IEEE ComSoc | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received 4 December 2022; revised 17 April 2023; accepted 12 September 2023.
Date of publication 18 September 2023; date of current version 10 October 2023.

The associate editor coordinating the review of this article and approving it for publication was J. J. P. C. Rodrigues.

Digital Object Identifier 10.1109/TMLCN.2023.3316150

3D Radio Map-Based GPS Spoofing Detection and Mitigation for Cellular-Connected UAVs

YONGCHAO DANG, ALP KARAKOC, SABA NORSHAHIDA, AND RIKU JÄNTTI (Senior Member, IEEE)

Department of Information and Communications Engineering, Aalto University, 00076 Espoo, Finland

Corresponding author: Y. Dang (yongchao.dang@aalto.fi)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Program through the 5G!Drones Project under Grant 857031, in part by the Academy of Finland through the ULTRA Project under Grant 328215, and in part by the Academy of Finland through the Backscatter Enabled Sustainable Monitoring Infrastructure for Assisted Living (BESIMAL) Project under Grant 334197.

ABSTRACT With the upcoming 5G and beyond wireless communication system, cellular-connected Unmanned Aerial Vehicles (UAVs) are emerging as a new pattern to give assistance for target searching, emergency rescue, and network recovery. Such cellular-connected UAV systems highly rely on accurate and secure navigation systems, e.g. the Globe Navigation System (GPS). However, civil GPS services are unencrypted and vulnerable to spoofing attacks that can manipulate UAVs' location and abort the UAVs' mission. This paper leverage 3D radio map and machine learning methods to detect and mitigate GPS spoofing attacks for cellular-connected UAVs. Precisely, the edge UAV flight controller uses ray tracing tools deterministic channel models, and Kriging methods to construct a theoretical 3D radio map. Then the machine learning methods, such as Multi-Layer Perceptrons (MLP), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN), are employed to detect GPS spoofing by analyzing the UAV/base station reported Received Signal Strength (RSS) values and the theoretical radio map RSS values. Once spoofing is detected, the particle filter is applied to relocate the UAV and mitigate GPS deviation. The experiment results indicate that the Universal Kriging (UK) with exponential covariance function has the lowest standard errors for radio map construction. Moreover, the MLP achieves the highest spoofing detection accuracy with different spoofing margins because of the statistic prepossessing relieving environmental impacts, while the CNN has a comparable detection accuracy with less training time than MLP since CNN inputs are raw RSS data. Furthermore, the particle filter-based GPS spoofing mitigation can relocate the UAV to the real position within an error of 10 meters using 100 particles.

INDEX TERMS Unmanned aerial vehicles (UAV), GPS spoofing, radio map, Kriging, machine learning.

I. INTRODUCTION

CELLULAR-CONNECTED Unmanned Aerial Vehicles (UAVs) are emerging as an integrated part of the upcoming 5G and beyond mobile wireless networks due to their mobility and flexibility [1]. As a flying platform, cellular-connected UAVs can provide diverse civilian applications efficiently and economically with the assistance of a safe and secure navigation system, such as material transportation, building inspection, and emergency communication [2]. However, as with any new technology, security concerns must be addressed to ensure that UAVs are not used for malicious purposes [3], [4]. As a global satellite navigation system, the Globe Navigation System (GPS), is mainly used by UAVs for navigation but it is unencrypted and vulnerable to spoofing attacks. In fact, the attacker can use the low-cost Software Defined Radio (SDR) tools to generate fake GPS messages and mislead the GPS receiver to compute the wrong position. In addition, even with encryption protection, the attacker can also collect the raw GPS signals and replayed these signals with a time delay on a higher power, resulting in consistent deviations in all pseudo-range calculations [5]. Indeed, GPS spoofing attacks deviate UAVs from their planned trajectories

and lead to undesirable events such as collisions among UAVs. Besides, the spoofed GPS positions also increase the risk of public safety for UAVs breaking the no-flying zone restriction. Thus, it is necessary to bestow UAVs the ability to verify their GPS locations and detect and mitigate GPS spoofing attacks.

There are four kinds of GPS spoofing detection approaches for cellular-connected UAVs, including GPS navigation signal analysis (e.g., [6], [7], [8], [9]), GPS navigation message encryption (e.g. [10], [11], [12], [13]), sensor fusion re-localization (e.g. [14], [15], [16], [17], [18], [19]) and mobile cellular network positioning (e.g. [20], [21], [22], [23]). The GPS navigation signal analysis methods use GPS signals features for spoofing detection, such as GPS satellites' signals fingerprints, the Direction of Arrival (DOA), or the Time of Arrival (TOA). Since the ionosphere interference makes the orbit GPS signals different from fake ones, the authors in [6] leveraged the pseudo-random noise sequence of GPS satellite signals to detect GPS spoofing attacks. Similarly, Milidu et al. in [7] used the DOA time-series estimation to detect GPS spoofing and Harshad et al. in [8] employed the extended Kalman filter to detect the fake GPS signals TOA and recover the legitimate signals from spoofed ones. Specifically, the authors in [9] utilized the signal phase to detect distance-decreasing GPS spoofing attacks. Unlike GPS navigation signal analysis, GPS navigation message encryption methods secure GPS navigation through the message cryptographic signature [10]. In [11], the authors used the elliptic curve digital signature algorithm to insert a signature into the navigation message for secure GPS navigation. Furthermore, Wu et al. leveraged the Shangyong-Mima cryptographic algorithm to counteract tamper attacks on navigation messages for BeiDou II [12]. Moreover, Nicola et al. evaluated the Timed Stream Loss Tolerant Authentication (TSLTA) protocol in the Galileo navigation system in order to against GPS spoofing attacks [13]. However, those GPS navigation signal analysis and message encryption methods can hardly be applied to the cellular-cellular UAVs economically, because they require either redesigned GPS receivers or updated GPS navigation systems.

Sensor fusion is another way to help UAVs detect and mitigate GPS spoofing attacks. Generally, GPS spoofing attacks result in unintended accelerations or extra mission distances and deviate UAVs from planned trajectories, which kinds of abnormal behaviors can be recorded by onboard sensors, such as accelerometers, gyroscopes, magnetometers, and cameras. In fact, the Inertial Navigation System (INS) can determine the UAVs' position using those sensors' data and compare it with the GPS position [14], [15], [16]. The GPS position is spoofed if the distance between the INS position and the GPS position is outside a preset margin. Similarly, the aerial camera and machine learning methods endow a UAV with the visual ability to recognize its surroundings and locate itself, which can also be used for GPS spoofing detection [17], [18], [19]. When environment coordinates are different from GPS positions, the GPS is spoofed. Although sensor fusion re-localization methods offer advantages, the accuracy of location can be reduced due to error accumulation in sensor measurements. While IMU sensors are useful for measuring acceleration and angular velocity, they have limitations such as the issue of integration drift that causes measurement errors to accumulate over time, resulting in inaccurate readings. Additionally, IMU sensors can be sensitive to temperature changes, and their limited range can lead to further inaccuracies and make them unsuitable for GPS spoofing detection. The operation of aerial cameras is also limited by weather conditions, and they cannot be used on cloudy or rainy days. Moreover, external factors such as vibrations, magnetic fields, and electromagnetic interference can affect camera readings. Therefore, when using sensor fusion re-localization methods, it is crucial to consider these limitations, ensure appropriate calibration and usage, and select the most suitable sensors and cameras for the application.

Mobile cellular network positioning is a new GPS spoofing detection method for cellular-connected UAVs. According to the 3rd Generation Partnership Project (3GPP), the upcoming 5G and beyond Long Term Evolution (LTE) system can provide identification and tracking service for cellular-connected UAVs [24]. As a response, the authors in [20] proposed the Adaptive Trustable Residence Area (ATRA) to verify the authenticity of GPS positions using the UAV up-link RSS data on the edge server. Notwithstanding the detection performance of the ATRA method, it requires at least three base stations at the same time. For that reason, Dang et al. in [21] used deep ensemble learning on edge servers to detect GPS spoofing with only a single base station. Simultaneously, Meles et al. did measurements in [22] and [23] that proved the 3D Angle of Arrival (AOA) of cellular signals can assist UAV self-localization and help UAV to detect and mitigate GPS spoofing attacks. Although the above mobile cellular network positioning methods demonstrate effectiveness in detecting GPS spoofing, those methods cannot be implemented on cellular-connected UAVs in the urban canyon because of the dense and irregular buildings with complex electromagnetic propagation environments.

To this end, we build a 3D radio map for an urban canyon environment and then use machine learning methods and a 3D radio map to detect and mitigate GPS spoofing attacks for cellular-connected UAVs. First, we construct a 3D radio map with the help of the deterministic channel mode and Kriging method. Then, machine learning methods, including MLP, CNN, and RNN, are employed to analyze the radio map data and UAV real-time RSS data. Explicitly, the radio map data is used as ground truth to indicate abnormal behaviors of the UAV caused by GPS spoofing attacks. Finally, the particle filter is applied to mitigate GPS spoofing attacks after the spoofing detection. The following are the major contributions of this paper.

- We have designed a system for detecting and mitigating GPS spoofing in cellular-connected UAVs operating in

urban canyon environments. This system allows us to monitor the UAV's GPS location using the theoretical RSS data and the real-time measured RSS data on an edge server, thereby reducing energy consumption and minimizing the additional load on the UAV.

- A 3D radio map is constructed to aid in the detection and mitigation of GPS spoofing by providing theoretical RSS values. Specifically, through our research, we discovered that the Universal Kriging (UK) method, which employs an exponential covariance function, can produce a highly detailed radio map with accurate RSS values and minimal computation and storage requirements.
- To detect GPS spoofing attacks, artificial neural networks such as MLP, CNN, and RNN, are implemented on the edge server. These neural networks can analyze the difference between the RSS values of the 3D radio map and real-time measurements and identify any deviations caused by a spoofing attack in the trajectory of the UAV. Importantly, these neural networks can achieve effective spoofing detection even with a single base station.
- To mitigate GPS spoofing attacks, it is essential to determine the true position of a UAV before establishing a recovery path. For this purpose, we designed a particle filter using the Wasserstein distance between radio map data and a set of RSS measurement data to determine the accurate location of the UAV. The particle filter is highly effective and can relocate the UAV within a minute with an error margin of only 10 meters.

