Remote ID Spoofing Attacks and Defenses

Bryce Bjorkman, Stanley Zheng, Austin Coursey, Cailani Lemieux-Mack, Samuel Gonzalez, Abel Diaz-Gonzalez, Noah Dahle, Neils Koroma, Robert Canady, Xenofon Koutsoukos, Gautam Biswas, Abenezer Taye, and Bryan C. Ward
Vanderbilt University, Nashville, TN, USA

As the use of unmanned aerial vehicles (UAVs) increases, so does the importance of ensuring their cybersecurity. As new requirements and regulations for their use are introduced, new cybersecurity challenges emerge. One such recent example in the U.S. is the requirement of small UAVs to broadcast their Remote Identification, a message containing information about the UAV and its operator. This message can be used by authorities to identify suspicious UAVs and by nearby UAVs as cheap sensory information about neighboring UAVs for tasks like collision avoidance. However, this message is only useful if its contents can be trusted. In this paper, we study the potential of Remote ID to be used for cyberattacks by spoofing their Remote ID message. We simulate multiple UAVs flying in an urban area, broadcasting their Remote IDs, and receiving other UAVs' Remote IDs in the OMNeT++ network discrete event simulation that simulates the dynamics of the Remote ID signal. We demonstrate how an adversarial UAV may broadcast a spoofed Remote ID, causing another UAV to perceive it in the wrong location. Additionally, we show that by analyzing the strength of the spoofed signal, it is possible to determine that it is spoofed.

I. Introduction

Small unmanned aerial systems (sUAS) are being increasingly used in daily applications such as precision agriculture [1], search and rescue [2], urban traffic monitoring [3], delivery [4], and more. Their popularity comes from their ability to support aerial tasks while being low-cost, versatile, agile, and easy to deploy [5]. Along with the growing number of sUAS flying in our skies, there is also an increasing need to ensure their cybersecurity [6]. Under a cyberattack, the aircraft may fail to complete their mission or lose communication with their operator, such as in a denial of service attack [7]. In situations where the aircraft is performing a critical mission, like medical supply delivery [8], this can impact human lives. Moreover, some cyberattacks may aim to crash the aircraft [9, 10], damaging expensive equipment and property, and even risking pedestrian safety.

With these challenges in mind, the topic of sUAS cybersecurity has been widely studied [6]. Much of this work has focused on ensuring sUAS and their communication systems are secure. These build security into the system design, such as in [11], where they present two case studies where 5G communication is secured using physical layer security mechanisms. Other examples include [12], where they introduce a security layer to encrypt MAVLink packets, and [13], where they introduce a trusted execution environment for federated learning using sUAS. Knowing that not all real systems are secure, many works have focused on attack classification and detection. In [14], they classify sUAS cyberattacks into hardware, software, sensor, network, and communication attacks. To detect attacks in these categories, recent approaches have used techniques like modified sliding innovation sequences [10], parameter distribution analysis [15], and machine learning [16]. Finally, others have considered how to defend against an attack when it is occurring. For example, [17] proposes a cyber defense strategy that formulates a UAS-edge computing network as a game.

While the rich literature on sUAS cybersecurity aims to address sUAS security concerns, as new regulations and use cases of sUAS emerge, so will new cybersecurity challenges not covered by current studies. One such regulation in the U.S., and the focus of this study, is drone Remote Identification (remote ID) [18]. Remote ID requires that drones locally broadcast a radio message containing information about the operator and the drone. These messages, concretely defined in Section II, are primarily intended for drone and operator identification by the Federal Aviation Administration (FAA) and law enforcement. If a drone is suspicious, unsafe, or flying in an incorrect area, the authorities can locate its control station and contact its operator. Since this broadcast also includes the position and velocity of the drone, it can also be used to sense surrounding drones. While LiDAR sensors are getting cheaper and lighter [19], they still may not be practical for small or budget sUAS. At the same time, cameras require a line of sight and computationally demanding computer vision algorithms to detect objects like nearby drones [20]. Additionally, cameras are heavily impacted by weather conditions [19]. Therefore, some sUAS operators may want to leverage local remote ID radio broadcasts to spatially isolate surrounding drones for tasks such as collision avoidance. The abilities of authorities

to locate a suspicious drone's operator or aircraft to effectively use the remote ID as an additional sensor rely on an assumption that *every drone will broadcast the correct remote ID message*. In situations where a sUAS is compromised or a malicious operator is flying a drone, this assumption may not hold.

In this paper, we study attacks and defenses of sUAS remote ID spoofing, where an attacker broadcasts an incorrect and malicious remote ID message to influence the behavior of surrounding drones or to hide their actual behavior. We simulate spoofed remote IDs broadcast over Wi-Fi and received by nearby aircraft's antennas for collision avoidance using OMNeT++, a network discrete-event simulator. This offers a simplified model of sUAS dynamics with a realistic simulation of the physical propagation of the remote ID radio signals. Using this, we demonstrate the ability of an attacker to spoof their remote ID and how this spoofed remote ID and how dynamics of the signal, such as signal strength, could be used to try and detect if a drone is spoofing their remote ID.

The contributions of this paper are as follows.

1) We develop an OMNeT++ network discrete event simulation for drone remote ID broadcasting and receiving.
2) We demonstrate how an attacker may spoof their remote ID and how this attack can be detected based on the message's signal strength. We apply multilateration and filtering methods as well as a simple multilayer perceptron.
3) We develop a data generation pipeline and present a rich data set for urban Remote ID Spoofing detection.

The remainder of this paper is structured as follows. In Section II, we provide the details of remote ID and provide some background into signal spoofing and collision avoidance. In Section III, we define the assumptions and threat model. In Section IV, we detail the design of our OMNeT++ remote ID simulation as well as the generation of our urban Remote ID spoofing data set. In Section V, we describe our detection approach and the intuition behind it. In Section VI, we evaluate our detection methods. In Section VII, we describe future work on this topic. Finally, in Section VIII, we conclude the paper.

II. Background

Wireless signal spoofing has been studied in-depth in contexts such as the Global Positioning System (GPS) [21], including specifically for UAVs [22]. The FAA has a regulation requiring small UAVs to be equipped with Remote ID [18]. The ASTM has a standard defining how this should be done [23]. A typical UAV deployed in urban airspace is not currently required by the FAA to perform object detection, and may not necessarily have onboard systems dedicated to object detection due to the financial and physical costs of the sensors and processing power such systems require. As such, a typical UAV in this context may rely on Remote ID as input for a collision avoidance algorithm such as Force Field Protocol [24].

Signal Spoofing and Detection In autonomous systems, signal-spoofing attacks are a form of sensor attack where a malicious actor sends falsified signals to compromise the target system. These attacks have been well documented with GPS Spoofing [21], which have been shown to be able to compromise a target UAV [22].

Detection of GPS spoofing attacks has included array methods where there are multiple GPS receivers that can make simultaneous measurements and coordinate their results making it more difficult to convincingly deceive all of them at the same time [21]. This can also be simulated with a single receiver that is moved over time forming a synthetic array.

