ConFIDe: A PWM-Driven Control-Fused Intrusion Detection System for Hardware Security in Unmanned Aerial Vehicles

Muneeba Asif
masif004@fiu.edu
ACyD Lab, Florida International University
Miami, Florida, USA

Ahmad Mohammad*
asm6t@mtmail.mtsu.edu
Middle Tennessee State University
Tennessee, USA

Mohammad Ashiqur Rahman
marahman@fiu.edu
ACyD Lab, Florida International University
Miami, Florida, USA

Kemal Akkaya
kakkaya@fiu.edu
ADWISE Lab, Florida International University
Miami, Florida, USA

ABSTRACT

With the rise in the application of unmanned aerial vehicles (UAVs), security concerns associated with them have become paramount. Similar to other cyber-physical systems, the primary working principle behind UAVs follows the sensor-controller-actuation cycle. Errors between the setpoints and sensor data are computed through a PID controller and translated to pulse width modulated (PWM) signals that control the orientation and movement of a UAV. Recent research has demonstrated intentional electromagnetic interference (IEMI)-based alteration of PWM signals causing unauthorized maneuvers and crashes in UAVs. PWM alteration attacks can be carried out in various ways. For instance, hardware Trojans (HTs) can manipulate the PWM signals, and given the untrusted supply chain, HTs are a critical threat. Adversaries can exploit the PWM signals to manipulate UAV operations subtly, bypassing traditional intrusion detection systems (IDSs) that only monitor sensor data. Therefore, ensuring the integrity of PWM signals and their correlation with sensor and controller data is crucial for end-to-end UAV security. We address this need by proposing ConFIDe (Control-Fused Intrusion Detection system), a novel defense technique for UAVs. It verifies the integrity of the flight controller-generated PWM signals, ensuring the motors receive the signals free from hidden exploits. We validated our proposed IDS on different PWM alteration attack scenarios. In particular, we implemented a hardware Trojan attack targeting the PWM signals on a PX4-UAV to test the efficacy of the proposed IDS on a real system. ConFIDe performed well on all the attack scenarios, achieving a high ROC-AUC, including sensor attacks like GPS spoofing.

CCS CONCEPTS

• **Security and privacy** → **Malicious design modifications**; **Embedded systems security**; • **Computer systems organization** → *Embedded systems.*

ASIA CCS '24, July 1–5, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0482-6/24/07…$15.00
https://doi.org/10.1145/3634737.3657014

KEYWORDS

Unmanned aerial vehicles; embedded systems; hardware attacks; PWM alteration attacks; hardware Trojans; intrusion detection systems; cyber-physical systems

ACM Reference Format:
Muneeba Asif, Ahmad Mohammad[1], Mohammad Ashiqur Rahman, and Kemal Akkaya. 2024. ConFIDe: A PWM-Driven Control-Fused Intrusion Detection System for Hardware Security in Unmanned Aerial Vehicles. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '24), July 1–5, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3634737.3657014

1 INTRODUCTION

The unmanned aerial vehicle (UAV) industry has experienced exponential growth, and by 2031, the global UAV market is projected to rise to $97.65 billion [1, 2]. UAVs are integral in varied applications such as surveillance for smart grids [3], crowd analysis [4], disaster monitoring [5], urban monitoring [6], agriculture [7], remote sensing [8], logistics [9], and defense sectors [10, 11]. To proficiently handle these tasks, UAVs must ensure end-to-end security and resiliency without sacrificing safety. As shown in Fig. 1, UAV's working principle follows the sensor-control-actuation technique. Initially, UAVs receive navigation targets or setpoints via inputs from radio controls or ground control stations. The onboard sensors then gather real-time data, which is essential for the proportional-integral-derivative (PID) controller. This controller evaluates the position and attitude errors between the actual sensor readings and the predefined setpoints. It then computes a control value to rectify these errors. Subsequently, the flight controller interprets this control value to generate pulse width modulated (PWM) or signals (throughout the paper, we use the term PWM and actuation signals interchangeably). These signals are essential in regulating the power supply to the UAV's motors, thereby controlling its orientation, direction, and speed. This working mechanism exposes UAVs to various cyber-physical attack vectors.

Intrusion detection is a hot research topic in the UAV security domain, and existing works have devised various AI/Machine Learning techniques to detect anomalous behaviors in UAVs [12], [13], and [14]. These works have proposed efficient security mechanisms for detecting and mitigating the impacts of sensor spoofing, sensor

*Ahmad Mohammad Interned at the Analytics for Cyber Defense (ACyD) Lab, Florida International University, during the summer of 2022 under the REU Program.

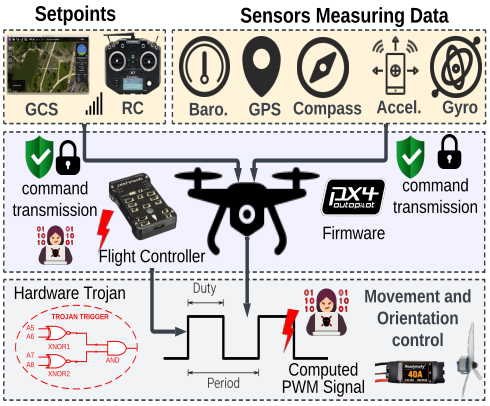


Figure 1: Sensor-control-actuation working principle in UAVs and potential attack vectors. Red shows vulnerabilities like PWM manipulations and hardware attacks, while green shows existing security solutions for sensor command data in UAVs.

jamming, and network intrusion attacks. However, most of these techniques predominantly focus on threats such as GPS spoofing, signal jamming, and network intrusions, which exploit the UAV's sensor-command domain, encompassing sensor data and mission commands transmitted via the MAVlink protocol. Attacks targeting the PWM signals (remotely or through hardware manipulation) are often overlooked. a state-of-the-art framework is the PID-Piper framework, created by Dash et al., which aims to safeguard UAVs from GPS spoofing and transduction attacks [15]. While it considers the sensor and controller behavior, it does not fully address the vulnerabilities that occur after the control signals are physically translated into PWM outputs, as PWM generation takes place after the PID control. This aspect is critical since alterations in PWM signals can covertly affect UAV operation, evading detection by systems focused solely on sensors and controllers. Additionally, Dayanikli et al. investigated how intentional electromagnetic interference (IEMI) can modify PWM signals, resulting in unauthorized movements and possible crashes in UAVs [16].

Moreover, given the complex and globally distributed supply chain of UAV components, manufacturing the flight controllers often involves multiple outsourced entities. This raises the potential for hardware Trojan insertion, which can be selectively triggered to alter the PWM signals as per their payload [17]. As the PWM signals govern the overall movement and orientation in UAVs, their adversarial alteration can result in significant deviations from intended UAV behavior, from subtle changes in the desired trajectory to complete operational failures. However, despite the significant risks associated with PWM signal manipulation, the security of these PWM signals has not been extensively studied. Modern hardware Trojans are no longer simple, always-active threats that can be easily identified during routine test flights or through basic operational checks. Instead, these Trojans can be intricately designed to remain dormant, undetectable through conventional means, and activated only under highly specific conditions that may not be replicated in standard testing environments. Trojans can be designed to remain dormant until triggered by specific conditions, such as altitude, geographic location, or even specific payloads, making their existence and activation far from straightforward. This sophistication facilitates the Trojans to remain undetected during routine test flights or inspections [18, 19]. For instance, activation could depend on a particular sequence of commands, specific geographic locations, altitudes, or even the UAV's interaction with certain wireless signals. This specificity ensures that the Trojan remains dormant during the test flight phase, thus bypassing detection mechanisms that do not replicate these unique conditions.

To overcome this research gap, we propose novel **Con**trol-**F**used **I**ntrusion **De**tection system (ConFIDe), which is specifically designed to secure the end-to-end process of UAV flight control. ConFIDe employs deep learning techniques to analyze the relationship between the sensing, controlling, and the subsequent generation of PWM signals. It effectively detects a wide spectrum of attacks ranging from GPS spoofing and jamming to intrusions in sensor and controller systems and, crucially, in the PWM signals. By employing a holistic approach that utilizes a deep learning-based auto-encoder for one-class classification, it verifies the PWM signal's integrity before sending it to the motors, ensuring the UAVs are secured against any tampering. The proposed approach ensures that any discrepancy originating from either FDIs or direct hardware interventions is detected based on the resultant PWM signals. We validated our IDS through experiments on an S500-Pixhawk 2.4.8 quad-copter UAV using a comprehensive dataset from multiple flights on different trajectories. We utilized synthetic attack samples to evaluate the performance of the approach. We also implemented a hardware Trojan attack emulated on a Pixhawk UAV to selectively alter the PWM signal in real time. We have made the datasets available at [20] to encourage further research. To summarize, we make the following contributions:

- We develop ConFIDe, a novel intrusion detection system for UAVs to ensure the integrity of PWM/actuation signals, which are crucial for UAV's orientation and movement. ConFIDe detects a broad spectrum of sensor, controller, actuation, or hardware attacks in UAVs.
- Using an S500-Pixhawk 2.4.8 quad-copter, we emulate hardware Trojan attacks targetting the alternating of PWM signal and evaluate ConFIDe's performance. The results show up to 92.5% ROC-AUC on synthetic data and 100% on real data.

The remainder of the paper is organized as follows. Section 2 reviews the related work. Section 3 explains the UAV workflow, sheds light on the PWM control in UAVs, and establishes our motivation for Control-Fused Intrusion Detection. We discuss the threat model, the attacker's intent, and our synthetic attack data generation in Section 4. Section 6 covers the details of the formation of our IDS, and Section 7 shows the implementation. We discuss the evaluation results and potential countermeasures in Section 8 and conclude the paper in Section 9.

2 RELATED WORK

Existing literature has extensively investigated IDS-based defense strategies for UAVs, adopting various techniques, including game theory, blockchain, spectral traffic analysis, and AI/Machine Learning, alongside behavior rule definitions, artificial immune systems (AIS), and hybrid approaches. We categorize these studies according to the technologies they utilize for the defense.

