Efficient Detection of GPS Spoofing Attacks on Unmanned Aerial Vehicles Using Deep Learning

Richmond Asiedu Agyapong, Mahmoud Nabil, Abdul-Rauf Nuhu,
Mushahid I. Rasul, Abdollah Homaifar
Department of Electrical and Computer Engineering
North Carolina A&T State University

2021 IEEE Symposium Series on Computational Intelligence (SSCI) | 978-1-7281-9048-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/SSCI50451.2021.9659972

***Abstract*—Unmanned Aerial Vehicles (UAV) have gained major attention in recent years due to its numerous benefits. The navigation of UAVs is heavily reliant on sensors such as the Global Positioning System (GPS). However, sensor attacks that specifically target the GPS are a major concern. Equipment such as the software-defined radio can be used to launch GPS spoofing attacks. Existing techniques used to monitor the GPS and channel characteristics as a way to detect these attacks are insufficient due to the clear knowledge of the structure of GPS and channel characteristics which gives adversaries a blueprint to launch stealthy attacks. In this work, we developed various deep learning models that rely on UAV flight logs and telemetry data to detect GPS spoofing attacks in real time. We generate UAV data for different UAV models and use the data to train the models to detect the GPS spoofing attacks through classification. We developed two types of detectors: a Long Short-Term Memory (LSTM) binary classifier and an LSTM autoencoder-based one-class classifier (OCC). For each type, we compared two variations namely a UAV-specific detector that is customized for different UAV types and a UAV-generalized detector that can work with any model of UAV. We test and compare the efficiency of detection models in simulation. Hardware validation experiments were also performed using Intel® NCS2. For the binary classifier, we get a detection accuracy of 97.79% for the UAV-generalized detector and up to 99.56% for the UAV-model-specific detector. While for the OCC, we get 94.98% for the UAV-generalized detector and up to 99.24% for the UAV-model-specific detector.**

***Index Terms*—GPS spoofing, NCS2, one-class classifier, inference, LSTM, deep learning**

I. INTRODUCTION

With the current ever-advancing revolution in autonomous systems, many tasks can now be done with little or no human intervention. Leveraging sensory information, autonomous systems can better inform their decision-making capabilities in tasks that seemed to be impossible before. Unmanned Aerial Vehicles (UAVs) are an example of these autonomous technologies that are equipped with multiple sensors such as gyroscopes, accelerometers, LiDAR, thermal sensors, optical cameras, and GPS/GNSS receivers. The use of UAVs originated from the military [1], where drones are used for dangerous missions such as detecting bombs [2], precision airstrikes [3], and surveillance and reconnaissance missions. Due to their speed, precision, and convenience, UAVs have grown in popularity in recent years among civilians industries such as health care systems, agricultural, retail, and delivery and surveillance of transportation sectors [4], [5], [6].

Despite their dominance, UAVs are vulnerable to a variety of onboard sensor spoofing attacks. In these attacks, adversaries aim to falsify the data collected by the sensors from the UAV's surroundings to make the UAV take false decisions. The GPS spoofing attack is an example where the precision of the mission is harmed when the onboard GPS sensor receives erroneous data. For example, if a UAV on a delivery mission receives the incorrect GPS signal, the delivery would be sent to the wrong location [7], resulting in a loss of money, and in military applications, errors in computing the position might result in the destruction of the wrong target [8]. Because the GPS signal is not encrypted, it is vulnerable to different attacks such as spoofing, meaconing, and jamming [9], [10], [11]. Humphery et al. demonstrated the simplicity of spoofing the GPS receiver of an $80M megayacht and in [12] the susceptibility of UAVs to GPS spoofing was evaluated. Software Defined Radio (SDR) is an available piece of equipment that can be used to launch GPS spoofing effectively [13]. SDR has contributed to the proliferation of GPS spoofing techniques such as meaconing and replay attacks [14].

Various research advancements have recently been achieved in the attempt to protect UAVs and detect GPS spoofing attacks. These include GPS signal and channel monitoring techniques [15], mathematical modeling or control-based techniques, and cryptography-based techniques [16], [17]. However, since the structure of GPS signal and its communication channels are well-documented [18], attackers can imitate the legitimate signals without detection, and hence existing methods that rely on monitoring GPS signal and modeling channel characteristics have proven to be challenging and borderline unreliable. Cryptography-based techniques [16], [17] on the other hand require restrictive infrastructure assumptions, which is unlikely to be satisfied.

Considering the limitations in literature, we aim to provide a more reliable and practical solution to detect GPS spoofing attacks. We aim at: (a) having a reliable dataset for building and testing GPS spoofing detector models for UAVs and (b) building an effective machine learning/data-driven approach to detecting GPS attacks on UAVs in real-time. We start by generating a dataset using the Gazebo simulator. We use the PX4 autopilot firmware to simulate UAVs and QGroundControl App for mission planning and control. During mission flights, we collect flight log data. Afterward, we use the data to train a machine learning model to detect GPS spoofing attacks on UAVs. We then test the model and validate the result by deployment onto hardware.