Alternatively, if there are external methods for measuring position, it might be possible to verify the received GPS data [21]. It is common that this verification is done against Inertial Measurement Unit (IMU) data in UAVs [25].

An advanced detection mechanism involves the observation that the source of spoofed signals is much closer than the normal GPS satellites. Thus, by measuring the signal strength of each received GPS message and measuring how that signal strength changes with the movement of the UAV, it can be observed that the variation in the signal strength is greater for spoofed signals due to the inverse square law [21].

In this work, we propose using a similar technique of measuring signal strength to detect Remote ID spoofing, instead of GPS signals.

Remote ID The FAA regulation requires that any UAV in urban airspace periodically broadcast a Remote ID message, which must contain each of the following elements, per § 14 CFR 89.305 [18].

Remote Identification Message Requirements

(a) The identity of the unmanned aircraft, consisting of:
    (1) A serial number assigned to the unmanned aircraft by the person responsible for the production of the standard remote identification unmanned aircraft; or
    (2) A session ID.
(b) An indication of the latitude and longitude of the control station.
(c) An indication of the geometric altitude of the control station.
(d) An indication of the latitude and longitude of the unmanned aircraft.
(e) An indication of the geometric altitude of the unmanned aircraft.
(f) An indication of the velocity of the unmanned aircraft.
(g) A time mark identifying the Coordinated Universal Time (UTC) time of applicability of a position source output.
(h) An indication of the emergency status of the unmanned aircraft.

The Remote ID broadcast must be on a radio frequency readable by personal devices, with a period no greater than one second, and with a maximum latency of one second from the time of location measurement. The options for wireless communication protocols are Bluetooth and Wi-Fi. When using Wi-Fi, the broadcasts can be done using either Neighbor Awareness Networking (NAN) or Beacon Frames (BF). The NAN approach relies on Wi-Fi Direct and may be more efficient for environments with multiple UAVs, while the BF approach is simpler to implement.

Importantly, Remote ID and similar technologies are required for the purpose of collision avoidance [24] as well as attribution in cases of law violations [26]. Many drones do not have the sensors or computational power to be able to observe nearby drones, and thus rely on Remote ID for collision avoidance.

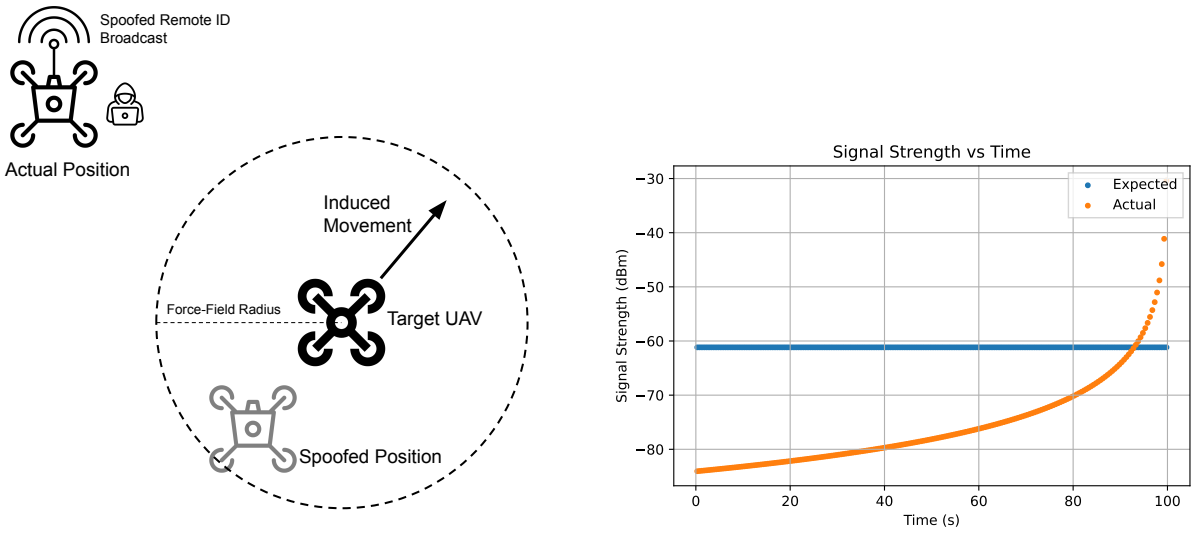
Collision Avoidance In urban environments or otherwise crowded airspaces, it is critical that the UAVs do not collide with each other, as that would endanger the vehicle, its mission, and any surrounding people or infrastructure. While things like buildings and trees do not move and thus can be planned around, UAVs operated by external entities may not be easily accounted for. Thus, collision-avoidance systems are an important piece of any safety-critical, urban UAV deployment.

In collision avoidance, there are two important problems to solve. The first is to know where all of the other UAV's in the area are and where they are going (i.e., the positions and velocities). The other problem is to find an acceptable course of action for the UAV such that it avoids a collision.

To solve the first problem, there are multiple solutions. There are camera-based techniques that involve computer-vision models [27]. While this solution is relatively inexpensive in terms of hardware costs, there is significant computational overhead in running vision models, which may be infeasible on many platforms given size, weight, and power (SWaP) constraints. Additionally, there are active-sensing approaches such as using LiDAR [28, 29], but such sensors are often expensive and may not be applicable to the majority of UAV applications. Alternatively, with wide enough cooperation, it is possible to adopt a UAV-to-UAV communication approach where each UAV self-reports its position and velocity in order to facilitate collision avoidance [24]. This approach is inexpensive in both hardware (wireless radios are already onboard many drones) and software. However, it assumes that the radio messages, such as Remote ID, are trusted. Thus, in this work, we consider the potential effects of spoofed messages and how to detect them.

To solve the second problem of collision avoidance, there are again many existing solutions. Among these are geometric methods, force-field methods, optimization methods, sense and avoid approaches, model predictive control methods, and more [29, 30]. One solution that uses both Remote ID and force-field collision avoidance is presented in [24]. Force-field approaches use a "virtual force function" to keep UAVs from getting too close to one another. If one UAV gets too close to another, a virtual force is calculated that will "push" them away from each other. Each UAV is expected to apply that force to itself in order to maintain spacing.

Force-field collision avoidance methods can be exploited using spoofed Remote IDs to take control of a target UAV. We present a simplified example in Figure 1a. In this example, a malicious attacker uses Remote ID spoofing to convince the target UAV that it is within the force-field radius. Thus, the target applies a virtual force on itself in order to avoid a possible collision. In this way, the attacker can induce a specific movement in the target. With sufficient prior knowledge of the specific collision avoidance technique and continued spoofing, it can be seen that the attacker is able to



(a) Remote ID spoofing attack on collision avoidance

(b) RSSI anomaly from spoofed position

Fig. 1 Motivating Remote ID spoofing threat and basis for detection. (a) An attacker uses spoofed Remote ID to induce movement in a target UAV via force-field collision avoidance. (b) When the actual transmitter position differs from the claimed position, the observed RSSI deviates from what would be expected if the claimed position were accurate.

induce arbitrary movement in the target. Thus, in order to be able to safely use Remote ID as an obstacle-detection mechanism, there must be a classification of Remote ID spoofing threats as well as detection mechanisms.

