RODAD: Resilience Oriented Decentralized Anomaly Detection for Urban Air Mobility Networks

Sixiao Wei, Hui Huang, Genshe Chen
Intelligent Fusion Technology, Inc.
20410 Century Blvd, Suite 230, Germantown, MD 20874
sixiao.wei, hui.huang, gchen@intfusiontech.com

Erik Blasch
Air Force Research Laboratory
Arlington, VA 22203
erik.blasch.1@us.af.mil

Yu Chen, Ronghua Xu
Department of Electrical and Computer Engineering,
Binghamton University, Binghamton, NY 13902
ychen, rxu22@binghamton.edu

Khanh Pham
Air Force Research Lab
Kirtland AFB, NM 87117
khanh.pham.1@us.af.mil

Abstract—**Urban air mobility (UAM) helps ease traffic congestion and offers cleaner, faster, and safer transportation, especially for densely populated areas. Recent events have shown that modern unmanned aerial vehicles (UAVs) are vulnerable to attacks through buggy or malicious devices, which raise concerns regarding performance, security, and privacy on UAM networks. Existing Air Traffic Service (ATS) providers mainly rely on a centralized system (e.g., Information Display System) for data aggregation, sharing, and security policy enforcement; and it incurs critical issues related to a bottleneck of data analysis, provenance, and consistency in terms of less efficiency with large computational resources, and high false positive with low flexibility. In this paper, we develop a *Resilience Oriented Decentralized Anomaly Detection (RODAD)* framework to maximize UAM capability to secure data access among aircraft and ATS service providers based on microservices technologies in an edge-fog-cloud computing paradigm. Machine learning based anomaly detection (MLAD) is developed to detect anomaly behaviors (e.g., aircraft route anomaly) against both single-feature and multi-feature spoofing attacks across avionics mission data. Two GPS spoofing attack scenarios (e.g., restricted and generalized) with four attacking types (e.g., continuous, interim, biased, random) are crafted for the performance evaluation. A hardware-in-the-loop (HITL) implementation is also developed to demonstrate the effectiveness of RODAD for supporting real-time resilient analysis. Our experiments validate the performance of RODAD in detection accuracy and efficiency against spoofing attacks for UAM.**

Keywords—Decentralized, Container-Based Microservices, Edge-Fog-Cloud Computing, Machine Learning, Anomaly Detection

I. Introduction

Air travel for missions and goods in the world's cities is now becoming a reality. Just 3 to 5 years from now, drones and electrical aircraft taking off and landing vertically will be commonplace in urban areas. There is an increase in burdens on avionics to support public safety and security due to the increase in air travel in terms of time, frequency, and space [20]. Urban air mobility (UAM) encompasses air traffic operations for manned and unmanned aircraft systems in a metropolitan area. The delivered services include but are not limited to, ground traffic flow evaluation, humanitarian missions, emergency medical evacuation, news reporting, package delivery, and passenger transport [1]. In this work, we focus on unmanned aircraft systems that provide on-demand automated transportation services. UAM could help ease traffic congestion and will offer cleaner, faster transport with a more integrated transportation system, especially for densely populated areas. The concept of operations describes the main systems and their interactions toward building a functioning UAM architecture. These systems are (1) autonomous aerial vehicles that can carry passengers, (2) landing and takeoff pads, called "vertiports", (3) operation centers for flight planning and monitoring, (4) vehicle maintenance centers, (5) governance structure to organize and manage the infrastructure, and (6) consumer applications that provide service access [2]. However, recent events have shown that modern unmanned aerial vehicles (UAVs) are vulnerable to attacks and subversions through buggy or sometimes malicious devices that are present on UAM communication networks. Citizens' concerns about safety, security, noise [28], and environmental impact must be addressed. This increases the need for cyber awareness including UAVs in airspace and at risk of cyber intrusion [4].

Unlike traditional aircraft, which fly mostly over sparse areas, many operations (e.g., ground traffic flow evaluation, emergency evacuation) will usually occur over metropolitan areas that are highly dense in population and property [2]. Concerning an aircraft system design and architecture, the greater complexity of onboard subsystems (e.g., sensors or autopilot), compared to drones and the increased connectivity between the aircraft and the ground elements (stations, beacons, etc.) increases the number and size of exposed interfaces (i.e., the attack surface) [19]. Attacks become more and more sophisticated and may target critical data. These attacks in turn, can affect the integrity and/or availability of critical aircraft functions. Attacks on vehicles used in an UAM context could thus lead to catastrophic outcomes. It is hence important to integrate security mechanisms to protect and assure vehicle data, critical navigation information, control commands, aircraft components, and sensor measurements.

One emerging challenge is that the conventional technologies mainly rely on a centralized system (e.g., IDS/IPS) for data aggregation, sharing, and security policy enforcement; and it incurs critical issues related to a bottleneck of data analysis, provenance, and consistency in terms of less efficiency with large computational resources, and high false-positive with

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

low flexibility. Since air vehicles can be compromised at a single point, effects can propagate across the entire UAM network. The sustainable use of unmanned aerial systems (UAS) increases the need for cyber awareness [21-22]. Cyber awareness concerns require new methods for security, e.g., decentralized Artificial Intelligence/Machine Learning (AI/ML) method, to enhance performance over reliability, resilience, and assurance [19].

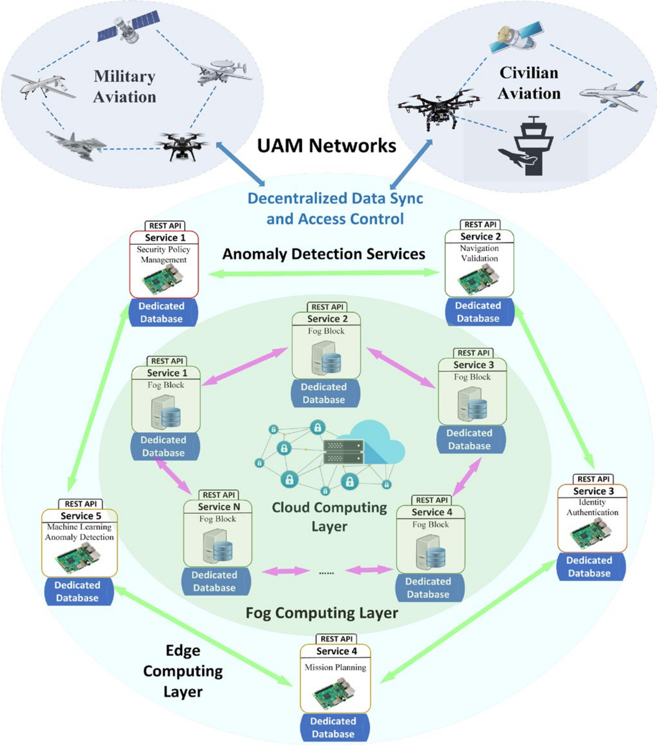


Fig. 1. RODAD System Architecture.

To address these pressing needs and to accommodate a wide range of data and challenges, in this paper, we developed a *Resilience Oriented Decentralized Anomaly Detection (RODAD)* framework to maximize UAM capability to secure data access among aircraft and Air Traffic Service (ATS) service providers (e.g., Aircraft Situation Display to Industry (ASDI)) based on microservices technologies in a hierarchical edge-fog-cloud computing paradigm (as shown in Fig. 1): *Edge Computing Layer* [31]: The on-site edge devices and sensors are responsible for enrolling and capturing raw mission data (e.g., Automatic Dependent Surveillance-Broadcast (ADS-B) or MAVLink messages [32]) that can be digitized and converted to key features, such as aircraft identification and trajectories. As collected raw data are transferred to edge computing nodes for pre-processing and low-level feature extraction, only the extracted features are transmitted to the fog layer for data aggregation and higher-level analytic services, such as route pattern recognition and anomalous event detection. In detail, we model and craft different types (e.g., continuous and interim) of GPS spoofing attacks on UAV sensors in a more generalized way and show their impact by performing a case study in UAM networks. *Fog Computing Layer*: Intermediate-level feature contextualization and smart decision-making for a surveillance system are available to support intelligent analytic functions [5][29]. Through analyzing extracted features, UAM navigation of mission planning is constructed and saved to the operational database. The machine learning (ML)-based detection algorithm performs aircraft authentication, either route verification or frame identification, based on a specified threshold of a scenario mission. As such, we trained the data-driven-based ML models (Long Short-Term Memory [6] (LSTM) and XceptionTime) and evaluate the detection performance of these models. The trained ML models and archived operation data could be shared among inter-domain UAM networks and uploaded to the cloud servers for high-level information fusion and analysis tasks. *Cloud Computing Layer*: This layer handles high-level tasks which are often computing extensive and big-data oriented, such as multi-airborne collaborative planning and decision-making reasoning [7]. Based on a thorough analysis of shared avionics data among different airborne Command and Control (C2) nodes, a comprehensive Target Identifiable Information (TII) could be constructed by associating C2 individuals with historical information.