The remainder of this paper is organized as follows. Section II reviews the related works about radio maps and neural networks. Section III provides the system model and problem formulation. Section IV gives details on Kriging-based radio map construction. Section V presents the processes of machine learning-based GPS spoofing detection as well as the particle filter-based spoofing mitigation. The simulation platform and results are shown in Section VI. Conclusion and future work are presented in Section VII.

II. RELATED WORKS

A. RADIO MAP

Radio maps are functions that provide both large-scale channel gain and small-scale channel fading information for a region of interest, and have been widely used for network planning, spectrum management, interference coordination, and indoor localization [25]. In fact, radio maps have been also used for UAV trajectory planning aiming to reduce communication interference and increase communication performance [26], [27], [28].

There are two kinds of radio map construction methods, including data-driven methods and model-driven methods. The former methods leverage the electromagnetic data of 3D space and inverse distance weighted interpolation or Kriging spatial interpolation to build a radio map directly [29]. The latter methods use the property of wireless channels' spatial correlation and radio propagation models (e.g., the exponential decay model for channel correlation and the log-normal model for shadowing) to build a radio map as a function of geographic locations [30], [31]. Data-driven methods show better performance when electromagnetic data are evenly distributed, while model-driven methods perform well with extra radiation information [32]. However, the urban area is crowded with a diversity of random radio interference that declines the accuracy of data-driven radio maps. Nevertheless, the shapes of the urban buildings are irregular, which makes the actual radio propagation complex and deteriorates model-driven radio map estimation. To overcome the above weakness, the combination of ray-tracing and interpolation is feasible for dynamic 3D radio map construction with less time consumption as well as high accuracy reservation [33].

B. NEURAL NETWORK

1) MLP

MLP is a deep learning neural network consisting of an input layer, a set of hidden layers, and an output layer. The input layer has a number of neurons the same as the size of features, and hidden layers are located between the input and output layer with an arbitrary number of neurons depending on the neural network function [34]. Functionally, the hidden layer neurons apply weights or nonlinear transformations to the input features and propagate an output to the next layer. Mathematically, the hidden layer with n neurons is formulated as

$$y = f(\sum_{j=1}^{n} \omega_j x_j + \theta_j) \tag{1}$$

where x_j denotes the input of the j^{th} neuron, ω_j is the weight and θ_j is the bias in j^{th} neuron. $f(\cdot)$ is the activation function that performs the nonlinear transformation. y is the output to the next layer. Although MLP shows good performance in classification, it requires well-designed features and labeled data for model training, updating and evaluation.

2) CNN

CNN is good at spatial pattern recognition, including image classification and voice recognition [35]. A typical CNN consists of four kinds of layers, which are the convolution layer, pooling layer, flatten layer, and fully-connected layers. The convolution layer is the first layer and the core layer of CNN, and it focuses on extracting the deep features from the raw input data using a convolutional kernel and a set of learnable parameters. Note that the convolution layer has a big output because of convolutional operations. In this vein, the pooling layer is used after convolution layers to down-sample convolutional outputs and reduces neural network size for controlling overfitting. Following the pooling layer, the flattened layer is used to reshape the down-sample features into a 1-D array that is the input of the fully-connected layers. Finally, fully-connected layers conduct classification and recognition, which works the same as MLP. Since the use of

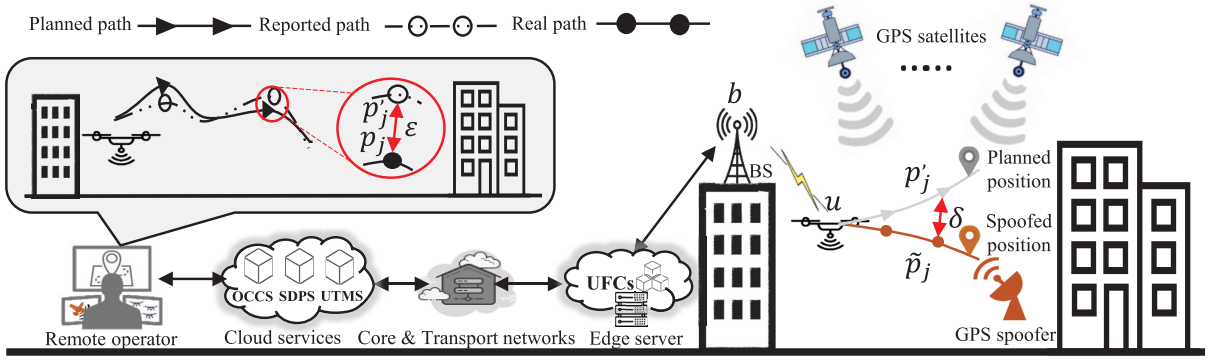


FIGURE 1. **The cellular-connected UAV network system.**

convolution layers, CNN can directly process the raw data according to its input formation.

3) RNN

RNN is famous for its feedback neural network architecture that can extract temporal features from time-sequence data and has been widely used for speech recognition and time series prediction [36]. Compared with one input in MLP and CNN neurons, RNN neurons have two inputs, one for current input and the other for past inputs, which allows exhibiting the temporal behaviors. However, the feedback neuron design leads to the vanishing or explosion of the gradient during neural network training because of the cycling of previous inputs [37]. To solve this problem, two RNN variants, the long-short-term memory (LSTM) model and the gated recurrent units (GRUs) model, have been designed based on a gated mechanism [36]. The LSTM model uses three gates including a forget gate to extract the temporal features from time sequences, while the GRU model has two input gates with activation functions for combining the current and recent previous features. In comparison to LSTM, GRU has a simple architecture that reduces the model training time as well as keeps model performance.

III. SYSTEM MODELING AND PROBLEM FORMULATION

A. SYSTEM MODEL

1) NETWORK MODEL

Fig. 1 illustrates the cellular-connected UAV network system, which consists of one remote operator, a set of UAVs' mission-related cloud services, the core and transport networks, the edge server, and mobile cellular networks [38]. The remote operator can interact with the UAV through the cloud services, where the Operator Command and Control Service (OCCS) provides interfaces for the UAV operator to access the Supplementary Data Provider Service (SDPS) and the Unmanned Aircraft System Traffic Management Services (UTMS). Specifically, the SDPS offers meteorological data and other information related to UAV flight planning and the UTMS manages UAVs' registrations, identifications, and airspace restrictions [39]. Additionally, core and transport networks are in charge of data transmission between the cloud service and the edge UAV flight Controllers (UFCs). Furthermore, the UFCs are deployed on edge servers and connected to a base station to execute UAV flight missions, including controlling and monitoring the UAVs. In such a cellular-connected UAV network system, the GPS serves as the main navigation and positioning solution for the UAVs. Therefore, the security of GPS is critical for the system, as all of the UAV flight operations, such as controlling, monitoring, and geofencing, are based on GPS positions.

2) CELLULAR-CONNECTED UAV MODEL

The cellular-connected UAV setup shown in Fig.1 includes GPS satellites, an edge server, a base station (denoted by b), and a UAV (denoted by u). The UAV is operated by the edge UFC through the base station wirelessly. Typically, the edge UFC receives waypoints from a remote controller and uses a mission planner to determine the UAV's trajectory, while also considering the coverage information of the base station. To ensure a successful mission, the UAV's trajectory is optimized to maintain a strong connection with the base station. The location of the base station is denoted as (x_b, y_b, z_b), and the planned position of the UAV is denoted as (x_j, y_j, h_j) for a particular position p_j. If the UAV follows the planned trajectory, it will be in the best connection positions with the base station during the mission.

3) ATTACK MODEL

In the attack scenario shown in Fig.1, there exist a GPS spoofer, a base station (b), and a victim UAV (u). The GPS spoofer has the ability to transmit falsified GPS signals, leading the UAV's GPS receiver to compute incorrect positions, ultimately resulting in the UAV following a spoofed trajectory instead of its planned trajectory. If there are no GPS spoofing or GPS errors, then the UAV should proceed along the planned trajectory and be located at p_j at time j. If GPS errors occur, then the UAV will report its position as p'_j at time j, which is ϵ distance units away from p_j. However, when GPS spoofing is present at time j, the UAV's actual location will be $\tilde{p}_j$, deviating from p_j by δ distance units. In this context,

we consider $\epsilon \leq dE < \delta$, where dE represents the maximum GPS error.

4) CHANNEL MODEL

It should be noted that GPS spoofing attacks can cause deviations that negatively impact the connection performance between the UAV and the base station. In this way, the performance of wireless connections is leveraged to detect GPS spoofing attacks. This paper adopts a deterministic radio propagation model to model the channel between the base station and its connected UAV. The use of the deterministic channel model allows for the sensing of physical effects in urban canyon environments, resulting in more accuracy for path loss values computation. Foundationally, the wireless link of cellular-connected UAVs includes both line-of-sight and non-line-of-sight paths. Specifically, Friis' free space radio propagation is used for modeling the line-of-sight directed path, while Huygens' secondary source is utilized for modeling the non-line-of-sight diffraction path. According to Friis' equation, the directed path loss between the base station and its connected UAV is modeled as:

$$\bar{L}_{bu} = 20log\frac{4\pi d_{ub}}{\lambda_{bu}}, \tag{2}$$

where d_{ub} is the distance between b and u, and λ_{bu} is the wavelength from base station. In addition, the knife-edge diffraction model follows the International Telecommunication Union (ITU) recommendation [40].

$$\hat{L}_{bu} = 6.9 + 20log(\sqrt{(\nu_{bu} - 0.1)^2 + 1} + \nu_{bu} - 0.1) \tag{3}$$

where the Fresnel diffraction parameter ν_{bu} can be written as

$$\nu_{bu} = z'_{bu}\sqrt{\frac{2(d^1_{bu} + d^2_{bu})}{\lambda_{bu} d^1_{bu} d^2_{bu}}}. \tag{4}$$