The main contributions of this paper are summarized as follows:

978-1-7281-9048-8/21/$31.00 ©2021 IEEE

- We generated benchmark telemetry data for testing and building GPS spoofing detectors for UAVs using the Gazebo simulator. Compared to the existing data [19], [20], we take into consideration a variety of environmental factors, such as wind gusts and various degrees of spoofing that are likely to occur. Telemetry data from several UAV types are also included in the dataset.
- We developed two LSTM-based classifiers to detect GPS spoofing attacks. One is a binary classifier that is trained using dataset that contains both benign and spoofed UAV data. The other is a one-class classifier that is trained using only benign data. We also train model-specific classifiers, as well as UAV-generalise classifiers from the aggregation of data from all selected UAV models. We obtained competitive results compared to existing GPS spoofing detection approaches.
- We validated the performance and feasibility of the detection models by deploying it on Intel® Neural Computing Stick (NCS2) for testing on hardware. We obtained competitive inference times for both models, which makes the models good for real-time usage.

The rest of this paper is organized as follows: in section II, we discuss related work and advances in recent research on GPS attacks as well as the detection and defense mechanisms against these attacks. In section III, we give the necessary background on the problem and discuss some technologies, tools, and algorithms that are used in this work. In section IV, we describe the experimental setup, the attack model, and the detection models that have been used in this work, and in section V, we give some more information about the structure of the data that has been generated and used for this work and then discuss results. Finally, section VI provides the recommendation for future work and the conclusions.

II. RELATED WORK

To protect UAVs against GPS spoofing attacks, different techniques have been proposed in the literature. These techniques may be divided into 3 categories. GPS signal and channel monitoring techniques, Cryptography-based approaches, and mathematical modeling or control techniques are all examples of these. GPS signal and channel monitoring techniques involve monitoring the GPS signal characteristics for observables changes such as signal strength, Doppler effect, Pseudorange, and signal-to-noise ratio (SNR). They then use a threshold or machine learning techniques to classify the signals. For instance, in [21] a GPS receiver that uses auxiliary peak tracking was proposed. The receiver that is proposed tracks both the strong and weak correlation peaks of a GPS satellite. This technique offers protection against sturdy adversaries that can launch seamless takeover attacks due to the receiver's ability to track weaker correlation peaks. In [22], data-driven techniques are used as a basis of spoofing detection. The data contained multiple GPS basic observables and their cross-correlation. However, using GPS signals and these observables as the underlying data for spoofing detection is not sufficient. The GPS signal structure is well documented and adversaries have found smart ways to replicate the signal and get away with spoofing mechanisms that use these observables as a part of their detection.

Cryptography-based techniques involve using cryptography approaches to validate the authenticity of GPS signals. O'Hanlon et al [16] use the characteristics of the encrypted military P(y) code signal to check for the presence of spoofing attacks. Their method makes use of correlations between the P(y) code from two GPS receivers to validate the civilian GPS signal. In their work, one of the two GPS receivers was secured from a spoofing source. However, for this to be applied, the detection system needs to have two receivers, one of which should be resistant to spoofing. Kerns et al detail the use of reserved and unallocated bits in the GPS navigation message for authentication [17]. Their research suggests adding digital stamps to the navigation message which will be used to authenticate the signal. This method also requires a change to GPS infrastructure. In [23], a technique fuses the signal timing based on statistical hypothesis tests with authentication of the GPS navigation message to validate GPS signals. However, the adoption of these cryptography-based techniques requires restrictive infrastructure assumptions, which are unlikely to be satisfied.

Mathematical modeling or control based techniques involves using mathematical and control framework to detect GPS spoofing. The detection approach in [24] is based on analysis of state estimation using SVM for detecting inconsistencies in the error distribution between GPS and inertial navigation system (INS). This method is only effective for a short-duration attack. As the attack duration grows longer, the performance suffers significantly. In [25], data from crowd-sourced air traffic monitors are used to detect GPS spoofing attacks. The discrepancies in the data reported by different UAVs are the main factor for the detection. A 15-minute time window is required to detect the attack when this method is used. This is enough time for the adversary to accomplish its mission.

Compared to these methods, we used telemetry data from different UAVs' models to test and build deep-learning-based GPS spoofing detectors. We take into consideration a variety of environmental factors, such as wind gusts and various degrees of spoofing that are likely to occur. We validated the performance and feasibility of the detection models by deploying them on Intel® NCS2 for testing on hardware.

III. BACKGROUND

This section discusses the necessary background on GPS signal structure, UAV navigation, and machine learning models that we will use in our detectors.