In UAM, the airborne C2 node will function as the control and reporting center and will be assigned responsibility for "flexing" Air Traffic Organization (ATO) to accommodate emergent targets [30]. Usually, the aircrews depart with a detailed plan developed using the most recent threat and other (e.g., weather, reconnaissance photos, etc.) data available in the hours before departure. The data used to plan the mission and the overall mission tasked by the ATO may therefore be hours old by the time that the missions are executed and highly vulnerable to malicious attack the mission redirection (including en route target assignments or mission change). The security mechanism of RODAD, regarding secured avionics data sharing and navigation verification, abnormal detection, and access management, is implemented as separate microservices [23][25]. Note that how to create and distributed the decentralized microservices in permissioned UAM networks has been introduced in our previous work [8]. This paper focuses on the development of machine learning / anomaly detection (MLAD) against spoofing attacks.

The rest of the paper is organized as follows. Section II briefly discusses the related works and background of GPS spoofing attacks on UAM networks. Section III introduces the details of spoofing attack craft and generation. Section IV presents our developed data-driven ML models for detecting the single-feature spoofing attack patterns in terms of multiple attacking scenarios, including restricted, generalized, and slow-shifting schemes. Additional analysis includes the relationship between detection accuracy and single-feature attack intensity. The method of detecting multi-feature spoofing attacks on compromised sensors to show “Which specific sensor outputs are attacked” and “How efficiency of RODAD against multi-feature attack intensity” is introduced in Section V. Section VI shows the hardware-in-the-loop (HITL) implementation of the developed system to accurately and real-timely detect GPS spoofing attacks and protect UAM networks. Section VII concludes the paper.

II. Related Works Background

A. Spoofing Attacks

Today's cybersecurity has become more than a necessity due to the widespread adoption of the Internet of Everything (IoE) [16]. hackers can now launch sophisticated spoofing attacks which can have unprecedented impacts. For instance, Tippenhauer et al. [9] showed a spoof attack scenario on GPS-enabled devices. In the GPS attack, a forged GPS signal is transmitted to the device to alter the perceived location. In this way, the true location of the device can be disguised and the attacker can fool the target to navigate to a specific location, and there perform a physical attack. In another work, Giannetos et al. [11] introduced an app named Spy-sense, which monitors the behaviors of several sensors in a device. The app can manipulate sensor data by deleting or modifying it. The Spy-sense exploits the active memory region in a device and relays sensitive data covertly. These studies show that spoofing attacks can be performed even without gaining direct access to a system.

GPS spoofing was introduced originally in the space domain, where an attacker compromises sensor readings in such a way that undetected errors are induced into the estimate of state variables and values [12]. Another work [14] considers cyber-attacks to be one of the reasons behind the two recent Boeing 737 Max 8 crashes. According to [14], a passenger, vehicle, or drone carrying a sonic device capable of impacting the Maneuvering Characteristics Augmentation System (MCAS) sensor controlling the plane could have been responsible for such an attack. There is an example of erroneous data from the angle of attack (AoA) sensor on Boeing 737 for the Lion Air flight JT610 right before its crash. It was described that an AoA sensor fed false information to the flight computer, which automatically initiates the plane's nose to pitch downward without any input from the pilot to prevent the flight from a potential stalling event [26]. The inconsistent reading from the two sensors is an indication of a potentially faulty device or a spoofing attack.

In this paper, we first craft and generalize the GPS spoofing utilizing a software simulator (Ardupilot) and make the attacks more realistic. Second, we investigate the detection of multi-feature spoofing attacks for validating the performance effectiveness of RODAD.

B. Time-series ML Models: LSTM and XceptionTime

Recent developments in deep learning techniques enable superior performance in time series modeling and as well as anomaly detection for time series signals. For time series, Recurrent Neural Networks (RNN) and some variants of Convolutional Neural Networks (CNN) are popularly used to model the time series data patterns. In this study, the LSTM and XceptionTime [15] model is used for the spoofing attack detection performance comparison and verification.

LSTM is a type of RNN that is designed for processing sequential signals by introducing the memory mechanism; they can explore the dependency information in the sequences and learn the representations of the sequences that distinguish them without manually designing the features. In addition to the hidden state in the ordinary RNN cell, LSTM introduces a cell state that acts as a "high-way" of the gradient by avoiding the interaction of nonlinearities with backpropagation. Moreover, LSTM employs multiple gates with nonlinearities to increase the expressive power of the network. [18] details the mathematical operations and training of LSTM, which is beyond the scope of this paper. In general, empirical tests are conducted to select the better approach [13]. In this paper, we implement and evaluate the LSTM model for the detection of GPS spoofing.

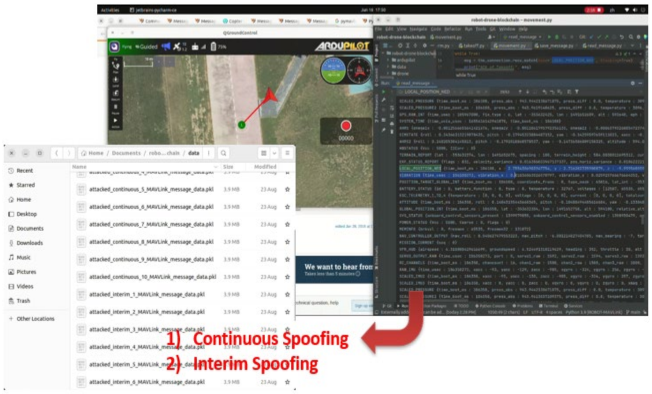
XceptionTime model is a variant of CNN architecture that is designed using the concept of Inception Networks [16]. In traditional CNN architectures, one of the challenging tasks in designing CNNs is selecting the right kernel size, which is crucial to extract global or local information. In the XceptionTime model, instead of picking a filter with a specific size, multiple one-dimensional filters with different kernel sizes are adopted to extract short and long-time series' features simultaneously with the resulting feature maps being concatenated to construct the output features. Moreover, to mitigate the computational cost problems, as well as lessen the overfitting problems, the bottleneck layer is used as the first component within the XceptionTime Module. In addition, by utilizing the techniques of depth-wise separable convolutions, adaptive average pooling, and non-linear normalization, the XceptionTime model achieves better classification performance in terms of time-series data. Therefore, it is used in this study for the detection of GPS spoofing.

To efficiently detect real-time GPS spoofing attacks, we implement and compare the LSTM and XceptionTime models in terms of single-feature and multi-feature attack scenarios.

III. Spoofing Attack Data Preparation

To better perform machine learning based anomaly detection (MLAD) among UAVs, we utilized the MAVLink (Micro Air Vehicle Link) Protocol to initialize the security analysis of UAM networks. MAVLink is an open-source protocol that enables UAVs to send waypoints, control commands, and telemetry data. There have two types of messages in the MAVLink format [17]: 1) **State messages** represent the messages sent from the unmanned platform to the ground station and contain information about the state of a system, such as ID, velocity, geolocation, etc. **Command messages** usually refer to the information from the ground station to the unmanned system to execute some actions by autopilot. Those messages can be transmitted via WiFi, Ethernet, or other serial telemetry channels. The software emulation tool (ArduPilot) [17] supports crafting and generating the cyber-attack scenarios for training the datasets (both benign and malicious samples) based on the acquired MAVLink message communication data.

Specifically, we run the ArduPilot directly on a local server without any special hardware for an initial simulation to acquire the raw data (Fig. 2). While running MAVLink, the sensor data comes from a flight dynamics model in a flight simulator. ArduPilot has a wide range of vehicle simulators built in, including multi-rotor aircraft, fixed-wing aircraft, ground vehicles, and underwater vehicles. It can interface with several external simulators as well. We have successfully implemented it on a Python platform, associated with this Pymavlink package to emulate and collect the MAVLink message data. We recorded and saved this key information for the MLAD training. For instance, *GPS_RAW_INT* refers to the absolute geolocation of GPS, latitude, longitude, and altitude. The *AHRS* refers to the Attitude and Heading Reference System, which consists of sensors on three axes that provide attitude information for aircraft, including roll, pitch, and yaw. *EKF_STATUS_REPORT* indicates that an Extended Kalman Filter (EKF) algorithm is used to estimate vehicle position, velocity, and angular orientation based on rate gyroscopes, accelerometer, compass, GPS, airspeed, and barometric pressure measurements. The *ESC telemetry* is a feature that allows displaying data from the Electronic Speed Controller (ESC) directly in the transmitter, some related key data like temperature, voltage, etc. [17].