Table 1: Related Work

Work	Defense Technique	Layer Targeted	Attacks Considered	Control Aware?
[21]	Game Theory	Sensor	DoS, Sybil, false alarm	×
[22]		Sensor	GPS spoofing	×
[23]		Network	DoS, Sybil, false alarm	×
[24]	Blockchain	Network	DoS, Remote to User, User to Remote, probing	×
[25]	Spectral Traffic Analysis	Network	DDoS	×
[26]	AI/ Machine Learning/ Deep Learning	Network	SYN-flood, de-authentication	×
[27]		Network	Eavesdropping, sniffing, buffer overflow	×
[28]		Network	DoS, Remote to User, User to Remote, probing	×
[29]		Hardware	GPS Spoofing, GPS Jamming	×
[15]		Hardware	GPS Spoofing,Transduction	✓
[30]	Behavior Rules	Network	Resource depletion, capturing, data corruption	×
[31]		Network	Physical Invariants	×
[32]		Network	Misbehavior	×
[33]	Artificial Immune System (AIS)	Network	Blackhole, sybil, flooding	×
[34]	Hybrid	Network	Replay, MiTM, impersonation	×
ConFIDe	Machine Learning	Hardware	Hardware Trojan, PWM manipulation	✓

Sun et al. apply Bayesian Game Theory-based IDS to spot malicious nodes in UAV networks, using hierarchical monitoring and optimal node counts for efficiency [21]. Eldosouky et al. introduce a cooperative localization countermeasure against GPS spoofing on UAVs, framing it as a Stackelberg game to optimize defense [22]. Meanwhile, Sedjelmaci et al. present a security game framework (SGF) based on Bayesian game theory, safeguarding mobile ground nodes with UAVs, addressing two attack-defense scenarios [23]. This involves suspicious units, leading to two attack-defense scenarios, employing unique and effective strategies to safeguard a robust UAV network against potential threats.

Khan et al. use blockchain and federated learning for IDS in multi-UAV networks, leveraging multiple ML algorithms for decentralized analytics [24]. Condomines et al. enhance drone fleet IDS by combining a linear controller/observer with traffic spectral analysis. Their wavelet approach detects intrusions, and the linear system gauges attack intensity [25].

Basan et al. introduce a neural network to spot denial-of-service attacks on UAVs by observing entropy changes in traffic patterns [26]. Al-Haija et al. apply deep convolutional neural networks, dubbed UAV-IDS-ConvNet, to detect threats in encrypted Wi-Fi traffic of notable UAVs such as Parrot Bebop and DJI Spark [27]. Meanwhile, Praveena et al. use a deep reinforcement learning method with black widow optimization (DRL-BWO) to bolster UAV network security, leveraging an enhanced deep belief network (DBN) IDS [28].

Furthermore, Whelan et al. present MAVIDS, employing novelty-based one-class classification to tackle the scarcity of labeled data for UAV IDS. Validated against GPS spoofing and jamming, it also enacts mitigation strategies [29]. Also, Mitchell and Chen utilize a behavior rule-based UAV-IDS (BRUIDS), derived from threat models, to guard against cyber-attacks, aiming for a balance between security and performance [30]. Kwon et al. introduce a real-time threat evaluation technique grounded in reachability assessment [32].

Fotohi et al. introduce SID-UAV, which employs a self-matching method in the MAPE-K loop to find secure UAV paths, utilizing

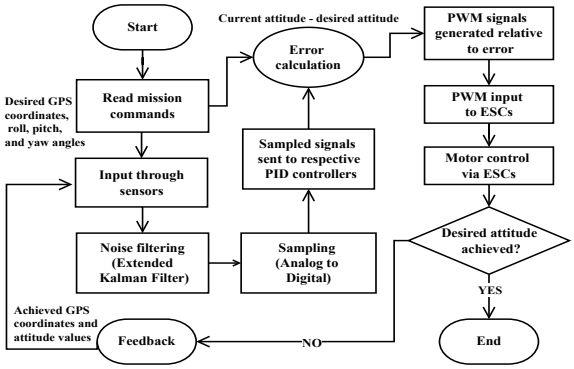


Figure 2: UAV's typical workflow: Flight controller receives input commands, processes them through its firmware, and produces PWM values to drive the motor, directing motion.

agents for analysis and defense against adversarial UAVs [33]. Kumar et al. suggest a blockchain and deep learning-based data-sharing system for UAVs, incorporating a Proof-of-Authentication consensus mechanism and a neural network flow analyzer to detect fraud, enhancing intrusion detection with SCSAE-ALSTM [34]. Quinonez et al.'s SAVIOR employs machine learning to leverage physical invariants in autonomous vehicle systems, preventing sensor and control system attacks [31]. Meanwhile, Dash et al.'s PID-Piper uses a feed-forward controller-based parallel PID control to recover robotic vehicles from sensor-based attacks, filtering attack-induced sensor and control disruptions [15] As shown in Fig. 2 and summarized in Table. 1, operate either only at the sensor /network level or up to the PID control level, but the physical translation of control signals to PWM signals takes place after it. None of the aforementioned techniques take the PWM signals into consideration, even though these are the primary signals governing the UAV's movement and orientation. To the best of our knowledge, this is the first work that addresses the critical security concerns related to the PWM actuation signals in UAVs. By integrating advanced intrusion detection mechanisms specifically targeting PWM signal manipulation, ConFIDe extends beyond the sensor/network level or PID control level security measures, directly safeguarding the physical translation of control signals to PWM outputs and detecting sensor, controller, actuator, or hardware attacks.

3 BACKGROUND

This section overviews the UAV's operation PWM control for BLDC motors and outlines the research motivation.

3.1 Working Principle in UAVs

A UAV's control and monitoring tasks are complex due to the nonlinear aerodynamics of the embedded system [35]. Four key terms clarify a UAV's operation: (1) *Roll*: the UAV's longitudinal rotation, moving left or right. (2) *Pitch*: lateral rotation tilting the UAV forward or backward. (3) *Yaw*: rotation about the vertical axis, pivoting the UAV clockwise or counterclockwise. (4) *Throttle*: controls the UAV's vertical motion, dictating its speed, illustrated in Fig. 3. UAVs, controlled remotely or autonomously, use inertial measurement units (IMUs) for sensor data refined by noise and Kalman filters. After ADC sampling, this data informs PID controllers to determine

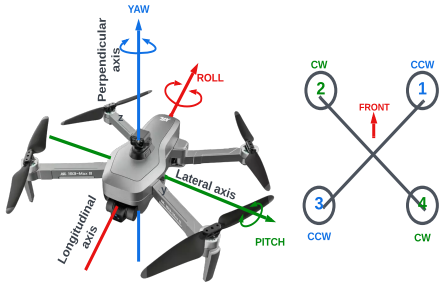


Figure 3: Illustration of a UAV's roll, pitch, and yaw axes for orientation and movement control and clockwise (CW) and counterclockwise (CCW) motors in a quadcopter UAV.

roll, pitch, and yaw error signals. These signals then guide PWM outputs for the motors, as shown in Fig. 2. Then, firmware acts as the intermediary software layer, guiding the UAV's operations by translating high-level commands into actionable hardware responses. By understanding the PWM output generated by this control process (being "control-fused"), an IDS can effectively identify a broad spectrum of attacks anywhere in the sensor-control-actuation process.

3.2 PWM Control and UAV Dynamics

The dynamics of UAV movement are fundamentally governed by the motors' response to PWM signals, which control their speed and, consequently, the thrust generated by each rotor. The angular velocity of motor i, denoted as ω_i, is a function of the PWM signal provided to it, as shown in the following Equation 1:

$$\omega_i = f(\text{PWM}_i) \tag{1}$$

where ω_i is the angular velocity of motor i, and PWM_i is the PWM signal to motor i. The function f maps PWM signals to motor speed. The thrust generated by each motor, F_i, is proportional to the square of its angular velocity, which is represented by the Equation 2:

$$F_i = k \cdot \omega_i^2 \tag{2}$$

where F_i is the thrust produced by motor i, and k is the thrust coefficient. Roll motion is controlled by creating a differential in the speed of motors on either side of the UAV's longitudinal axis, as shown in Equation 3.

$$\Delta\omega_{\text{roll}} = g_{\text{roll}}(\text{PWM}_2 + \text{PWM}_4 - \text{PWM}_1 - \text{PWM}_3) \tag{3}$$

where $\Delta\omega_{\text{roll}}$ is the change in roll motion, and g_{roll} is a gain factor for roll. Pitch motion is similarly controlled through a differential in the speed of front and back motors, as described by Equation 4:

$$\Delta\omega_{\text{pitch}} = g_{\text{pitch}}(\text{PWM}_1 + \text{PWM}_2 - \text{PWM}_3 - \text{PWM}_4) \tag{4}$$

where $\Delta\omega_{\text{pitch}}$ is the change in pitch motion, and g_{pitch} is a gain factor for pitch. Yaw motion is achieved by varying the speed of motors spinning in opposite directions, which is mathematically formulated as in Equation 5:

$$\Delta\omega_{\text{yaw}} = g_{\text{yaw}}\left(\sum_{i\in\text{CW}} \omega_i - \sum_{i\in\text{CCW}} \omega_i\right) \tag{5}$$

where $\Delta\omega_{\text{yaw}}$ represents the change in yaw motion, and g_{yaw} is a gain factor for yaw. CW and CCW denote the sets of motors spinning clockwise and counterclockwise, respectively.

$$T_{\text{total}} = \sum_{i=1}^{4} F_i \tag{6}$$

Finally, the overall thrust, which controls the UAV's altitude, is the sum of the thrusts from all four motors as in Equation 6 where T_{total} is the total thrust for altitude control. For further understanding of PWM-governed movement control in UAVs, refer to Appendix A. These equations collectively illustrate how PWM signals are critical to achieving precise control over the UAV's orientation and altitude, enabling it to perform complex maneuvers and maintain stable flight. The PWM signal consists of alternating high (T_{ON}) and low (T_{OFF}) pulses. The motor speed depends on the pulse duration: a longer pulse indicates greater voltage and faster rotation. Typically, a motor expects a pulse between 1ms and 2ms in a 400Hz waveform. For UAVs, especially those with Pixhawk flight controllers, the pulse duration typically ranges from 1.1ms to 1.9ms. When T_{ON} is high 100% of the time, full bus voltage drives the motor. At 50%, half the bus voltage is applied and none during T_{OFF}.

Pixhawk Control Pipeline: In a Pixhawk UAV, the PWM value, ranging from 1100µs-1900µs, is derived from sensor input, the desired UAV attitude, and control commands. Pixhawk employs a control pipeline wherein control groups (inputs) map to output groups (PWM outputs) via a mixer. This mixer translates force commands, like turning left, into actuator commands influencing roll, pitch, or yaw. For instance, a control group might indicate a desired vehicle attitude, scaled from -1 to +1. The mixer then maps this to a PWM output channel, such as 1500µs [36].

