Insights on Using Deep Learning to Spoof Inertial Measurement Units for Stealthy Attacks on UAVs

Kyo Hyun Kim*, Denizkhan Kara*, Vineetha Paruchuri†, Sibin Mohan†,
Greg Kimberly‡, Denis Osipychev‡, Jae H. Kim‡, Josh D. Eckhardt‡ and Miroslav Pajic§
*Department of Computer Science, University of Illinois at Urbana-Champaign, {kkim103, kara4}@illinois.edu
†Department of Computer Science, The George Washington University, {vineetha.paruchuri, sibin.mohan}@gwu.edu
‡Boeing Research & Technology, {greg.kimberly, denis.osipychev, jae.h.kim, josh.d.eckhardt}@boeing.com
§Department of Electrical and Computer Engineering, Duke University, {miroslav.pajic}@duke.edu

***Abstract*—Unmanned Aerial Vehicles (UAVs) find increasing use in mission critical tasks both in civilian and military operations. Most UAVs rely on Inertial Measurement Units (IMUs) to calculate vehicle attitude and track vehicle position. Therefore, an incorrect IMU reading can cause a vehicle to destabilize, and possibly even crash. In this paper, we describe how a strategic adversary might be able to introduce spurious IMU values that can deviate a vehicle from its mission-specified path while at the same time evade customary anomaly detection mechanisms, thereby effectively perpetuating a "stealthy attack" on the system. We explore the feasibility of a Deep Neural Network (DNN) that uses a vehicle's state information to calculate the applicable IMU values to perpetrate such an attack. The eventual goal is to cause a vehicle to perturb enough from its mission parameters to compromise mission reliability, while, from the operator's perspective, the vehicle still appears to be operating normally.**

***Index Terms*—UAV, IMU, Deep Learning**

I. Introduction

There is a growing interest in exploring the use of Unmanned Aerial Vehicles (UAVs) in contexts such as agriculture [1], logistics [2], military [3]–[5] etc. In the current (2022) Ukraine conflict, various types of UAVs, including commercial-grade UAVs [6] are being used for military surveillance [7] and attack [8] purposes. The Internet of Things (IoT) All signs point to continued, if not increased, use of UAVs in various use cases, both in civilian and military contexts. Given this status quo, understanding the vulnerabilities of UAV systems is crucial for identifying potential threats and finding countermeasures against threats.

There are various types of UAVs, but for the purposes of our current work we limit our focus to quadrotors: a multi-copter aircraft with four rotors. Since most UAVs use Inertial Measurement Units (IMUs) for navigation, we start by exploring whether the IMUs can be manipulated to exploit any resulting vulnerabilities. Others [9], [10] have looked into manipulating IMUs a few different ways, and also found [11] that naive manipulation of IMUs can alert the anomaly detection system thereby triggering the recovery mechanisms.

This work is supported by a grant from Boeing Research Technology (BRT), Office of Naval Research (ONR) grant N00014-21-1-2217 as well as the National Science Foundation (NSF) grant CPS-2145787. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of Boeing, ONR or NSF.

Existing attacks on IMUs lack fine control of the system. We want to exercise precise control of UAVs by manipulating the IMU values more effectively. To this end, we explore the possibility of an adversary introducing spurious IMU values that evade the anomaly detection system. Specifically, we aim to *(i)* examine and establish the necessary conditions for UAVs to deviate from mission parameters, and *(ii)* explore the range of control capabilities of an adversary that can inject malicious values in readings from IMU sensors [12]. Injecting spurious or malicious values in a system can be termed as "spoofing". A UAV sensor is considered spoofed when the attacker can deviate the UAV's position and velocity estimates (hence, mission parameters) by injecting malicious sensor readings.

Our goal is to develop IMU spoofing attacks that are *stealthy*, *i.e.,* attacks that can practically drive UAVs off their mission without raising alarms based on state estimators commonly used in UAV systems to detect sensor anomalies [13]. We analyze the dynamics of UAV control loops to assess the practicality of launching such stealthy attacks, and test our hypothesis using realistic simulations.

A real-world instance of stealthy IMU spoofing can be deployed in a scenario similar to the multi-UAV surveillance mission described by Manyam et al. [14] where UAVs patrol a designated area and report back to the base. The mission involves UAVs searching for sightings of enemy units using various cameras (*e.g.,* optical, infrared, and ultraviolet). By compromising one or more UAVs via IMU spoofing in such a mission, an enemy can evade detection by the UAVs that are deployed to search and report enemy units.