Name	Size	Modified
attacked_continuous_5_MAVLink_message_data.pkl	3.9 MB	23 Aug
attacked_continuous_6_MAVLink_message_data.pkl	3.9 MB	23 Aug
attacked_continuous_7_MAVLink_message_data.pkl	3.9 MB	23 Aug
attacked_continuous_8_MAVLink_message_data.pkl	3.9 MB	23 Aug
attacked_continuous_9_MAVLink_message_data.pkl	3.9 MB	23 Aug
attacked_continuous_10_MAVLink_message_data.pkl	3.9 MB	23 Aug
attacked_interim_1_MAVLink_message_data.pkl	3.9 MB	23 Aug
attacked_interim_2_MAVLink_message_data.pkl	3.9 MB	23 Aug
attacked_interim_3_MAVLink_message_data.pkl	3.9 MB	23 Aug
attacked_interim_4_MAVLink_message_data.pkl	3.9 MB	23 Aug
attacked_interim_5_MAVLink_message_data.pkl	3.9 MB	23 Aug
attacked_interim_6_MAVLink_message_data.pkl	3.9 MB	23 Aug

Fig.2. Spoofing Attack Generation.

To study the impact of spoofing attacks on UAM networks, we consider attack scenarios where the attacker has access to jam and compromise the UAVs. Thus, resulting in incorrect predictions of the GPS location of the UAVs. It is possible to launch a spoofing attack without direct access to the UAVs, using another drone carrying a special jamming device that is capable of interfering and modifying with the onboard sensor measurements. Figure 2 presents a screenshot of crafted GPS spoofing attacks under multiple attacking scenarios.

IV. Detection of the Single-Feature Spoofing Attacks

According to our previous study [4], we developed four types of GPS spoofing attacks based on data from the software simulator (ArduPilot), namely interim-biased, interim-random, continuous-biased, and continuous-random (Fig. 3). Wherein, interim/continuous refers to the duration of the attack. In an interim attack, each attack is initiated after several operational cycles and continues into the next 400 timecycles. In a continuous attack, the attack is initiated at a certain operational cycle and continues to the end-of-mission of a drone. Biased/random attack refers to the mode of attack. A biased attack prefers to manipulate sensor readings by adding/subtracting a constant value, while a random attack contaminates the sensor readings with random noise.

A. Detection of the Restricted Spoofing Attacks

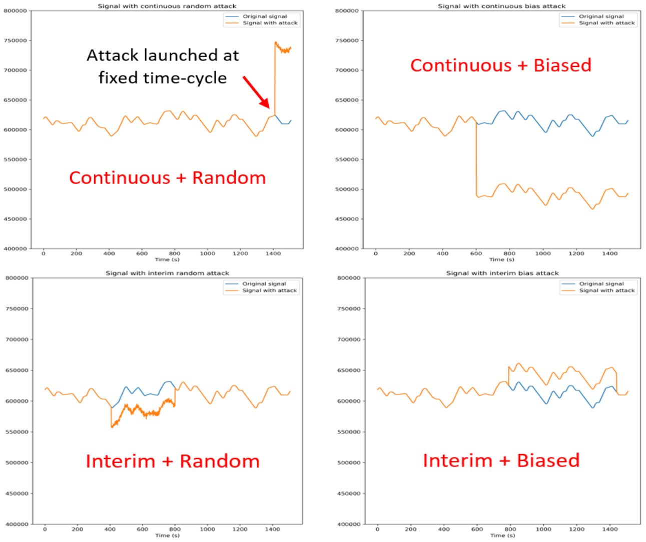


Fig.3. Restricted spoofing attacks with additive white Gaussian noise as a constant amplitude shift for a biased type and a random type.

For spoofing attacks with restricted assumptions, a specific noise type is randomly simulated and added to the original signals. Take the altitude data as an example. Fig. 3 shows the restricted spoofing attacks with additive white Gaussian noise (AWGN) as the random type and a constant amplitude shift for the biased type. The attacks are applied on the time series data of altitude feature with AWGN and a constant shift of amplitude, which can be formulated as follows,

$$S_i^r = \begin{cases} S_i & , i \notin [d_k, l_k] \\ S_i + N_i + c_k & , d_k \leq i \leq l_k, 0 < k < K \end{cases}$$

where $S_i \in S$ and $S_i^r \in S^r$ are the value of i-th data sample in the original and attacked altitude time series data with restricted assumptions, respectively. K is the number of intervals being attacked and k indicates the k-th interval. $N_i \sim \mathcal{N}(0, \sigma)$ is a zero-mean white noise, c_k is a random number for shift bias in the k-th attack interval, d_k is the start time index of the k-th attack interval when the attack is initiated and l_k $(0 < d_k < l_k < L_k, L_k \leq L)$ is the end time index when the attack ends. L is the total length of the time series data. In this study, d_k and l_k are random variables that determine the interval of the attack applied to the time series data corresponding to the interim attacks. If l_k equals to L, the attack is considered as a continuous attack. In total, the datasets contain 30000 data samples with original data and attacked data half by half. A sliding window of 10 samples is applied to make up the training samples (size of 10 by 1) and labels (1 for original data and 0 for attacked data) for the deep learning (DL) model. The dataset was split as 70% for training, 20% for validation, and 10% for testing.

TABLE I. THE EXPERIMENTAL LSTM AND XCEPTIONTIME MODEL ARCHITECTURE

	LSTM
Loss	Cross-entropy
Model Parameter	Linear(in_feature=64, out_feature=2, bias=True)
RNN cell	LSTM(in_feature=10, out_feature=64, num_layers=2, dropout=0.5)
	XceptionTime
Loss	Cross-entropy
Model Parameter	4-series with residual connection, depth-wise convolution filters = 16, 32, 64, 128
Layers	Adaptive average pooling, Conv 1x1 layers

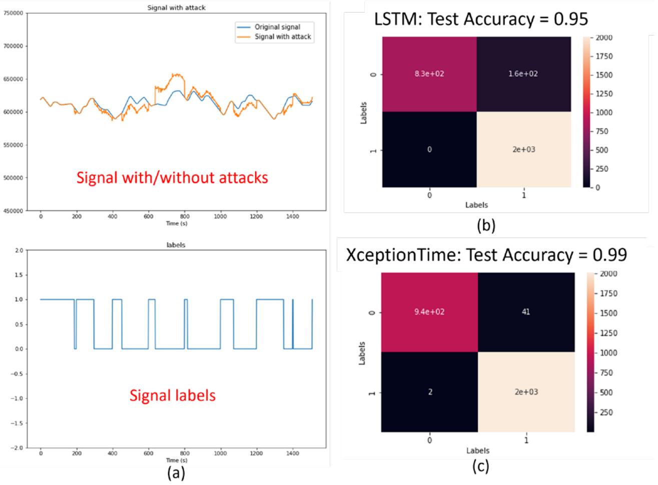


Fig. 4. (a) Signal with restricted assumption and the signal labels with 1 representing the original signal and 0 representing attacks. (b) The confusion matrix of the attack detection with accuracy by LSTM (c) The confusion matrix of the attack detection with accuracy by XceptionTime.

For single-feature attack detection, different attack types are evaluated individually. Both LSTM and XceptionTime are trained and tested to verify the performance of RODAD. The LSTM model leverages a linear layer and two LSTM layers with an *out_feature* of 64 and a dropout rate of 0.5. The loss function is cross-entropy. For the XceptionTime model, four series of modules with residual connections followed by Adaptive Average Pooling layers are utilized. In the four fused XceptionTime modules, the numbers of depth-wise separable convolution filters are 16, 32, 64, and 128 respectively. The cross-entropy loss is considered in the experiment. The architectures of the two models are shown in Table I. The training epochs are set as 150 and the initial learning rate is 10^{-5}. The optimizer is Adam. Fig. 4 shows attack detection performance under restricted assumptions. It is noticed that the XceptionTime model performs better than LSTM model with restricted spoofing attacks. To evaluate more attack types, the slow shifting process can be represented as the following equation,

$$S_i^r = \begin{cases} S_i, & i \notin [d_k, l_k] \\ S_i + N_i + (i - d_k)\dfrac{\alpha \bar{S}}{l_k - d_k + 1}, & d_k \le i \le l_k, 0 < k < K \end{cases}$$

where α is the amplitude shift ratio, and $\bar{S}$ is the mean of S. We use $\alpha = 0.05$ for the slow shifting process in this study. Fig. 5 displays the attack detection performance with the slow-shifting attack added. Compared with the previous results, adding a slow shifting process seems a little difficult for the models to detect the attack patterns. Still, the LSTM and XceptionTime model achieves excellent attack detection performance with accuracies of 83% and 98%.