A. GPS and UAV Navigation

The Global positioning system (GPS) consists of a constellation of satellites that transmit signals down to the earth to help vehicles calculate their positions for navigation purposes. The satellites transmit NAV/SYSTEM data which contains information about the satellite timing and synchronization

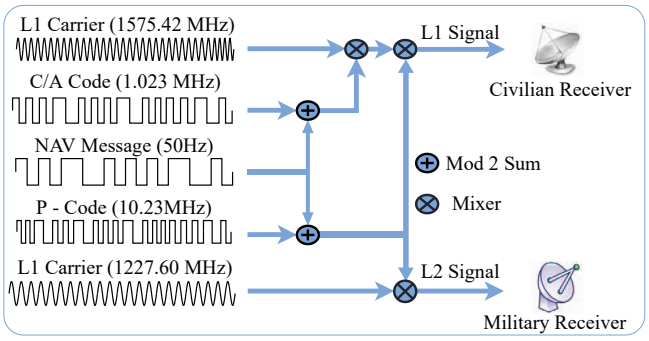

Figure 1. GPS Satellite signal composition.

of the signals, time correction information, precise orbital (ephemeris), and approximate orbital (almanac) data, correction signal, and ionospheric data. The NAV/SYSTEM data helps the receiver to determine the transmission time of each signal and the exact position of every satellite. From this information, the exact location of the receiver is inferred. GPS satellites generate two pseudo-random noise codes, the coarse acquisition (C/A) code for traditional positioning service and the precision (P) code for precise positioning service. Each GPS satellite has a unique C/A code used to identify them. The P-code has been encrypted with the y code as P(y), which is only useful to the military. GPS satellites have two main channels, L1 carrier at 1575.42MHz and L2 carrier at 1227.60MHz. On the L2 channel are the P(y) code and the NAV/SYSTEM data. on the L1 channel are the C/A code, the P(y) code, and the NAV/SYSTEM data. The P(y) code can not be decrypted by civilian receivers, hence the L2 channel is for military use only.

B. Long Short-Term Memory RNNs

Recurrent Neural Networks (RNNs) are a class of neural networks that take a sequence of data as input to generate the output. RNNs have a hidden state that allows them to remembers past inputs within the sequence and use them to make decisions. The Long Short-Term Memory (LSTM) improves on the RNN to solve vanishing and exploding gradient problem [26] by substituting the hidden state for an LSTM cell. Figure 2 shows the LSTM cell structure [27].

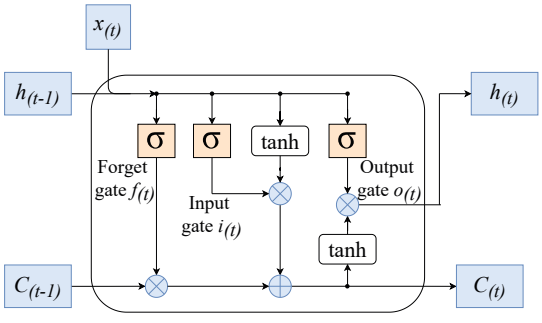

Figure 2. The LSTM cell

The LSTM cell has gates that determine the information to be kept or dropped at each step and the cell state, which is the long-term memory. The input gate chooses what new information should go into the cell state, the forget gate used the sigmoid function to decide what information from the previous hidden state and current input should be kept and propagated to the next time step. Inferring from Fig. 2, the below equations gives the calculations at each gate:

$$f_{(t)} = \sigma\left(W_f x_{(t)} + W_f h_{(t-1)} + b_f\right) \tag{1}$$

$$i_{(t)} = \sigma\left(W_i x_{(t)} + W_i h_{(t-1)} + b_i\right) \tag{2}$$

where $f_{(t)}$, $i_{(t)}$, $x_{(t)}$ and σ are the forget gate, input gate, input and the sigmoid function respectively. W_f, W_i, denote the weights, and b_f, b_i denote the biases of the forget and input gates. The output gate also decides the next hidden state.

$$o_{(t)} = \sigma\left(W_o x_{(t)} + W_o h_{(t-1)} + b_o\right) \tag{3}$$

$$C_{(t)} = f_{(t)} * C_{(t-1)} + i_{(t)} * \tanh\left(W_c x_{(t)} + W_c h_{(t-1)} + b_c\right) \tag{4}$$

$$h_t = o_{(t)} * \tanh\left(C_{(t)}\right) \tag{5}$$

where $o_{(t)}$, $C_{(t)}$ and h_t are respectively the output gate, cell state and hidden state. W_o and W_c denote the weights, while b_o and b_c denote the biases of the output gate and cell state.

C. Autoencoder

Auto-encoders are a class of neural networks that takes an input ($\boldsymbol{x}$) and gives an output ($\boldsymbol{x'}$) that is a replica of the input. It compresses the input with an encoder to obtain an encoded or compressed version of the input and then reconstructs it with a decoder. The compressed representation of the input is necessary for efficient dimensionality reduction [28].

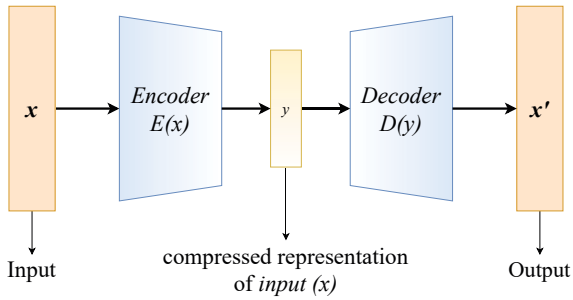

Figure 3. The Autoencoder

Auto-encoders can be used efficiently to detect anomalies in data leveraging the idea of one-class classification (OCC) [29]. In specific, the autoencoder learns the reconstruction of one data class of the input by minimizing the reconstruction error that indicates how close the output is to the input. Anomalies can be identified by setting a static threshold, or a dynamic threshold of the reconstruction error with techniques such as rolling mean or exponential mean [30]. When an input has a reconstruction error that is greater than the threshold, this input will be marked as an anomaly.