Motivating Observation Using our customized simulation environment, we produced preliminary results for a scenario in which an adversary broadcasts that its position is tracking in parallel to a detector drone, when in reality its position is actually tracking perpendicular to the detector drone and is on a collision course.

We recorded the signal strength of spoofed Remote ID messages over time (the actual signal strength) as well as a baseline of what the signal strength would have been if the Subject UAV was actually where the Remote ID claims them to be (the expected signal strength). Figure 1b shows these recorded signal strengths over time. We can see that the actual signal strength of the spoofed messages (orange) has a much greater variation than the expected strength (blue). This is because the actual broadcast source is moving closer to the detector instead of being a constant distance away.

Such discrepancy between actual signal dynamics and the expectation implied by Remote ID messages forms the basis for the detection approaches studied in this paper.

III. Threat Model

We consider UAVs that fly at altitudes less than 400 feet and are equipped with Wi-Fi radios using omnidirectional antennas. We assume a high-density urban environment with many drones, each needing to avoid collisions, and employing Remote ID for such purpose.

We consider an attacker that is spoofing Remote ID messages. As described previously, spoofing could be for the purpose of diverting other vehicles to off course (or even into a collision with another drone), or for concealing the malicious drone's own flight patterns. There are many other attacker goals that could be achieved through spoofing, but the details of such attacker goals are outside the scope of this paper. We merely consider an attacker that is trying to spoof Remote ID messages. Therefore, we assume that the attacker has control of a UAV with Remote ID broadcast capabilities. Specifically, we assume the attacker has control of the mobility of the UAV, the timing of its Remote ID broadcasts, and the contents of its Remote ID packets, Our goal in this work is to be able to simulate the signal dynamics of remote ID messages, and develop approaches to be able to use the radio signal dynamics to be able to detect whether Remote ID messages are spoofed or not.

IV. Design

We employ the OMNeT++ discrete-event simulator [31] with the INET Framework [32] to simulate Remote ID spoofing and detection. The INET Framework provides modules for IEEE 802.11 networking stacks, including physical-layer radio propagation models that compute received signal strength indicator (RSSI) based on distance, transmission power, and environmental factors.

A. Simulation Architecture

Our simulation is built around a class hierarchy for Remote ID beacon management. The base `RidBeaconMgmt` class extends INET's `Ieee80211MgmtApBase` to transmit and receive Wi-Fi beacon frames containing Remote ID data. Each beacon includes the transmitter's claimed position, velocity, heading, serial number, and timestamp. On reception, the module records RSSI and both the receiver's actual position and the claimed transmitter position from the message.

The `KalmanFilterDetectMgmt` class extends `RidBeaconMgmt` to implement online transmission (TX) power estimation. It maintains a per-transmitter Kalman Filter (keyed by serial number) using the Eigen linear algebra library. On each reception (RX), it computes the TX power measurement from RSSI and claimed distance, updates the Kalman Filter state, and records the Normalized Innovation Squared (NIS) for later analysis.

B. Spoofer Implementation

We implement a dynamic trajectory spoofer that impersonates another drone. The `DynamicTrajectorySpooferMgmt` class overrides the `fillRidMsg` method to populate Remote ID fields with a target drone's position rather than its own. The target is a "ghost" drone (`GhostHost`)—a module that flies a trajectory but has no wireless interface, representing a physical drone presence without RF emissions. At each beacon transmission, the spoofer queries the ghost's mobility module and broadcasts the ghost's current position while physically located elsewhere. This creates a realistic spoofing scenario where RSSI reflects the spoofer's actual position but the message claims the ghost's position.

C. Urban Environment Generation

To generate diverse evaluation scenarios, we developed a pipeline for creating urban airspace environments. The pipeline generates:

- **Flight corridors:** East-west and north-south corridors with configurable width and spacing, representing designated flight paths between buildings.
- **Buildings:** Randomly placed obstacles that affect radio propagation, with configurable density and height distributions.
- **Trajectories:** Corridor-constrained UAV flight paths using INET's `TurtleMobility` module, with randomized waypoints, speeds, and altitudes.
- **Radio parameters:** Randomized transmission power, beacon intervals, and startup jitter to create realistic variation.

Figure 2 illustrates the urban environment generation process. Flight corridors (Fig. 2a) define the permissible airspace where UAVs may operate, modeling designated flight paths in urban air mobility scenarios. Buildings (Fig. 2b) are placed in the regions between corridors, affecting radio propagation through the INET obstacle loss model. Finally, UAV trajectories (Fig. 2c) are generated to navigate within the corridor network, with each drone assigned randomized speed and altitude parameters.

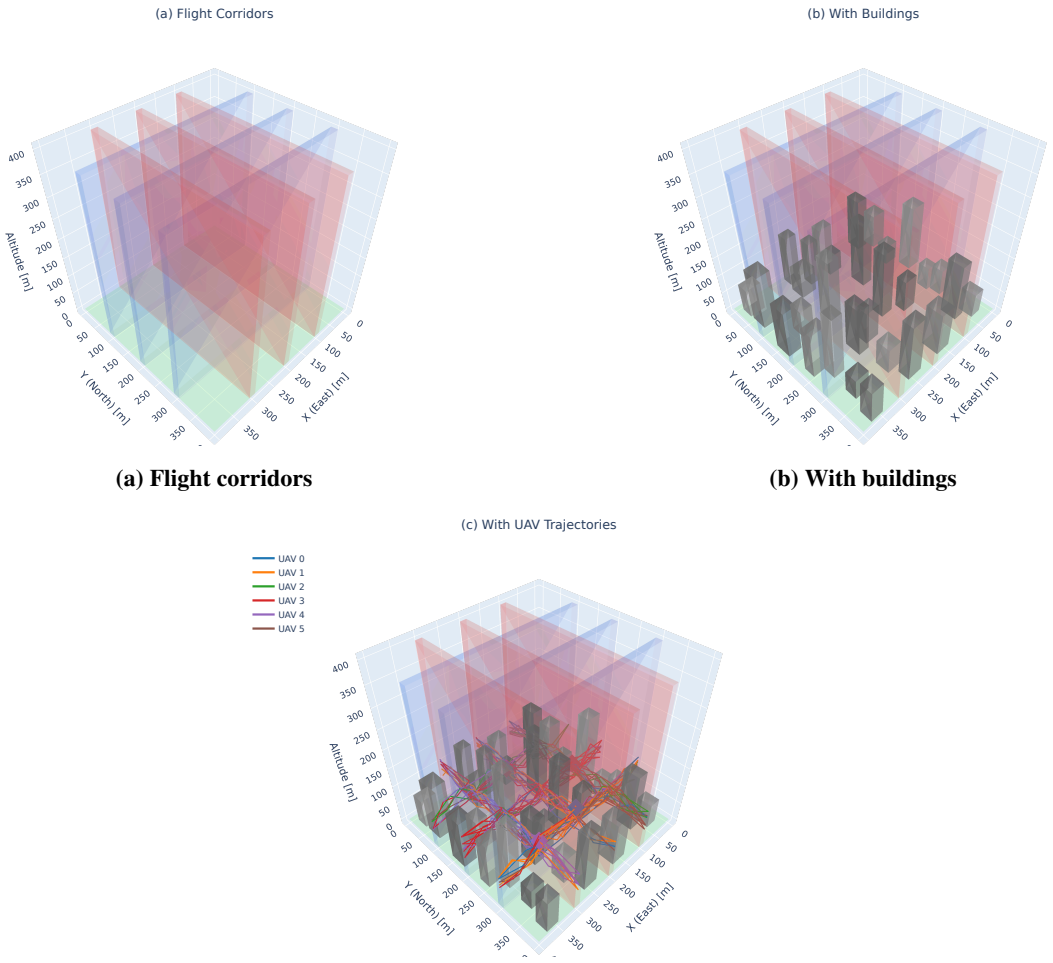
Each scenario configuration is recorded in a manifest file that enables reproducible regeneration of any dataset.