Figure 1 shows an example of such a mission and a possible result of the attack. When the mission is executed in the absence of attacks, the UAVs detect the enemies, as shown in Figure 1-a. An ideal manipulation resulting from an attack will cause the UAV to take a similar route but entirely miss the enemies, as shown in Figure 1-b. A successful instance of IMU spoofing attack by the enemy can perturb the trajectory of the compromised UAV just enough for the enemy's ground units to evade detection, as shown in Figure 1-c.

A. Contributions

Our contributions through this paper are as follows:

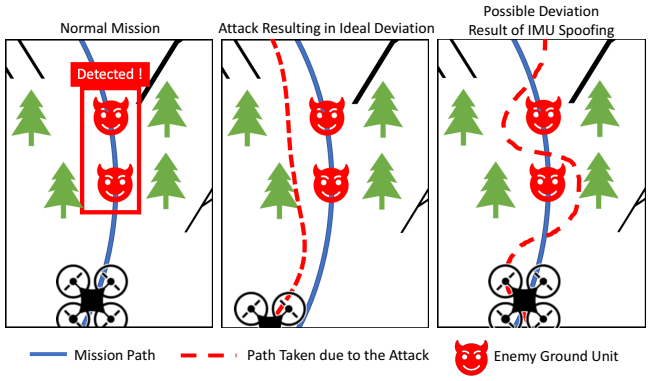

Fig. 1: **(Left)** UAV following the mission path in a surveillance mission and successfully detecting the enemy ground units. **(Middle)** The ideal UAV deviation, away from the mission path, resulting in enemy units evading the detection. **(Right)** The deviation path possible due to IMU spoofing but still resulting in enemy units evading the detection.

- We investigate the presence and extent of vulnerabilities of UAV control systems through adversarial deviations in IMU sensor readings.
- We propose a metric to measure the influence of IMU spoofing in standard UAV systems through formalizing the coupled dynamics of IMU spoofing attacks and UAV control systems.

II. System Models

In this section, we present the system model and how the IMU is utilized in a flight controller. Then we discuss the details of how Extended Kalman Filter (EKF) is utilized in a flight controller.

Control Architecture. Inertial measurement units (IMUs) measure the linear and angular acceleration which is used to calculate the orientation and the trajectory of the vehicle. The IMU conjunction with a set sensors (*e.g.,* GPS, Barometer, and Magnetometer) and using Extended Kalman Filter (EKF), they can provide position and velocity along with the attitude.

Figure 2 shows the general diagram on how the UAVs operate. Therefore, ❶ given a mission set point and current physical state of the vehicle, ❷ the controller calculates the actuation command required to keep the UAV stable and ensure that the UAV is heading towards the set point. ❸ The actuation command is sent to the actuators which actuates the motors/engine. ❹ The actuation movement is picked up by the IMU which is used to predict the position, velocity, and attitude of the vehicle. ❺ The prediction is compared against arriving sensor values and the updated estimated state is published. ❻ The controller receives the newly updated estimated state and the cycle repeats.

Extended Kalman Filter. The state of the vehicle refers to the physical state / kinematics of the UAV (*e.g.,* attitude, position, velocity). The UAV control system uses an anomaly detector to detect state deviations. In order to ensure the validity of the sensory information and the corresponding control actions, a state estimator is used to process the sensor data and the previous state to calculate the approximate state for the current condition. Extended Kalman Filter-Based state estimators are often used to combine the information from the sensors to ensure there are no anomalies (or attacks) directed at the UAV sensors [15].

The core principle of a Kalman Filter (KF) is to combine information from multiple sensors to estimate the state in a linear fashion. An Extended Kalman Filter (EKF) is a Kalman Filter variant to handle non-linear state estimation. An EKF is composed of two stages: *(a)* predict (Equation 1) and *(b)* update (Equation 2).