IV. METHODOLOGY

In this section, we discuss the workflow to generate a spoofing attack dataset (i.e., attack model), the simulation setup, and the spoofing attack detection algorithms used.

A. Proposed Framework and Simulation Setup

The workflow is presented in Figure 4. We first generate and collect datasets using the Gazebo simulator and QGroundControl, then we preprocess that data to obtain optimal features for training and testing the spoofing detectors. After building the model, we then deploy the model for test on the simulated UAV in the Gazebo simulator. This is done by developing the model as a plugin in the Gazebo simulator. To further validate the inference time of the models, we deploy them on Intel® NCS2 hardware, an inference device used to run deep learning models on the edge. [31].

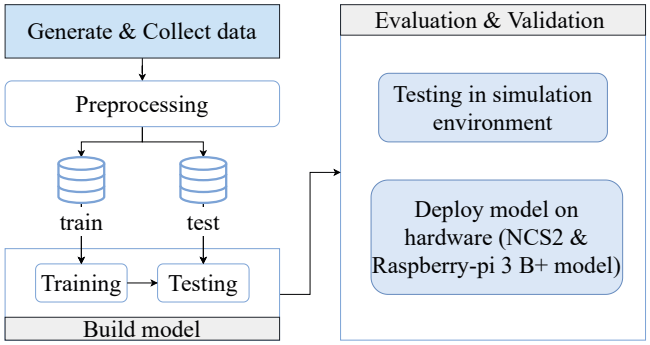

Figure 4. Workflow and overview of proposed work

To obtain data for training the detection models, we simulate UAVs using PX4 autopilot firmware and Gazebo simulator. QGroundControl App (QGC) was used as the ground station. We employed PX4 autopilot firmware in both HITL (hardware in the loop) mode and SITL (software in the loop) mode. The PX4 firmware is run in HITL mode on real flight controller hardware connected via USB/UART to the development machine running the Gazebo simulator. Whereas in SITL mode, PX4 autopilot firmware is run on a simulated drone in a simulated environment. The UAV models for SITL simulation include Typhoon H480, Standard Plane, Standard VTOL, Tailsitter VTOL, the default Quadrotor, while Intel® Aero Ready to Fly Drone was used for HITL simulation. The QGC was used for mission control and mission planning. The adversarial GPS spoofing attack was simulated in the Gazebo environment as discussed in Section IV-B. Using the workflow outlined, we design and conduct various missions while recording telemetry data during the mission flight.

B. Attack Model: Generating The Spoofing Source in Gazebo

The goal of GPS spoofing is to deceive the UAV into believing it is at a different location than it is. Thus, when there is no GPS spoofing attack, the UAV's receiver computes its correct location as *loc* - this describes the longitude and the latitude of the current position. However, during the spoofing attack, the adversary sends a signal that deceives the UAV's

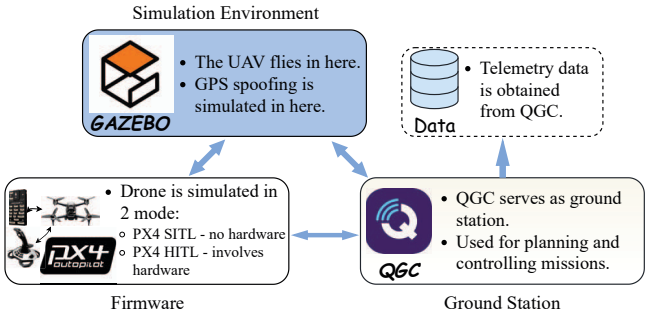

Figure 5. Setup for generating Data

receiver to compute its location as $(loc+\varepsilon)$. The ε constitutes an offset in the longitude and the latitude and it denotes the amount of deviation (in meters) that is caused by the spoofing. To achieve this illusion in the Gazebo simulator, the correct GPS message that is transmitted by the GPS plugin (i.e., a Gazebo shared library) has to be compromised and modified. We achieve this by modifying the GPS plugin that generates GPS readings. This modification creates a fake message, which is transmitted to the simulation environment. Algorithm 1 gives the steps taken to create and transmit the GPS spoofing in the Gazebo simulator. In this algorithm, the amount of deviation is denoted as (ε) while (d_M) denotes the distance of UAV from the takeoff, at which the spoofing starts. While spoofing has been activated, the algorithm gets the current position (loc) of the UAV. The initial position, (A) of the UAV at takeoff is stored. Then the fake position $(loc+\varepsilon)$ is transmitted while the UAV's current position (loc) is d_M away from the takeoff (A). Steps 6-10 ensures that the spoofing is occurring at d_M from the takeoff. This is to mimic real-world spoofing which usually occurs far away from the takeoff point of vehicles. Figure 6 shows a GPS spoofing attack on a UAV.