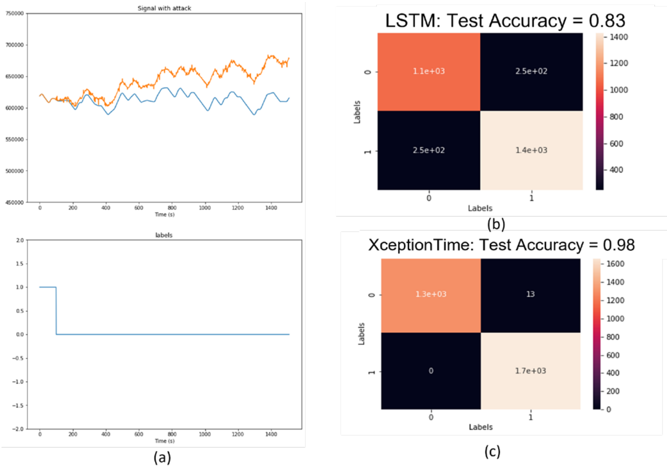


Fig. 5. (a) Signal with restricted assumptions and slow-shifting attacks. (b) The confusion matrix of attack detection with accuracy by LSTM (c) The confusion matrix of attack detection with accuracy by XceptionTime.

B. Detection of the Generalized Spoofing Attacks

According to the results with restricted spoofing attacks, it is quite easy to detect attacks with DL models with restricted assumptions, which make strong assumptions that may be far from realistic spoofing attacks. From the viewpoint of an attacker, attacks similar to the signal itself may be more realistic and more difficult to be detected. To simulate more realistic scenarios and evaluate the ability to detect more stealthy attacks, some time series signal generation techniques are used to generate more stealthy attack data. A simple stealthy attack is a replay attack that the original signal is maliciously or fraudulently repeated or delayed [24]. The replay attack is implemented by intercepting the signals and re-transmiting them, thereby fooling the receiver for interference purposes. In addition, other time-series generation schemes such as regression and ML methods can also be utilized to generate more stealthy attack scenarios and further evaluate the attack detection capability. In this study, we generalize the assumptions of the spoofing attacks by the following rules: 1) the attacker observes the GPS signal for a certain amount of time; 2) the attacks are initialized randomly; 3) the duration of the interim attacks is randomly selected; 4) the shift bias is either positive or negative with a 50% probability; and 5) the observed GPS message will be replayed with the above rules to interfere with the original GPS signals. For example, the attack duration is first determined by random integers defining the start and end index in the time series such as an interval with 1000 data samples. Within the attack duration of 1000 data samples, small parts of original data samples such as 200 data samples are corrupted by replaying various previously observed data samples that are randomly selected. For the bias, a random value is selected to determine the amount to shift the amplitude. The following equation formulates the generalized spoofing attacks.

$$S_i^r = \begin{cases} S_i & ,i \notin [d_k, l_k] \\ S_i + S_{i'} + c_k & ,d_k \le i \le l_k, 0 < k < K \end{cases}$$

where $S_{i'}$ is the replayed S_i from previously observed data. Fig. 6 shows the generalized assumption attack type with replay attacks coupled with the previous four attack types.

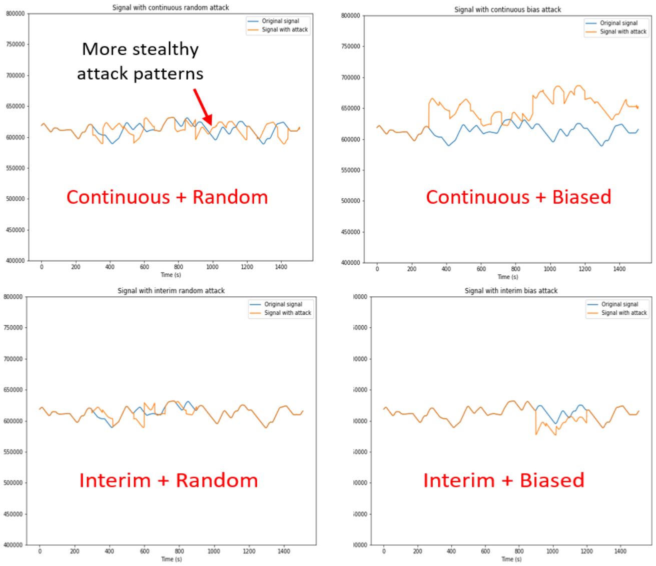


Fig. 6. Generalized spoofing attacks with different attacking types.

Similar to the restricted spoofing attacks, the dataset is also based on altitude data and contains the same number of data samples, and the data split setting is consistent with the restricted spoofing attacks. Compared with the restricted spoofing attacks, one can visually see that the generalized versions are stealthier and may be difficult to judge if an attack exists. Both LSTM and XceptionTime are trained and tested with this dataset. Fig. 7 displays the results for the generalized spoofing attacks. As shown in Fig. 7, the test accuracies were 90% and 94% for LSTM and XceptionTime models, respectively. It is noticed that the attack detection accuracy decreases compared to the ones in Section B since more generalized assumptions are used, which are stealthier and more difficult to be detected. Similar to Section IV.A, the slow shifting process is added with the generalized spoofing attacks, which can be formulated as follows,

$$S_i^g = \begin{cases} S_i & ,i \notin [d_k, l_k] \\ S_i + S_{i'} + (i - d_k)\dfrac{\alpha \bar{S}}{l_k - d_k + 1} & ,d_k \le i \le l_k, 0 < k < K \end{cases}$$

where α is the amplitude shift ratio, and $\bar{S}$ is the mean of S. To be consistent with the restricted version, we use $\alpha = 0.05$ as well for emulating the slow shifting process in the generalized version. Models are retrained and evaluated with the same architecture and parameter settings. Fig. 8 shows the results of the model performance with slow shifting added. It is noticed that the accuracies of both models decrease to 75% for LSTM and 83% for XceptionTime after adding the slow shifting process which is consistent with previous restricted spoofing attacks.

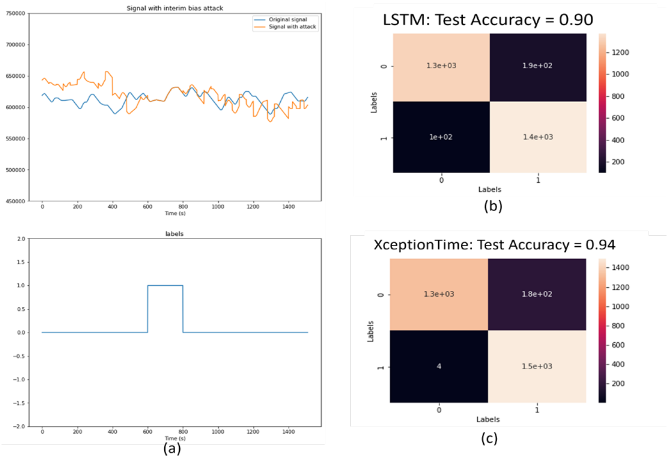


Fig. 7. (a) Signal with generalized assumption and the signal labels with 1 representing the original signal and 0 representing attacks. (b) The confusion matrix of the attack detection with accuracy by LSTM (c) The confusion matrix of the attack detection with accuracy by XceptionTime.

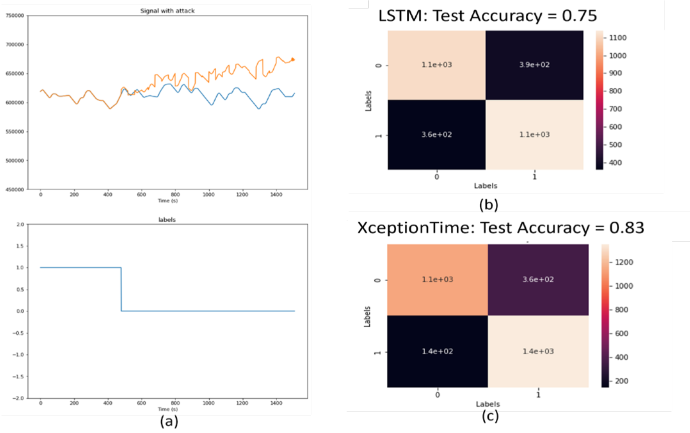


Figure 8. (a) Signal with generalized assumption and slow shifting attack. (b) The confusion matrix of the attack detection with accuracy by LSTM (c) The confusion matrix of the attack detection with accuracy by XceptionTime.

