with respect to the green one. Normalizing data helps models converge faster and adding noise makes the model more robust against long-term noise. Correlated AP selection, noise addition, and data normalization bring the MLE of the proposed scheme below 2 meters, outperforming existing machine learning schemes [4].

B. Malicious AP Detection

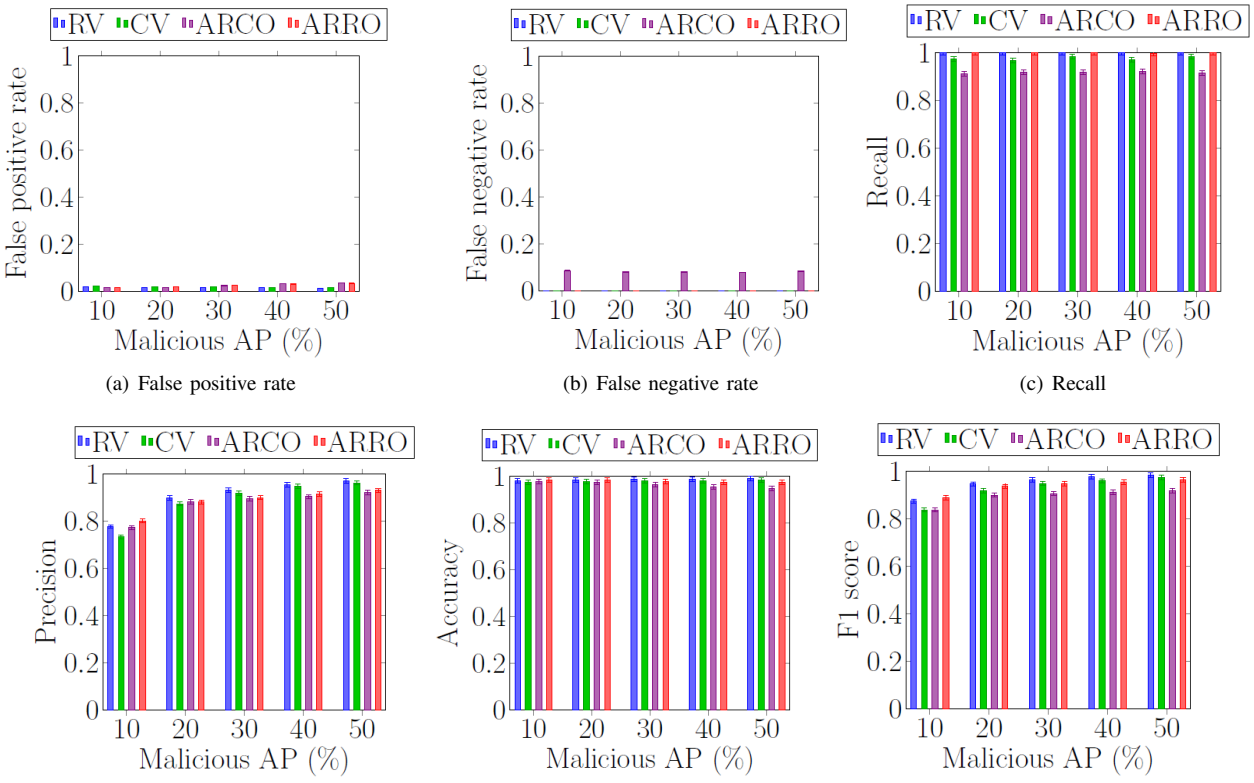
We considered four malicious behaviors of an AP following the studies of [18], [15], [7], [29]: *constant RSSI value*, *random RSSI value*, *constant offset alteration of the actual RSSI value*, and *random offset alteration of the actual RSSI value* as shown in Table IV. We evaluated the malicious AP detection capability of the proposed scheme using test data comprising 1560 samples from the first month's data (floor 3). We chose this data because the first month's data is less susceptible to errors caused by undetected malicious APs and environmental changes. For each attack type, each experiment was iterated 100 times with malicious APs in the range $10\% \sim 50\%$. Each data point in the result was averaged over 100 iterations.