In (4), z'_{bu} is the effective height of the building knife-edge with infinite width between b and u at the distance d^1_{bu} and d^2_{bu}, $d^1_{bu}, d^2_{bu} \gg z'_{bu}$ and $d^1_{bu}, d^2_{bu} \gg \lambda_{bu}$, seeing in Fig.2(a). In addition to the directed path and diffracted path, the reflection also has impacts on the deterministic model. Let $\check{L}_{bu}$ denote the reflection path loss between b and u, and $\check{L}_{bu}$ is formulated as

$$\check{L}_{bu} = 20log\frac{4\pi d'_{bu}}{\alpha_{p_r}\lambda_{bu}}. \tag{5}$$

As shown in Fig.2(b), d'_{bu} is the total traveling distance from b to u with a signal reflection at position p_r, $d'_{bu} = d_{bp_r} + d_{p_r u}$ respectively, and α_{p_r} is the reflection coefficient that depends on the transmitters' positions, the reflection medium permittivity and the signal polar position [41]. The used path loss from b to u is determined by

$$L_{bu} = \min(\bar{L}_{bu}, \hat{L}_{bu}, \check{L}_{bu}), \tag{6}$$

where L_{bu} is considered as the smallest path loss of the directed, diffracted or reflected path. Regarding multi-path fading caused by the scattering effect, small-scale fading is considered in the urban canyon environment. In contrast to

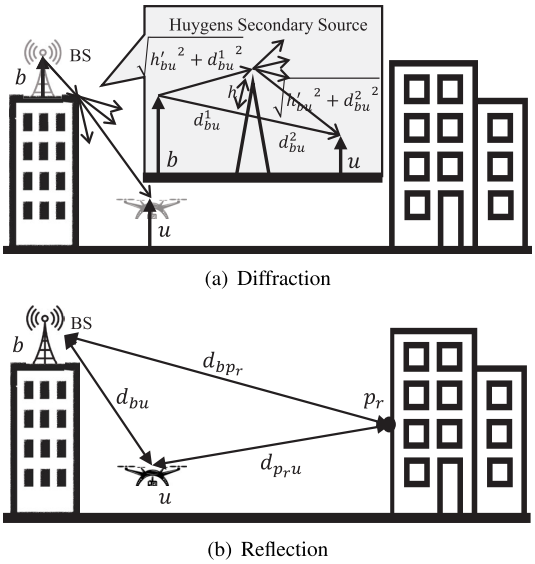


FIGURE 2. Diffraction and reflection.

stochastic models, which are typically described using statistical distributions (such as the Rayleigh and Rician fading models), deterministic channel models utilize the reflection coefficient and random noise to model small-scale fading. Thus, at a given position, p_j, the UAV RSS is

$$RSS(b, u, p_j) = T_b - L_{bj} + G_b + G_u + \aleph_{bu} \tag{7}$$

where T_b is the base station transmission power, and L_{bj} is the path loss value when the UAV locates at p_j, and G_b and G_u represent the base station antenna gain and the UAV antenna gain separately, and $\aleph_{bu}$ serves as the noise caused by the scattering effect in dBm. To address the impact of dynamic environments on RSS values, one approach is to use statistical analysis techniques to minimize their influence. Another effective method is to utilize neural network layers such as convolutional layers to eliminate environmental effects from RSS data, seen in Section V.

5) RADIO MAP MODEL

Let's assume that the UAV locates at p_j and plans to p_{j+1} at a given time period τ. When the UAV is located on the rooftop of the building, the probability of having a line-of-sight connection increases, making it easier to model the channel using statistical models as stated by the 3GPP in [42]. However, in the urban canyon environment, due to the scarcity of direct connections, it is difficult to model the wireless channel using statistical models. In this case, the radio map is used to model the channel when the base station is under the rooftop and the UAV is limited below rooftop heights. To construct the radio map $\mathcal{M}_\tau$, an amount of K positions are randomly picked from the region $\mathcal{S}_\tau$ that is centered at p_c with a diameter d_c, p_c locating at the middle position between p_j and p_{j+1}, and $d_c = d_{j,j+1} + dE$, $d_{j,j+1}$ indicating the distance from p_j to p_{j+1} respectively. Accordingly, the k^{th} position, p_k, is assigned

with a corresponding RSS value $\mathfrak{r}_k$ using the deterministic channel model in (7). Then, the radio map $\mathcal{M}_\tau$ is initiated as

$$\mathcal{M}_\tau \leftarrow \{p_k, \mathfrak{r}_k\}, \quad \forall k, k \in [1, K]. \tag{8}$$

Given a new location $p_i, p_i \in \mathcal{S}_\tau, p_i \neq p_k, k \in [1, K]$, $\mathcal{M}_\tau$ can estimate $\mathfrak{r}_i$ for p_i using the interpolation method, seen in Section IV.

B. PROBLEM FORMULATION

As shown in Fig.1, the GPS spoofer deceives the UAV with false GPS positions that make the UAV deviate from the planned trajectory. Along with GPS spoofing deviation, the real distance between the UAV and the base station is not the same as the reported GPS distance, which can be determined by the difference between the theoretical path loss and the real-time path loss. In fact, the radio map can offer theoretical path losses and the base station can provide real-time path losses. In addition, the edge UFC can use the radio map and the historical path losses to indicate the UAV's true position.

1) GPS SPOOFING DETECTION PROBLEM

Let $\tilde{L}_{bu}(t)$ present the real-time path loss at time t and $L_{bu}(t)$ denote the corresponding theoretical path loss for the reported position at time t. Theoretically, the path loss is determined by the radio propagation distance from the base station and UAV, so that the absolute difference ΔL_{bu}, ($\Delta L_{bu} = \left|\tilde{L}_{bu} - L_{bu}\right|$), can indicate the deviation between real position and reported position. Hence, the GPS spoofing attack can be detected through hypothesis testing, seen in (9).

$$\begin{cases} H_0: & \Delta L_{bu} > T, \\ H_1: & \Delta L_{bu} \leq T, \end{cases} \tag{9}$$

where H_0 stands that the reported position is spoofed when ΔL_{bu} is over the preset threshold T, while H_1 represents the reported position without GPS spoofing. It is noteworthy that the real path loss $\tilde{L}_{bu}(t)$ is shifting with environment changes, which results in the hypothesis testing in (9) may not accurately indicate the deviation between the reported position and the spoofed position. In addition, the threshold T requires an appropriate setting. Otherwise, the hypothesis testing may raise a higher false alarm because of a smaller T or give a lot of missed detection due to a bigger threshold.

2) GPS SPOOFING MITIGATION PROBLEM

The goal of GPS spoofing mitigation is to help the UFC identify the correct position of the UAV and redirect it to the planned path, even after a spoofing attack. This involves an analysis of both historical real path losses and theoretical radio map path losses to relocate the UAV and generate a recovery path based on the actual and intended position of the UAV. Furthermore, motion sensors can be utilized to monitor the UAV, which can ensure the UAV returns to its intended location [43].

Let d_τ denote the UAV traveling distance from the initial waypoint p_j to the final waypoint p_{j+1}, and $\mathfrak{r}_{1:M}$ represents the UAV RSS measurements in this time period, $\mathfrak{r}_{1:M} = \{\mathfrak{r}_i; i = 1, \ldots, M\}$ correspondingly. So, the true position can be found in the positions that are away from p_j with distance d_τ. Let $\mathbb{P}_N$ denote N positions in $\mathcal{S}_\tau$ that have a distance d_τ from p_j, $\mathbb{P}_N = \{\tilde{p}_n; n = 1, \ldots, N\}$, and $\tilde{\mathfrak{r}}^n_{1:M}$ stand for the theoretical RSS provided by $\mathcal{M}_\tau$ for the path from p_j to $\tilde{p}_n$, $\tilde{\mathfrak{r}}^n_{1:M} = \left\{\tilde{\mathfrak{r}}^n_i; i = 1, \ldots, M\right\}$ respectively. Hence, the mitigation problem is to find $\tilde{p}_n$ that has the highest similarity on the distribution of path loss values.

$$\min_{\mathfrak{s}_1, \ldots, \mathfrak{s}_N} \quad \{\mathfrak{s}_n; n = 1, \ldots, N\}, \tag{10}$$

$$\text{Subject to:} \quad \mathfrak{s}_n = \Phi(\mathfrak{r}_{1:M}, \tilde{\mathfrak{r}}^n_{1:M}), \tag{10a}$$

$$0 \leq n \leqslant N, \tag{10b}$$

$$0 \leq \mathfrak{s}_n \leqslant 1; \tag{10c}$$

where $\mathfrak{s}_n$ is the similarity between $\mathfrak{r}_{1:M}$ and $\tilde{\mathfrak{r}}^n_{1:M}$. It is worth mentioning that the number of N has a direct impact on the mitigation performance, a bigger N with higher accuracy as well as greater computation. Precisely determining the actual position of the UAV is vital to mitigate GPS spoofing, as it enables the UFC to create a recovery plan that takes into account different constraints and guides the UAV back to its intended trajectory. Commonly used methods for UAV path planning include A* search, Dijkstra's algorithm, Rapidly-Exploring Random Trees (RRT), and Probabilistic Roadmaps (PRM) [44]. Thus, the primary challenge in GPS mitigation is pinpointing the true position, as it is a prerequisite for planning a recovery path.

To address the issues with spoofing detection and mitigation, we propose a solution that involves constructing a 3D radio map to aid in detecting and mitigating GPS spoofing. The radio map provides theoretical RSS values that can be compared to real-time RSS measurements using artificial neural networks such as MLP, CNN, and RNN implemented on the edge server. By analyzing the difference between the RSS values of the 3D radio map and real-time measurements, any deviations caused by a spoofing attack in the UAV's trajectory can be identified. Once spoofing is detected, the particle filter estimates the true location of the UAV by calculating the Wasserstein distance between the radio map data and a set of RSS measurement data. This estimation helps to mitigate the spoofing attack.

IV. KRIGING BASED RADIO MAP CONSTRUCTION

In this part, the Kriging spatial interpolation method is applied to the ray-tracing samples in order to build a radio map accurately and efficiently. With the help of the Kriging method, the fine-grained 3D Radio Map construction algorithm is designed for both UFC and UAVs, which allows constantly detecting and mitigating GPS spoofing attacks in the discontinuous communication environment.