Figure 3 shows the complete data generation pipeline architecture, from parameter specification through final labeled CSV output.

D. Data Collection

The simulation outputs timestamped CSV files containing transmission (TX) and reception (RX) events. Each RX event includes the receiver's position, the claimed transmitter position from the Remote ID message, RSSI, and pre-computed KF state (estimate, covariance, NIS). Ground truth labels indicate whether each event originated from the spoofer. This format supports offline evaluation of detection algorithms without re-running simulations.

We note that the OMNeT++ simulator can present virtual network devices to external processes, enabling integration with our concurrent work on large-scale UAV simulations [33] to conduct spoofing simulations with physics-aware collision-avoidance algorithms.



(a) Flight corridors **(b) With buildings**

(c) With UAV trajectories

Fig. 2 Urban environment generation pipeline. East-west corridors (blue) and north-south corridors (red) define flyable airspace. Buildings (gray) are placed between corridors and affect radio propagation. UAV trajectories (colored lines) are constrained to corridors with randomized speeds and altitudes.

Source code is available at `https://github.com/brycethebjorkman/uli-net-sim` along with instructions for reproducing the results included in this paper.

V. Approach

As illustrated in Figure 1b, when a UAV spoofs its Remote ID position, the observed RSSI differs from what would be expected if the claimed position were accurate. If the detector knows the expected signal strength at the claimed distance, it can detect this discrepancy. However, the detector does not know the transmission power of the Remote ID messages, nor exactly how environmental factors like humidity or obstacles will affect the signal. Although the detector can expect that signal strength should be consistent with the claimed distance, it cannot know what that strength *should* be in absolute terms. Nevertheless, the *dynamics* of the signal strength over time can reveal spoofing—a spoofer claiming constant distance but actually approaching will exhibit increasing RSSI.

We hypothesize that for an adversary using fixed transmission power, spoofing can be detected due to the discrepancy between observed and expected signal strength dynamics. However, a sophisticated adversary with the ability to vary transmission power appropriately may be able to perform spoofing in a manner undetectable by a single UAV, requiring the cooperation of multiple UAVs to defend against.

With these observations in mind, we have implemented two deterministic rule-based detection methods that are

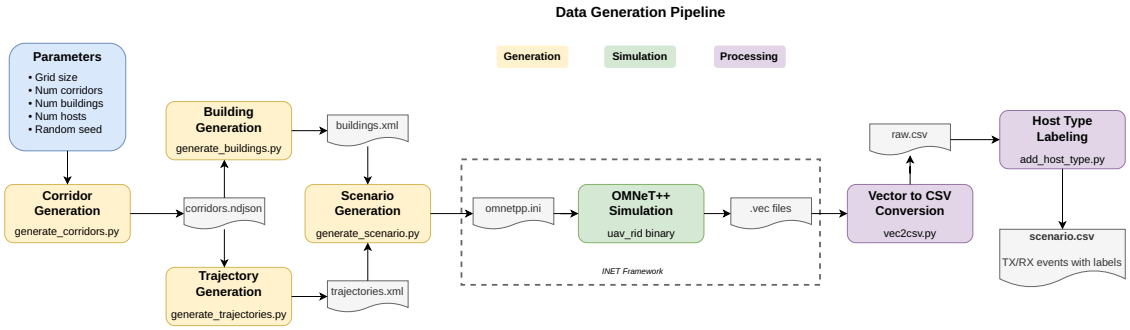


Fig. 3 Data generation pipeline. Parameters drive corridor, building, and trajectory generation. The resulting scenario configuration is simulated in OMNeT++/INET, and output vectors are converted to labeled CSV files for offline analysis.

based entirely on Remote ID message fields and RSSI. The first uses the received transmissions of one drone to update a Kalman Filter tracking the estimated transmission power for each drone that it receives a Remote ID transmission from. The second uses the received transmissions of four drones in a federation to perform multilateration for each drone that the federation receives a Remote ID transmission from. The Kalman Filter approach has the benefit of being computationally lightweight such that it can be performed onboard a SWaP-constrained UAV without requiring extra communication overhead. The federated multilateration approach has the advantage of combining more data from different observation points, and while it does require some additional communication, this may be accomplished by simply streaming the observations from each federate back to a common ground control station (GCS) where the multilateration computation occurs. Additionally, we have implemented a machine learning approach as a baseline for comparison.

A. Kalman Filter Transmission Power Estimation

The Kalman Filter (KF) detector uses a simple free space path loss model for 2.4GHz Wi-Fi to relate received signal strength indicator (RSSI) to distance:

$$RSSI = P_{tx} - 20 \cdot \log_{10}(d) - 40.04 \tag{1}$$

where P_{tx} is TX power in dBm and d is the distance between transmitter and receiver in meters. Given an RSSI and claimed distance d_{claimed} as computed from the difference between transmitter claimed position and receiver position, we can estimate the transmitter's TX power:

$$\hat{P}_{tx} = RSSI + 20 \cdot \log_{10}(d_{\text{claimed}}) + 40.04 \tag{2}$$

For an honest transmitter, the claimed position is accurate, so successive TX power estimates should be consistent. For a spoofer, the actual RSSI depends on the real distance, which differs from the claimed distance, causing the TX power estimate to be inconsistent.