1) Performance Analysis of the Proposed Scheme: Figure 4 shows the performance of the proposed detection module.

1) *False Positive Rate (FPR):* Figure 4(a) shows that the FPR consistently remains below 4%. The legitimate APs sometimes exhibit signal variation due to temporal effect and are marked as malicious by the detection system.
2) *False Negative Rate (FNR):* Figure 4(b) shows that except for the ARCO attack, all other attacks show a negligible FNR. As marking an AP as malicious in our experiment was a random process, in very few cases, the selected malicious AP was assigned the same original RSSI value, leading to a false negative outcome. In ARCO attacks, FNR stays nearly 10%. In this attack, we add or subtract a fixed value and also add an offset in the noise addition module, leading to very little change in the mean that can not be distinguished with the mean difference checker, resulting in an increased FNR.
3) *Recall:* The recall is consistently 100% for RV and ARRO attacks, almost 98% for CV attacks, and nearly 92% for ARCO attacks, as shown in Fig. 4(c). In ARCO attacks, the recall is lower for the higher FNR than other attacks.
4) *Precision:* Figure 4(d) shows that the detection rate increases with escalating malicious APs for all attack types. The proposed scheme achieves a satisfactory precision of nearly 98%, 97%, 95%, and 96% for the RV, CV, ARCO, and ARRO attacks, respectively, for 50% malicious APs.
5) *Accuracy:* Figure 4(e) shows that the accuracy of the detection module consistently remains above 95% for all types of attack, underscoring the robustness and reliability of the detection system across various scenarios.
6) *F1 Score:* Figure 4(f) shows that the F1 score increases with malicious APs for all attack types. The proposed scheme exhibits an F1 score within 84%∼99%.

2) Comparative Analysis of AAIFU and Proposed Schemes: In this section, we compare the performance of the AAIFU [15] and proposed schemes for 50% malicious APs.

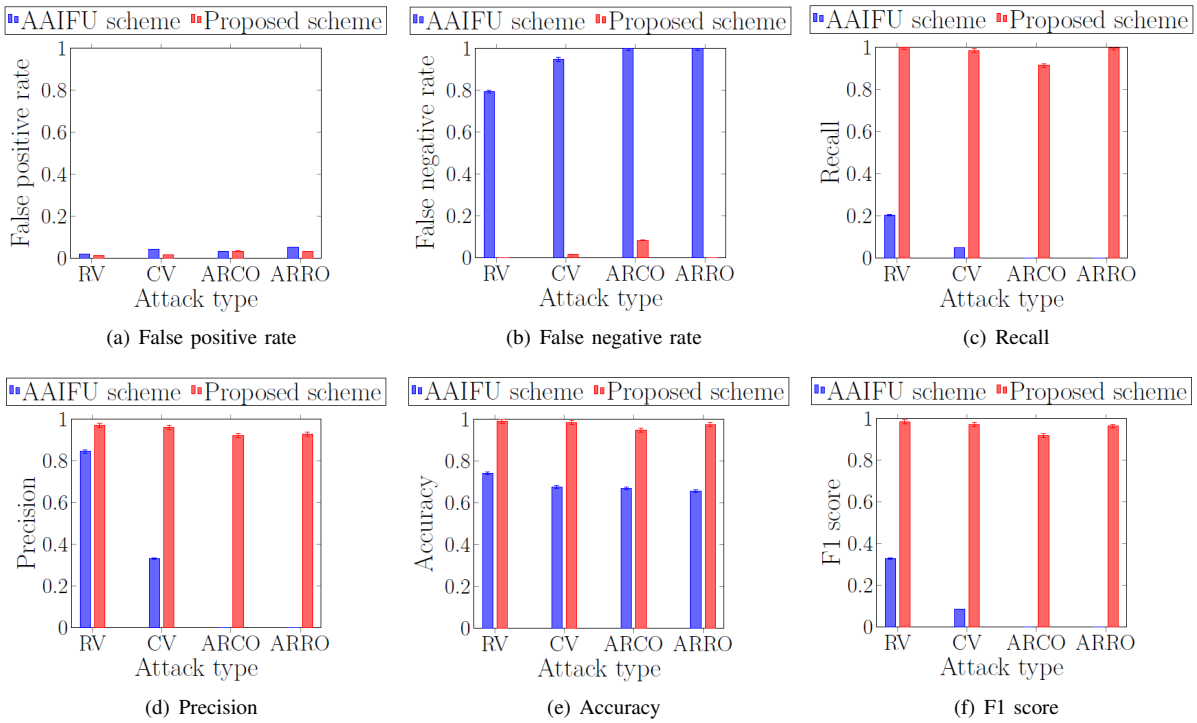
1) *False positive rate (FPR):* Figure 5(a) shows that the FPR ranges between $2\% \sim 6\%$ and $2\% \sim 4\%$ for AAIFU and proposed schemes, respectively. Due to the temporal effect both schemes exhibit FPR.
2) *False negative rate (FNR):* Figure 5(b) shows that the FNR in the AAIFU scheme is dramatically higher, ranging between 80%∼100%. The proposed scheme never misses any malicious AP in RV and ARRO attacks. It shows nearly 9% and 2% for ARCO and CV attacks, respectively. In the AAIFU scheme, FNR is visible when the alarm frequency of altered APs is lower than that of unaltered APs. From the alarm frequency distribution, the AAIFU scheme uses a clustering method to find the actual altered APs, ignoring the altered (malicious) APs with low alert frequency, causing a significant FNR.
3) *Recall:* The AAIFU scheme exhibits a recall of nearly 0% for ARRO and ARCO attacks, visible from Fig. 5(c). For RV and CV attacks, the AAIFU scheme shows lower recall values of around 21% and 5%, respectively. On