A. KRIGING METHOD

Kriging is a famous interpolation method for geostatistic measurements due to its productivity in terms of dealing with spatial variations [45]. Compared with the other interpolation methods, such as the inverse distance weighted interpolation method and the spline interpolation method, Kriging does not depends on deterministic mathematical formulas but focuses on statistical models that can not only produce a surface but also provide predictions to uncertain points [46]. There are two kinds of Kriging methods, Ordinary Kriging (OK) and Universal Kriging (UK). The OK is for data with constant trends meanwhile the UK is for data with deterministic trends, for example, the large-scale fading [47]. Both OK and UK follow the same procedures, including data analyzing, variogram modeling, surface creating, and predicting [48].

The Kriging method is mathematically defined in (11).

$$\hat{R}(p_i) = \sum_{k=1}^{K} \omega_k \mathfrak{r}_k, \tag{11}$$

where $\hat{R}(p_i)$ denotes the predicted RSS for a given position p_i, $p_i \in \mathcal{S}_\tau, p_i \neq p_k, k \in [1, K]$, and $\mathfrak{r}_k$ is the measured RSS value at position p_k, ω_k representing an unknown weight for $\mathfrak{r}_k$, K standing for the number of known RSS values. Using those known positions and RSS data, Kriging employs variograms and covariance functions to estimate the known data statistical dependence and then fit those data as an empirical semivariogram model for making a prediction [45]. According to the data statistical dependence, the empirical semivariogram model can be a circular, spherical, exponential, Gaussian, or linear function respectively [45]. Carefully, each empirical semivariogram model is designed for different tasks, which makes the prediction more accurate.

B. FINE-GRAINED 3D RADIO MAP CONSTRUCTION

Although the use of ray-tracing and Kriging methods can build a 3D radio map accurately, it spends too much computation and storage in the edge server for constructing the whole region's fine-grained 3D radio map. In addition, due to the UAV movement and the urban environment changes, the radio map needs to keep fresh with the UAV mission, which requires a large number of updating processes on the edge server. Considering the limited resources of the edge server, the mission-based fine-grained 3D radio map is proposed that focuses on a region for a segment of the planned trajectory, seeing in Fig.3.

In Fig.3, the UAV u plans to move from p_j to p_{j+1} at the given time period τ, and $\mathcal{M}_\tau$ is the fine-grained 3D radio map for the region where includes the UAV moving space during time period τ.

Algorithm.1 summarizes the fine-grained 3D radio map construction methods, including the region selection, ray tracing, and Kriging prediction. The region selection aims to find the UAV movement space that depends not only on the planned trajectory but also on the motion sensors (lines 2-5). Specifically, the motion sensors supervise the UAV to follow its planned trajectory but also infuse the accumulated error because of its intrinsic imperfection [21], which can be further discriminated by the radio map in the region. After the region selection, ray tracing is applied to get the coarse radio map for the region (lines 6-11). Particularly, all rays start

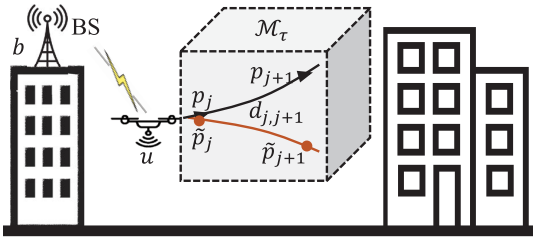


FIGURE 3. The concept of fine-grained 3D radio map.

Algorithm 1 Fine-grained 3D radio map constructing.

Input:

p_j: The planned waypoint, $p_j = (x_j, y_j, z_j)$;
p_{j+1}: The planned waypoint, $p_{j+1} = (x_{j+1}, y_{j+1}, z_{j+1})$;
dE: The GPS error margin in meter;
v: The resolution of the fine-grained radio map in meter;
τ: The planned time period from p_j to p_{j+1};

Output:

$\mathcal{M}_\tau$: The fine-grained radio map;

1: Initial the fine-grained radio map, $\mathcal{M}_\tau \leftarrow \{\}$;
/********** Region selection ********** /
2: Calculate the distance $d_{j,j+1}$, $d_{j,j+1} = |p_j, p_{j+1}|_{3D}$;
3: Compute the region center p_c, $p_c = (x_c, y_c, z_c)$, where $x_c = \frac{x_j + x_{j+1}}{2}$, $y_c = \frac{y_j + y_{j+1}}{2}$, $z_c = \frac{z_j + z_{j+1}}{2}$;
4: Compute the region radius d_c, $d_c = d_{j,j+1} + dE$;
5: Get the region $\mathcal{S}_\tau$, $\mathcal{S}_\tau = \{(x, y, z); x \in \mathbb{X}, y \in \mathbb{Y}, z \in \mathbb{Z}\}$, where $\mathbb{X} = [x_c - d_c, x_c + d_c]$, $\mathbb{Y} = [y_c - d_c, y_c + d_c]$, $\mathbb{Z} = [z_c - d_c, z_c + d_c]$;
6: Sample K positions from $\mathcal{S}_\tau$ randomly, $\mathcal{P}_K = \{p_k; k = 1, \ldots, K\}$, where $p_k \in \mathcal{S}_\tau, p_k \neq p_{k'}, 1 \leq k \neq k' \leq K$;
/********** Ray Tracing (RT) ********** /
7: Initial the ray tracing radio map, $\mathcal{M}_K \leftarrow \{\}$;
8: **for** $k = 1 \rightarrow K$ **do**
9: Get the RSS value for p_k, $\mathfrak{r}_k = RT(p_k)$;
10: $\mathcal{M}_K \leftarrow \{p_k, \mathfrak{r}_k\}$;
11: **end for**
/********** Kriging predict ********** /
12: **for** $x_v \in [x_c - d_c, x_c + d_c, v]$ **do**
13: **for** $y_v \in [y_c - d_c, y_c + d_c, v]$ **do**
14: **for** $z_v \in [z_c - d_c, z_c + d_c, v]$ **do**
15: Construct the position p_v, $p_v = (x_v, y_v, z_v)$;
16: Predict $\mathfrak{r}_v$, $\mathfrak{r}_v = Kriging(\mathcal{M}_K, p_v)$;
17: Update $\mathcal{M}_\tau$, $\mathcal{M}_\tau \leftarrow \{p_v, \mathfrak{r}_v\}$;
18: **end for**
19: **end for**
20: **end for**
21: **return** $\mathcal{M}_\tau$.

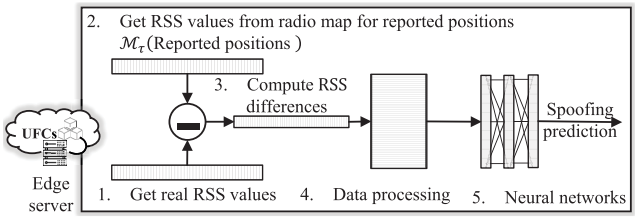


FIGURE 4. GPS spoofing detection.

at the base station position and end at the randomly chosen positions from the region. In the end, Kriging uses the coarse radio map to interpolate the RSS values for the selected region until achieving the resolution requirement of the fine-grained 3D radio map (lines 12-20). Expressly, the resolution is related to the spoofing detection system requirements, and the higher resolution comes with better detection and mitigation performance.

It is notable that ray tracing is a heavy program and is only deployed on the edge server in order to interact with the real dynamic environment. Compared with ray tracing, Kriging is a light program that can also work on the UAV onboard computer. The combination of ray tracing and Kriging provides an affordable method for both the edge server and the UAV that constructs a fine-grained 3D radio map with high resolution and compatible computation on demand. In particular, the UAV constructs the radio map with a lower resolution if and only if it loses the connection to the base station because of the spoofing attack.

V. GPS SPOOFING DETECTION AND MITIGATION

GPS spoofing attack happens continuously for manipulating the UAV to deviate from its planned trajectory gradually without raising any notice on the motion sensors. However, the hypothesis testing in (9) can not indicate the true deviation caused by the GPS spoofing attack, because the wireless communication between the base station and UAV is dynamic and disturbed by spectrum noise. In addition, the threshold T in (9) has a significant impact on the detection results, a smaller T with a higher false alarm while a bigger T rising a lot of missed detection. To overcome those problems, this section provides machine-learning methods for GPS spoofing detection and mitigation, where different neural networks are applied to detect GPS spoofing, including MLP, CNN, and RNN, and then the particle filter is introduced to estimate the true position after the spoofing detection.

A. GPS SPOOFING DETECTION

As shown in Fig.4, the edge server GPS spoofing detection consists of five steps, including getting real RSS values, getting theoretical RSS values from the fine-grained radio map, computing RSS differences, data processing, and neural network training and predicting. As for the UAV, the GPS spoofing detection has the same processes as the edge server, but, instead of training neural networks, the UAV can download neural networks from the edge server directly. The edge server GPS spoofing detection is processed as follows.

- Firstly, the real RSS values are reported by the UAV in the telemetry data or the base station through 3GPP-defined services. Notably, the cellular connections between the UAV and base station are not impacted by the spoofing attack because of different communication bandwidths and security standards, so the real RSS values can indicate the spoofing attacks.
- Secondly, the theoretical RSS values are provided by the fine-grained radio map constructed in Section.IV. It is worth mentioning that the fine-grained radio map may not have the reported position RSS value but can provide the theoretical RSS using the nearest position in the high-resolution map.
- Thirdly, the differences between real RSS values and theoretical RSS values can demonstrate the UAV trajectory deviation caused by GPS spoofing attacks. Due to the dynamic environment and spectrum interference, the real RSS values include uncertain noises that are biased from the theoretical radio map. Therefore, data processing and neural network are employed on the RSS differences for mitigating the noise effects on the spoofing detection.
- Fourthly, data processing aims to process the data with statistical analysis or reshaping operations. The statistical analysis can help to reduce the noise effects while the reshaping operations can feed the data into a neural network directly.
- Fifthly, the neural networks are responsible for spoofing detection using prepossessed data. Generally, the neural networks need to be trained before using for prediction. In fact, the training data can be guaranteed without GPS spoofing attacks by leveraging higher-level protection, such as GPS signal analysis.