C. Detection Efficiency with Attack Intensity

Since the slow shifting process brings another issue for the MLAD, a quantified analysis needs to be performed for the amplitude shifting or attack intensity evaluation. For the slow shifting process, we use ten values of α, evenly distributed from 0 to 0.45, to evaluate the model performance. A challenge is how the signal amplitude shifts affect the detection performance. The evaluation is analyzed in Fig. 9. It is noticed that while the attack intensity increases, the performance of attack detection also increases. In Fig. 9, the signal amplitude is shifted with the slow shifting up to 0.45 of the average amplitude for both restricted and generalized spoofing attacks. The attack detection accuracy increases from 75% to 95% with restricted assumptions and 62% to 91% with generalized assumptions.

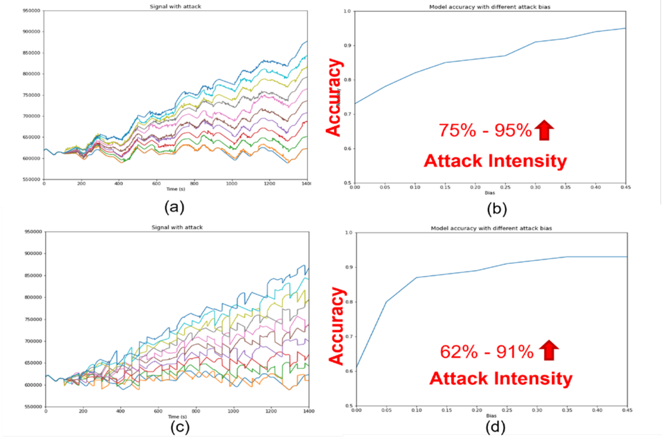


Fig. 9. Signal with (a) restricted assumptions and (c) generalized assumptions and slow shifting by adding amplitude from 0 to 0.45 of the average amplitude of the original signal. The attack detection accuracy with different shifting amounts by (b) LSTM and (d) XceptionTime.

V. Detection of the Multi-Feature Spoofing Attacks

For the multi-feature spoofing attack detection scenario, all the feature data are subject to four different attack types (e.g., interim/continuous, biased/random) introduced in Section IV.A. Similar to the attacks in altitude feature, the other three features (e.g., longitude, latitude, and ground speed) are all included to evaluate the attack detection performance under the four different attack types.

A. Detection of the Restricted Spoofing Attacks

For the restricted spoofing with multi-feature attacks, the utilized datasets are the same as the single-feature version with the four attacking types introduced in Section IV.A. Instead of the single-channel time series, the multi-feature time series has four channels of time series. For each channel, the restricted spoofing attacks are applied with AWGN and random amplitude shifts. Fig. 10 presents the restricted spoofing attacks with four compromised features.

To evaluate the performance of MLAD dealing with multi-feature scenarios, a fused model with four separate trained models with the corresponding feature channels is used. Similarly, ten samples sliding window is performed to obtain the training and testing data samples as well as the labels. The total testing data samples including the four features are used for the performance evaluation. Figs. 14(a) and (b) show the detection performance with confusion matrix and accuracy (LSTM of 85% and XceptionTime of 96%). In addition, the slow shifting process is also added for the multi-feature detection evaluation. Fig. 11 shows the slow shifting added with the restricted spoofing attacks and Figs. 14(e) and (f) show decreased accuracy performance (LSTM of 67% and XceptionTime of 92%).

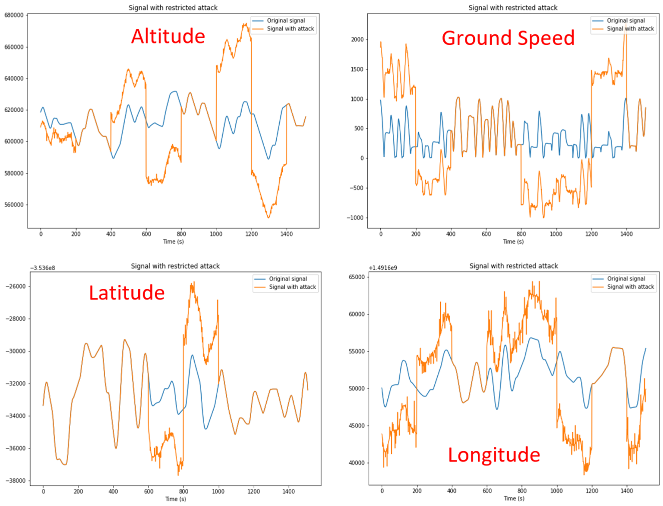


Fig. 10. Multi-feature signal with restricted assumptions and corresponding labels.

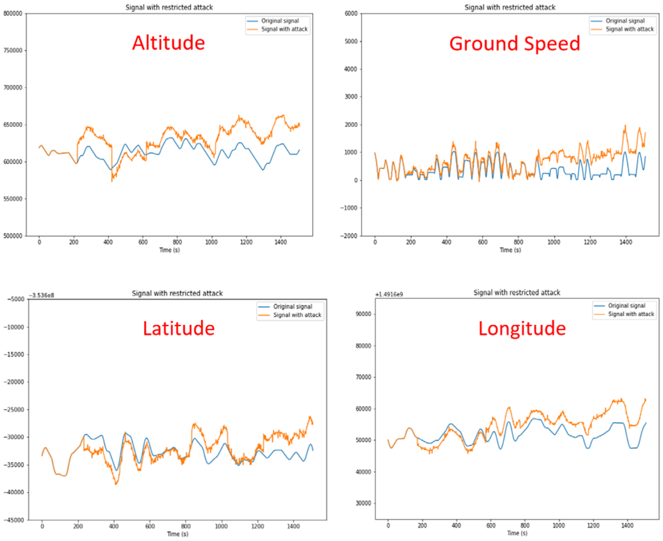


Fig. 11. A slow shifting process is added for the multi-feature signal with restricted assumptions and the corresponding labels.

B. Detection of the Generalized Spoofing Attacks

In Section IV.B, single-feature generalized spoofing attacks are applied in the altitude feature, which is also performed with the other three compromised features to form the multi-feature generalized spoofing attacks. For the *attack swapping intervals*, different lengths are used such as 200 samples with altitude, 250 samples with longitude, 300 samples with latitude, and 350 samples with ground speed. Other random settings are consistent with the single-feature generalized spoofing attacks in Section IV.B. Fig. 12 shows the generalized spoofing attacks with the four compromised features. Similar to the restricted spoofing attacks, the four channels are separately used for training and testing the sub-models and fused to a model for overall generalized spoofing attack detection. Figure 14(c) and (d) shows the detection performance for the generalized spoofing attacks with an accuracy of 78% using LSTM and 81% using XceptionTime. A slow shifting process is also added with the generalized spoofing attacks as shown in Fig. 13. The performance indicates a decrease with a slow shifting process added with

LSTM of 72% and XceptionTime of 80% as shown in Figs. 14(g) and (h).

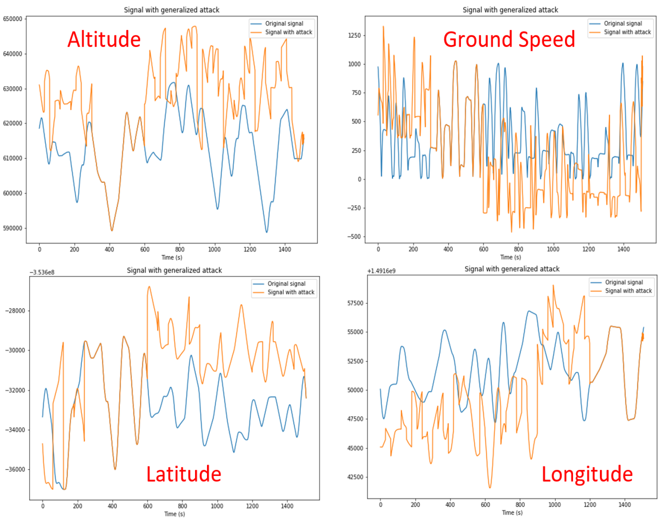


Fig. 12. Multi-feature signal with generalized assumptions with corresponding labels.