(a) False positive rate (b) False negative rate (c) Recall

(d) Precision (e) Accuracy (f) F1 score

Fig. 4. Performance analysis of the proposed scheme.



(a) False positive rate (b) False negative rate (c) Recall

(d) Precision (e) Accuracy (f) F1 score

Fig. 5. Performance comparison between AAIFU and proposed schemes.

the other hand, the proposed scheme exhibits a recall of 100% for RV and ARRO attacks and always maintains a recall $\geq$ 92% for other attacks. The ineffective clustering of the AAIFU scheme lead to higher FNR and 0 TPs.

4) Precision: As shown in Fig 5(d), the AAIFU scheme shows a precision of nearly 0% for ARRO and ARCO attacks due to 0% TP for nearly 100% FNR. It displays a precision of approximately 85% and 34% for RV and CV attacks, respectively. In contrast, the proposed scheme shows precision values $\geq$ 93% for all attack types.
5) Accuracy: Figure 5(e) shows that the highest accuracy for the AAIFU and proposed schemes are approximately 75% and 99%, respectively. Despite the lower TP, the AAIFU scheme exhibits a higher TN value, resulting in greater accuracy than recall and precision.
6) F1 score: The AAIFU scheme exhibits an F1 score of 0% for ARCO and ARRO attacks and approximately 33% and 9% F1 scores for RV and CV attacks, respectively, visible from Fig 5(f). On the contrary, the proposed scheme maintains a good balance between precision and recall, with an F1 score $\geq$ 92% for all attack types. The 0 in precision and recall results in 0 in the F1 score for the AAIFU scheme.

C. Impact of Malicious Effect Mitigation

Five Cumulative Distribution Function (CDF) curves in Fig. 6 show the localization error using 8112 samples for 25 months from the UJI dataset [25] for the following scenarios:

1) *Without malicious AP* – shows the MLE without the presence of malicious AP. Here, the malicious AP detection and effect mitigation modules were not deployed.
2) *50% malicious AP* – presents the MLE when 50% of the APs are malicious. Here, malicious AP detection and effect mitigation modules were not used.
3) *AAIFU Scheme* – shows the MLE for the AAIFU scheme [15], where 50% APs were malicious.
4) *Proposed Scheme without offline database updation and model retraining* – displays the MLE for the proposed scheme where a malicious AP detection module was used and online queries were updated. Here, the offline database was not updated, the model was not retrained, and 50% of the APs were malicious.
5) *Proposed scheme with offline database updation and model retraining* – shows the MLE for the proposed scheme where the malicious AP detection and effect mitigation modules were used. The offline database was updated, the model was retrained, and 50% of the APs were malicious.