There are two kinds of neural networks used in our previous works for GPS spoofing detection, including MLP-based and CNN-based GPS spoofing detection [21], [38]. In [21], we proposed using path loss statistics features as inputs to an MLP for detecting GPS spoofing. To remove environmental variance from the path loss computation, statistical methods were used. The MLP was then trained on these statistics to find an appropriate threshold and improve detection accuracy. In contrast, [38] employed a CNN to detect GPS spoofing. The CNN used grouped data and convolution layers to extract path loss differences caused by spoofing attacks, and fully-connected layers were used to address threshold issues. The convolution layers allowed for the extraction of deep features directly from the raw data. However, statistical processing can cause visible latency and may impede GPS spoofing detection in a UAV environment. Additionally, to reduce neural network size and avoid overfitting, dropout was added to the CNN layers, which may decrease detection accuracy.

In addition to MLP and CNN, RNN-based GPS spoofing detection is an alternative solution. Rather than using

statistics or convolution layers, RNN uses the bidirectional layer to extract abnormal behavior features in the RSS differences. Those features are further delivered to the fully connected layers for spoofing detection. The spoofing detection performance of MLP, CNN, and RNN are shown in Section.VI.

B. GPS SPOOFING MITIGATION

The particle filter is an efficient method for solving the estimation problem in (10), which can help to find the UAV's true position after spoofing detection. Following the hands-on tutorial in [49], the particle filter-based GPS spoofing mitigation is summarized in Algorithm.2.

In Algorithm.2, the particle filter estimates the UAVs' position requiring four steps, including initialization, sampling, selection, and resampling. The first initialization step is to randomly choose N particles on the circle that center at p_j with radius d_{ij}. Specifically, the Mises distribution, or named circular normal distribution, is used for picking an angle φ from the circle with a Probability Density Function (PDF)

$$f(x \mid \mu_\varphi, \kappa_\varphi) = \frac{exp(\kappa_\varphi \cos(\varphi - \mu_\varphi))}{2\pi I_0(\kappa_\varphi)}, \tag{12}$$

where μ_φ is the mean of the locations and κ_φ is the concentration of the locations and $I_0(\kappa_\varphi)$ is the modified Bessel function to scale the chosen angle [50]. In addition, $\mu_\varphi = 0, \kappa_\varphi = 0$ represent the Mises distribution is uniform. Then, the sampling step uses the chosen N angles and d_{ij} to compute particles' coordinates. In particular, each position is assigned with the possibility of being the final position, which depends on the similarity between the real RSS values and the radio map theoretical RSS values. Especially, the Wasserstein Distance (WD) is used to indicate the distance between the real RSS values and the theoretical RSS values, and a smaller WD is corresponding to a higher possibility that the estimated position is more closed to the real position [21]. Following sampling, the selection step picks up the smallest Z elements from the similarity set and saves indexes for those elements in a predefined index set. In line 18, $\overrightarrow{ArgPartition}$ performs sorting and partition along the given set and returns all indexes in the partition that contains the smallest Z elements. Obviously, the true position is more likely at the position with a smaller WD distance. After selection, the resampling uses the selected positions to update $\mu_\varphi, \kappa_\varphi$ and go back to another sampling round in order to make particle samples toward the final position. As shown in Fig.5, the particle filter is more concentrated on the true position after resampling. In the end, the estimated position is the center of the resampling selected positions.

Theorem 1: The time complexity of the particle filer-based GPS spoofing mitigation algorithm is $O(N^2)$.

Proof: The time complexity of the particle filer-based GPS spoofing mitigation algorithm depends on the number of particles N and the number of RSS samples M, where the time complexity is $O(N)$ for particles and is $O(M)$ for radio map-based RSS predictions. Since position estimation needs a lot of trials, the number of particles is more than that of RSS values. Thus, the time complexity is round to $O(N^2)$ for Algorithm.2. □

GPS spoofing detection is typically performed at the edge server in a sequence way, and the frequency of detection depends on the speed of the UAV and the quality of its wireless connection. First, the edge needs to construct a 3D radio map in the UAV movement space, and then uses the neural networks to keep monitoring the UAV GPS position as well as detecting GPS spoofing. Once the spoofing is detected,

Algorithm 2 Particle filer-based GPS spoofing mitigation.

Input:

p_j: The planned waypoint, $p_j = (x_j, y_j, z_j)$;
p_i: The reported waypoint, $p_i = (x_i, y_i, z_i)$;
$\mathcal{M}_\tau$: The fine-grained radio map;
$\mathfrak{r}_{1:M}$: The real RSS values, $\mathfrak{r}_{1:M} = \{\mathfrak{r}_i; i = 1, \ldots, M\}$;
N: The number of sampling particles;
Z: The number of selected particles;

Output:

$\tilde{p}^*_{j+1}$: The estimated position;

/********** Initialization ********** /

1: Calculate the distance d_{ij}, $d_{ij} = \left|p_i, p_j\right|_{3D}$;
2: Initial Mises distribution $\mathcal{T}_\varphi(\mu_\varphi, \kappa_\varphi)$, $\mu_\varphi = 0, \kappa_\varphi = 0$;
3: Random N samples for φ using $\mathcal{T}_\varphi$, $\varphi_{1:N} = \{\varphi_n; n = 1, \ldots, N\}$;
4: Initial the position set $\mathbb{P}_N$, $\mathbb{P}_N \leftarrow \{\}$;
5: Initial the similarity set $\mathbb{S}_N$, $\mathbb{S}_N \leftarrow \{\}$;

/********** Sampling********** /

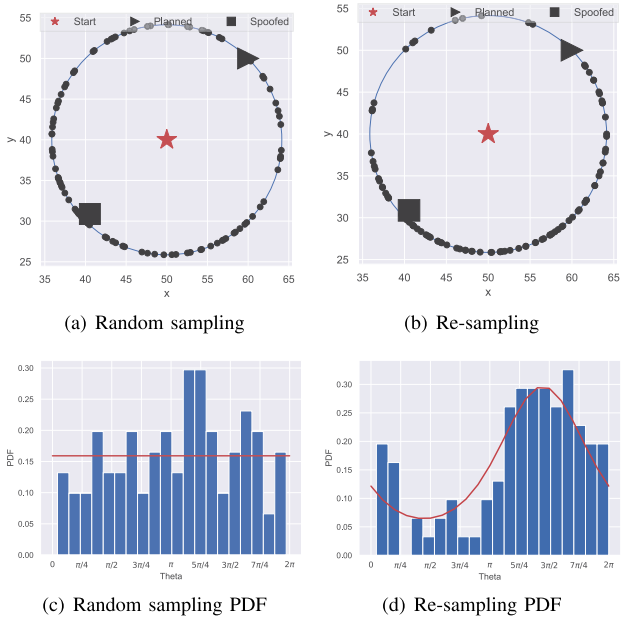
6: **for** $n \in [1, N]$ **do**
7: $\tilde{p}_n = (x_j + d_{ij}\cos\varphi_n, y_j + d_{ij}\sin\varphi_n, z_j)$
8: $\mathbb{P}_N \leftarrow \{\tilde{p}_n\}$;
9: Initial the trajectory $\mathfrak{p}_{1:M}$, $\mathfrak{p}_{1:M} = \{\mathfrak{p}_i; i = 1, \ldots, M\}$, where the positions are evenly distributed between p_j and $\tilde{p}_n$, $\mathfrak{p}_1 = p_j$, $\mathfrak{p}_M = \tilde{p}_n$ particularly;
10: Initial the RSS predictions $\tilde{\mathfrak{r}}^n_{1:M}$, $\tilde{\mathfrak{r}}^n_{1:M} \leftarrow \{\}$;
11: **for** $\mathfrak{p}_i \in \mathfrak{p}_{1:M}$ **do**
12: $\tilde{\mathfrak{r}}^n_{1:M} \leftarrow \{\mathcal{M}_\tau(\mathfrak{p}_i)\}$;
13: **end for**
14: Compute the similarity $\mathfrak{s}_n$, $\mathfrak{s}_n = WD(\mathfrak{r}_{1:M}, \tilde{\mathfrak{r}}^n_{1:M})$
15: Update the similarity set, $\mathbb{S}_N \leftarrow \{\mathfrak{s}_n\}$;
16: **end for**

/********** Selection ********** /

17: Initial the indexes set with size Z $\mathbb{I}_Z$, $\mathbb{I}_Z \leftarrow \{\}$;
18: Select Z indexes with the smallest elements from $\mathbb{S}_N$ and save the records to $\mathbb{I}_Z$, $\mathbb{I}_Z = \overrightarrow{ArgPartition}(\mathbb{S}_N, Z)$;
19: Choose Z elements from $\varphi_{1:N}$, $\varphi_{\mathbb{Z}} = \varphi_{1:N}[\mathbb{I}_Z]$

/********** Resampling ********** /

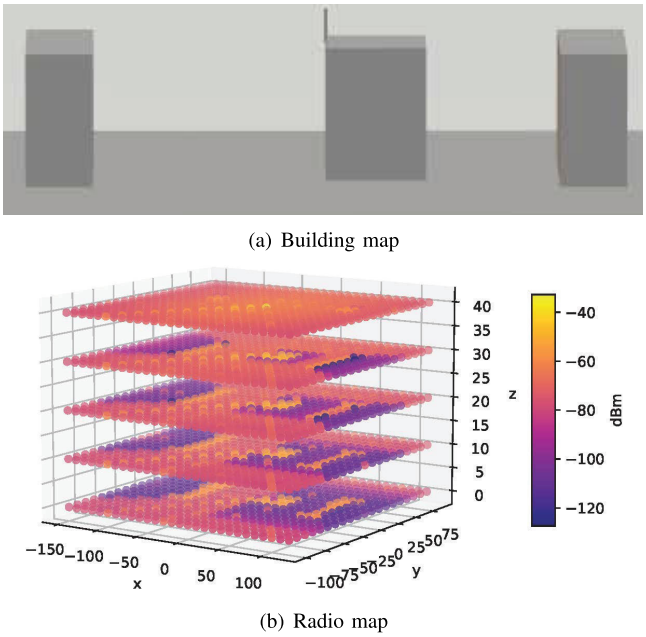
20: Update μ_φ and κ_φ, $\mu_\varphi = Mean(\varphi_{\mathbb{Z}})$, $\kappa_\varphi = Stand(\varphi_{\mathbb{Z}})$;
21: Go to line 3;
22: $\tilde{p}^*_{j+1} = Center(\mathbb{P}_N[\mathbb{I}_Z])$;
23: **return** $\tilde{p}^*_{j+1}$.