We use a Kalman Filter [34] to track these TX power estimates over time. The state is the estimated TX power, $x = P_{\text{tx}}$, and each measurement corresponds to a single RX event. Since our goal is to detect changes in the transmitted signal power, we assume in the prediction phase that the transmitted power remains constant over time. Under this assumption, the Kalman Filter prediction and update equations become:

$$\hat{x}^- = \hat{x}, \quad P^- = P + Q \tag{3}$$

$$K = P^-(P^- + R)^{-1} \tag{4}$$

$$\hat{x} = \hat{x}^- + K(z - \hat{x}^-) \tag{5}$$

$$P = (1 - K)P^- \tag{6}$$

where Q is the process noise variance, R is the measurement noise variance, and z is the TX power measurement. In addition, $\hat{x}^-$ represents the predicted TX power at the next timestep, and $\hat{x}$ represents the estimated TX power at the

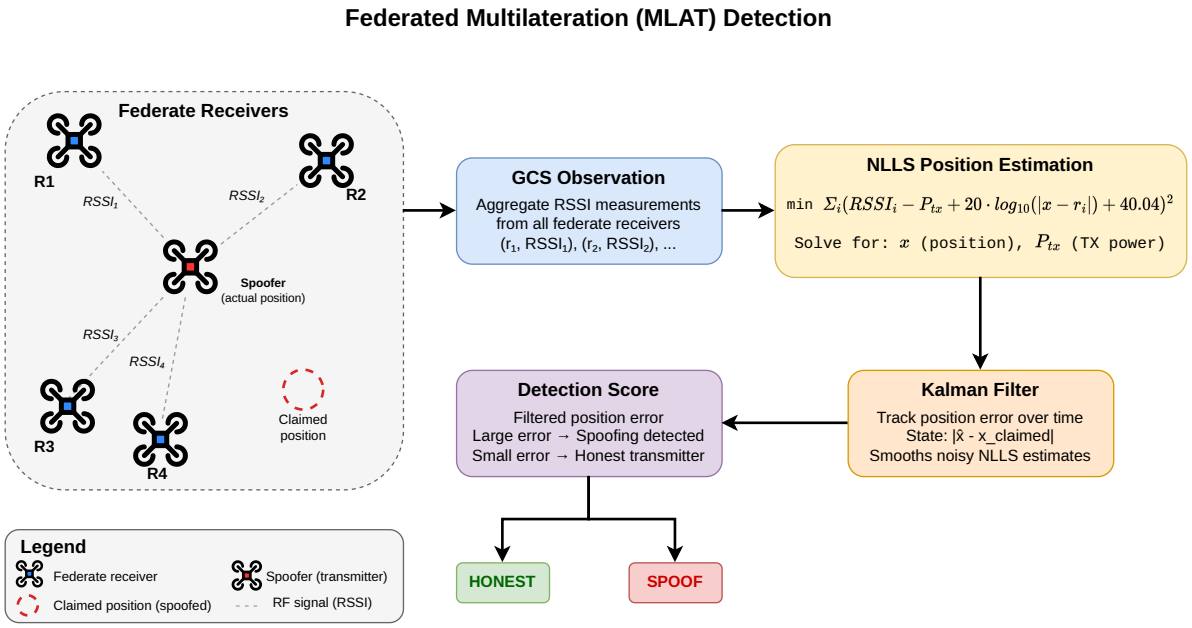


Fig. 4 Federated multilateration detection architecture. Multiple receivers measure RSSI from a transmitter and stream observations to a ground control station. The GCS performs nonlinear least squares position estimation, applies Kalman filtering to smooth the position error, and flags transmitters whose estimated position differs significantly from their claimed position.

current timestep. Equation 3 describes the *prediction* phase of the Kalman filter, while Equations 4, 5, and 6 represent the *correction* phase. Once all the parameters of the Kalman filter are obtained, we use the Normalized Innovation Squared (NIS), which is given by the following equation, as a detection statistic.

$$NIS = \frac{(z - \hat{x}^-)^2}{P^- + R} \tag{7}$$

Under the null hypothesis of an honest transmitter with correctly-specified model, NIS follows a $\chi^2(1)$ distribution. Large NIS values indicate model mismatch, suggesting spoofing.

B. Federated Multilateration

The Federated Multilateration (MLAT) detector uses RSSI measurements from multiple federate receivers to jointly estimate the transmitter's position and TX power, then compares to the claimed position. Figure 4 illustrates the MLAT detection architecture.

Given N federate receivers at known positions $\mathbf{r}_i$ with RSSI measurements $RSSI_i$, we solve for the transmitter position $\mathbf{x} = (x, y, z)$ and TX power P_{tx} using nonlinear least squares with the same path loss model as the KF detector (Eq. 1):

$$\min_{\mathbf{x}, P_{tx}} \sum_i \left(RSSI_i - P_{tx} + 20 \log_{10}(|\mathbf{x} - \mathbf{r}_i|) + 40.04 \right)^2 \tag{8}$$

This formulation avoids assuming the claimed position is correct when estimating TX power. With 4 or more receivers, the system is overdetermined (4 unknowns, $N \geq 4$ measurements).

The raw position estimates from multilateration are noisy, so we apply a Kalman Filter to track the position error magnitude over time for each transmitter (identified by serial number). The detection score is the filtered position error $|\hat{\mathbf{x}} - \mathbf{x}_{claimed}|$.

For an honest transmitter, the estimated position should be close to the claimed position (small filtered error). For a spoofer, the estimated position reflects the actual location based on RSSI, which differs significantly from the claimed position (large filtered error).

C. Machine Learning Baseline

We train a Multilayer Perceptron (MLP) neural network model as a data-driven, machine learning baseline. This model is a two-layer fully-connected neural network with ReLU activation functions. It takes in data from the four federate receiver drones to decide whether the current Remote ID signal is spoofed as a supervised binary classification task. To do so, it uses the following features

- RSSI (Eq. (1)),
- X Distance = $|x_{\text{federate}_i} - x_{\text{rid}}|$,
- Y Distance = $|y_{\text{federate}_i} - y_{\text{rid}}|$,
- Z Distance = $|z_{\text{federate}_i} - z_{\text{rid}}|$,

where $i = \{0, 1, 2, 3\}$ are the four federate drones reporting working together to detect spoofers, and rid is the data received from a Remote ID broadcast. Intuitively, the received signal strength of a Remote ID broadcast from a drone that is spoofing its position should conflict with the claimed position. For example, one that claims to be far away but has a high signal strength may be spoofing. By training a simple data-driven model solely on these features, we can determine whether this relationship holds in complex urban settings with many non-spoofing drones, buildings, and varying broadcast strengths.

As there are fewer spoofing drones than benign drones, this is an imbalanced classification problem, and we use the following weighted binary cross-entropy loss function.

$$\text{WBCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[w_i\, y_i\, \log(\hat{p}_i) + (1 - y_i)\, \log(1 - \hat{p}_i)\right] \tag{9}$$

where:

N is the number of samples in the batch,

$y_i \in \{0, 1\}$ is the true label of the i-th sample,

$\hat{p}_i \in [0, 1]$ is the predicted probability of the positive class for the i-th sample,

$$w_i = \begin{cases} \text{pos\_weight}, & \text{if } y_i = 1 \text{ (positive sample)}, \\ 1, & \text{if } y_i = 0 \text{ (negative sample)}. \end{cases}$$

Here, pos_weight is a scalar hyperparameter used to upweight the loss contribution from the minority (positive) class to mitigate class imbalance. To determine the value of this hyperparameter, we divide the number of negative (non-spoofed) samples by the number of positive (spoofed) samples.