Figure. 6 shows that the curve for 50% malicious AP exhibits significant degradation from the curve with no malicious AP across all types of attacks. The proposed scheme, which includes database update and model retraining, shows an almost unnoticeable deviation from the curve with no malicious AP for every type of attack. However, the proposed scheme that does not include the database updation and model retraining exhibits degraded performance compared to the proposed scheme with the database updation curve. By updating the database and retraining the model after identifying malicious APs, the localization error is minimized. The AAIFU scheme [15] focuses on the steady change in the behavior of APs rather than the dynamic malicious effects during the online phase. This scheme produces results similar to our proposed scheme without requiring the database update, as shown in Fig. 6. The CDF curve for the AAIFU scheme is smoother for the use of a regressor and the CDF curve for the proposed scheme appears more like a staircase for the use of a classifier.

It is observed from experiments that the proposed scheme effectively handles at most 60% malicious APs. After that the performance deviates to a greater extent.

D. Execution Time

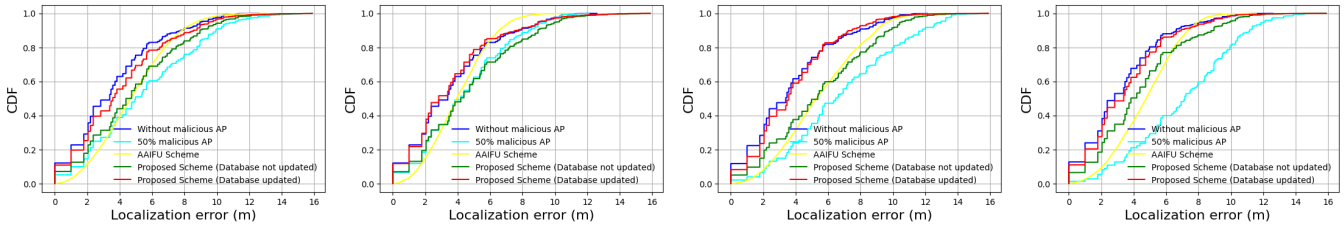
The proposed scheme exhibits a gradual increase in runtime with the increasing number of malicious APs as shown in Table V. The AAIFU scheme randomly detects malicious APs, which is displayed by the non-gradual increase of runtime. The AAIFU scheme spends a significant amount of time in the detection phase than the proposed scheme as the proposed scheme uses only some simple statistical calculations whereas the AAIFU scheme uses GBDTRegressors. The AAIFU scheme detects fewer APs than the proposed scheme. Hence, even though both schemes use regressors for updating database, the proposed scheme takes more time than the AAIFU scheme. The proposed scheme takes more time than the AAIFU scheme in model retraining, indicating that training a classifier is more time-consuming than a regressor. Lastly, in the inference phase, the proposed scheme involves regressors to predict the values of malicious APs in online queries, resulting in greater time consumption compared to the AAIFU scheme. The overall time consumed by the proposed scheme is approximately 94% less than the AAIFU scheme.

VI. Conclusion

We have proposed a reliable and efficient long-term indoor localization system by incorporating malicious AP detection and effect mitigation modules. Experimental results demonstrate that the proposed scheme achieves detection accuracy above 95% for all attack types. Besides, the malicious effect mitigation module ensures the accurate functioning of the indoor localization system by restoring the system's performance to the initial state where the system operates without any malicious APs. In addition, the proposed indoor localization system cuts down the execution time by approximately 94% compared to existing works. We believe the proposed scheme enhances the existing literature by offering a faster and more reliable indoor localization system. We plan to extend this work by considering device diversity and attack scenarios for online and offline databases.