3.3 PWM Security and Research Motivation

As intricate cyber-physical systems, UAVs often undertake mission-critical tasks where precision and adherence to a specific trajectory are paramount. While tampering with the hardware supply chain demands profound knowledge, resources, and access, some alarming instances of such breaches have been reported. Notably, the 2018 compromise of Supermicro servers used by tech giants resulted from illicitly embedded chips during manufacturing, facilitating unauthorized data access [37]. Similarly, concealed hardware Trojans have shown potential for remote deactivation of sophisticated defense systems and insidious privilege escalation attacks, as evidenced by Yang et al.[38]. Intel's Management Engine (ME), revealed in 2018, epitomizes another latent vulnerability, granting unmitigated control over computers and undermining user security[39]. This requires the UAVs to be secured from end-to-end and be resilient to attacks in their network, firmware, and hardware. While existing IDSs effectively counter input-space threats, such as GPS spoofing and DDoS attacks, they predominantly rely on sensor data. This leaves them oblivious to a UAV's flight control. For instance, a covert hardware Trojan embedded within a flight controller could manipulate the PWM values, altering motor speeds and disrupting the UAV's intended trajectory. Moreover, traditional IDS solutions adeptly address sensor-based threats like GPS spoofing and network-based DDoS attacks but fall short in detecting PWM-manipulative attacks. The PWM signals can be manipulated in the following ways to disorient UAVs:

- *Jamming*: Adversarial interference within the PWM communication frequency causing UAV control loss.
- *Replay Attacks*: Replaying previously captured legitimate PWM signals to disorient UAVs.
- *Malware or Firmware Attacks* Altering PWM signals via firmware infiltration.
- *Electromagnetic Interference (EMI)*: Stealthy modification of PWM signals via induction.
- *Physical Tampering*: A direct threat to PWM integrity through hardware Trojan insertion.

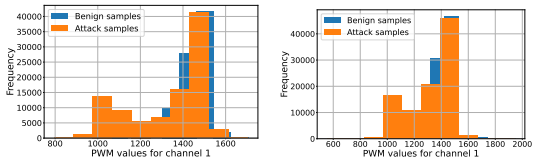
A PWM-exploitative attack can have ranging impacts. The attack's efficacy is demonstrated by directly manipulating PWM values, introducing a sizeable increment, and drastically altering the expected motor outputs. This calculated injection of erroneous control signals can lead to pronounced alterations in roll, pitch, or yaw movements, compelling the UAV to subtly deviate from its prescribed trajectory at first, then more noticeably over time. Unless finely tuned to detect such anomalies, the UAV's control system may not immediately recognize the malicious intent, attributing the deviations to environmental factors or sensor errors, thus allowing the adversary to achieve their objective of redirecting the UAV without raising immediate alarms. It's important to note that while traditional intrusion detection systems (IDSs) are designed to correct positional and attitude errors, they typically operate before the generation of PWM signals. Therefore, if an attack targets these positional errors, any corrective actions taken by these IDSs can be effectively negated. This happens because the manipulated PWM signals generated after the IDS intervention continue to direct the UAV erroneously, undermining the IDS's corrective measures and leading to severe consequences. Hence, an IDS integrating advanced intrusion detection mechanisms specifically targeting PWM signal manipulation is essential to address this.

4 THREAT MODEL

This section briefly discusses the threat model considered in this research. We list the assumptions, explain the adversary's knowledge, analyze the attack goal, summarise our attack techniques, and explain the synthetic attack sample generation to validate the IDS's detection performance.

4.1 Assumptions

- The PWM attacks faced by UAVs manifest as stealthily installed hardware Trojan that manipulates the PWM output values after the flight controller generates them.
- The hardware Trojan insertion is an insider attack, meaning the attacker in the untrusted supply chain has physical access to the target components of UAVs.
- The attacker has the knowledge of the PWM control of motors in a UAV (detailed in Section. 3.2) and can exploit vulnerabilities accordingly.
- The manipulated PWM values cannot be out of band (1100µs - 1900µs). (Detailed in Appendix B).
- ConFIDe is integrated into the UAV ecosystem at the user (trusted) side after the conventional supply chain. Hence, it is free from supply chain attacks. To ensure this, ConFIDe will be installed in a trusted computing base (TCB) [40].



(a) (b)

Figure 4: Benign and malicious samples for a PWM-manipulative hardware Trojan attack in (a) channel 4 and (b) 1.

4.2 Knowledge of the Adversary

In our study, we assume an adversary knowledgeable of the relationship between the PWM signals generated by the flight controller and the servo motor functions in a UAV. Whether guided remotely or autonomously, a UAV's flight is executed by its PWM signals, providing the adversary a great motivation to exploit.

4.3 Attack Goal

The primary attack goal considered in this research is the stealthy alteration of the PWM signals to lead the UAV to stray from its intended behavior. To alter the signal, the attacker can use the techniques mentioned in Section 3.3. However, as a test case for our research, we implement a hardware Trojan to alter the PWM selectively. We also simulate the impacts of other hardware Trojans-based PWM alterations via software modifications due to equipment restrictions. These techniques are described in the next subsection.

4.4 Attack Technique

This subsection examines hardware Trojan and firmware modification attacks, highlighting their impact on UAV security and emphasizing the need for ConFIDe.

4.4.1 Hardware Trojan (HT). A hardware Trojan is an intentional modification within an integrated circuit (IC) that consists of a trigger and an associated action known as a payload. The trigger is activated when specific conditions are met, leading the payload to execute its malicious operation. These modifications can bypass security mechanisms, impairing or completely disabling parts of the IC. Despite advancements in semiconductor technology, ICs remain susceptible to HTs placed by adversaries. By subtly altering the PWM signal timings, even a minuscule Trojan can have significant repercussions. Our research demonstrates a PWM-focused HT attack in two ways: first, through synthetic generation, and second, via practical implementation on an actual UAV. We assume that the Trojan is active 25% of the time. As a payload, we introduce a selective signal inversion, altering the PWM signal's duty cycle and affecting motor speed and, subsequently, UAV movement.

$$\mathcal{S}_{\text{orig}}(t) = \begin{cases} 1 & 0 \leq (t \bmod T_{\text{total}}) < D \cdot T_{\text{total}} \\ 0 & D \cdot T_{\text{total}} \leq (t \bmod T_{\text{total}}) < T_{\text{total}} \end{cases} \tag{7}$$

The original PWM signal, $\mathcal{S}_{\text{orig}}(t)$, defined in Equation 7, is defined by its duty cycle D, which represents the fraction of T_{total} during which the signal is active. Specifically, for time t, the signal is "on" between 0 and $D \times T_{\text{total}}$. The Trojan-infected PWM signal has a duty cycle, D_{att}, altered by ΔD from the original one (Equation 8). The infected PWM signal is expressed by Equation 9.

$$D_{\text{att}} = D + \Delta D \tag{8}$$

Algorithm 1: Synthetic Attack Sample Generation

```
Input: BenignData, Trigger
Result: BenignData_inverted
X ←Trigger
for every X/100 sample in BenignData do
    PWM_Inverted = abs(2500μs − sample.PWM)
    sample.PWM ← PWM_Inverted end
    for sample in BenignData do
        if sample.PWM < 1100μs then
            sample.PWM = 1100μs
        end
        if sample.PWM > 1900μs then
            sample.PWM = 1900μs
        end
    end
    return BenignData;
```

$$\mathcal{S}_{\text{att}}(t) = \begin{cases} 1 & 0 \le (t \bmod T_{\text{total}}) < D_{\text{att}} \cdot T_{\text{total}} \\ 0 & D_{\text{att}} \cdot T_{\text{total}} \le (t \bmod T_{\text{total}}) < T_{\text{total}} \end{cases} \tag{9}$$

4.4.2 Simulation Through Firmware Modification. Other hardware Trojan-based PWM attacks can be simulated by software/firmware modifications. For example, an attack to invert a motor's PWM outputs employs a trigger-payload logic: motor outputs are inverted only when a specific condition is met. Algorithm 2 provides a concise outline. It inputs current motor outputs and their count, then iterates through each motor output and examines for reversibility. If the trigger condition is satisfied, it adjusts the value accordingly, storing the result in the *inverted_outputs* array.

4.5 Attack Data Generation

Our study also simulated internal attacks on UAVs, like hardware Trojans, by creating synthetic PWM anomalies within the operational range of 1100μs to 1900μs for Pixhawk-based UAVs. This approach maintained the data's integrity and tested our IDS's ability to detect subtle control disruptions within the normal PWM signal range. Given that T_{ON} and T_{OFF} are respective on and off times of the PWM waveform, the total period of the waveform T_{total} can be computed as shown in Equation 10.

$$T_{\text{total}} = T_{ON} + T_{OFF} \tag{10}$$

We use the given equation to produce synthetic attack samples based on hardware Trojan logic. Given the PWM waveform period for a Pixhawk flight controller is 2500μs, a benign PWM value, denoted as PWM_{Benign}, can be inverted to yield $PWM_{Inverted}$, as illustrated in Equation 11.

$$PWM_{Inverted} = abs(2500\mu s - PWM_{Benign}) \tag{11}$$

A Pixhawk flight controller limits the band of its computed PWM values as shown in Equation 12, ensuring that the synthetic attack samples are within the specified bounds of PWM.

$$effective\_PWM = control\_value \times \frac{max\_PWM - min\_PWM}{2} + \frac{max\_PWM + min\_PWM}{2} \tag{12}$$

The mixer sets control values, ensuring that PWM signals adhere to the UAV's operational range. Any value outside this range, such as below 1100s, is auto-corrected to the minimum limit. The process is outlined in Algorithm 1. Attack samples for channels 4 and 1 are illustrated in Figs. 4(a) and 4(b), showing subtly aligned benign and altered PWM values.

Algorithm 2: Firmware Modification Attack

```
Input: outputs[MAX_ACTUATORS], num_outputs
Output: inverted_outputs[MAX_ACTUATORS]
for i = 0 to num_outputs - 1 do
    function = _mixing_output.outputFunction(i)
    is_reversible = reversible_outputs & (1u << i)
    output = outputs[i]
    if ((int)function ≥ (int)OutputFunction::Motor1) &&
     ((int)function ≤ (int)OutputFunction::MotorMax) &&
     !is_reversible then
        if function == OutputFunction::Motor1 || function ==
         OutputFunction::Motor2 then
            output = (output - PWM_SIM_PWM_MIN_MAGIC) /
             (PWM_SIM_PWM_MAX_MAGIC -
             PWM_SIM_PWM_MIN_MAGIC)
        end
    end
    inverted_outputs[i] = output
end
```

5 CASE STUDY: ATTACK IMPACTS

We implemented hardware Trojan attacks, both practically and through software modification in jMAVsim simulator.

Physical Attack Implementation The attack circuit is shown in Fig. 21 and impact in Fig. 22 in Appendix D. To execute the practical Trojan attack, we undertake the following procedure:

(1) Use a NAND logic gate inverter IC to modify the PWM waveform from the flight controller's output.
(2) Direct both the original and modified PWM values to a switch.
(3) Connect the switch outputs to the ESCs.
(4) Activate the switch every 4 seconds, simulating the hardware Trojan's trigger-payload mechanism.