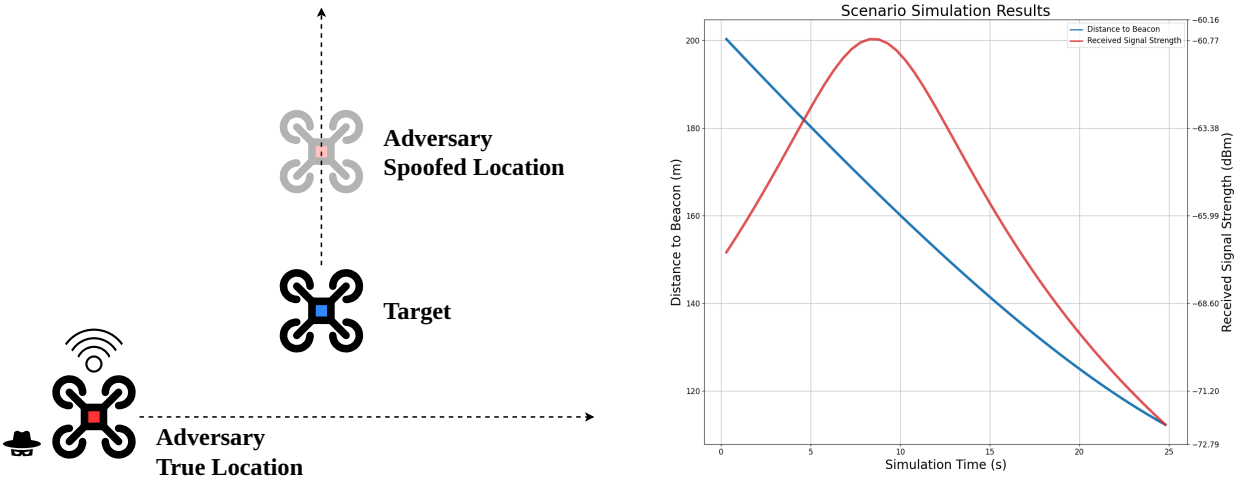
We trained the MLP on 500 scenarios from the training dataset presented in the following section. First, the training dataset was normalized. Then, the model was trained on an NVIDIA GeForce RTX 3060 GPU until convergence, for around 10 epochs over the shuffled ≈ 1.9 million sample training dataset. The model was trained with the following manually selected hyperparameters: hidden layer neurons = $[128, 64]$, learning rate = 1×10^{-3}, and batch size = 32. Results for this and all other spoofing detection methods are presented in the following section.

This method serves as a first step machine learning baseline for Remote ID spoofing detection. Using the data developed in this work and described in the following section, future work can benchmark other data-driven methods such as recurrent neural networks, transformers, and graph neural networks. In particular, the temporal methods will need to address the challenge of infrequent Remote ID receiving caused by the drones moving through the environment and leading to a non-fixed-interval time series.

D. Illustrative Spoofing Scenario

To demonstrate the feasibility of RSSI-based spoofing detection, we present a concrete scenario implemented in our simulator. Figure 5a illustrates an adversarial UAV that broadcasts a spoofed Remote ID claiming to hover directly in front of the target, when in reality the adversary is flying perpendicularly behind the target. The adversary's true location (bottom left) differs from its claimed spoofed location (top), creating a discrepancy that can be detected through RSSI analysis.

Figure 5b shows the simulation results for this scenario. As the target approaches the location claimed in the spoofed Remote ID beacon of the adversary, the distance from target to beacon decreases (blue line). If the target were to compute the expected signal strength based on the adversary's *claimed* position, it would expect the RSSI to increase significantly. However, it actually decreases dramatically (red line) once the adversary flies past the target. This discrepancy between observed and expected RSSI dynamics forms the basis for our detection methods.



(a) Attack scenario **(b) Simulation results**

Fig. 5 Illustrative spoofing scenario. (a) The adversary broadcasts Remote ID claiming to be at a spoofed location directly in front of the target while actually flying on a course behind the target. (b) As the adversary approaches, distance to the spoofed location decreases, but RSSI decreases after initially increasing.

VI. Evaluation

We evaluate our detection methods using a synthetic dataset generated from our OMNeT++ simulation environment.

A. Dataset

Our dataset consists of simulated urban airspace scenarios with UAVs flying constrained trajectories along corridors that run between buildings. Each scenario includes one dynamic trajectory spoofer that claims to be at the position of a silent "ghost" drone while actually flying its own trajectory. The dataset is generated with the following parameter ranges:

- **Environment:** 500–1000m grid with 4–6 East-West and 4–6 North-South flight corridors with 10–50m width, 60–120m spacing, and 10–20 buildings per scenario of 10–100m height
- **Hosts:** 6–12 UAVs per scenario, including one ghost and one spoofer
- **Trajectories:** 5–15 m/s speed, 30–100m altitude, 5–10 minute duration
- **Radio:** 10–25 dBm transmission power, 0.75–1.0s beacon interval, −90 dBm background noise

The dataset contains 1,920 unique scenario configurations, each producing two variants: one in open space and one with building obstructions. These scenarios were then split 80/20 into training and test sets.

The CSV file for a scenario contains timestamped reception (RX) events with columns for receiver position, claimed transmitter position (from the Remote ID message), received signal strength (RSSI), and pre-computed Kalman Filter state. Ground truth labels indicate whether each RX event originated from the spoofer. The CSV files also contain transmission (TX) event data including the true position of the transmitter.

B. Metrics

We evaluate detection performance using the following metrics:

- **AUC:** Area under the receiver operating characteristic (ROC) curve, measuring threshold-independent classification accuracy
- **TPR/FPR:** True positive rate (TPR) and false positive rate (FPR) at the operating threshold

The operating threshold is selected on the training set by maximizing Youden's J statistic (TPR − FPR).

C. Results

Table 1 shows the detection performance of all three methods on our test set, and Figure 6 shows the corresponding ROC curves. All methods are evaluated on using four benign hosts as federates in each test scenario. The KF detector is

Table 1 Detection performance on test set

Detector	AUC	TPR	FPR
KF	0.799	0.660	0.170
MLAT	0.838	0.784	0.183
MLP	0.940	0.823	0.056

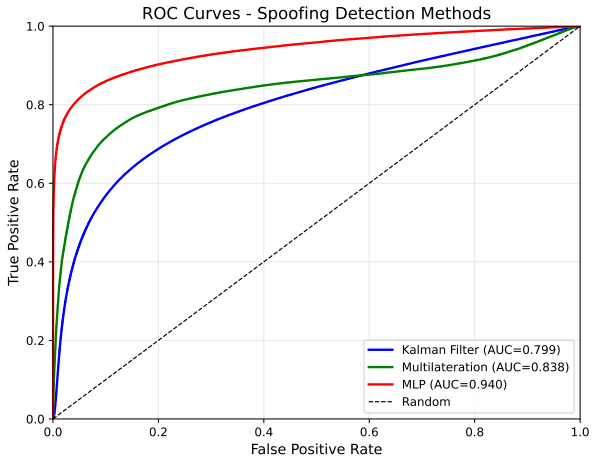


Fig. 6 Detection results. Left: AUC, TPR, and FPR for each method. Right: ROC curves showing the trade-off between true and false positive rates.

evaluated per-RX-event (each federate receiver independently detects anomalies), while MLAT and MLP are evaluated per-transmission (aggregating information across federate receivers).