References

[1] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Commun. Surv. Tut.*, vol. 19, no. 2, pp. 1327–1346, 2017.



(a) RV (b) CV (c) ARCO (d) ARRO

Fig. 6. CDF curves of localization error for different attacks.

TABLE V
EXECUTION TIME (SEC.) OF THE AAIFU AND PROPOSED SCHEMES FOR 10% $\sim$ 50% MALICIOUS AP

Step	AAIFU scheme					Proposed scheme				
	10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
Detection	599.69	604.14	575.38	602.94	554.47	1.43	1.39	2.19	1.77	1.32
Database updation	3.46	1.76	1.92	0.72	0.95	8.38	6.85	13.84	13.13	17.30
Model retraining	0.91	0.67	1.05	0.77	0.65	8.56	12.19	9.50	10.34	10.67
Inference	0.05	0.05	0.07	0.08	0.04	9.05	9.13	9.07	9.38	9.49
Total	604.11	606.61	578.42	604.51	556.12	27.42	29.56	34.61	34.62	38.78

[2] Y. Hu, F. Qian, Z. Yin, Z. Li, Z. Ji, Y. Han, Q. Xu, and W. Jiang, "Experience: Practical indoor localization for malls," in *Proc. 28th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom'22)*, p. 82–93, 2022.

[3] J. Shin, G. An, J.-S. Park, S. J. Baek, and K. Lee, "Application of precise indoor position tracking to immersive virtual reality with translational movement support," *Multimedia Tools Appl.*, p. 12331–12350, 2016.

[4] S. Tiku and S. Pasricha, "Siamese neural encoders for long-term indoor localization with mobile devices," in *Proc. Des. Automat. Test in Europe Conf. Exhib. (DATE)*, pp. 1215–1220, 2022.

[5] C. Wang, J. Luo, X. Liu, and X. He, "Secure and reliable indoor localization based on multitask collaborative learning for large-scale buildings," *IEEE Internet of Things J.*, vol. 9, no. 22, pp. 22291–22303, 2021.

[6] Q. Ye, X. Fan, H. Bie, D. Puthal, T. Wu, X. Song, and G. Fang, "SE-Loc: security-enhanced indoor localization with semi-supervised deep learning," *IEEE Trans. Netw. Sci. Eng.*, 2022.

[7] X. Sun, H. Ai, J. Tao, T. Hu, and Y. Cheng, "BERT-ADLOC: A secure crowdsourced indoor localization system based on ble fingerprints," *Appl. Soft Comput.*, vol. 104, p. 107237, 2021.

[8] S. Tiku and S. Pasricha, "Overcoming security vulnerabilities in deep learning–based indoor localization frameworks on mobile devices," *ACM Trans. Embed. Comput. Syst.*, vol. 18, nov 2019.

[9] J. Yin, Q. Yang, and L. Ni, "Adaptive temporal radio maps for indoor location estimation," in *Proc. Third IEEE Int. Conf. Pervasive Comput. Commun.*, pp. 85–94, 2005.

[10] D. Li, J. Xu, Z. Yang, and C. Tang, "Train once, locate anytime for anyone: Adversarial learning-based wireless localization," *ACM Trans. Sensor Netw.*, vol. 20, no. 2, pp. 1–21, 2024.

[11] B. Mukhopadhyay, S. Srirangarajan, and S. Kar, "RSSI-based secure localization in the presence of malicious nodes in sensor networks," *arXiv preprint arXiv:1912.06362*, 2019.

[12] A. González-Tablas Ferreres, B. Álvarez, and A. Ribagorda, "Guaranteeing the authenticity of location information," *IEEE Pervasive Computing*, vol. 7, no. 3, pp. 72–80, 2008.

[13] B. Huang, Z. Xu, B. Jia, and G. Mao, "An online radio map update scheme for WiFi fingerprint-based localization," *IEEE Internet of Things J.*, vol. 6, no. 4, pp. 6909–6918, 2019.