Algorithm 1: Creating GPS spoofing in Gazebo

```
Data: Amount of deviation to be caused by
      spoofing, ε
      Distance at which spoofing starts,
      d_M
      Set init to 1 to indicate start of
      mission
Result: The UAV in the Gazebo environment
        will deviate from its original
        position by ε
1  initialization;
2  while spoofing is activated do
3  |  get current position of UAV, (loc)
   |   if init = 1 then
4  |  |  store initial position of UAV, (A)
   |  |   set init to 0
5  |  end
6  |  if (Distance(A,loc) > d_M) then
7  |  |  Transmit (loc+ε) to UAV
8  |  else
9  |  |  Transmit the true position, X
10 |  end
11 end
```

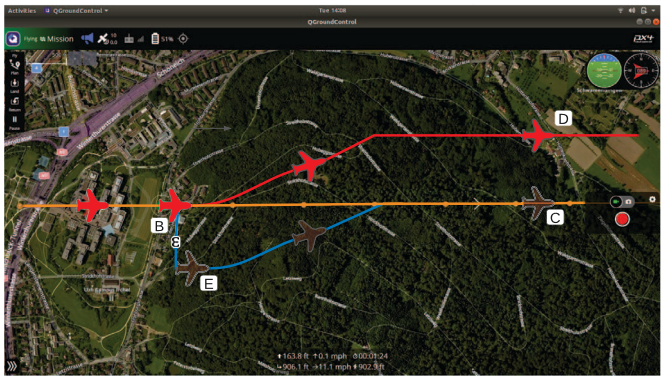

Figure 6. Mission trajectory: The yellow line indicates the planned mission trajectory, whereas the red line indicates the reset path At point, B where the spoofing is introduced, we see a deviation of ε from the planned trajectory to point E. In an attempt of the UAV to go back to the planned mission trajectory, its path is set to a different trajectory (reset path), hence landing the UAV at D instead of C.

C. Defense Models: GPS Spoofing Attack Detection

To provide benchmark results for detecting GPS spoofing on the generated dataset, we use data-driven approaches. The detection was carried out using two LSTM-based algorithms namely: *(1)* an LSTM-based binary classifier for sequence classification and *(2)* an LSTM-based autoencoder one-class classifier. Since the input data is a time series that represents different UAVs' sensor readings sorted in chronological order, the LSTM-based models can exploit the temporal correlations between the input features of the data.

1) LSTM-based Binary Classifier: The model consists of four LSTM layers, four dropout layers, and a fully connected layer with softmax as the activation function. The hyper-parameters were optimized using grid search for each UAV dataset [32]. Binary Cross-Entropy is used for the loss function. Table I shows the architecture and the number of cells for each layer. The model was configured with the parameters shown in Table IV.

2) LSTM-based Autoencoder One-Class Classifier (OCC): The model consists of an encoder with 3 LSTM layers, 3-layer decoder, and a `RepeatVector` layer to serve as encoder-decoder bridge. The Mean absolute error (MAE) was chosen as the loss function due to its robustness to outliers [33]. The resulting model can detect GPS spoofing by flagging the input as an anomaly if the mean absolute error (MAE) is above the set threshold. The exponential Moving Average technique is used to determine the threshold for detecting the anomalies. Table II gives the model architecture and the parameters of the LSTM-based autoencoder OCC model. The model was configured with the parameters shown in Table IV.

D. Testing and Validation

To validate the GPS spoofing detection model, we test the model by deploying it in simulation in the GAZEBO environment. We then prepare the model for deployment on edge hardware. We use the Intel® Neural Compute Stick 2 and Raspberry pi 3 B+. To run the model on the NCS2, we prepare and optimize the pre-trained model and convert it into a suitable representation that contains information about the weights biases and the topology of the model. We then deploy the model on the Intel® NCS2 and connect to the Raspberry Pi and Flight controller for testing. We evaluate the model's performance by measuring the inference time in a real-time application.

Table I
MODEL SUMMARY (LSTM-BASED BINARY CLASSIFIER).

Model: "Binary Classifier" Layer (type)	Output Shape	Param #
lstm_5 (LSTM)	(None, 10, 7)	420
dropout_5 (Dropout)	(None, 10, 7)	0
lstm_6 (LSTM)	(None, 10, 20)	2240
dropout_6 (Dropout)	(None, 10, 20)	0
lstm_7 (LSTM)	(None, 10, 10)	1240
dropout_7 (Dropout)	(None, 10, 10)	0
lstm_8 (LSTM)	(None, 5)	320
dropout_8 (Dropout)	(None, 5)	0
dense_2 (Dense)	(None, 2)	12
Total params: 4,232 Trainable params: 4,232 Non-trainable params: 0		

Table II
MODEL SUMMARY (LSTM-BASED AUTOENCODER ONE CLASS CLASSIFIER).