All three detectors achieve AUC scores of 0.80 or higher, indicating good discrimination between spoofed and benign transmissions. The MLP approach achieves the highest AUC (0.940) with the lowest false positive rate (5.6%), demonstrating that learned features can outperform hand-crafted detection rules. The MLAT detector achieves higher AUC (0.838) and TPR (78.4%) than the KF detector, but requires coordination among multiple federate receivers. The KF detector achieves comparable FPR (17.0%) to MLAT (18.3%) with the advantage of requiring only a single receiver, making it suitable for onboard deployment on resource-constrained UAVs.

The TPR/FPR trade-off has important implications for the intended application. For collision avoidance, false negatives (missed spoofing detections) could lead to dangerous situations where a UAV trusts incorrect position information. Conversely, false positives (incorrectly flagging benign transmissions as spoofed) could cause nuisance alerts that degrade operator trust or trigger unnecessary evasive maneuvers. The MLP's low FPR (5.6%) makes it preferable in operational settings where false alarms are costly, while the rule-based detectors (KF and MLAT) offer interpretability without requiring training data.

These results suggest that machine learning approaches can effectively combine information from multiple detection features to achieve superior performance. The KF and MLAT detectors serve as interpretable baselines that require no training data, while the MLP leverages learned patterns to achieve the best overall discrimination. For deployment, the choice depends on operational constraints: the MLP requires training data and computational resources, while the rule-based detectors can be deployed immediately with tuned thresholds.

We note several caveats regarding these results. First, our evaluation uses simulated data with idealized radio propagation models; real-world performance may differ due to multipath effects, interference, and environmental factors not captured in simulation. Second, our dataset assumes the spoofer uses constant transmission power. This is a reasonable first-order assumption for several reasons: dynamically varying TX power to match claimed distance requires the spoofer to know the receiver's position and the path loss characteristics in real-time—a significantly more sophisticated attack. Additionally, varying TX power requires low-level control of the radio hardware, which may not be possible if the drone uses a dedicated commercial off-the-shelf (COTS) Remote ID module. Nevertheless, an adaptive adversary could potentially evade these detectors and is an important direction for future work. Third, we assume honest serial numbers. Since serial numbers are used to track per-transmitter state in both detectors, a spoofer that frequently changes serial numbers could reset the Kalman Filter state and evade detection. However, serial number spoofing is orthogonal to position spoofing and could be addressed through complementary mechanisms such as cryptographic authentication.

VII. Future Work

While our current dataset generation pipeline is intended to support offline training, we plan to implement APIs to support online training and implementation of more advanced spoofing and detection logic.

Barring multiple coordinated UAVs, it may be possible to approximate the "synthetic array" technique for GPS spoofing detection by instructing a detector UAV to make unpredictable but controlled movements in order to foil "undetectable" spoofing scenarios where an adversary matches their transmission power to their claimed position.

Our current simulation uses INET's built-in radio propagation models, which do not natively support multipath effects such as reflections and scattering from buildings. While our obstacle loss models account for attenuation through buildings, they do not capture the complex signal dynamics of urban environments. For future work, we plan to implement a custom `IObstacleLoss` module that integrates ray tracing tools to provide more realistic radio propagation in dense urban scenarios.

Currently we are working on studying spoofing detection in our OMNeT++ simulator, but we plan to further study the problem via our integration with uav-cyber-sim [33], a distributed simulator designed for simulating a wide array of cyberattacks on multiple UAS in an urban air environment.

VIII. Conclusion

The FAA has mandated that all UAVs broadcast Remote IDs, which can be used for collision avoidance. This provides a lightweight mechanism for UAVs to track nearby vehicles without expensive sensors or computationally intensive processing, but requires that Remote ID messages are trusted. This work considers the possibility that nearby drones can spoof their Remote ID messages and develops detection methods based on RSSI signal dynamics.

We presented two rule-based detection approaches: a Kalman Filter method that tracks TX power consistency from a single receiver, and a federated multilateration method that jointly estimates transmitter position and TX power from multiple receivers. We also evaluated a machine learning baseline using a multilayer perceptron (MLP) trained on RSSI and position features. Our evaluation on simulated urban airspace scenarios shows that all three approaches achieve AUC scores of 0.80 or higher. The MLP achieves the best performance (AUC 0.940, FPR 5.6%), while the rule-based KF and MLAT detectors offer interpretability without requiring training data. The KF detector is suitable for onboard deployment on resource-constrained UAVs, while the MLAT detector achieves higher accuracy (AUC 0.838) at the cost of requiring coordination among multiple receivers.

These results demonstrate that RSSI-based spoofing detection is feasible for Remote ID, providing a foundation for securing UAV collision avoidance systems against position spoofing attacks.

Acknowledgments

This material is based upon work supported by the NASA Aeronautics Research Mission Directorate (ARMD) University Leadership Initiative (ULI) under cooperative agreement number 80NSSC24M0070. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration.

References

[1] Radoglou-Grammatikis, P., Sarigiannidis, P., Lagkas, T., and Moscholios, I., "A compilation of UAV applications for precision agriculture," *Computer Networks*, Vol. 172, 2020, p. 107148.

[2] Lyu, M., Zhao, Y., Huang, C., and Huang, H., "Unmanned aerial vehicles for search and rescue: A survey," *Remote Sensing*, Vol. 15, No. 13, 2023, p. 3266.

[3] Butilă, E. V., and Boboc, R. G., "Urban traffic monitoring and analysis using unmanned aerial vehicles (UAVs): A systematic literature review," *Remote Sensing*, Vol. 14, No. 3, 2022, p. 620.

[4] Betti Sorbelli, F., "UAV-based delivery systems: A systematic review, current trends, and research challenges," *Journal on Autonomous Transportation Systems*, Vol. 1, No. 3, 2024, pp. 1–40.

[5] Wang, H., Zhao, H., Zhang, J., Ma, D., Li, J., and Wei, J., "Survey on Unmanned Aerial Vehicle Networks: A Cyber Physical System Perspective," *IEEE Communications Surveys & Tutorials*, Vol. 22, No. 2, 2020, p. 1027–1070. https://doi.org/10.1109/COMST.2019.2962207, URL https://ieeexplore.ieee.org/document/8943319/.

[6] Yu, Z., Wang, Z., Yu, J., Liu, D., Song, H. H., and Li, Z., "Cybersecurity of unmanned aerial vehicles: A survey," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 39, No. 9, 2023, pp. 182–215.

[7] de Carvalho Bertoli, G., Pereira, L. A., and Saotome, O., "Classification of denial of service attacks on Wi-Fi-based unmanned aerial vehicle," *2021 10th Latin-American Symposium on Dependable Computing (LADC)*, IEEE, 2021, pp. 1–6.