Predicted State — Prediction Function

$$\hat{x}_{t|t-1} = f(\hat{x}_{t-1}, u_{t-1}) \quad (1)$$

Previous State — IMU

Updated State — Sensor — Transformation Function

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t \cdot (y_t - h(\hat{x}_{t|t-1})) \quad (2)$$

Kalman gain — Innovation

During the prediction stage at time step t, estimated state of previous time-step, $\hat{x}_{t-1}$ and the actuation command of previous time-step u_{t-1} are used to predict the current state $\hat{x}_{t|t-1}$ using the prediction function $f(\cdot)$. The predicted state provides a context into what incoming sensor value should be. That is accomplished by converting the predicted state value to a sensor value using the transformation function $h(\cdot)$. For instance, if the predicted x-position is 1m, then the expected GPS value should correspond to 1m in x-position. The difference in the predicted and the observed sensor value is the *innovation*.

Kalman gain K_t is the set of weights that define how much the innovation should influence the state estimation. Therefore, sensor values with consistently higher innovation would have lower Kalman gain and vice versa.

If the difference between the estimated and the actual state is above a certain threshold, the anomaly detector raises an alarm to the system to indicate an attack or sensor anomaly and overrides the current control commands.

For instance, if the anomaly threshold was set to $\theta = 0.1$ and the predicted state x-position was 1.2m but the GPS shows that the x-position was 1m, the resulting innovation for x-position is $0.2 > \theta$ which is an anomaly, therefore an alarm is activated. EKF implementations have checks for this kind of anomalous behavior. *To evade this baseline behavior, the prediction due to the spoofed IMUs must stay close enough to the upcoming sensor value.*

III. Adversary Model.

We assume a *grey-box attack model* (*i.e.,* the attacker does not have the complete knowledge about the UAV). Specifically, the adversary has the following knowledge:

- Mission parameter of the victim UAV
- Set of sensors used by the EKF
- Timing of when the sensors arrive

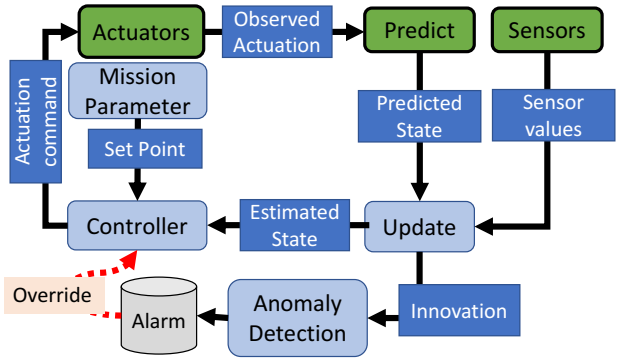

Fig. 2: Operation of UAV controls using EKF

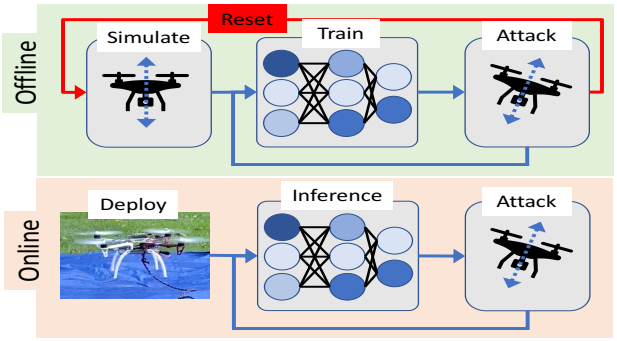

Fig. 3: Pipeline for training and deploying attack.

- Values of the sensors
- Partial EKF knowledge
  - Reported EKF estimation
  - Predicted state resulting from IMU
  - Innovation resulting from EKF

The knowledge about the EKF is partial because the adversary is only aware of the input and the output of the prediction and the state update. The adversary is not aware of any implementation specific latent variables. The adversary can:

- Manipulate the IMU readings by the sensor
- Calculate and inject the spoof values before new IMU is used by the EKF

Therefore, the equations (3) and (4) describe the state of the EKF during the attack

$$\hat{x}^c_{t|t-1} = f(\hat{x}_{t-1}, u_{t-1} + a_t) \quad (3)$$

(a_t: Injected Value; $\hat{x}^c_{t|t-1}$: Resulting Predicted State)

$$\hat{x}_t = \hat{x}^c_{t|t-1} + K_t \cdot (y_t - h(\hat{x}^c_{t|t-1})) \quad (4)$$

($y_t - h(\hat{x}^c_{t|t-1})$: Innovation Resulting From Attack)

We define IMU spoofing as injecting a value, a_t, into the IMU, u_{t-1}, before prediction occurs. Therefore when the IMU is spoofed, the resulting predicted state, $\hat{x}^c_{t|t-1}$, must ensure that the corresponding innovation $r^c_t = y_t - h(\hat{x}^c_{t|t-1})$ is less than θ. We consider an attack to be *successful* if the adversarial influenced predicted state is different from uninfluenced predicted state (*i.e.,* $|\hat{x}_{t|t-1} - \hat{x}^c_{t|t-1}| > 0$) while $r^c_t < \theta$. The result of the attack causes the control to make unnecessary adjustments.