Simulated Attack We simulated hardware Trojan attacks with two characteristics: (i) random alteration of PWM values for the motors and (ii) targeted alteration in PWM values with an incremental offset, ultimately crashing the drone. Fig. 5(a) shows the stealthiness of attack (i)impacting UAV's duty cycle, angular velocity, and causing unhealthy vibrations as seen in Fig. 5(c) and Fig. 5(d), which may result in increased motor wear and compromised mission outcomes. As Fig. 5(b) illustrates, a UAV might complete its mission without flagging the intrusion. For (ii), the attack applies a large, oscillating offset to the PWM signals, incrementing or decrementing by 500 units in Fig. 6(a), and 525 in Fig. 6(b) with each iteration to induce an increment in the offset. The impacts are shown in Fig. 6(a), which causes increased system vibration, but still the mission is carried out, and Fig. 6(b), where the UAV crashes.

Discussion: (i) Random alteration of PWM values for the motors, subtly impacting the UAV's duty cycle and angular velocity. This can result in unhealthy vibrations and increased wear on the motors,

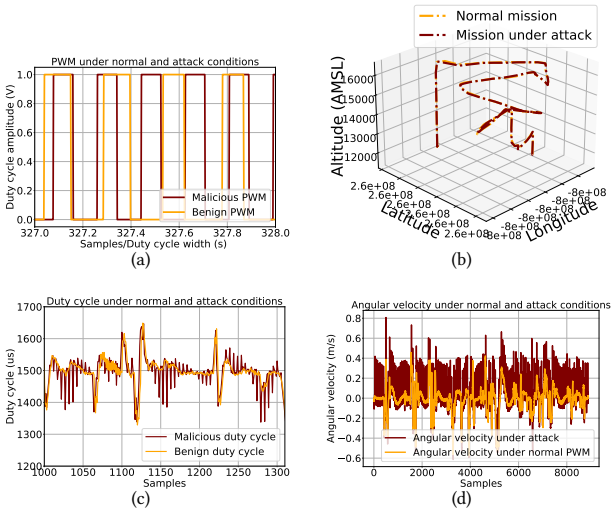


Figure 5: Simulated Trojan-based PWM attack with random PWM alteration. (a) PWM signals (minor differences in attacked waveforms), (b) UAV mission (completed without detecting the attack), (c) duty cycle, and (d) angular velocity (attack introduces noise, increasing system vibration).

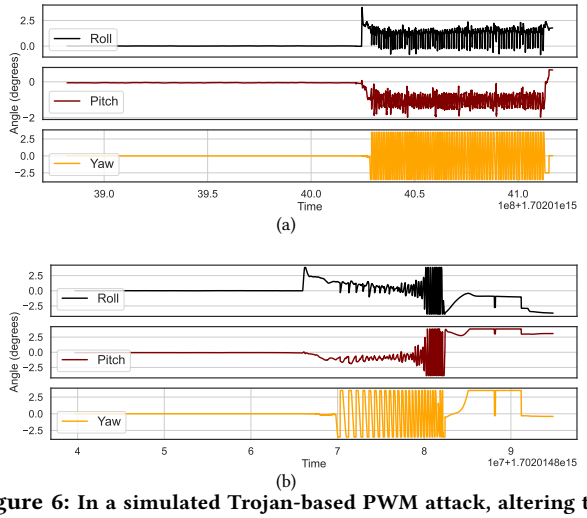


Figure 6: In a simulated Trojan-based PWM attack, altering the duty cycle value by 500 units (a) allowed the mission to proceed with increased system vibration, whereas an alteration of 525 units (b) resulted in a UAV crash.

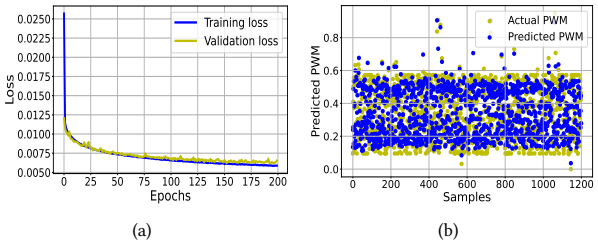


Figure 7: (a) Regression model's training and validation losses show effective PWM prediction. (b) Strong correlation underscores the effectiveness of the chosen features in anomaly detection.

as seen in Fig. 5(c) and Fig. 5(d). A UAV might complete its mission under such conditions without triggering internal failsafes or flagging the intrusion, as illustrated in Fig. 5(b). (ii) Targeted alteration in PWM values with incremental offsets, potentially leading to a UAV crash. The alteration amount, denoted by the attack parameter (Δ), is varied to observe different outcomes. With an incremental offset, we witness increased vibrations within the system but not enough to halt the mission, as shown in Fig. 6(a). Conversely, a larger offset of $\Delta = 525$ units in Fig. 6(b) results in a UAV crash.

It's critical to note that the attack parameter (Δ) directly correlates with the impact severity. A minimal Δ may cause the UAV to experience only slight disruptions, whereas a moderate Δ can lead to operational instability. As demonstrated, a significant Δ leads to a loss of control and subsequent crash. This shows that the impact of PWM signal manipulation is a function of both the magnitude of alteration and the operational context, including flight conditions and environmental factors. Thus, the attack model, as summarized in Equation 8 and 9, albeit appearing simplistic, can yield a spectrum of operational consequences, substantiating the complex nature of such cyber-physical attacks.

6 PROPOSED CONTROL-FUSED IDS

The proposed IDS has three phases, each discussed at length along with technical details in this section.

6.1 Flight-Control and PWM Mapping phase

Selecting accurate features for flight controller modeling is crucial for simulating a UAV's flight control process. Precision in PWM value prediction is achieved by minimizing the root mean squared error (RMSE). Given the non-linear relationships inherent in UAV features, we use a neural network-based Keras regressor [41]. The link between control features and PWM values stems from their use as inputs to the flight controller's firmware, generating motor-driving PWM signals. ConFIDe's regressor, as described in Section C.2, models this relationship using training data. Deviations from expected PWM outputs are thus identified as anomalies. This mapping, which helps integrate the entire sensor-control-actuation cycle, is at the heart of ConFIDe's real-time intrusion detection mechanism. This mapping was developed using a diverse dataset collected from real-world missions, covering a wide spectrum of UAV behaviors and environmental conditions. This dataset was specifically designed to encapsulate the variability in control feature-to-PWM value mappings that might arise due to differences in varied flight parameters.

Regressor Architecture: Our model, featuring 33 flight control inputs and five hidden layers, predicts four PWM outputs. The model was trained over 200 epochs with a batch size of 270 and yields a mean absolute error of 0.058, MSE of 0.0066, and RMSE of 0.0814. The training loss and PWM predictions are illustrated in Fig.7(a) and Fig.7(b), respectively.

6.2 Training phase

Our approach uses one-class classification algorithms to create a decision boundary from existing data. Significant deviations are flagged as anomalies in testing and real-time operations [42]. We

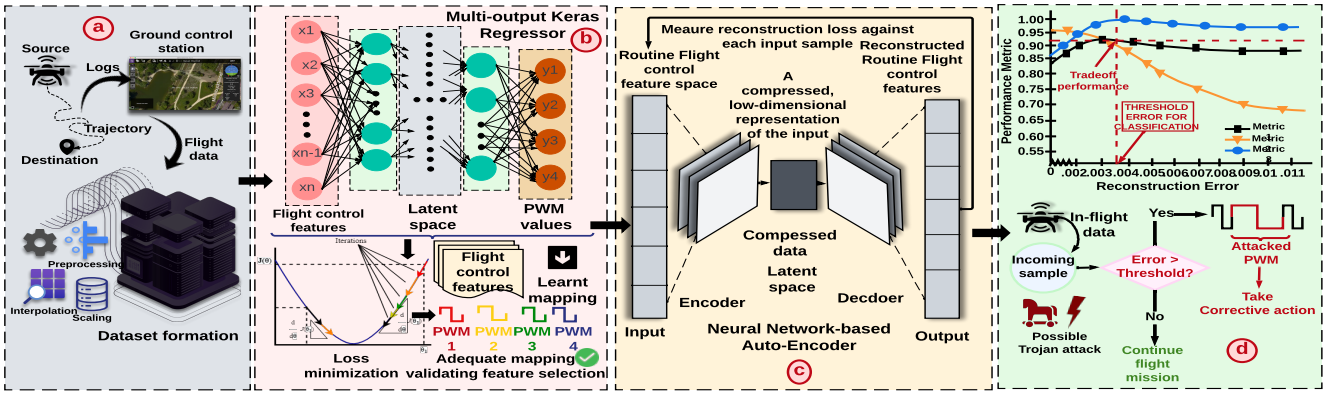


Figure 8: ConFIDe Overview: (a) Dataset Formation - Data collection and preparation. (b) PWM Mapping - Regression for control optimization. (c) Training - Utilizing an AutoEncoder (AE). (d) Classification - Setting thresholds and categorizing data.

employ neural network-based Auto-Encoder (AE), as alternatives like OCSVM and DBSCAN are computationally intensive and less adept at understanding the nuanced non-linear relationships in UAV flight data. The autoencoder provides a more robust framework for understanding the intricate dynamics of UAV flight control data. This is primarily due to its ability to learn a dense, low-dimensional representation of the data, which inherently captures the complex relationships within the UAV's operational signals. The autoencoder reconstructs original network traffic at its output layer by learning intrinsic network traffic attributes [43].

Network Architecture: Autoencoder comprises sequentially connected encoder and decoder networks. Encoder, using function f with parameters W and b, maps input X to a feature representation (Equation13). Decoder, with g and parameters W' and b', reconstructs the input from this (Equation14).

$$H = f(WX + b) \tag{13}$$

$$Z = g(W'H + b') \tag{14}$$

Hyperparameter tuning: To minimize MSE loss, we adjust the architecture. It features input and output layers with 37 nodes and five palindromic hidden layers consisting of 52, 40, and 24 nodes, getting MSE of 4.76e-04 after 1000 epochs.

6.3 Classification phase

The model's decision-making phase checks anomalies based on learned flight control and PWM mapping.

Classification: Using the autoencoder, classification hinges on the reconstruction error. Test samples are reconstructed, and their MSE against the original data is determined. If this error exceeds the trained threshold, the sample is marked as an outlier; otherwise, it's considered benign.