(a) Random sampling (b) Re-sampling

(c) Random sampling PDF (d) Re-sampling PDF

FIGURE 5. **Particle filter-based GPS spoofing mitigation.**



(a) Building map

(b) Radio map

FIGURE 6. **3D city environment.**

the edge server runs the particle filter to estimate the true position of the UAV and mitigate the spoofing with the aid of a recovery path. Accordingly, when the UAV is traveling at a higher speed or the wireless connection quality decreases, the spoofing detection procedure is performed more frequently. Once spoofing is detected, the GPS mitigation procedure is conducted to correct the UAV's trajectory. During GPS spoofing mitigation, the system checks for spoofing with a high frequency and continuously monitors the UAV until it returns to its intended trajectory. The simulation results for neural network-based GPS spoofing detection and particle filter-based GPS spoofing mitigation are given in the next section.

VI. SIMULATION AND RESULTS

In this section, we first build a 3D city map for cellular-connected UAVs. Then, ray tracing and Kriging methods are leveraged to construct the 3D radio map for this city. After that, the proposed neural networks and particle filters are performed on the 3D radio map for GPS spoofing detection and mitigation.

A. SIMULATION SETTING

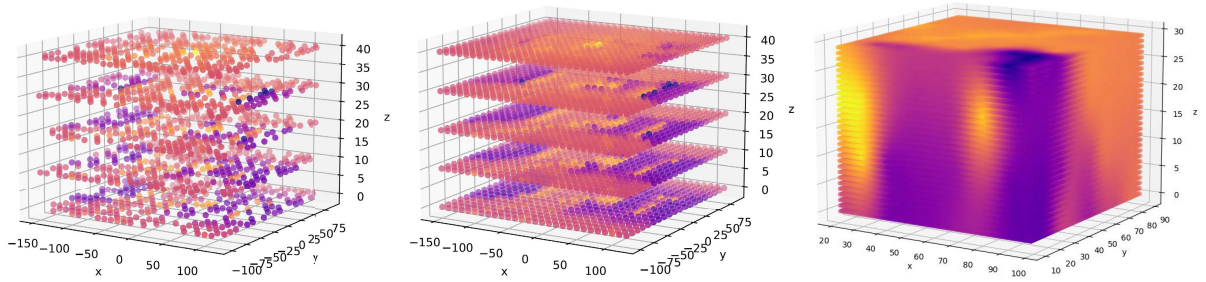
We develop a simulation platform for cellular-connected UAVs in an urban canyon environment, where buildings are constructed with Blender 3.3 and ray tracing tools are from the radio gym in [51]. In addition, Python 3.8.10 is used to set the environment and Tensorflow 2.9.0 is adopted to evaluate the performance of the designed neural networks for GPS spoofing detection and particle filter for GPS spoofing mitigation.

Fig.6 shows a 3D city environment including a building map and a radio map. In particular, Fig.6(a) is a building map that has one base station and three constructions in a 3D 300 × 300 × 50 m^3 space, where the base station is on the top of the middle building. Fig.6(b) is the city radio map developed with the deterministic channel model and ray tracing tools. Notably, this radio map is set as the ground truth to test the Kriging methods that are used for constructing the fine-grained 3D radio map.

TABLE 1. **Parameter settings of the simulation platform.**

Methods	Parameter	Definition	Value(s)
Ray Tracing	f_c	Communication frequency	$2.5Ghz$
	λ_{bu}	Wave length	0.12 m
	α_{p_r}	Reflection coefficient	5.31
	$\aleph_{bu}$	Environment noise	$\mathcal{N}(0,5)$
Kriging	p_j	Start waypoint	(60,20,50)
	p_{j+1}	End waypoint	(80,20,75)
	dE	GPS error margin	10 m
	v	Radio map resolution	1 m

Table.1 illustrate parameter settings for ray tracing and Kriging. The communication frequency is set at 2.5 GHz (wavelength round to 0.12 meters) in the simulation platform, which is used by ray tracing in the radio propagation models. In addition, the reflection coefficient is 5.31 standing for the urban concrete reflection surface [52] and the environmental



(a) Sampling (b) UK interpolation (c) UK fine-grained radio map

FIGURE 7. The performance of the UK with an exponential kernel.

noise follows the normal distribution $\mathcal{N}(0,5)$. The Kriging method is applied to the UAV flying region for constructing a fine-grain radio map. Particularly, the UAV locates at $p_j(60, 20, 50)$ and plans to $p_{j+1}(80, 20, 75)$. The GPS error dE is preset as $\{10, 20, 30, 40, 50\}$ to simulate the GPS error for different scenarios. The resolution of the fine-grained radio map, v, is 1 meter, which can meet the requirement of GPS spoofing detection and mitigation.

Neural networks are applied to analyze the fine-grained map and the reported data for GPS spoofing detection. Specifically, the MLP uses 10 neurons with the ReLU activation function in the input layer for capturing statistical results, while the CNN employs two conventional layers with the ReLU activation function for extracting deep spoofing features from the reshaped RSS data, whilst the RNN utilizes the Bidirectional layer with LSTM unit on the raw RSS sequence data directly. Individually, CNN has a MaxPool layer for reducing the neural network seize and a flatten layer for reshaping the deep conventional features, where the dropout ratio is set as 0.1 for protecting the neural network from overfitting. Typically, all neural networks have one output layer with a sigmoid activation function for predicting the possibility of GPS spoofing attacks and are trained with the Adam optimizer with a learning rate of 0.001 for 200 epochs. In addition, the training and evaluation data set come from two trajectories, one planned and one spoofed, where those two trajectories have the same length and start position but different end positions.

The particle filter is leveraged to estimate the true GPS position after the spoofing attack. In particular, the number of sampling particles N is set as 200, and the number of selected particles Z is initialed with 10 for GPS spoofing mitigation. Additionally, the particle filter has been tested 10 times with different positions for evaluating its accuracy in terms of GPS mitigation.

B. PERFORMANCE METRICS

1) STANDARD ERROR

Standard error is used for evaluating the performance of the Kriging method, which is expressed as:

$$\mathcal{L} = \frac{1}{\|\mathcal{M}_\tau\|} \sum_{p_t \in \mathcal{M}_\tau} (\mathcal{M}_\tau(p_t) - \hat{\mathcal{M}}_\tau(p_t))^2, \tag{13}$$

where $\mathcal{L}$ denotes the standard error, and $\mathcal{M}_\tau(p_t)$ represents the Kriging RSS value at p_t while $\hat{\mathcal{M}}_\tau(p_t)$ stands for the reference ground true RSS at p_t.

2) ACCURACY

Accuracy is utilized to perform the evaluation on neural networks, which is defined as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \tag{14}$$

where the true positive (TP) is the correct detection for the spoofed position while the false positive (FP) stands for the wrong detection of spoofing, the true negative (TN) represents the correct prediction for GPS position not spoofed, and, on the contrary, the false negative (FN) presents the spoofed position detected as no spoofing.

C. PERFORMANCE RESULTS

We first investigate the Kriging performance in terms of radio map construction. Then, GPS spoofing detection neural networks are trained and tested with the real RSS data and the theoretical radio map data. In the end, the particle filter is evaluated to mitigate the GPS spoofing.

1) KRIGING BASED RADIO MAP CONSTRUCTION

We evaluate both OK and UK with five kinds of kernels for radio map construction to find the best method and kernel with the lowest stand error and the shortest time consumption. The results are summarized in Table.2 and Fig.7.

Table.2 illustrates that the UK with exponential kernel has the best performance for radio map construction. Compared with the OK, the UK shows better performance on RSS data interpolation because of the deterministic trends of radio propagation. In addition, the exponential kernel has a smaller standard error than the other kernels, due to the fact that the radio large-scale fading follows the Log model. Furthermore, the standard error is decreasing with increasing the number of ray tracing, because the accuracy of the kriging interpolation depends on the number of initialization samples and the more samples with the more accurate prediction. Moreover, it is notable that the train time is increasing with the initialization samples. Although the linear kernel needs less time for fitting the samples, it makes more errors in predictions. Thus, the

TABLE 2. Kriging-based radio map construction.

Kriging method	Kernel	Standard error					Train time (millisecond)				
		10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
OK	linear	180	160	139	121	100	**497**	**3984**	**14425**	**38940**	**76388**
	power	160	147	140	120	97	1471	10378	33347	81252	148322
	gaussian	156	318	375	174	191	857	6749	20592	61628	123360
	spherical	**97**	78	55	**44**	39	1275	7689	26060	60334	111674
	exponential	105	**58**	**46**	**44**	**28**	816	6231	22392	49532	115916
UK	linear	183	160	139	121	100	**552**	**3918**	**15124**	**36094**	**72582**
	power	159	154	138	121	101	1443	10385	33964	75964	153501
	gaussian	210	176	428	331	363	1035	7267	24215	66108	117023
	spherical	94	76	**56**	50	36	1315	7913	27431	50692	117705
	exponential	**89**	**58**	**56**	**35**	**28**	835	6426	21816	55222	106336

UK with an exponential kernel is finally adopted for radio map construction.

Fig.7 demonstrate the UK with an exponential kernel for radio map construction, which includes sampling and Kriging interpolating. Fig.7(a) show the sampling that generates the positions in the 3D city environment randomly and then uses ray tracing and the deterministic model to compute the RSS values for those positions. In Fig.7(b), the UK with an exponential kernel is trained on those samples and predicts the RSS values for the unknown positions. It can be observed from Fig.7(b) that the use of Kriging can achieve the same results as ray tracing in Fig.6(b). Fig.7(c) depicts the fine-grained 3D radio map using the UK with an exponential kernel, where the radio map focuses on the UAV movement region with a higher resolution. The fine-grained 3D radio map can help to increase the GPS spoofing detection and mitigation performance as well as decrease the ray tracing and Kriging region.