IV. Framework

The construction of our detection system consists of two major stages as presented in Figure 3: *(i)* Offline training phase for the DNN to learn stealthy injections *(ii)* An online phase to run the pre-trained DNN model on a deployed UAV. In the following, we present the details of each of the stages.

Offline Phase. Offline phase primarily deals with simulation. To ensure that the simulation is reflective of the real-world its parameters must be adjusted (*i.e.,* set of sensors, sensor noise distribution and bias, sensor update frequency). After the adjustment, the compromised UAV is tasked to run a mission such as patrolling a designated area. During the mission, the simulated UAV sends the information needed to train the DNN model. Specifically, the information includes IMU values, predicted state and other sensors. Therefore, the model must maximize for the following:

$$\max_{a_t} || x_t - \tilde{x}_t ||$$

(x_t: True State; $\tilde{x}_t$: Mission Specified State)

$$\text{s.t. } y_t - h(\hat{x}^c_{t|t-1}) < \theta$$

$$x_t = A(C(\hat{x}^c_{t|t-1}, \tilde{x}_t), x_{t-1})$$

(A: Actuation; C: Control)

The model is trained to generate values that successfully spoof the IMU to cause the UAV to deviate from its set point. In the event the spoofing is unsuccessful (*i.e.,* caught by anomaly detector), the simulation is reset.

Online Phase. In this phase, the simulation starts but the DNN model only performs inference. The pre-trained attack model from the offline phase is used to generate attack values during inference time. The attack must not raise any alerts at any point during the mission; otherwise, the attack will be deemed a failure.

V. Initial Evaluation

To evaluate our approach, first we must consider the type of mission. In this paper, we consider three types of mission with the following movement: stationary, lateral movement, and vertical movement. Then we define the metric to measure the influence of the attack: state error, prediction difference, and innovation. The attack will be tested on a PX4 Software-In-The-Loop(SITL) [16] flight controller simulation conjunction with gazebo [17] physics simulator. Then we will examine how the attack can influence in each mission types.

A. Mission Parameters

We consider three types of mission with the following movement: stationary, lateral movement, and vertical movement. Vertical movement mission primarily use barometer to

check to see if the UAV is in the mission specified altitude and similarly lateral movement mission use GPS. Stationary mission would rely on both.

Stationary. This requires the UAV to hold position at 20m altitude while holding the same longitude and latitude as the starting position.

Lateral Movement. This is when the UAV moves from side-to-side. Specifically, the UAV must first lift off to 20m altitude then move 5m to the east and then 10m west.

Vertical Movement. This is when the UAV moves from side-to-side. Specifically, the UAV must first lift off to 20m altitude then move 5m to the east and then 10m west.

B. Metrics

We measure the spoofing effectiveness by observing the difference from the ground truth state vs the predicted state which we call *state error*.

$$|| \underbrace{x_t}_{\text{True State}} - \underbrace{\tilde{x}_{t|t-1}}_{\text{Predicted State}} || \tag{5}$$

We also need to measure the how much the state prediction changed as a result of the spoofing.

$$|| \underbrace{\hat{x}^c_{t|t-1}}_{\text{Predicted State due to IMU Spoofing}} - \underbrace{\hat{x}_{t|t-1}}_{\text{Predicted State}} || \tag{6}$$

The innovation due to spoofing, shown in equation 4, must also be recorded and measured. For the attack to be stealthy, the innovation resulting from the attack must also remain below the threshold δ. We use these metrics to measure the quality of the model. Therefore, we compared the state error in nominal operation vs adversarial influenced operation as well as the innovation. We should see some noticeable difference between the state error while keeping the innovation (*i.e.,* difference between the predicted and the observed value) below the threshold.