[8] Lammers, D. T., Williams, J. M., Conner, J. R., Baird, E., Rokayak, O., McClellan, J. M., Bingham, J. R., Betzold, R., and Eckert, M. J., "Airborne! UAV delivery of blood products and medical logistics for combat zones," *Transfusion*, Vol. 63, 2023, pp. S96–S104.

[9] Kong, P.-Y., "A survey of cyberattack countermeasures for unmanned aerial vehicles," *IEEE Access*, Vol. 9, 2021, pp. 148244–148263.

[10] Xiao, J., and Feroskhan, M., "Cyber attack detection and isolation for a quadrotor UAV with modified sliding innovation sequences," *IEEE Transactions on Vehicular Technology*, Vol. 71, No. 7, 2022, pp. 7202–7214.

[11] Li, B., Fei, Z., Zhang, Y., and Guizani, M., "Secure UAV communication networks over 5G," *IEEE Wireless Communications*, Vol. 26, No. 5, 2019, pp. 114–120.

[12] Khan, N. A., Jhanjhi, N., Brohi, S. N., Almazroi, A. A., and Almazroi, A. A., "A secure communication protocol for unmanned aerial vehicles," *CMC-Computers Materials & Continua*, Vol. 70, No. 1, 2022, pp. 601–618.

[13] Liao, J., Jiang, B., Zhao, P., Ning, L., and Chen, L., "Unmanned aerial vehicle-assisted federated learning method based on a trusted execution environment," *Electronics*, Vol. 12, No. 18, 2023, p. 3938.

[14] Cosar, M., "Cyber attacks on unmanned aerial vehicles and cyber security measures," *The Eurasia Proceedings of Science Technology Engineering and Mathematics*, Vol. 21, 2022, pp. 258–265.

[15] Basan, E., Basan, A., Nekrasov, A., Fidge, C., Gamec, J., and Gamcová, M., "A self-diagnosis method for detecting UAV cyber attacks based on analysis of parameter changes," *Sensors*, Vol. 21, No. 2, 2021, p. 509.

[16] Baig, Z., Syed, N., and Mohammad, N., "Securing the smart city airspace: Drone cyber attack detection through machine learning," *Future Internet*, Vol. 14, No. 7, 2022, p. 205.

[17] Sedjelmaci, H., Boudguiga, A., Jemaa, I. B., and Senouci, S. M., "An efficient cyber defense framework for UAV-Edge computing network," *Ad Hoc Networks*, Vol. 94, 2019, p. 101970.

[18] Federal Aviation Administration, D. o. T., "14 CFR Part 89 – Remote Identification of Unmanned Aircraft," Regulation, Federal Aviation Administration, Department of Transportation, 2021. URL https://www.ecfr.gov/current/title-14/part-89.

[19] Yasin, J. N., Mohamed, S. A., Haghbayan, M.-H., Heikkonen, J., Tenhunen, H., and Plosila, J., "Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches," *IEEE access*, Vol. 8, 2020, pp. 105139–105155.

[20] Zhai, X., Huang, Z., Li, T., Liu, H., and Wang, S., "YOLO-Drone: an optimized YOLOv8 network for tiny UAV object detection," *Electronics*, Vol. 12, No. 17, 2023, p. 3664.

[21] Tippenhauer, N. O., Pöpper, C., Rasmussen, K. B., and Capkun, S., "On the requirements for successful GPS spoofing attacks," *Proceedings of the 18th ACM Conference on Computer and Communications Security*, Association for Computing Machinery, New York, NY, USA, 2011, p. 75–86. https://doi.org/10.1145/2046707.2046719, URL https://doi.org/10.1145/2046707.2046719.

[22] Horton, E., and Ranganathan, P., "Development of a gps spoofing apparatus to attack a dji matrice 100 quadcopter," *Journal of Global Positioning Systems*, Vol. 16, 2018. https://doi.org/10.1186/s41445-018-0018-3.

[23] International, A., "Standard Specification for Remote ID and Tracking," Standard, ASTM International, jun 2022.

[24] Wubben, J., Calafate, C. T., Cano, J.-C., and Manzoni, P., "FFP: A Force Field Protocol for the tactical management of UAV conflicts," *Ad Hoc Networks*, Vol. 140, 2023, p. 103078. https://doi.org/https://doi.org/10.1016/j.adhoc.2022.103078, URL https://www.sciencedirect.com/science/article/pii/S1570870522002505.

[25] Ardupilot Documentation, *Extended Kalman Filter*, n.d. URL https://ardupilot.org/dev/docs/extended-kalman-filter.html#extended-kalman-filter, accessed: 2025-05-21.

[26] Schiller, N., Chlosta, M., Schloegel, M., Bars, N., Eisenhofer, T., Scharnowski, T., Domke, F., Schönherr, L., and Holz, T., "Drone Security and the Mysterious Case of DJI's DroneID," 2023. https://doi.org/10.14722/ndss.2023.24217.

[27] Hossain Shad, M. I., Sadid Ifty, J., and Rahman, M. T., "Cost-Effective and Low-Latency Computer Vision-Based Collision Avoidance with Remote Monitoring and Intervention," *2024 IEEE International Conference on Computing, Applications and Systems (COMPAS)*, 2024, pp. 1–6. https://doi.org/10.1109/COMPAS60761.2024.10796555.

[28] Liang, Q., Wang, Z., Yin, Y., Xiong, W., Zhang, J., and Yang, Z., "Autonomous aerial obstacle avoidance using LiDAR sensor fusion," *PLOS ONE*, Vol. 18, No. 6, 2023, pp. 1–16. https://doi.org/10.1371/journal.pone.0287177, URL https://doi.org/10.1371/journal.pone.0287177.

[29] Yasin, J. N., Mohamed, S. A. S., Haghbayan, M.-H., Heikkonen, J., Tenhunen, H., and Plosila, J., "Unmanned Aerial Vehicles (UAVs): Collision Avoidance Systems and Approaches," *IEEE Access*, Vol. 8, 2020, pp. 105139–105155. https://doi.org/10.1109/ACCESS.2020.3000064.

[30] Huang, S., Teo, R. S. H., and Tan, K. K., "Collision avoidance of multi unmanned aerial vehicles: A review," *Annual Reviews in Control*, Vol. 48, 2019, pp. 147–164. https://doi.org/https://doi.org/10.1016/j.arcontrol.2019.10.001, URL https://www.sciencedirect.com/science/article/pii/S1367578819300598.

[31] Varga, A., "The OMNeT++ Discrete Event Simulation System," *Proceedings of the European Simulation Multiconference (ESM'2001)*, 2001.

[32] "INET Framework," , 2024. URL https://inet.omnetpp.org/, [Accessed 01-05-2024].

[33] Diaz-Gonzalez, A., Coursey, A., Bjorkman, B., Shatokhin, D., Lemieux-Mack, C., Dahle, N., Taye, A., Canady, R., Koutsoukos, X., Biswas, G., and Ward, B. C., "Networked Simulation for Cybersecurity Evaluation of Small Unmanned Aircraft Systems in Dense Urban Environments," *AIAA SCITECH 2026 Forum*, 2026.

[34] Welch, G., Bishop, G., et al., "An introduction to the Kalman filter," 1995.