Model: "Autoencoder" Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 1, 9)	0
lstm (LSTM)	(None, 1, 16)	1664
dropout (Dropout)	(None, 1, 16)	0
lstm_1 (LSTM)	(None, 4)	336
dropout_1 (Dropout)	(None, 4)	0
tf.expand_dims (TFOpLambda)	(None, 1, 4)	0
tf.tile (TFOpLambda)	(None, 1, 4)	0
lstm_2 (LSTM)	(None, 1, 4)	144
dropout_2 (Dropout)	(None, 1, 4)	0
lstm_3 (LSTM)	(None, 1, 16)	1344
dropout_3 (Dropout)	(None, 1, 16)	0
time_distributed (TimeDistri	(None, 1, 9)	153
Total params: 3,641 Trainable params: 3,641 Non-trainable params: 0		

V. EVALUATION & RESULTS

In this section, we discuss the properties of the generated dataset, the performance metrics used in the evaluation, the developed model experimental results, and the deployment results.

A. Description of Dataset

The generated telemetry dataset consists of both benign and spoofed trajectory data and its associated sensory readings of six UAVs models that include: Typhoon H480, Standard Plane, Standard VTOL, Tailsitter VTOL, the default Quadrotor, and Intel® Aero RTF. The spoofed data was collected while the spoofing plugin in Gazebo was active. The amount of

deviation, ε, that was introduced ranges from as low as 5 meters up to 150 meters. Table III shows a summary of the UAV models used and the number of the sampled missions. The active flight time for each mission was between twenty to thirty minutes. For spoofed missions, the spoofing is activated after three minutes from take-off. All trajectories are assumed to be straight lines.

Telemetry readings are time series with eighty-eight features that include the pitch, roll, heading, groundspeed, and other features that are related to the inertia measurement unit (IMU), other sensors that are affected by the position of the UAV. To pre-process the dataset, we performed principal component analysis on the data to reduce the dimensionality while retaining the strongest features set. This process also helps us to filter out the noise from the data.

Table III
DATASET DESCRIPTION: UAVS' MISSIONS SUMMARY.

UAV Model	Spoofed Missions	Benign Missions
Quadrotor	80403	89180
Typhoon H480	7980	16560
Standard Plane	11166	20825
Tailsitter VTOL	15481	14948
Standard VTOL	13682	17041
Intel® Aero	6907	13989

B. Performance Metrics

We utilize conventional metrics such as recall, precision, accuracy, and the F-1 score to assess the performance of the developed models. Precision is the percentage of predicted positive observations that are truly positive, whereas recall is the percentage of predicted positive observations made out of all positive examples in the dataset. The accuracy gives a measure of all the observations, both positive and negative, that are predicted correctly. The last metric is the F1-score, which is the harmonic mean of Precision and Recall. F1-score is a more useful metric when data is unbalanced.

C. Evaluation of the Proposed Detection Model

1) Models' Evaluation: We trained the Binary classifier with both benign and spoofed data while training the Autoencoder one-class classifier (OCC) with only benign data. All models were trained on an Intel(R) Core i7-9700 CPU with RAM of 16GB and a 3.00GHz processor. All models were trained for 100 epochs. The window size of the LSTM input is 60 seconds. Table IV describes the hyper-parameters that were used after the grid search.

For both the binary classifier and autoencoder OCC architectures, we trained a generalized detector using the dataset from all selected UAV models. We also trained UAV-specific detectors for each model of UAV. Both UAV-specific detectors and generalized detectors were trained using the same model architecture described in subsection IV-C. The results in Table V show that training a UAV-specific detector gives a higher detection accuracy than a generalized detector. For the generalized detector, we obtained detection accuracy of 97.79% & 94.98% for the Binary classifier and autoencoder OCC models respectively, while for UAV-specific detectors we recorded up to 99.56% accuracy for the binary classifier architecture and up to 99.24% for the autoencoder OCC architecture. Table VI shows the average recall, precision, accuracy, and F1-score of the UAV-generalized detectors for binary classifier and OCC architectures.

Table IV
MODEL PARAMETERS.

Parameter	Binary Classifier	Autoencoder OCC
Activation function	Sigmoid	Softmax
Loss function	Categorical cross-entropy	MAE
Optimizer	Adam	Adam
Sequence length	50	50
Learning rate	0.002	0.002
Epochs	100	100
Dropout	0.2	0.2

Table V
TEST ACCURACY OF LSTM DETECTORS ON DATASET FROM VARIOUS UAV MODELS.

UAV Model	Binary classifier	Autoencoder OCC
Quadrotor	99.56%	99.24%
Typhoon H480	98.67%	98.84%
Standard Plane	96.95%	97.09%
Standard VTOL	98.81%	98.11%
Tailsitter VTOL	97.22%	98.47%
Intel® Aero	97.17%	94.67%
UAV-generalized detector	97.79%	94.98%

We also test the sensitivity of the model to stealthy spoofing by testing for spoofing attacks where the deviation caused is less than 10 meters. Figures 7 and 8 compare how well the detectors perform during short-range spoofing. The models have competitive results in detecting GPS spoofing attacks that cause minor deviation, obtaining a 93.97% accuracy for the generalized detector in the case of the binary classifier and 90.94% in the case of the autoencoder one-class classifier.