VI. Insights

To study the effect of different network hyperparameters for training the attack models, we considered two training configurations for a DNN with 50 fully connected layers: *(i)* ReLU activation [18] and *(ii)* ELU activation [19]. ReLU configuration is for preventing the exploding gradient problem, which can result in unrealistically high attack value generations [20]. However, it maps to a limited range of values, decreasing model capability. On the contrary, ELU generates a smoother activation and can map to a wider range of values but is more prone to suffer from the exploding gradient problem. As expected during the offline phase, the ReLU-activated model trained better than the ELU configuration due to its mapping simplicity. However, during the online phase, some models from both configurations were numerically unstable (*i.e.,* the model outputs a very large number). This is likely due to the exploding gradient problem; therefore, we need to make additional changes to the model's architecture to avoid this issue [20].

For the IMU spoofing to be stealthy, the resulting spoof must be within θ of the sensor values. The expected result is that the UAV will oscillate around the mission-specified set point. Therefore, successful stealthy IMU spoofing is bound by the timing interval between each arrival of sensor values.

The bound implies the UAV stays in the deviated path for a longer distance if the velocity is higher. Therefore in the previously described surveillance evasion scenario, it is possible for the UAV to achieve the ideal deviation path if the UAV velocity is high enough.

The impact of the IMU spoof is greater when the rate of the external positioning system values are lower. For instance, if the GPS signal is infrequent enough (especially if the UAV has to pass through GPS denied environment), the attacker may have enough time to deviate the UAV and return back to the mission path. Therefore, when the UAV is in a dead-reckoning state, IMU-spoofing will have full control over the vehicle's movement.

Even if the IMU spoofing does not result in large path deviations, if the UAV has to aim for a particular target or direction, even a slight change in the attitude can be sufficient to compromise the mission. IMU is a crucial sensor in the UAV, and more work is needed to secure the UAV against such attacks.

A. Limitations.

Spoofing only the IMU cannot cause the UAV to permanently deviate away from the mission path because by the time other sensors arrive, the EKF adjusts its state based on the sensors and the control corrects based on the adjusted state.

There are also practical limitations. The attacker needs to know exact timing of the attack as well as ensure that the inference can run at 250Hz which means the inference must run quicker than 4ms. There are two places to run the inference: on the drone or on a remote machine. If the model were to run on the drone, the model need to significantly minimize its computational footprint to avoid affecting other functionalities of the UAV. Currently, a single-board computer such as raspberry-pi cannot handle 250 inferences a second. If the model were to run on a remote machine, the model must account for the latency where if the latency is large enough, the received data from the UAV is outdated. Therefore, the model needs to predict incoming sensors values to generate the spoof values.

VII. Related Work

In this section, we highlight the work related to cyber-physical systems (CPS) safety and anomaly detection mechanisms and recent work on the adversarial attacks to evade these mechanisms, particularly on UAVs.

Spoofing the sensors of safety-critical systems to explore security risks recently attracted much interest. For instance, stealthy manipulation attacks against road navigation system

controllers to trigger fake navigation turns and deviate the system were proposed against intelligent vehicles [21]. Mendes *et. al.* [22] explored the possible effects of sensor spoofing attacks with different attack models. However, the attacks shown in this work are not strategically executed to tamper with the UAVs for the specific purpose of impacting mission objectives such as connectivity. Similar work [13] showed a take-over attack on UAVs by spoofing optical flow sensors. By spoofing optical flow sensor inputs to manipulate the perceived environment of the victim, they assume implicit control over the mission route. However, their attack is directed against optical flow sensors and does not tamper with industry-grade controllers and corresponding anomaly detectors for UAVs. Khazrei *et. al.* [23] utilizes deep-learning-based models to perform vulnerability analysis in various cyber-physical systems, including UAVs. Our work is different as we aim to find semantic deviation traces that would impact a specific mission objective and parameter, such as connectivity and surveillance visibility.

Recently, it has been demonstrated that taking over UAV command and control capabilities is possible through spoofing UAV sensors. Kerns *et. al.* [24] showed that taking over UAVs is possible through strategic GPS spoofings. However, their work focuses primarily on GPS sensor spoofings and does not address mission-specific deviation capabilities. Similarly, Gaspar *et. al.* [25] proposed a Software-Defined Radio with GPS spoofing capabilities to deviate UAVs from mission traces and assert control over unauthorized UAVs. In contrast, we demonstrate the strategic spoofing capabilities to deviate mission traces through IMU, an inertial sensors module. Moreover, we consider an adversary that performs stealthy strategic spoofs to deviate UAVs from realistic mission task element traces.