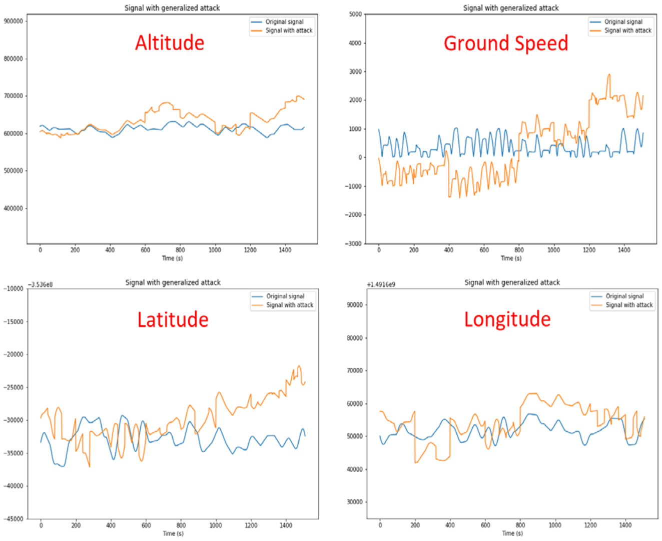
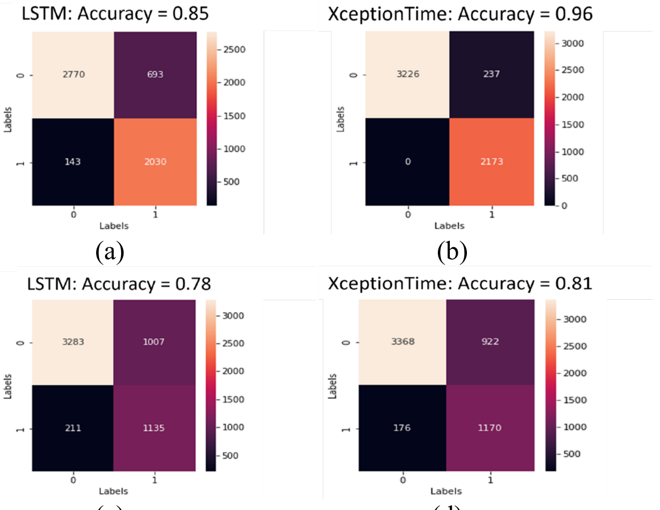
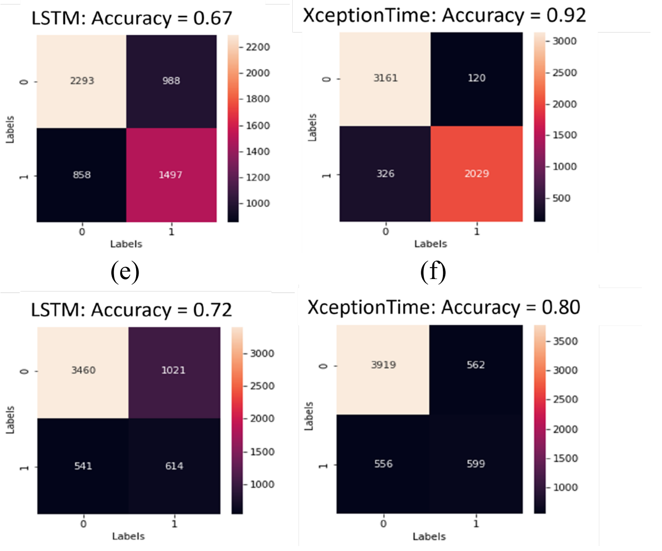


Fig. 13. A slow shifting process is added for the multi-feature signal with generalized assumptions and corresponding labels.



(a) (b)

(c) (d)



(e) (f)

(g) (h)

Fig. 14. Multi-features spoofing attack detection confusion matrix and accuracy of (a) LSTM with restricted assumptions, (b) XceptionTime with restricted assumptions, (c) LSTM with generalized assumptions, (d) XceptionTime with generalized assumptions, (e) LSTM with restricted and slow shifting attacks, (f) XceptionTime with restricted and slow shifting attacks, (g) LSTM with generalized and slow shifting attacks, and (h) XceptionTime with generalized and slow shifting attacks.

C. Detection Efficiency with Attack Intensity

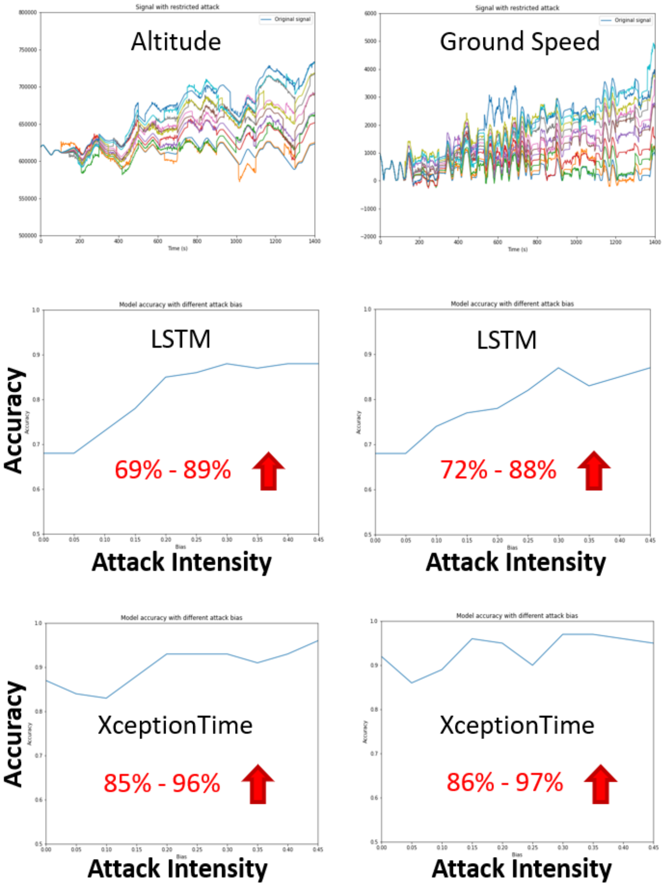


Fig. 15. Multi-feature spoofing attack detection accuracy with restricted assumptions and slow shifting from 0 to 0.45 of the average amplitude of the original signal.

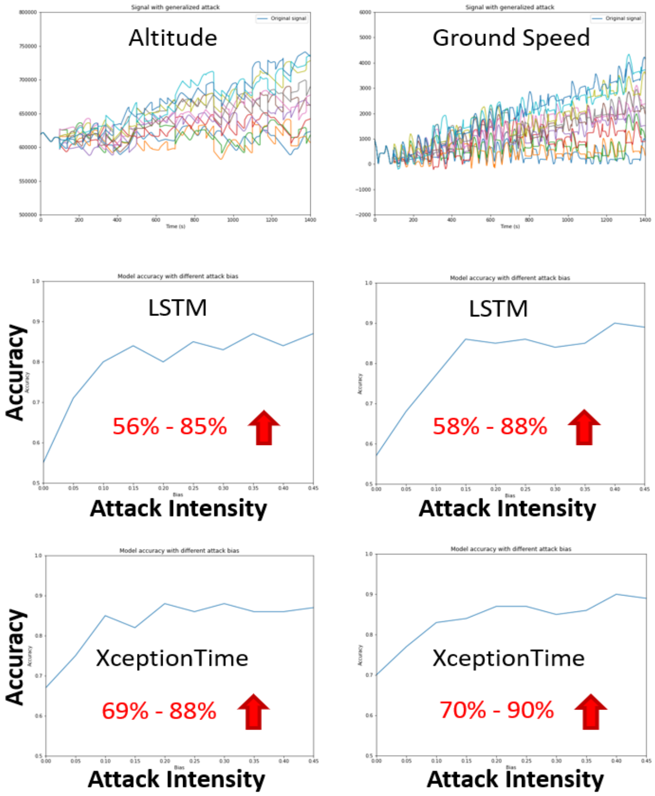


Fig. 16. Multi-feature spoofing attack detection accuracy with generalized assumptions and slow shifting from 0 to 0.45 of the average amplitude of the original signal.

To further verify the signal amplitude shifting effects for the performance of MLAD, different amplitude shifting amount is investigated by increasing the amplitude amount from 0 to 0.45 to evaluate the multi-feature spoofing attack detection performance. Fig. 15 and Fig. 16 shows how the detection accuracy changes for both LSTM and XceptionTime model under restricted assumptions and generalized assumptions. The results are consistent with the previous single-feature version that increasing the attack intensity would result in a significant improvement in the detection performance. The evaluation of the spoofing attack detection performance with different attack types is summarized in Table II.

TABLE II. ATTACK DETECTION PERFORMANCE

	Single Feature Attack Detection			*Multi-Feature Attack Detection*			
Models	**Restricted**	**Generalized**	**Slow Shifting**	**Restricted**	**Generalized**	**Slow Shifting**	**Overall**
LSTM	95%	90%	75%	85%	78%	72%	70%
XceptionTime	98%	94%	83%	96%	89%	80%	85%

VI. HARDWARE-IN-THE-LOOP IMPLEMENTATION

This Section presents our hardware-in-the-loop (HITL) implementation design to support real-time system-aware anomaly detection. The detection applications are decoupled as multiple decentralized microservices that are developed as Docker images and are deployed on fog and edge computing platforms. An iIntel Next Unit of Computing (NUC) minicomputer acts as an edge device while a laptop is adopted as the fog node. Key metrics are assessed for performance and effectiveness over awareness, support, veracity [33], and relevance [10]. RODAD can accept sensor data from different types of assets or unmanned platforms, while one individual asset may contain several subsystems [27]. Fig. 17 shows the detailed system implementation architecture of both the front-end and back-end.