[14] X. Jiang, J. Liu, Y. Chen, D. Liu, Y. Gu, and Z. Chen, "Feature adaptive online sequential extreme learning machine for lifelong indoor localization," *Neural Comput. Appl.*, vol. 27, pp. 215–225, Jan 2016.

[15] J. Yang, X. Zhao, and Z. Li, "Crowdsourcing indoor positioning by light-weight automatic fingerprint updating via ensemble learning," *IEEE Access*, vol. 7, pp. 26255–26267, 2019.

[16] T. Li, Y. Chen, R. Zhang, Y. Zhang, and T. Hedgpeth, "Secure crowd-sourced indoor positioning systems," in *Proc. IEEE Conf. Comput. Commun.*, pp. 1034–1042, 2018.

[17] H. Zou, C.-L. Chen, M. Li, J. Yang, Y. Zhou, L. Xie, and C. J. Spanos, "Adversarial learning-enabled automatic wifi indoor radio map construction and adaptation with mobile robot," *IEEE Internet of Things J.*, vol. 7, no. 8, pp. 6946–6954, 2020.

[18] M. Yan, J. Wang, and Z. Zhao, "Online detection of Wi-Fi fingerprint alteration strength via deep learning," in *IEEE 45th Int. Conf. Local Comput. Netw. (LCN)*, pp. 321–324, 2020.

[19] Y. Chen, W. Liu, H. Zhao, S. Cao, S. Fu, and D. Jiang, "Bisecting k-means based fingerprint indoor localization," *Wireless Netw.*, vol. 27, pp. 3497–3506, 2021.

[20] A. K. Panja, S. F. Karim, S. Neogy, and C. Chowdhury, "A novel feature based ensemble learning model for indoor localization of smartphone users," *Eng. Appl. Artif. Intell.*, vol. 107, p. 104538, 2022.

[21] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. 31st Int. Conf. Neural Inf. Processing Syst. (NIPS'17)*, p. 3149–3157, 2017.

[22] G. Félix, M. Siller, and E. N. Álvarez, "A fingerprinting indoor localization algorithm based deep learning," in *Proc. 8th Int. Conf. Ubiquitous Future Netw. (ICUFN'2016)*, pp. 1006–1011, 2016.

[23] P. Bahl and V. Padmanabhan, "RADAR: an in-building rf-based user location and tracking system," in *Proc. IEEE INFOCOM 2000. Conf. Comput. Commun. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc. (Cat. No.00CH37064)*, vol. 2, pp. 775–784, 2000.

[24] C. Spearman, "The proof and measurement of association between two things," *The American J. Psychol.*, vol. 100, no. 3-4, pp. 441–471, 1987.

[25] G. M. Mendoza-Silva, P. Richter, J. Torres-Sospedra, E. S. Lohan, and J. Huerta, "Long-Term Wi-Fi fingerprinting dataset and supporting material (2.2)," Apr. 2020. Zenodo repository, https://doi.org/10.5281/zenodo.3748719.

[26] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl. Soft Comput.*, vol. 97, p. 105524, 2020.

[27] R. Montoliu, E. Sansano, O. Belmonte, and J. Torres-Sospedra, "A new methodology for long-term maintenance of WiFi fingerprinting radio maps," in *Porc. Int. Conf. Indoor Positioning and Indoor Navigation (IPIN'18)*, pp. 1–7, 2018.

[28] G. M. Mendoza-Silva, P. Richter, J. Torres-Sospedra, E. S. Lohan, and J. Huerta, "Long-term WiFi fingerprinting dataset for research on robust indoor positioning," *Data*, vol. 3, no. 1, p. Article 3, 2018.

[29] J. Kamel, M. Wolf, R. W. van der Hei, A. Kaiser, P. Urien, and F. Kargl, "VeReMi Extension: A dataset for comparable evaluation of misbehavior detection in VANETs," in *Proc. IEEE Int. Conf. Commun. (ICC'20)*, pp. 1–6, 2020.