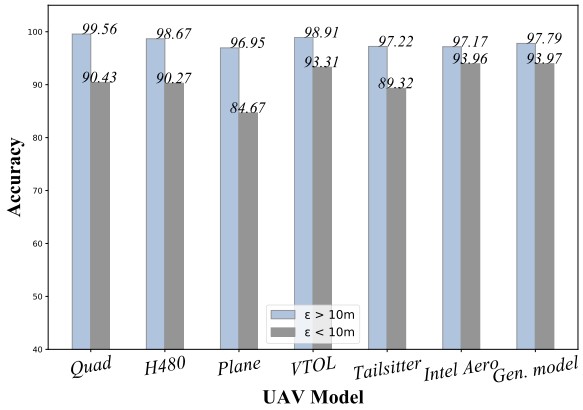

Figure 7. Comparison of detection accuracy of the LSTM-based binary classifier for different deviations (ε).

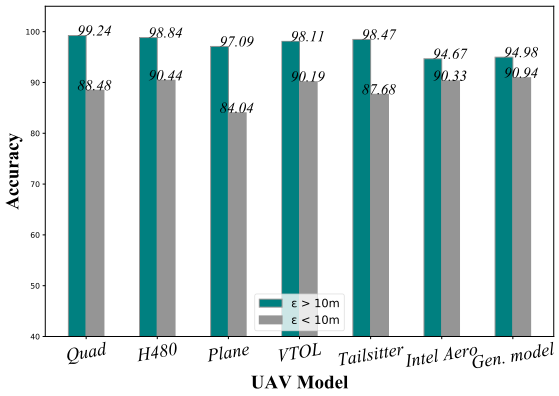

Figure 8. Comparison of detection accuracy of the LSTM-Auto-encoder one-class classifier (OCC) for different deviations (ε).

Table VI
AVERAGE RECALL, PRECISION, AND F1-SCORE OF THE UAV-GENERALIZED DETECTORS FOR BOTH ARCHITECTURES.

Metric	Binary Classifier	Autoencoder OCC
Recall	97.79%	94.93%
Precision	97.89%	95.03%
F1-score	97.81%	94.93%
Accuracy	97.79%	94.98%

2) Inference on Hardware: To test how effectively the GPS spoofing detectors operate in real-time and determine the overhead that the model adds to the UAV operation, we measured the inference time on edge hardware. We deployed the model and ran it on the Intel® NCS2 to gauge the inference time. The average inference time for the LSTM–based autoencoder OCC is 4.32 ms while that of the binary classifier is 3.986 ms. Fig. 9 shows the average inference execution time of 100 queries for both LSTM–based autoencoder OCC and binary classifier models. The results show the effectiveness of the developed models to give accurate decisions in real-time.

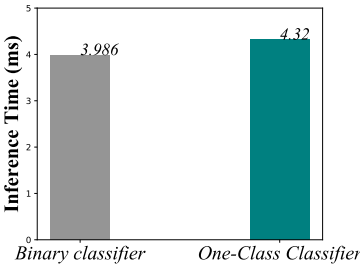

Figure 9. Total inference time for running LSTM–based autoencoder OCC and binary classifier models.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a benchmark telemetry dataset generated from UAV flights using the Gazebo simulator as the simulation environment. We used PX4 Autopilot firmware to simulate drones in both SITL and HITL modes. We planned and executed mission flights using QGroundControl App. During some missions, GPS spoofing was introduced to obtain malicious data, whereas other flights were benign without spoofing. Using this data, we train various deep learning models. In addition, we validated the overhead added by the models on Intel® NCS2 by measuring the inference time. The results show the effectiveness of the developed models to detect deviations caused by GPS spoofing with competitive accuracy and minor overhead. This work provides an opportunity to use telemetry data that is readily available during flights missions to detect GPS spoofing attacks. Future work includes proposing effective ways to respond when a GPS spoofing attack is detected.

ACKNOWLEDGMENT

This work is supported by the NASA University Leadership Initiative (ULI) under grant number 80NSSC20M0161 and partly supported by the National Science Foundation (NSF) award number 2018879.

REFERENCES

[1] Haque Nawaz, Husnain Mansoor Ali, et al. “Applications of Unmanned Aerial Vehicles: A Review”. In: (2019).

[2] Dinesh Sathyamoorthy. “A review of security threats of unmanned aerial vehicles and mitigation steps”. In: *J. Def. Secur* 6.1 (2015), pp. 81–97.

[3] Lynn E Davis et al. *Armed and dangerous? UAVs and US security*. Tech. rep. RAND Corp Santa Monica Ca, 2014.

[4] MA Jiménez-Bello et al. “Assessment of drip irrigation sub-units using airborne thermal imagery acquired with an Unmanned Aerial Vehicle (UAV)”. In: *Precision agriculture’13*. Springer, 2013, pp. 705–711.

[5] Sudarshan Sreeram and Lokesh Shanmugam. “Autonomous robotic system based environmental assessment and dengue hot-spot identification”. In: *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*. IEEE. 2018, pp. 1–6.

[6] Kimberly M Fornace et al. “Mapping infectious disease landscapes: unmanned aerial vehicles and epidemiology”. In: *Trends in parasitology* 30.11 (2014), pp. 514–519.