REFERENCES

[1] P. Skobelev, D. Budaev, N. Gusev, and G. Voschuk, "Designing multi-agent swarm of uav for precise agriculture," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2018, pp. 47–59.

[2] J. Grzybowski, K. Latos, and R. Czyba, "Low-cost autonomous uav-based solutions to package delivery logistics," in *Advanced, Contemporary Control*. Springer, 2020, pp. 500–507.

[3] Z. Xiaoning, "Analysis of military application of uav swarm technology," in *2020 3rd International Conference on Unmanned Systems (ICUS)*, 2020, pp. 1200–1204.

[4] A. Utsav, A. Abhishek, P. Suraj, and R. K. Badhai, "An iot based uav network for military applications," in *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2021, pp. 122–125.

[5] S. Wu, B. He, and X. Liu, "Combined model of principal component analysis and general regression neural network for military aircraft cost estimation," in *2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI)*, 2022, pp. 301–306.

[6] J. Ismay, "The iranian drones in ukraine's already crowded skies," *NYT*. [Online]. Available: https://www.nytimes.com/2022/10/19/us/politics/ukraine-drones-iran-russia.html

[7] "How are 'kamikaze' drones being used by russia and ukraine?" Oct 2022. [Online]. Available: https://www.bbc.com/news/world-62225830

[8] W. Davis, "What are the suicide drones bombarding ukraine, and where did russia get them?" *NPR*. [Online]. Available: https://www.npr.org/2022/10/18/1129576360/suicide-drones-ukraine-russia-iran-shahed-kamikaze

[9] Y. Son, H. Shin, D. Kim, Y. Park, J. Noh, K. Choi, J. Choi, and Y. Kim, "Rocking drones with intentional sound noise on gyroscopic sensors," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 881–896.

[10] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks," in *2017 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2017, pp. 3–18.

[11] J. E. Martin, V. Saul, D. Novick, and D. Allen, "Assessing the vulnerability of unmanned aircraft systems to directed acoustic energy," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2020.

[12] J. Lee, M. Kim, J. Lee, and S. Pullen, "Integrity assurance of kalman-filter based gnss/imu integrated systems against imu faults for uav applications," in *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, 2018, pp. 2484–2500.

[13] D. Davidson, H. Wu, R. Jellinek, V. Singh, and T. Ristenpart, "Controlling {UAVs} with sensor input spoofing attacks," in *10th USENIX workshop on offensive technologies (WOOT 16)*, 2016.

[14] S. G. Manyam, S. Rasmussen, D. W. Casbeer, K. Kalyanam, and S. Manickam, "Multi-uav routing for persistent intelligence surveillance & reconnaissance missions," in *2017 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2017, pp. 573–580.

[15] H. Wang, A. Meng, Y. Liu, X. Fu, and G. Cao, "Unscented kalman filter based interval state estimation of cyber physical energy system for detection of dynamic attack," *Energy*, vol. 188, p. 116036, 2019.

[16] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6235–6240.

[17] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[18] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[19] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[20] B. Hanin, "Which neural net architectures give rise to exploding and vanishing gradients?" *Advances in neural information processing systems*, vol. 31, 2018.

[21] K. C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, and Y. Yang, "All your {GPS} are belong to us: Towards stealthy manipulation of road navigation systems," in *27th USENIX security symposium (USENIX security 18)*, 2018, pp. 1527–1544.

[22] D. Mendes, N. Ivaki, and H. Madeira, "Effects of gps spoofing on unmanned aerial vehicles," in *2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE, 2018, pp. 155–160.

[23] A. Khazraei, S. Hallyburton, Q. Gao, Y. Wang, and M. Pajic, "Learning-based vulnerability analysis of cyber-physical systems," in *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2022, pp. 259–269.

[24] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[25] J. Gaspar, R. Ferreira, P. Sebastião, and N. Souto, "Capture of uavs through gps spoofing," in *2018 Global Wireless Summit (GWS)*. IEEE, 2018, pp. 21–26.