2) NEURAL NETWORKS BASED GPS SPOOFING DETECTION

Neural networks based GPS spoofing detection approaches are evaluated with the train and test data set in terms of the model accuracy and training time, as shown in Fig.8 and Fig.9. Specifically, the GPS spoofing detection performance is not only influenced by input data size or step size but also by the system GPS spoofing margin, where the input data size is the number of data points in statistic analysis for MLP while the step size is the number of new data points in the input sequence of CNN and RNN (e.g., the LSTM). Additionally, the model training and updating time also has impacts on GPS spoofing detection performance, since a higher model training or updating latency may hinder the GPS spoofing detection on time and increase the collision risk amongst UAVs.

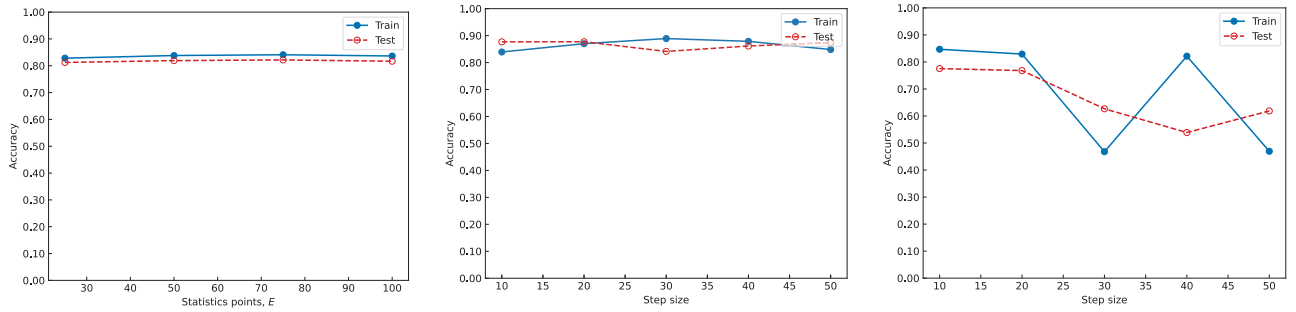
Fig.8 illustrates the accuracy of neural networks on GPS spoofing detection. Fig.8(a) shows the input data size has insignificant impacts on the MLP model because the statistic processing can extract the spoofing features from the raw data. In Fig.8(b) and Fig.8(c), the step size shows fewer impacts on the CNN but more on the LSTM. The reason is that CNN uses convolutional layers to extract the deep features while LSTM leverage the bidirectional layers to excerpt the temporal features, the temporal features influenced by the environment more than the deep features. In addition, MLP accuracy is increasing but the CNN and LSTM accuracy is decreasing when turning up the spoofing margin. In practice, the statistics mitigate the environmental dynamics while the convolutional layers and the bidirectional layers record the environmental changes. In addition, LSTM has overfitting with a bigger step size or spoofing margin, due to the fact that LSTM needs more data for training but a bigger step reduces the data points in train data.

Fig.9 summarizes the average training time spent by MLP, CNN, and LSTM for 10 rounds. Expressly, the MLP training time also includes statistical processing. Remarkably, it can be observed from Fig.9 that CNN spends less time than MLP and LSTM. In addition, the CNN training time can be further reduced by using transfer learning [38]. Overall, CNN can achieve comparable accuracy as well as model training time for detecting GPS spoofing attacks on cellular-connected UAVs.

3) PARTICLE FILTER BASED GPS SPOOFING MITIGATION

We evaluate the particle filer-based GPS spoofing mitigation method and its performance in Fig.10 and Fig.11.

Fig.10 depicts the GPS spoofing mitigation processes, where the GPS spoofer deviates the UAV from its intended path and the particle filter is utilized to estimate the UAV's true position. Specifically, the GPS signal is considered as not spoofed when the UAV is at its starting position or the distance between the actual position and the intended position is within the spoofing margin. As evident from Fig.10, the particle filter can effectively mitigate GPS spoofing and relocate the UAV close to its actual position even with significant GPS deviations. This is because the particle filter utilizes both the theoretical and real Received Signal Strength (RSS) values to estimate the UAV's position. Rather than following the planned positions, the mitigated positions are positioned



(a) Data size on MLP accuracy with $d_E = 10$. (b) Step size on CNN accuracy with $d_E = 10$. (c) Step size on LSTM accuracy with $d_E = 10$.

(d) GPS error on MLP accuracy. (e) GPS error on CNN accuracy. (f) GPS error on LSTM accuracy.

FIGURE 8. The accuracy performance of MLP, CNN, and LSTM on spoofing detection. The input sequence size is 10 x 10 for the CNN and 100 x 1 for the LSTM and the GPS error on MLP accuracy uses 100 x 1 data points for statistical analysis, distinctively.

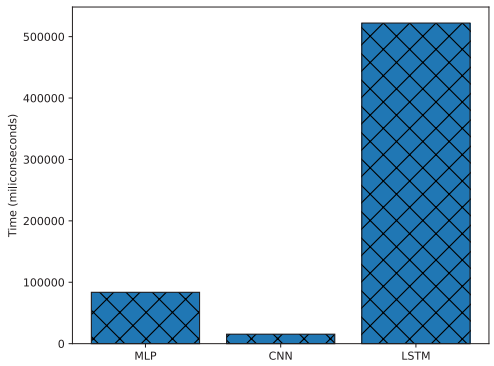


FIGURE 9. Average training time for 10 rounds with *dE* = 10.

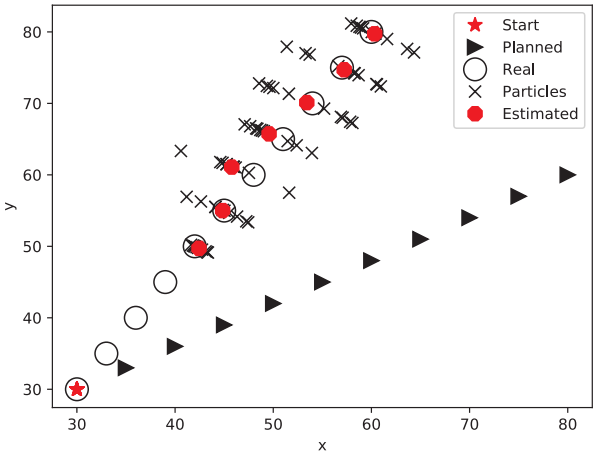
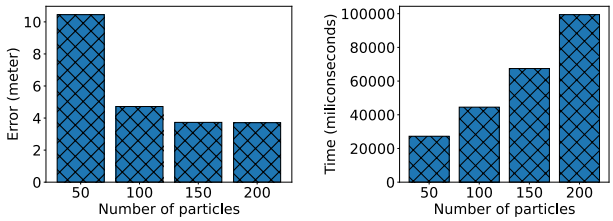


FIGURE 10. GPS spoofing mitigation with *N* = 100 and *dE* = 10(*m*) .

around the UAV's actual location. This is crucial as accurately determining the UAV's true position is crucial not only for mitigating GPS spoofing but also for planning the recovery path. Instead of following the planned positions, the mitigated positions are positioned around the actual location of the UAV. This is crucial as accurately determining the UAV's true position is essential not only for mitigating GPS spoofing but also for planning the recovery path.

Fig.11 demonstrates the estimation error and the computation time of different particle filters. It can be observed from Fig.11(a) that the estimation error decreases from above 10 meters to below 4 meters with increasing the number of particles from 50 to 200. In addition, the estimation error converges to around 3.75 meters after using 200 particles. Indeed, the particle filter is based on a normal distribution that induces inevitable errors when sampling positions. Moreover, Fig.11(b) indicates the estimation time of particle filter has an exponential relationship with particles because the time complexity of the particle filer-based GPS spoofing is $O(N^2)$. Therefore, the particle filter is for mitigating the GPS spoofing rather than replacing the GPS localization method due to its limited accuracy and high latency.

In conclusion, the utilization of a trained CNN can enable timely and ongoing detection of GPS spoofing, while the particle filter can accurately locate the true position of the UAV within one minute with an estimation error of 10 meters. The integration of CNN and particle filter can deliver a



(a) GPS spoofing mitigation error (b) GPS spoofing mitigation time.

FIGURE 11. GPS spoofing mitigation performance.

high-performance GPS spoofing detection and mitigation system. To guide the UAV to its planned path, a new recovery path planning is necessary, which will be addressed in our future work. While the particle filter offers numerous benefits, it has a drawback that it relies on the UAV being connected to the cellular network in order to obtain the radio map data and real-time measurements. To address this limitation, it is worth exploring alternative methods for UAV navigation that do not rely on GPS. In this regard, we will elaborate more on non-GPS navigation including the secure sensor fusion method in our future work.

VII. CONCLUSION AND FUTURE WORK

This paper investigates radio map and machine learning-based GPS spoofing detection and mitigation methods for cellular-connected UAVs. Firstly, the edge UFC or the UAV leverages ray tracing tools, deterministic channel models and the Kriging method to construct a 3D radio map. Then, machine learning methods, such as MLP, CNN and RNN, are used to detect the GPS spoofing by analyzing the real time RSS values reported by the base station or the UAV and the theoretical RSS values provided by the 3D radio map. Once the spoofing detection, the particle filter is applied to relocate the UAV and find the true position. The experiment results indicate that the Universal Kriging (UK) with an exponential kernel is the best radio map construction method regarding standard errors. Moreover, the MLP achieves the best spoofing detection accuracy because of the use of statistic features while the CNN needs less training time since the CNN inputs are raw RSS data. Furthermore, the particle filter can recover the UAV position to its real position within an error of 10 meters in one minute.

Despite the radio map demonstrating effectiveness in GPS spoofing detection and mitigation, the proposed solution is only for cellular-connected UAVs in an urban canyon environment. However, the cellular-connected UAV could be in different environments with different connections. Because radio map construction consumes a lot of computation and storage resources, it is difficult to build a radio map in a large region within an edge server. In the future, we will elaborate more on radio map combinations across different edge servers and build a digital twin platform that focuses on both data-driven and model-driven radio maps for seamlessly monitoring the UAV and UAV swarms. In the future, we will provide detailed information on the recovery path planning process that leverages the mitigated true position and the INS-based navigation to guide the UAV back to its intended trajectory.