[7] Juhwan Noh et al. “Tractor beam: Safe-hijacking of consumer drones with adaptive GPS spoofing”. In: *ACM Transactions on Privacy and Security (TOPS)* 22.2 (2019), pp. 1–26.

[8] Donna A Dulo. “Unmanned aircraft: the rising risk of hostile takeover [leading edge]”. In: *IEEE Technology and Society Magazine* 34.3 (2015), pp. 17–19.

[9] Jon S Warner and Roger G Johnston. *Think GPS cargo tracking= high security*. Tech. rep. Think again. Technical report, Los Alamos National Laboratory, 2003.

[10] Panagiotis Papadimitratos and Aleksandar Jovanovic. "GNSS-based positioning: Attacks and countermeasures". In: *MILCOM 2008-2008 IEEE Military Communications Conference*. IEEE. 2008, pp. 1–7.

[11] Nils Ole Tippenhauer et al. "On the requirements for successful GPS spoofing attacks". In: *Proceedings of the 18th ACM conference on Computer and communications security*. 2011, pp. 75–86.

[12] Daniel P Shepard et al. "Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks". In: *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*. 2012, pp. 3591–3605.

[13] Weike Feng et al. "Software-Defined Radio Implemented GPS Spoofing and Its Computationally Efficient Detection and Suppression". In: *IEEE Aerospace and Electronic Systems Magazine* 36.3 (2021), pp. 36–52.

[14] Mark L Psiaki and Todd E Humphreys. "GNSS spoofing and detection". In: *Proceedings of the IEEE* 104.6 (2016), pp. 1258–1270.

[15] Desmond Schmidt et al. "A survey and analysis of the GNSS spoofing threat and countermeasures". In: *ACM Computing Surveys (CSUR)* 48.4 (2016), pp. 1–31.

[16] Brady W O'Hanlon et al. "Real-time GPS spoofing detection via correlation of encrypted signals". In: *NAVIGATION, Journal of the Institute of Navigation* 60.4 (2013), pp. 267–278.

[17] Andrew J Kerns, Kyle D Wesson, and Todd E Humphreys. "A blueprint for civil GPS navigation message authentication". In: *2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014*. IEEE. 2014, pp. 262–269.

[18] James J Spilker Jr. "GPS signal structure and performance characteristics". In: *Navigation* 25.2 (1978), pp. 121–146.

[19] Jason Whelan et al. "Novelty-based Intrusion Detection of Sensor Attacks on Unmanned Aerial Vehicles". In: *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. 2020, pp. 23–28.

[20] Jason Whelan et al. "UAV Attack Dataset". In: (2020). DOI: 10.21227/00dg-0d12. URL: https://dx.doi.org/10.21227/00dg-0d12.

[21] Aanjhan Ranganathan, Hildur Ólafsdóttir, and Srdjan Capkun. "Spree: A spoofing resistant gps receiver". In: *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 2016, pp. 348–360.

[22] Silvio Semanjski et al. "Use and validation of supervised machine learning approach for detection of GNSS signal spoofing". In: *2019 International Conference on Localization and GNSS (ICL-GNSS)*. IEEE. 2019, pp. 1–6.

[23] Kyle Wesson, Mark Rothlisberger, and Todd Humphreys. "Practical cryptographic civil GPS signal authentication". In: *NAVIGATION, Journal of the Institute of Navigation* 59.3 (2012), pp. 177–193.

[24] G Panice et al. "A SVM-based detection approach for GPS spoofing attacks to UAV". In: *2017 23rd International Conference on Automation and Computing (ICAC)*. IEEE. 2017, pp. 1–11.

[25] Kai Jansen et al. "Crowd-gps-sec: Leveraging crowdsourcing to detect and localize gps spoofing attacks". In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 1018–1031.

[26] Yuhuang Hu et al. "Overcoming the vanishing gradient problem in plain recurrent networks". In: *arXiv preprint arXiv:1801.06105* (2018).

[27] Xiaofeng Yuan, Lin Li, and Yalin Wang. "Nonlinear dynamic soft sensor modeling with supervised long short-term memory network". In: *IEEE transactions on industrial informatics* 16.5 (2019), pp. 3168–3176.

[28] Jeremy Jordan. "Introduction to autoencoders". In: (2018). Jeremyjordan.me/ [Online; posted 19 MAR 2018].

[29] Seung Yeop Shin and Han-joon Kim. "Autoencoder-based one-class classification technique for event prediction". In: *Proceedings of the 2019 4th International Conference on Cloud Computing and Internet of Things*. 2019, pp. 54–58.

[30] Gali Katz. "Anomaly detection using LSTM with Autoencoder". In: (2020). taboola.com [Online; posted 14 September 2020].

[31] "Object detection using the Intel Neural Compute Stick 2 and OpenVINO". In: (2018). goodaudience.com [Online; posted 30 May 2019].

[32] Pier Paolo Ippolito. "Hyperparameters Optimization". In: https://towardsdatascience.com/ [Online; posted 26 Sept 2019]. Towards Data Science, 2019.

[33] Tianfeng Chai and Roland R Draxler. "Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature". In: *Geoscientific model development* 7.3 (2014), pp. 1247–1250.