Threshold Selection: To determine the anomaly detection threshold, we analyze MSEs from training samples, with the highest quantile setting the threshold $\mathcal{T}$. We select a threshold ensuring 98.99% of data is benign, as shown in Equation15. Based on this threshold, classification rules are specified in Equation16. Here, *MSE* represents the MSE per training sample, and q indicates the benign data

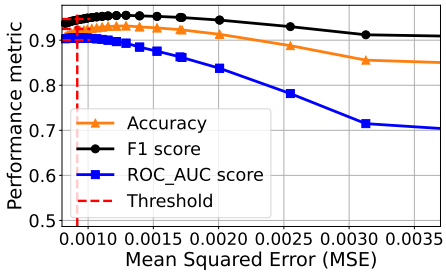


Figure 9: AE-based classification thresholding using quantile method. Balancing ROC_AUC score with accuracy/F1; higher ROC signifies superior performance

fraction (0.9899 in our scenario). Classification rules are further detailed in Equation 16.

$$\mathcal{T} = \text{quantile}(MSE, q) \tag{15}$$

$$Label = \begin{cases} \text{“Anomaly", if } MSE_{\text{sample}} > \mathcal{T} \\ \text{“Benign", if } MSE_{\text{sample}} \leq \mathcal{T} \end{cases} \tag{16}$$

Our selected threshold, 0.000917, balances ROC-AUC (0.9049) with accuracy and F1 scores. This threshold, as shown in Fig.9, guides ConFIDe's performance in real-time UAV monitoring, detecting hardware/PWM threats by analyzing control data and PWM values. For reproducibility, our code is available at[44]. ConFIDe system, illustrated in Fig. 8, operates in real-time on hardware, processing control data from the flight controller and PWM values directing the ESCs.

7 IMPLEMENTATION OF CONFIDE

Implementing ConFIDe in UAVs requires real-time data management and classification. This section outlines hardware specifications and flight control data capturing, with its Pixhawk UAV application as in Fig.10(a) and Fig.10(b).

7.1 Layer of Implementation

Unlike conventional IDSs in the UAV's sensor/network layer, ConFIDe integrates the knowledge of PWM signals in its system. Hence, it operates after the flight controller has generated the PWM signals. ConFIDe monitors the PWM outputs from the flight controller,

(a) (b)

Figure 10: ConFIDe was evaluated in-lab on an S500 Pixhawk 2.4.8 UAV, propellers removed for safety. The setup included a UAV, radio controller, and a hardware Trojan for PWM inversion. Readings were processed via Arduino and Pymavlink

verifying their legitimacy based on the flight control. Only non-anomalous PWM outputs then reach the ESCs for motor operation. ConFIDe's hardware framework consists of (1) the target UAV, (2) a data collection unit, and (3) a computational device for either its training or classification.

7.2 Hardware Specification

Here, we detail the technical specifications for the components central to the ConFIDe system.

7.2.1 UAV to be defended. ConFIDe was implemented on an S500 Pixhawk 2.4.8 quadcopter UAV, powered by a 32-bit ARM CortexM4 processor and running the NuttX Real-Time Operating System (RTOS). For navigation, the UAV was equipped with a Neo-M8N GPS and an integrated compass. Additionally, an ESP8266-NodeMCU WiFi module was integrated for telemetry purposes, facilitating communication with ground control at a baud rate of 921600.

7.2.2 Data collector module. ConFIDe utilizes two datasets for detection: 1) flight control data encompassing modules like control, estimator status, and position attributes, and 2) post-routine execution PWM outputs. We leverage the Pymavlink Python library for intra-UAV communication [45]. This library facilitates real-time UAV data transfer. MAVlink connection is initiated, and data is continuously fetched. We analyze PWM outputs with varied pulse durations. For real-time PWM analysis, we use the Arduino-Mega2560, employing ArduinoIDE routines for data automation.

7.2.3 Computational Device. We use an 11th Gen Intel(R) Core(TM) i7-1195G7 @ 2.90GHz with 16.0 GB RAM. This 64-bit system processes data from Pymavlink and Arduino and manages ConFIDe's training and classification.

8 EXPERIMENTAL EVALUATION

ConFIDe's effectiveness was validated theoretically using synthetic attack data and practically via a Trojan emulation of UAV hardware. We evaluated its performance using standard metrics (1) *Accuracy*, representing the ratio of correct identifications to all points; (2) *Precision*, indicating the fraction of correct classifications out of all classified instances; (3) *Recall*, denoting the fraction of correctly identified cases among all instances; (4) *F1 score*, which balances recall and precision. We used the *ROC-AUC* score to assess the model's ability to differentiate between classes. Higher the AUC, better the model's discriminatory power. The following nine research questions (RQs) guided our evaluation process.

Table 2: Performance metrics evaluated on different combinations of channels attacked

No. of channels attacked	Channels	Accuracy	F1-Score	ROC-AUC score	Precision	Recall
4	1234	0.925	0.949	0.927	0.975	0.924
3	123	0.909	0.938	0.893	0.953	0.924
	124	0.899	0.933	0.875	0.941	0.923
	134	0.932	0.953	0.939	0.984	0.924
	234	0.933	0.954	0.941	0.985	0.924
2	12	0.783	0.864	0.641	0.812	0.924
	13	0.904	0.935	0.885	0.947	0.924
	14	0.905	0.936	0.887	0.948	0.924
	23	0.905	0.936	0.886	0.948	0.924
	24	0.896	0.930	0.868	0.936	0.924
	34	0.931	0.953	0.934	0.983	0.924
1	1	0.836	0.894	0.747	0.866	0.924
	2	0.822	0.886	0.719	0.851	0.924
	3	0.905	0.936	0.886	0.949	0.924
	4	0.889	0.926	0.855	0.928	0.924

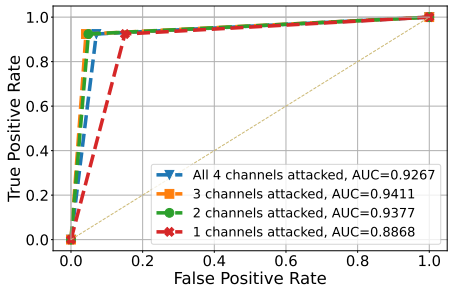


Figure 11: Performance varies based on the number of attacked channels, with the highest efficacy observed when three channels are simultaneously targeted.

RQ1: How effectively can ConFIDe perform when different combinations of PWM channels are under attack?

RQ2: Is ConFIDe able to detect PWM attacks when the trigger frequency of the hardware Trojan is varied?

RQ3: Is the chosen feature set optimal for training ConFIDe? What are the effects of using more or fewer features?

RQ4: How does ConFIDe compare to other ML models?

RQ5: What is ConFIDe's computation time?

RQ6: Can ConFIDe detect real-time hardware attacks?

RQ7: Can ConFIDe detect simulated PWM-based attacks??

RQ8: How can removing individual modules affect ConFIDe's performance, and how can this be optimized?

RQ9: Can ConFIDe detect common sensor attacks?

8.1 Evaluation Results

RQ1 - Performance evaluation when different combinations of PWM channels are attacked: A quadcopter UAV has four PWM channels, each driving the corresponding BLDC motor. An adversary can carry out a PWM-manipulative hardware Trojan attack on either one, two, three, or all four channels. Furthermore, attacking two and three sets of channels can be carried out in various combinations. As summarized in Section 4.4, we carry out PWM inversion with a trigger of 25% in our test set to generate synthetic attack data for each possible combination of the channels. The accuracy, F1-score, ROC-AUC score, precision, and recall for all these scenarios are summarized in Table 2. Fig. 11 visualizes the ROC-AUC scores for the same. It can be seen that ConFIDe successfully classified the

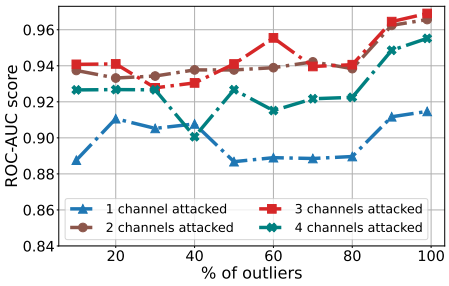


Figure 12: ConFIDe's performance was assessed by altering the percentage of outliers in the test set, indicative of Trojan's trigger frequency. ConFIDe effectively identifies even covert attacks.

attacks on all different combinations of channels with the highest AUC for a combination of attacks on three channels.

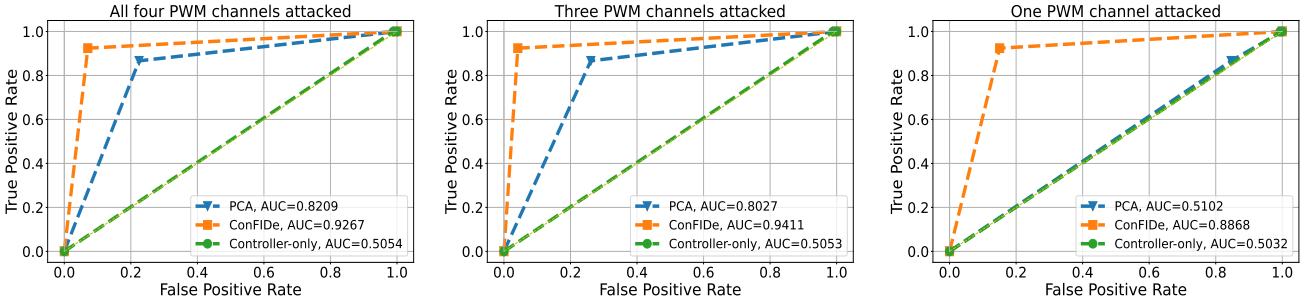
RQ2 - Performance evaluation when the trigger frequency of the hardware Trojan attack is varied: As entailed in Section 4.4, a hardware Trojan attack has a payload (the attack impact), which is launched whenever the trigger is satisfied. An adversary can vary the trigger frequency based on their attack goal. The less frequent the trigger, the stealthier the attack, and vice versa. Synthetically, this variation in the trigger frequency can be represented by varying the number of outliers or the attack data in the test set. For example, in a test set consisting of 100,000 benign samples, a hardware Trojan triggered 10% of the time would correspond to 10,000 malicious samples and 90,000 benign ones. We evaluated ConFIDe's performance across various trigger frequencies and channel combinations, generating multiple test sets with varying numbers of malicious PWM samples. The results, visualized in Fig. 12, demonstrate that ConFIDe effectively detects even stealthy attacks with less frequent triggers. Notably, the detection accuracy remains robust even at lower trigger frequencies, a scenario typically challenging for IDS systems due to reduced attack signatures.

RQ3 - Performance comparison of ConFIDe with different sets of features: ConFIDe has been trained on a set of features pertaining to the flight control in a UAV. The initial feature set, comprising 33 pivotal features, was carefully chosen based on their relevance to UAV flight control dynamics and their potential impact on identifying anomalous behaviors. These features encompass control inputs, positional accuracy, GPS data, and core motion angles, among others, which are crucial for the real-time detection capabilities of ConFIDe. These features were manually filtered out from the flight logs based on the documentation of the working mechanism of the flight controller [36]. The number of features accounts for the dimensionality of a model. Higher dimensionality can often result in models being unable to distinguish between classes adequately. To validate the feature selection for ConFIDe, we train two other AE models with a set of features obtained from principal component analysis (PCA) on the flight logs, accounting for 95% of the variance, and a set of features from the controller module of the flight controller, respectively, and compare their performances for different combinations of channels attacked. The ROC-AUC scores representing the performance of each of these models can be seen in Fig. 13. As shown in Fig. 13(a), Fig. 13(b), and Fig. 13(c), ConFIDe outperforms both the models when all four, three, and/or one channel(s) are attacked, respectively.