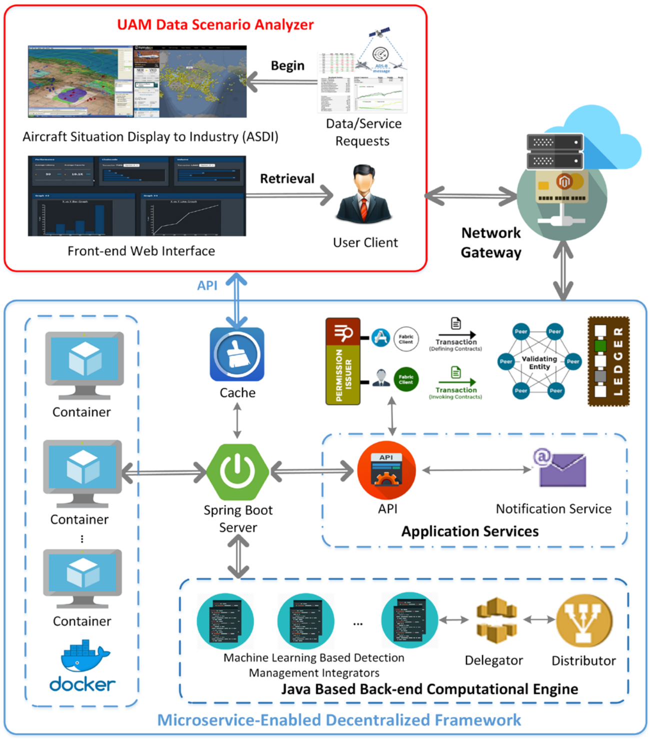


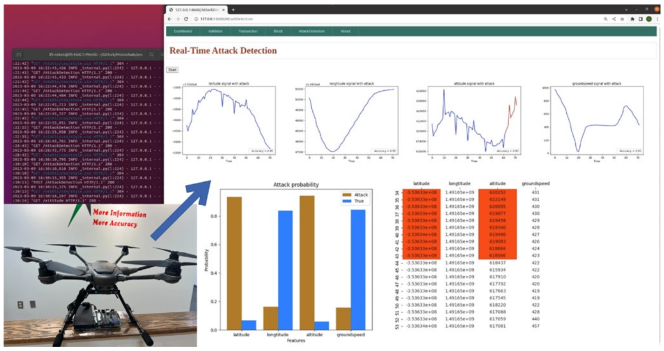
Fig. 17. RODAD Implementation Architecture with HITL.

The MLAD services are built on a decentralized network including three miners and three non-miner participants. One miner is deployed on a laptop and the other two miners are distributed on two desktops. Each miner just uses one CPU core to ensure they have the same computation resource. One desktop and two Intel NUC minicomputers are non-miners to run MLAD microservices. The MLAD microservices are developed as Docker containers providing a client application to interact with UAM networks and Flask-based webservice [5] to handle detection service requests. To evaluate the detection performance of the MLAD mechanism, a service access experiment is carried out on a physical network environment that includes two Intel NUC minicomputers and one desktop. One Intel NUC minicomputer has removed the case and attached on a hardware UAV (Yuneec International Typhoon H) that works as a client to send a service request, as shown in Fig. 18, while the server-side is the navigation service provider, which has been hosted on fog (desktop) nodes.

To demonstrate the effectiveness of RODAD, a real-time attack detection scheme is implemented as a web application, which works with the configured scenarios consisting of servers and clients using TCP/IP protocol with WIFI. As a client node, the UAV sends a GPS signal back to a server node in which the real-time signals are plotted and displayed in a front-end GUI. Simultaneously, the server node performs the attack detection for the incoming signals by assigning the streaming signals into a trained attack detection model that could present the probability of attacks for each feature and determine if attacks exist in the received GPS signals. Concretely, as shown in Fig. 18, the GPS signals sent by UAV are displayed on a webpage followed by the attack detection results which are showing the probability of the malicious patterns. The real-time sensor signal is also highlighted as red if it is determined to contain attack patterns. The web GUI is developed using the FLASK platform [5]. The GPS signal data is first serialized using a pickle package and further transmitted via socket communication to the server node. We deployed the well-trained XceptionTime model in the back-end to perform real-time detection by iteratively importing the incoming signals through the detection process of RODAD. Once the model outputs the results, it will be transferred to the front-end in terms of the inputs and the probability of attacks in each feature channel. The front-end GUI will plot and display the signal curves and the attack probability dynamically. Table III describes the configuration of the Intel NUC minicomputer that is used for the HITL implementation. The implementation further demonstrates that RODAD performs accurately and efficiently in GPS spoofing attack detection.

TABLE III. Configuration of Experimental Devices

Device	Intel NUC 13 Pro Mini PC
CPU	Intel Core i5 (2 cores), 3GHz
Memory	4GB DDR4
Storage	250GB SSD
OS	Ubuntu 20.04



```
:22:42] "GET /static/css/style.css HTTP/1.1" 304 -
2023-03-09 16:22:43,426 INFO _internal.py(l:224) - 127.0.0.1 - -
:22:43] "GET /AttackDetection HTTP/1.1" 200 -
2023-03-09 16:22:43,433 INFO _internal.py(l:224) - 127.0.0.1 - -
:22:43] "GET /static/css/style.css HTTP/1.1" 304 -
2023-03-09 16:22:44,476 INFO _internal.py(l:224) - 127.0.0.1 - -
:22:44] "GET /AttackDetection HTTP/1.1" 200 -
2023-03-09 16:22:44,484 INFO _internal.py(l:224) - 127.0.0.1 - -
:22:44] "GET /static/css/style.css HTTP/1.1" 304 -
2023-03-09 16:22:45,713 INFO _internal.py(l:224) - 127.0.0.1 - -
:22:45] "GET /AttackDetection HTTP/1.1" 200 -
2023-03-09 16:22:45,727 INFO _internal.py(l:224) - 127.0.0.1 - -
:22:45] "GET /static/css/style.css HTTP/1.1" 304 -
2023-03-09 16:22:55,051 INFO _internal.py(l:224) - 127.0.0.1 - -
:22:55] "GET /AttackDetection HTTP/1.1" 200 -
2023-03-09 16:22:55,058 INFO _internal.py(l:224) - 127.0.0.1 - -
:22:55] "GET /static/css/style.css HTTP/1.1" 304 -
2023-03-09 16:28:43,761 INFO _internal.py(l:224) - 127.0.0.1 - -
:28:43] "GET /AttackDetection HTTP/1.1" 200 -
2023-03-09 16:28:43,775 INFO _internal.py(l:224) - 127.0.0.1 - -
:28:43] "GET /static/css/style.css HTTP/1.1" 304 -
2023-03-09 16:30:10,795 INFO _internal.py(l:224) - 127.0.0.1 - -
:30:10] "GET /AttackDetection HTTP/1.1" 200 -
2023-03-09 16:30:10,810 INFO _internal.py(l:224) - 127.0.0.1 - -
:30:10] "GET /static/css/style.css HTTP/1.1" 304 -
2023-03-09 16:30:13,355 INFO _internal.py(l:224) - 127.0.0.1 - -
:30:13] "POST /AttackDetection HTTP/1.1" 200 -
2023-03-09 16:30:13,375 INFO _internal.py(l:224) - 127.0.0.1 - -
:30:13] "GET /static/css/style.css HTTP/1.1" 304 -
2023-03-09 16:30:14,297 INFO _internal.py(l:224) - 127.0.0.1 - -
:30:14] "GET /altitude HTTP/1.1" 200 -
```

```
      latitude       longtitude    altitude  groundspeed
34  -3.53633e+08   1.49165e+09   620252    431
35  -3.53633e+08   1.49165e+09   622149    431
36  -3.53633e+08   1.49165e+09   620005    430
37  -3.53633e+08   1.49165e+09   619877    430
38  -3.53633e+08   1.49165e+09   619458    429
39  -3.53633e+08   1.49165e+09   619340    428
40  -3.53634e+08   1.49165e+09   619490    427
41  -3.53633e+08   1.49165e+09   619093    426
42  -3.53633e+08   1.49165e+09   618684    424
43  -3.53633e+08   1.49165e+09   618566    423
44  -3.53633e+08   1.49165e+09   618437    422
45  -3.53633e+08   1.49165e+09   615934    422
46  -3.53633e+08   1.49165e+09   617910    420
47  -3.53633e+08   1.49165e+09   617792    420
48  -3.53633e+08   1.49165e+09   617663    419
49  -3.53633e+08   1.49165e+09   617545    419
50  -3.53633e+08   1.49165e+09   618220    422
51  -3.53633e+08   1.49165e+09   617088    428
52  -3.53633e+08   1.49165e+09   617059    440
53  -3.53634e+08   1.49165e+09   617081    457
```

Fig. 18. Front-end GUI of decentralized MLAD.