REFERENCES

[1] P. M. Ghari, M. Sabbaghian, and H. Yanikomeroglu, "Moving aerial anchors assisted network localization," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 7839–7851, Oct. 2022.

[2] Y. Li, A. H. Aghvami, and D. Dong, "Path planning for cellular-connected UAV: A DRL solution with quantum-inspired experience replay," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 7897–7912, Oct. 2022.

[3] Y. Zhi, Z. Fu, X. Sun, and J. Yu, "Security and privacy issues of UAV: A survey," *Mobile Netw. Appl.*, vol. 25, no. 1, pp. 95–101, Feb. 2020.

[4] K.-Y. Tsao, T. Girdler, and V. G. Vassilakis, "A survey of cyber security threats and solutions for UAV communications and flying ad-hoc networks," *Ad Hoc Netw.*, vol. 133, Aug. 2022, Art. no. 102894.

[5] S.-H. Seo, G.-I. Jee, and B.-H. Lee, "Spoofing signal generation based on manipulation of code delay and Doppler frequency of authentic GPS signal," *Int. J. Control, Autom. Syst.*, vol. 19, no. 2, pp. 1026–1040, Feb. 2021.

[6] M. Foruhandeh, A. Z. Mohammed, G. Kildow, P. Berges, and R. Gerdes, "Spotr: GPS spoofing detection via device fingerprinting," in *Proc. 13th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Jul. 2020, pp. 242–253.

[7] M. Jayaweera, "A novel deep learning GPS anti-spoofing system with DOA time-series estimation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 1–6.

[8] H. Sathaye, G. LaMountain, P. Closas, and A. Ranganathan, "SemperFi: Anti-spoofing GPS receiver for UAVs," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2022, pp. 1–17.

[9] K. Zhang, E. G. Larsson, and P. Papadimitratos, "Protecting GNSS open service navigation message authentication against distance-decreasing attacks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 2, pp. 1224–1240, Apr. 2022.

[10] D. Y. Jeon, T. Gaybullaev, J. H. Noh, J. M. Joo, S. J. Lee, and M.-K. Lee, "Performance analysis of authentication protocols of GPS, Galileo and BeiDou," *J. Positioning, Navigat., Timing*, vol. 11, no. 1, pp. 1–9, 2022.

[11] Z. Wu, R. Liu, and H. Cao, "ECDSA-based message authentication scheme for BeiDou-II navigation satellite system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 4, pp. 1666–1682, Aug. 2019.

[12] Z. Wu, Y. Zhang, and R. Liu, "BD-II NMA&SSI: An scheme of anti-spoofing and open BeiDou II D2 navigation message authentication," *IEEE Access*, vol. 8, pp. 23759–23775, 2020.

[13] M. Nicola, B. Motella, M. Pini, and E. Falletti, "Galileo OSNMA public observation phase: Signal testing and validation," *IEEE Access*, vol. 10, pp. 27960–27969, 2022.

[14] K.-C. Kwon and D.-S. Shim, "Performance analysis of direct GPS spoofing detection method with AHRS/accelerometer," *Sensors*, vol. 20, no. 4, p. 954, Feb. 2020.

[15] E. Basan, A. Basan, A. Nekrasov, C. Fidge, N. Sushkin, and O. Peskova, "GPS-spoofing attack detection technology for UAVs based on Kullback–Leibler divergence," *Drones*, vol. 6, no. 1, p. 8, Dec. 2021.

[16] W. Liang, K. Li, and Q. Li, "Anti-spoofing Kalman filter for GPS/rotational INS integration," *Measurement*, vol. 193, Apr. 2022, Art. no. 110962.

[17] H. Zhu, M. L. Cummings, M. Elfar, Z. Wang, and M. Pajic, "Operator strategy model development in UAV hacking detection," *IEEE Trans. Human–Mach. Syst.*, vol. 49, no. 6, pp. 540–549, Dec. 2019.

[18] A. A. Cabrera-Ponce and J. Martinez-Carranza, "Aerial geo-localisation for MAVs using PoseNet," in *Proc. Workshop Res., Educ. Develop. Unmanned Aerial Syst. (RED UAS)*, Nov. 2019, pp. 192–198.

[19] K. Amer, M. Samy, M. Shaker, and M. ElHelw, "Deep convolutional neural network based autonomous drone navigation," *Proc. SPIE*, vol. 11605, Jan. 2021, Art. no. 1160503.

[20] Y. Dang, C. Benzaïd, Y. Shen, and T. Taleb, "GPS spoofing detector with adaptive trustable residence area for cellular based-UAVs," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

[21] Y. Dang, C. Benzaïd, B. Yang, T. Taleb, and Y. Shen, "Deep-ensemble-learning-based GPS spoofing detection for cellular-connected UAVs," *IEEE Internet Things J.*, vol. 9, no. 24, pp. 25068–25085, Dec. 2022.

[22] M. Meles, A. Rajasekaran, K. Ruttik, R. Virrankoski, and R. Jäntti, "Measurement based performance evaluation of drone self-localization using AoA of cellular signals," in *Proc. 24th Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, Dec. 2021, pp. 1–5.

[23] M. Meles, L. Mela, A. Rajasekaran, K. Ruttik, and R. Jäntti, "Drone localization based on 3D-AoA signal measurements," in *Proc. IEEE 95th Veh. Technol. Conf. (VTC-Spring)*, Jun. 2022, pp. 1–5.
[24] *Study on Supporting Unmanned Aerial Systems (UAS) Connectivity, Identification and Tracking*, document 3GPP TS 23.754, Dec. 2018.
[25] S. Zhang and R. Zhang, "Radio map-based 3D path planning for cellular-connected UAV," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1975–1989, Mar. 2021.
[26] H. Xie, D. Yang, L. Xiao, and J. Lyu, "Connectivity-aware 3D UAV path design with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13022–13034, Dec. 2021.
[27] Y. Dong, C. He, Z. Wang, and L. Zhang, "Radio map assisted path planning for UAV anti-jamming communications," *IEEE Signal Process. Lett.*, vol. 29, pp. 607–611, 2022.
[28] Y. Zeng, X. Xu, S. Jin, and R. Zhang, "Simultaneous navigation and radio mapping for cellular-connected UAV with deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4205–4220, Jul. 2021.
[29] P. Zeng and J. Chen, "UAV-aided joint radio map and 3D environment reconstruction using deep learning approaches," in *Proc. ICC - IEEE Int. Conf. Commun.*, May 2022, pp. 5341–5346.
[30] J. Chen, U. Yatnalli, and D. Gesbert, "Learning radio maps for UAV-aided wireless networks: A segmented regression approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.
[31] R. Deng, Z. Jiang, S. Zhou, S. Cui, and Z. Niu, "A two-step learning and interpolation method for location-based channel database construction," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.
[32] Q. Zhu et al., "DEMO abstract: An UAV-based 3D spectrum real-time mapping system," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2022, pp. 1–2.
[33] A. E. C. Redondi, "Radio map interpolation using graph signal processing," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 153–156, Jan. 2018.
[34] H. Ramchoun, M. Amine, J. Idrissi, Y. Ghanou, and M. Ettaouil, "Multilayer Perceptron: Architecture optimization and training," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, no. 1, p. 26, 2016.
[35] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.
[36] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.
[37] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
[38] Y. Dang, C. Benzaïd, T. Taleb, B. Yang, and Y. Shen, "Transfer learning based GPS spoofing detection for cellular-connected UAVs," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, May/Jun. 2022, pp. 629–634.
[39] FAA. (Jan. 2021). *Unmanned Aircraft System Traffic Management, Federal Aviation Administration (FAA)*. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/unmanned-aerial-vehicles-uav-market-662.html
[40] *Propagation by Diffraction*, document Recommendation ITU-R P. 526-14, P. Series, 2018.
[41] K. Haneda et al., "5G 3GPP-like channel models for outdoor urban microcellular and macrocellular environments," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–7.
[42] *Enhanced LTE Support for Aerial Vehicles*, document 3GPP TR 36.777, Dec. 2017.
[43] A. Couturier and M. A. Akhloufi, "A review on absolute visual localization for UAV," *Robot. Auto. Syst.*, vol. 135, Jan. 2021, Art. no. 103666.
[44] Indu and R. Singh, "Trajectory planning and optimization for UAV communication: A review," *J. Discrete Math. Sci. Cryptogr.*, vol. 23, no. 2, pp. 475–483, Feb. 2020.
[45] M. A. Oliver and R. Webster, "Kriging: A method of interpolation for geographical information systems," *Int. J. geographical Inf. Syst.*, vol. 4, no. 3, pp. 313–332, Jul. 1990.
[46] D. Mao, W. Shao, Z. Qian, H. Xue, X. Lu, and H. Wu, "Constructing accurate radio environment maps with Kriging interpolation in cognitive radio networks," in *Proc. Cross Strait Quad-Regional Radio Sci. Wireless Technol. Conf. (CSQRWC)*, Jul. 2018, pp. 1–3.
[47] P.-W. Son, J. H. Rhee, J. Hwang, and J. Seo, "Universal Kriging for Loran ASF map generation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 4, pp. 1828–1842, Aug. 2019.
[48] S. Bi, J. Lyu, Z. Ding, and R. Zhang, "Engineering radio maps for wireless resource management," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 133–141, Apr. 2019.
[49] J. Elfring, E. Torta, and R. van de Molengraft, "Particle filters: A hands-on tutorial," *Sensors*, vol. 21, no. 2, p. 438, Jan. 2021.
[50] Y. Li, N. Li, and C. Han, "Ray-tracing simulation and hybrid channel modeling for low-terahertz UAV communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.
[51] S. Tamsri. (2022). *Radio Gyms*. [Online]. Available: https://github.com/intelek-ai/radio-gyms
[52] K. Haneda et al., "Radio propagation modeling methods and tools," in *Inclusive Radio Communications for 5G and Beyond*. Amsterdam, The Netherlands: Elsevier, 2021, pp. 7–48.