RQ4 - Comparative effectiveness of ConFIDe with other machine learning (ML) models: Comparing AE-based ConFIDe's performance with other ML models is essential to determine its effectiveness. One such algorithm is One-Class Support Vector Machine (OCSVM), widely used in anomaly detection applications. This comparison can provide insights into the strengths and weaknesses of each technique. We implemented OCSVM at the third phase of ConFIDe (Fig. 8(c)) and compared the performance with AE implemented in the third phase. As shown in Fig. 14, ConFIDe outperforms OCSVM due to OCSVM's inability to capture the non-linear relationship between UAV control features. While OCSVM is adaptable with radial basis function (RBF) for non-linear scenarios, our findings show that ConFIDe exhibits superior performance in the context of UAV security. This is due to its deep learning architecture, which effectively captures and analyzes the intricate patterns and dependencies characteristic of UAV control data. his discrepancy can be attributed to the unique challenges of UAV control signals, which exhibit highly complex and dynamic behaviors. The potential limitations of OCSVM in this context stem from its reliance on a predefined kernel function to transform the input space, which might not fully encapsulate the intricate dynamics of PWM signals in UAVs. Moreover, recent studies have shown that autoencoders can outperform traditional SVMs offering higher accuracy and reliability [46].

RQ5 - Computation time for ConFIDe: We record the time it takes for ConFIDe to make an attack or benign prediction on each sample. The total computation time is the sum of the time taken to reconstruct the incoming sample by the AE and the time taken by the classifier phase. These times are calculated for each combination of the channels attacked. Hence, for one channel under attack, ConFIDe detected all the malicious samples in 2.42ms. Similarly, two, three, and four channels under attack take 2.56ms, 2.56ms, and 2.98ms, respectively. On average, the time ConFIDe takes to predict whether an incoming sample is benign or malicious is approximately 2.63ms. The experiments were carried out on the device specified in Section. 7.2.3. It is to be noted that in an in-flight system, this IDS will be implemented using field programmable gate arrays (FPGA) technology within a TCB, which can reduce the detection latency and increase the computational speed significantly, as demonstrated by Zhang et al., who were able to increase the detection speed by 128 times [47]. It must be noted that implementing ConFIDe on an FPGA directly within the system's secure processing framework does not introduce new supply chain vulnerabilities [48]. This approach capitalizes on the inherent capabilities of FPGAs for high-speed processing while ensuring system integrity through a trusted configuration and verification process. The implementation will be carefully designed to utilize the FPGA's flexibility and speed in a secure manner, effectively strengthening the system's defenses without complicating the supply chain.

RQ6 - Practical performance evaluation of ConFIDe: To answer RQ6, we carried out multiple experiments on real-life UAVs. As explained in Section 4, the hardware Trojan attack is emulated using an inverter IC. In total, 2 data sets were formed in real-time: An attack data set with channel 1's PWM values attacked (20 samples) and a routine operation data set (10 samples). After receiving the flight control data in real time, ConFIDe activates the classification



(a) (b) (c)

Figure 13: Performance comparison of ConFIDe with models trained on different feature sets, i.e., PCA-based and controller-only models. (a) Shows the ROC-AUC score of all the models when all four channels, (b) three channels, and (c) only one channel is/are under a PWM-manipulative hardware Trojan attack, respectively. The attack selectively alters the duty cycle of the PWM signals sent to the motors. ConFIDe outperforms all the other models.

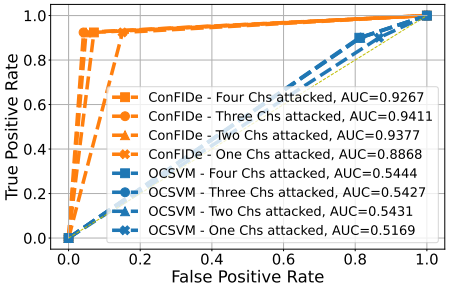


Figure 14: In the performance evaluation, ConFIDe outperforms OCSVM due to its ability to capture the non-linear relationship between UAV control features, as OCSVM falls short in this regard.

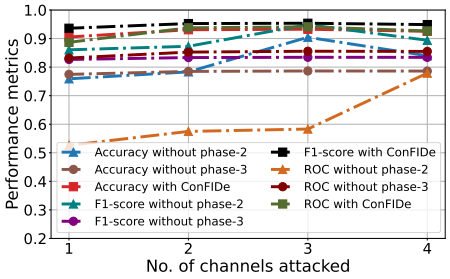


Figure 15: Results from an ablation study evaluating individual ConFIDe modules' performance under various combinations of attacked channels consistently show ConFIDe outperforming other methods with the highest scores.

module. It was seen that ConFIDe correctly classified all 20 of the attack samples in real time. Moreover, all 10 of the routine samples (benign) were also correctly classified. For both cases, the number of false negatives was zero, achieving 100% accuracy on practical implementation. Fig. 10 shows the implementation ofConFIDe system on an S500 Pixhawk 2.4.8 UAV. The experiments were conducted in a lab setup with the propellers removed for safety.

RQ7 - Performance evaluation of ConFIDe under simulated attacks: To answer RQ7, we simulated PWM-altering attacks through firmware modification on PX4 firmware as entailed in Section 5 where the PWM output for motors 1 and 2 is changed in a trigger-payload fashion, altering the duty cycle of the PWM signal. Despite high vibration in the system, as shown in Fig. 5(d), this attack went undetected as no failsafe was internally triggered by the system. Nonetheless, ConFIDe detected 30 out of the 31 attack samples with an accuracy of 99.2% and a ROC-AUC score of 98.38% with one false positive and zero false negatives as illustrated in the confusion matrix in Fig. 17. The implications of such attacks extend beyond immediate threats. Fig. 16 shows the actuator controls' frequency peaks exceeding 20Hz, indicating detrimental vibrations. These not only affect drone performance but also cause motor wear and tear. ConFIDe's ability to detect these anomalies highlights its effectiveness in identifying firmware manipulations that can silently degrade UAV hardware health.

RQ8 - Impact of eliminating individual modules on ConFIDe's performance: To thoroughly assess ConFIDe's overall performance and the contribution of its components, we conduct an ablation study comprising two parts. This study aims to scientifically analyze the impact of removing key elements within ConFIDe and evaluate the performance degradation resulting from these changes. This approach enables us to understand each component's importance and effectiveness in intrusion detection. The first part involves removing phase-2 (Fig. 8(b)), the flight control and PWM mapping. With phase 2 removed, we used feature engineering techniques to design a feature vector for the autoencoder (phase 3). The second part of this ablation study investigates the removal of the autoencoder (phase-3) in (Fig. 8(c)). Since the autoencoder carries out the main detection for ConFIDe, in case of its removal, we set a threshold of mean squared error of the PWM outputs predicted by the regressor in the flight control and PWM mapping (phase-2) for attack detection. We evaluate the performance of eliminating these individual modules under different combinations of channels attacked. The performance metrics, i.e., accuracy, F1, and ROC-AUC scores, are shown in Fig. 15. ConFIDe performs better in all the cases, with the highest scores for all performance metrics.

RQ9 - Sesnor attacks: To ensure ConFIDe offers end-to-end security, we tested it against common sensor attacks such as GPS spoofing. We launched the attack with three different deviation levels: small, medium, and large. The attack goal was to ultimately deviate the UAV from its planned trajectory. The attack data files and graphs (Fig. 23, Appendix D) can be found at [44]. As seen in Table. 3, ConFIDe has a detection accuracy of up to 100% when the attack becomes more evident.

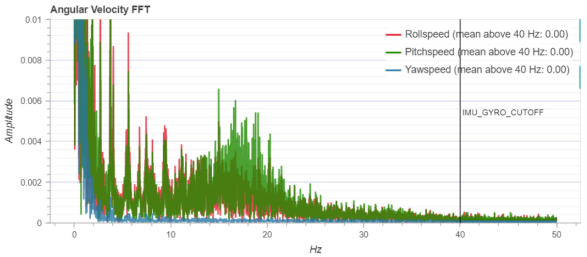


Figure 16: Frequency plot of roll, pitch, and yaw axes from PID actuator controls during a randomly PWM-altered flight. Peaks above 20Hz indicate harmful vibrations and potential motor wear.

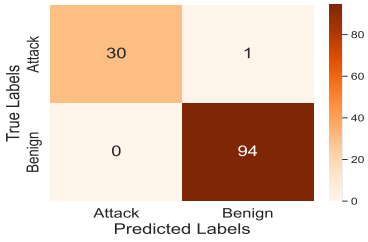


Figure 17: ConFIDe's performance on simulated PWM attack. 30 out of 31 attack samples were correctly predicted with zero FNs.

8.2 Discussion

After an attack is detected by ConFIDe, mitigation can involve triggering the failsafe so necessary remedial actions may be taken. A possible mitigation strategy can involve imputation of the PWM signals based on the learning of ConFIDe (using a time-series model such as LSTM or ARIMA as used by Dash et al. [15], such that at the point where anomalies are detected, ConFIDe can provide PWM signals to keep the mission going. The scope of this paper is detection and not mitigation, but this will be taken on as a future work where the timing criticality of imputing data on attack detection will also be evaluated. In the case of UAV security, false positives are considered bearable and better because they trigger a failsafe mechanism that prevents a potentially compromised UAV from continuing its mission. The failsafe mechanism can send the UAV back to the base station or take other remedial actions to ensure its data are not compromised, i.e., the mission might fail. Contrarily, a false negative, or the inability to identify an actual breach, can have disastrous effects because it would permit a compromised UAV to carry out its mission, possibly harming people or releasing critical information. Hence, false positives are preferable over false negatives in UAV security because they add extra protection. The false-positive threshold in determining security levels depends on the specific use case; for higher data security, a lower false-positive rate can be achieved by increasing the detection threshold. Striking a balance between security and operational efficiency is crucial, as overly conservative thresholds can lead to increased false alarms, potentially disrupting UAV operations unnecessarily.