VII. Conclusion

This paper highlights the Resilience Oriented Decentralized Anomaly Detection (RODAD) framework using the decentralized machine learning based data analytic methods (LSTM and XceptionTime) to characterize the detection performance against spoofing attacks. In particular, to generate the source data (MAVLink message) for creating the cyber-resiliency scenarios, we utilized and implemented a software simulator and associated demonstration package (pymavlink) in a Python environment to emulate the communications among UAVs. Two GPS spoofing attack scenarios (e.g., restricted and generalized) with four attacking types (e.g., continuous, interim, biased & random) are crafted for the performance evaluation. In the preliminary results, both LSTM and XceptionTime obtained good performances in distinguishing the attack patterns from crafted sensor reading sequences. The XceptionTime outperforms LSTM by more generalized spoofing scenarios in terms of classification accuracy and sensitivity. Hence, we can conclude that RODAD can identify and classify sources of spoofing attacks in both single-feature and multi-feature scenarios (e.g., which sensors are compromised) accurately and efficiently. In addition, we also analyzed the potential relationship between detection accuracy and attack intensity to further reveal the effectiveness of RODAD against slow-shifting schemes. A hardware-in-the-loop (HITL) implementation is developed to support real-time resilient analysis. Our experiments demonstrate the performance of RODAD in detection accuracy and efficiency against spoofing attacks.

Acknowledgments

This material is based upon work supported by the DoD under contract FA955022P0011. Any opinions, findings, and conclusions, or recommendations expressed in these materials are those of the authors and do not necessarily reflect the views of the Air Force. Also, thanks for Mr. Zelin Wan from Virginia Tech to research on software emulation tool (Ardupilot) during his internship at IFT and Ms. Maria Rios Sandoval from Intelligent Fusion Technology, Inc. (IFT) to review and organize the paper.

References

[1] K. Antcliff,"Baseline Assumptions and Future Research Areas for Urban Air Mobility Vehicles", In AIAA Scitech Forum, pp. 0528, 2019.

[2] R. Rothfeld,"Agent-based Simulation of Urban Air Mobility". In: 2018 Modeling and Simulation Technologies Conference. pp. 3891, 2018.

[3] Z. Chen, Y. Wei, et al., "A cloud/edge computing streaming system for network traffic monitoring and threat detection," International Journal of Security and Networks, vol. 13, pp. 169-186, 2018.

[4] S. Wei, L. Li, et al., "ROSIS: Resilience Oriented Security Inspection System against False Data Injection Attacks," 2023 IEEE Aerospace Conference, pp. 2-7, 2023.

[5] R. Xu, Y. Chen, E. Blasch, and G. Chen."BlendCAC: A BLockchain-ENabled Decentralized Capability-based Access Control for IoTs." 2018 IEEE Internal Conference on Blockchain, July 30-Augest 3, 2018.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, 9(8):1735–1780, 1997.

[7] R. Xu, Y. Chen, et al., "Blendcac: A smart contract enabled decentralized capability-based access control mechanism for the IoT," Computers 2018, 7(3), 39; 2018.

[8] R. Xu, et al., "LightMAN: A Lightweight Microchained Fabric for Assurance-and Resilience-Oriented Urban Air Mobility Networks," in Multidisciplinary Digital Publishing Institute. vol. 6, pp. 421, 2022.

[9] N. O. Tippenhauer, C. P¨opper, K. B. Rasmussen, and S. Capkun, “On the requirements for successful gps spoofing attacks,” in Proceedings of the 18th ACM conference on Computer and communications security. pp. 75–86, 2011.

[10] S. Wei, B. Jia, et al., "High performance enabled space object tracking via cloud computing," 2018 IEEE Aerospace Conference, pp. 1-9, 2018.

[11] T. Giannetsos and T. Dimitriou, “Spy-sense: spyware tool for executing stealthy exploits against sensor networks,” in Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy, ACM, pp. 7–12, 2013.

[12] H. M., Chen, A. Savakis, A. Diehl, E. Blasch, S. Wei, and G. Chen, “Targeted adversarial discriminative domain adaptation”, Journal of Applied Remote Sensing, vol. 15, 2021.

[13] L. Li, T. Bu, Y. Li, S. Wei, A. Harris, “Machine learning based tool chain solution for free space optical communication (FSOC) propagation modeling,” 2021 IEEE Aerospace Conference (50100), pp. 1-8, 2021.

[14] Lee Neubecker, “Could a sonic weapon have caused the two recent boeing 737 max 8 crashes?” 2019, [Online; accessed 22-March-2019]. Available: https://leeneubecker.com/sonic-weaponattack-boeing/.

[15] R., Elahe, S. Zabihi, S. Ff Atashzar, and A. Mohammadi. "Xceptiontime: A novel deep architecture based on depthwise separable convolutions for hand gesture classification." arXiv preprint arXiv:1911.03803, 2019.

[16] S., Christian, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the inception architecture for computer vision." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818-2826. 2016.

[17] Software in the Loop Simulator - Ardupilot. Available: https://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html#sitl-simulator-software-in-the-loop.

[18] Understanding LSTM networks. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[19] Y. Xu, G. Wang, S. Wei, E. Blasch, K. Pham and G. Chen, "High-throughput, cyber-secure multiuser superposition covert avionics system," in IEEE Aerospace and Electronic Systems Magazine, vol. 33, no. 2, pp. 4-15, February 2018.

[20] E. Blasch, R. Sabatini, A. Roy, K. A. Kramer, et al., “Cyber Awareness Trends in Avionics,” IEEE/AIAA 38th Digital Avionics Systems Conference, 2019.

[21] S. Wei, D. Shen, et al., "On effectiveness of game theoretic modeling and analysis against cyber threats for avionic systems," 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), 2015

[22] D. Yu, Y. Jin, Y. Zhang, and X. Zheng, “A survey on security issues in services communication of microservices-enabled fog applications,” Concurrency and Computation: Practice and Experience, p. 4436, 2018

[23] A. Krylovskiy, M. Jahn, and E. Patti, “Designing a smart city internet of things platform with microservice architecture,” in Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on IEEE, pp. 25–30, 2015.

[24] S. Wei, H. Zhang, G. Chen, "On resilience studies of system detection and recovery techniques against stealthy insider attacks," in Proceedings of SPIE, Sensors and Systems for Space Applications IX, 98380G, 2016.

[25] R. Xu, Y. Chen, et al., “Hybrid blockchain-enabled secure microservices fabric for decentralized multi-domain avionics systems,” in Proc SPIE 11422, 2020.

[26] S. Imai, E. Blasch, A. Galli, “Airplane Flight Safety Using Error-Tolerant Data Stream Processing,” IEEE Aerospace and Elec. Syst. Mag., 32(4): 4-17, 2017.

[27] S. Wei, D. Shen and G. Chen, "Secured network sensor-based defense system," in Proceedings of SPIE 9469, Sensors and Systems for Space Applications VIII, 946909, 2015.

[28] D. Homola, J. Boril, V. Smrz, et al., “Aviation Noise-Pollution Mitigation through Redesign of Aircraft Departures,” Journal of Aircraft, 56(5):1-13, Sept: 2019.

[29] A. Munir, et al., “FogSurv: A Fog-Assisted Architecture for Urban Surveillance Using Artificial Intelligence and Data Fusion,” IEEE Access, 9:111938-111959, 2021.

[30] C. Insaurralde, E. Blasch, “Situation Awareness Decision Support System for Air Traffic Management Using Ontological Reasoning,” AIAA Journal of Aerospace Information Systems 19(3), 224, 2022.

[31] N. Chen, et al., “Enabling Smart Urban Surveillance at The Edge,” SmartCloud, 2017.

[32] MAVLink: https://ardupilot.org/dev/docs/mavlink-basics.html.

[33] C. C. Insaurralde, et al. “Veracity Metrics for Ontological Decision-Making Support in Avionics Analytics,” IEEE/AIAA Digital Avionics Systems Conf., 2017.