Moreover, it is to be noted that the control-PWM mapping phase is a one-time offline DL model in the design of ConFIDe. Once this mapping is understood, it is leveraged to further the ConFIDe IDS design in its detection of any anomalous behavior. Moreover, the training for the autoencoder module in Fig. 8(c) is also offline. Only the testing and classification phase of the autoencoder in Fig. 8(c) is in real-time. For future work, the effectiveness of ConFIDe on more diverse UAV platforms and attack scenarios can be evaluated further to validate its robustness and reliability in real-world situations. Furthermore, we can incorporate flight control from various models and investigate how hardware limitations and environmental factors affect ConFIDe's performance, offering optimization insights. Furthermore, we acknowledge the critical importance of minimizing delay in UAV operations. Our approach introduces a latency of 2+ms, as identified in our evaluation. This latency is within the operational parameters' tolerable limits for UAV systems, ensuring no compromise to mission-critical functionalities. Arya et al. elaborate on ground-to-UAV communication challenges, emphasizing that slight delays are often acceptable in exchange for enhanced security and reliability, even in low-latency network designs [49]. To enhance ConFIDe's generalizability, our future work will utilize a training dataset to encompass multiple UAV models, capturing a broad spectrum of operational nuances and manufacturing variances. This will be essential for our neural network to generalize PWM mapping patterns effectively, avoiding model-specific biases. Simultaneously, we will refine our anomaly detection threshold through rigorous statistical analysis of PWM values across varied UAV models. This will ensure the threshold's broad applicability, accurately distinguishing between normal and anomalous behaviors in a way that accounts for the inherent variability in UAV systems.

Table 3: Performance in Detecting GPS Spoofing Attacks

Metric	Small Deviation	Medium	Large
Accuracy	0.9972	0.9993	1.0
F1 Score	0.9986	0.9995	1.0
ROC AUC	0.9986	0.9995	1.0
Precision	1.0	1.0	1.0

9 CONCLUSION

In this work, we have proposed a Control-Fused Intrusion Detection (ConFIDe) system that can defend against insider attacks in the hardware/firmware of UAVs instead of the existing IDSs securing the sensor/network space. We trained ConFIDe on a comprehensive UAV dataset consisting of multiple flight controls for PWM signal duty cycle computation. It verifies the integrity of the flight controller- generated PWM signals, ensuring the motors receive the signals free from hidden exploits. Further, to experimentally validate our proposed IDS, we simulate and emulate a hardware Trojan attack synthetically and in a real-life UAV system. The performance results are evaluated under different attack scenarios. Overall, ConFIDe performs well in all these scenarios achieving a ROC-AUC score up to 92.5% on synthetic attack samples, 99.2% on simulated data, and 100% accuracy when applied to real-time data.

10 ACKNOWLEDGMENT

This research was supported in part by the National Security Agency (NSA) under award H98230-22-1-0327, the Department of Energy (DOE) under award DE-CR0000024, and the National Science Foundation (NSF) under award 2150248.

REFERENCES

[1] Ed Alvarado. Industry leading drone market analysis 2022-2030: Droneii. https://droneii.com/drone-market-analysis-2022-2030, Sep 2022.

[2] Faa grants two more uas exemptions. https://www.faa.gov/newsroom/faa-grants-two-more-uas-exemptions, Jan 2015.

[3] Sayan Paramanik, Partha Sarathi Sarkar, Koustav Kumar Mondol, Avijit Chakraborty, Sajib Chakraborty, and Krishna Sarker. Survey of smart grid network using drone ptz camera. In *2019 Devices for Integrated Circuit (DevIC)*, pages 361–364, 2019.

[4] Muhammad Afif Husman, Waleed Albattah, Zulkifli Zainal Abidin, Yasir Mohd. Mustafah, Kushsairy Kadir, Shabana Habib, Muhammad Islam, and Sheroz Khan. Unmanned aerial vehicles for crowd monitoring and analysis. *Electronics*, 10(23), 2021.

[5] Andreas Kamilaris and Francesc X. Prenafeta-Boldú. Disaster monitoring using unmanned aerial vehicles and deep learning. https://arxiv.org/abs/1807.11805, Aug 2018.

[6] Konstantinos Kanistras, Goncalo Martins, Matthew J Rutherford, and Kimon P Valavanis. A survey of unmanned aerial vehicles (uavs) for traffic monitoring. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 221–234. IEEE, 2013.

[7] CYN Norasma, MA Fadzilah, NA Roslin, ZWN Zanariah, Z Tarmidi, and FS Candra. Unmanned aerial vehicle applications in agriculture. In *IOP Conference Series: Materials Science and Engineering*, volume 506, page 012063. IOP Publishing, 2019.

[8] Hazim Shakhatreh, Ahmad H Sawalmeh, Ala Al-Fuqaha, Zuochao Dou, Eyad Almaita, Issa Khalil, Noor Shamsiah Othman, Abdallah Khreishah, and Mohsen Guizani. Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges. *Ieee Access*, 7:48572–48634, 2019.

[9] Manuel Patchou, Benjamin Sliwa, and Christian Wietfeld. Unmanned aerial vehicles in logistics: Efficiency gains and communication performance of hybrid combinations of ground and aerial vehicles. In *2019 IEEE Vehicular Networking Conference (VNC)*, pages 1–8, 2019.

[10] Tariq Samad, John S. Bay, and Datta Godbole. Network-centric systems for military operations in urban terrain: The role of uavs. *Proceedings of the IEEE*, 95(1):92–107, 2007.

[11] Marco Calderón, Wilbert G Aguilar, and Darwin Merizalde. Visual-based real-time detection using neural networks and micro-uavs for military operations. In *International Conference of Research Applied to Defense and Security*, pages 55–64. Springer, 2020.

[12] Menaka Pushpa Arthur. Detecting signal spoofing and jamming attacks in uav networks using a lightweight ids. In *2019 international conference on computer, information and telecommunication systems (CITS)*, pages 1–5. IEEE, 2019.

[13] Omar Bouhamed, Ouns Bouachir, Moayad Aloqaily, and Ismaeel Al Ridhawi. Lightweight ids for uav networks: A periodic deep reinforcement learning-based approach. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 1032–1037. IEEE, 2021.

[14] Rakesh Shrestha, Atefeh Omidkar, Sajjad Ahmadi Roudi, Robert Abbas, and Shiho Kim. Machine-learning-enabled intrusion detection system for cellular connected uav networks. *Electronics*, 10(13):1549, 2021.

[15] Pritam Dash, Guanpeng Li, Zitao Chen, Mehdi Karimibiuki, and Karthik Pattabiraman. Pid-piper: Recovering robotic vehicles from physical attacks. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 26–38. IEEE, 2021.

[16] Gökçen Yılmaz Dayanıklı, Sourav Sinha, Devaprakash Muniraj, Ryan M Gerdes, Mazen Farhood, and Mani Mina. Physical-layer attacks against pulse width {Modulation-Controlled} actuators. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 953–970, 2022.

[17] Asia Perspective. China's thriving drone industry. https://www.asiaperspective.com/china-thriving-drone-industry/, Oct 2021.

[18] Tony F Wu, Karthik Ganesan, Yunqing Alexander Hu, H-S Philip Wong, Simon Wong, and Subhasish Mitra. Tpad: Hardware trojan prevention and detection for trusted integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(4):521–534, 2015.

[19] Trey Reece and William H Robinson. Detection of hardware trojans in third-party intellectual property using untrusted modules. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(3):357–366, 2015.

[20] Confide-datasets. https://sites.google.com/view/raid-datasets/home.

[21] Jianguo Sun, Wenshan Wang, Qingan Da, Liang Kou, Guodong Zhao, Liguo Zhang, and Qilong Han. An intrusion detection based on bayesian game theory for uav network. In *Proceedings of the 11th EAI International Conference on Mobile Multimedia Communications*, MOBIMEDIA'18, page 56–67, Brussels, BEL, 2018. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[22] AbdelRahman Eldosouky, Aidin Ferdowsi, and Walid Saad. Drones in distress: A game-theoretic countermeasure for protecting uavs against gps spoofing. *IEEE Internet of Things Journal*, 7(4):2840–2854, 2020.

[23] Hichem Sedjelmaci, Mohamed-Ayoub Messous, Sidi Senouci, and Horiya Imane Brahmi. Toward a lightweight and efficient uav-aided vanet. *Transactions on Emerging Telecommunications Technologies*, 30:e3520, 08 2019.

[24] Ammar Ahmed Khan, Muhammad Mubashir Khan, Kashif Mehboob Khan, Junaid Arshad, and Farhan Ahmad. A blockchain-based decentralized machine learning framework for collaborative intrusion detection within uavs. *Computer Networks*, 196:108217, 2021.

[25] Jean-Philippe Condomines, Ruohao Zhang, and Nicolas Larrieu. Network intrusion detection system for uav ad-hoc communication: From methodology design to real test validation. *Ad Hoc Networks*, 90:101759, 2019. Recent advances on security and privacy in Intelligent Transportation Systems.

[26] Elena Basan, Maria Lapina, Nikita Mudruk, and Evgeny Abramov. Intelligent intrusion detection system for a group of uavs. In Ying Tan and Yuhui Shi, editors, *Advances in Swarm Intelligence*, pages 230–240, Cham, 2021. Springer International Publishing.

[27] Qasem Abu Al-Haija and Ahmad Al Badawi. High-performance intrusion detection system for networked uavs via deep learning - neural computing and applications. https://link.springer.com/article/10.1007/s00521-022-07015-9, Feb 2022.

[28] V. Praveena, A Vijayaraj, Chinnasamy Ponnusamy, Ali Ihsan, Roobaea Alroobaea, Saleh Yahya, and Muhammad Raza. Optimal deep reinforcement learning for intrusion detection in uavs. *Computers, Materials and Continua*, 70:2639–2653, 09 2021.

[29] Jason Whelan, Abdulaziz Almehmadi, and Khalil El-Khatib. Artificial intelligence for intrusion detection systems in unmanned aerial vehicles. *Computers and Electrical Engineering*, 99:107784, 2022.

[30] Robert Mitchell and Ray Chen. Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications. *IEEE transactions on systems, man, and cybernetics: systems*, 44(5):593–604, 2013.

[31] Raul Quinonez, Jairo Giraldo, Luis Salazar, Erick Bauman, Alvaro Cardenas, and Zhiqiang Lin. Savior: Securing autonomous vehicles with robust physical invariants. In *Usenix Security*, 2020.

[32] Cheolhyeon Kwon, Scott Yantek, and Inseok Hwang. Real-time safety assessment of unmanned aircraft systems against stealthy cyber attacks. *Journal of Aerospace Information Systems*, 13(1):27–45, 2016.

[33] Reza Fotohi, Masoud Abdan, and Sanaz Ghasemi. A self-adaptive intrusion detection system for securing uav-to-uav communications based on the human immune system in uav networks. *Journal of Grid Computing*, 20(3):1–26, 2022.

[34] Prabhat Kumar, Randhir Kumar, Abhinav Kumar, A. Antony Franklin, and Alireza Jolfaei. Blockchain and deep learning empowered secure data sharing framework for softwarized uavs. In *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 770–775, 2022.

[35] Mike Ball. Utilizing embedded systems for uavs. https://www.unmannedsystemstechnology.com/2020/05/utilizing-embedded-systems-for-uavs/, Feb 2021.

[36] Mixing and actuators. https://docs.px4.io/v1.12/en/concept/mixing.html, Jun 2021.

[37] Jordan Robertson and Michael Riley. The big hack: How china used a tiny chip to infiltrate u.s. companies, Oct 2018.

[38] Kaiyuan Yang, Matthew Hicks, Qing Dong, Todd Austin, and Dennis Sylvester. A2: Analog malicious hardware. In *2016 IEEE symposium on security and privacy (SP)*, pages 18–37. IEEE, 2016.

[39] D Zammit. Intel x86s hide another cpu that can take over your machine.

[40] Stephen Hopkins, Carolyn Henry, Sikha Bagui, Amitabh Mishra, Ezhil Kalaimannan, and Caroline Sangeetha John. Applying a verified trusted computing base to cyber protect a vulnerable traffic control cyber-physical system. In *2020 SoutheastCon*, pages 1–8. IEEE, 2020.

[41] Adrian Garcia Badaracco. Advanced usage of scikeras wrappers. https://www.adriangb.com/scikeras/stable/advanced.html.

[42] Dionysios Sotiropoulos, Christos Giannoulis, and George A. Tsihrintzis. A comparative study of one-class classifiers in machine learning problems with extreme class imbalance. In *IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications*, pages 362–364, 2014.

[43] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[44] Control-fused intrusion detection system for uavs. https://sites.google.com/view/routine-aware-ids/home.

[45] Python (mavgen). https://mavlink.io/en/mavgen_python/.

[46] Hadhami Aouani and Yassine Ben Ayed. Emotion recognition in speech using mfcc with svm, dsvm and auto-encoder. In *2018 4th International conference on advanced technologies for signal and image processing (ATSIP)*, pages 1–5. IEEE, 2018.

[47] Linxi Zhang, Xuke Yan, and Di Ma. Accelerating in-vehicle network intrusion detection system using binarized neural network. Technical report, 2022.

[48] Mohammad Tehranipoor and Cliff Wang. *Introduction to hardware security and trust*. Springer Science & Business Media, 2011.

[49] Sudhanshu Arya, Jingda Yang, and Ying Wang. Towards the designing of low-latency sagin: Ground-to-uav communications over interference channel. *Drones*, 7(7):479, 2023.

A UAV MOVEMENT CONTROL VIA PWM

In Fig. 3, motors 1 and 3 rotate counterclockwise (CCW), while motors 2 and 4 rotate clockwise (CW). To control the UAV's motion, PWM signals are varied as follows:

Forward Movement (Pitch Control): Increase the PWM signal to motors 3 and 4 and decrease to motors 1 and 2.

$$\Delta\text{PWM}_{\text{forward}} = \text{PWM}_{3,4} - \text{PWM}_{1,2} \tag{17}$$

Backward Movement (Pitch Control): Increase the PWM signal to motors 1 and 2 and decrease to motors 3 and 4.

$$\Delta\text{PWM}_{\text{backward}} = \text{PWM}_{1,2} - \text{PWM}_{3,4} \tag{18}$$

Rightward Movement (Roll Control): Increase the PWM signal to motors 1 and 3 and decrease to motors 2 and 4.

$$\Delta\text{PWM}_{\text{right}} = \text{PWM}_{1,3} - \text{PWM}_{2,4} \tag{19}$$

Leftward Movement (Roll Control): Increase the PWM signal to motors 2 and 4 and decrease to motors 1 and 3.

$$\Delta\text{PWM}_{\text{left}} = \text{PWM}_{2,4} - \text{PWM}_{1,3} \tag{20}$$

Increase Altitude (Thrust Control): Increase the PWM signal equally to all motors.

$$\Delta\text{PWM}_{\text{up}} = \uparrow \text{PWM}_{1,2,3,4} \tag{21}$$

Decrease Altitude (Thrust Control): Decrease the PWM signal equally to all motors.

$$\Delta\text{PWM}_{\text{down}} = \downarrow \text{PWM}_{1,2,3,4} \tag{22}$$

Yaw Control (Rotation Control): For right (clockwise) rotation, increase PWM to motors 1 and 4 and decrease to motors 2 and 3.

$$\Delta\text{PWM}_{\text{yaw-right}} = (\uparrow \text{PWM}_{1,4}) - (\downarrow \text{PWM}_{2,3}) \tag{23}$$

For left (counterclockwise) rotation, increase PWM to motors 2 and 3 and decrease to motors 1 and 4.

$$\Delta\text{PWM}_{\text{yaw-left}} = (\uparrow \text{PWM}_{2,3}) - (\downarrow \text{PWM}_{1,4}) \tag{24}$$

In these equations, $\Delta\text{PWM}_{\text{movement}}$ represents the change in the PWM signal required for a specific movement. The symbol $\uparrow$ indicates an increase and $\downarrow$ indicates a decrease in the PWM signal's duty cycle. The magnitude of PWM adjustments depends on the desired movement intensity, quadcopter characteristics, and motor response.

B PWM OUTPUT CONSTRAINT CALCULATION

This function scales the input value according to the motor's configuration. Fig. 18 shows the code snippet in the mixer_module code that limits the PWM outputs to ensure they are not out of the specified band. This function, `output_limit_calc_single`, takes two inputs: the index `i` for the motor channel and the normalized control input value. It calculates the effective output for the motor channel by scaling the input value based on the motor's minimum and maximum values, which are stored in the arrays `_min_value` and `_max_value`, respectively. First, the function checks for invalid or disabled channels by verifying if the input value is finite. If the input value is not finite, the function returns the disarmed value for the motor channel, which is stored in the array `_disarmed_value`. Next, the function checks if the motor output should be reversed by examining the `_reverse_output_mask`. If the corresponding bit for the motor channel is set, the input value is multiplied by -1 to reverse its direction. The function then calculates the effective output by scaling the input value according to the motor's minimum and maximum values. This scaling ensures that the output value is within the valid range for the motor. Finally, the function uses the `math::constrain` function as a last line of defense to ensure that the calculated effective output is within the motor's valid range. The function returns the constrained effective output value so the PWM values are within range.

C DATASET FORMATION

Existing UAV datasets primarily feature camera images, lacking control data vital for IDS training. Hence, we developed a dataset with essential flight control attributes.

C.1 Flight Data Collection

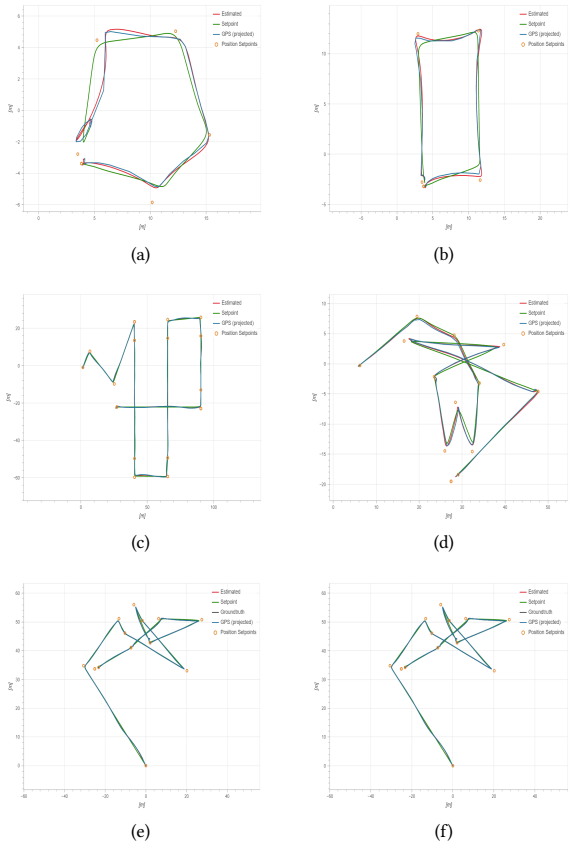
We experimented with various flights from a Pixhawk 2.4.8 UAV, closely emulating real-world missions, which include circular paths, polygonal paths, paths with multiple waypoints with increasing or decreasing speed and altitude, and survey missions in which the UAV flies through various obstacles. Throughout these complex flights, the flight controller logs the sensor, control, actuation, and other data, which will facilitate understanding the mapping of the flight control with PWM signals. After the flights, we download the flight logs and begin the preprocessing. The trajectories are shown in Appendix. D.

C.2 Data Preprocessing

Data preprocessing is pivotal for IDS efficacy because imbalanced datasets can skew classifications. We extracted *.ulg* files from our seven trajectories' logs via QGroundControl and converted these to *.csv* format, yielding 495 files—around 70 for each trajectory. These files, documenting varying features at distinct flight controller instances, present asynchronous data recordings. E.g., a sensor data logged at time t_1 might have its corresponding controller action recorded at t_{1+x}, where x represents the delay in timeslots. To provide a coherent view, we combined individual files per trajectory into a single file, encompassing timestamps and features.

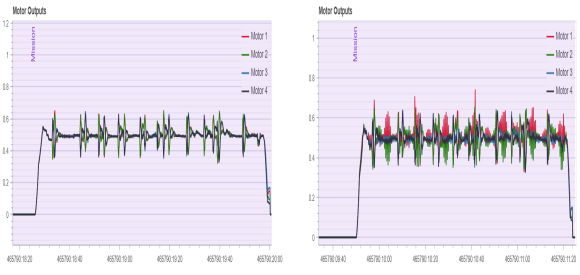
```
uint16_t MixingOutput::output_limit_calc_single(int i, float value) const
{
        // check for invalid / disabled channels
        if (!PX4_ISFINITE(value)) {
                return _disarmed_value[i];
        }

        if (_reverse_output_mask & (1 << i)) {
                value = -1.f * value;
        }

        uint16_t effective_output = value * (_max_value[i] - _min_value[i]) / 2
        + (_max_value[i] + _min_value[i]) / 2;

        // last line of defense against invalid inputs
        return math::constrain(effective_output, _min_value[i], _max_value[i]);
}
```

Figure 18: Snippet of the Output Constraint Calculation Function illustrating the method used to limit and calculate the effective output for a given channel.



(a) (b) (c) (d) (e) (f)

Figure 19: Data collection trajectories for ConFIDe: (a)-(b) polygonal paths, (e) complex route with varied altitudes/speeds, (f) survey path, and (g)-(h) jMAVsim simulations. While (g) is under normal conditions, (h) depicts PWM-manipulation attack effects. The trajectories remain consistent, but attack impacts are evident in Fig. 20(a) and 20(b).



(a) (b)

Figure 20: (a) shows the PWM outputs achieved under a normal trajectory as shown in Fig. 19(f), whereas (b) under a PWM-manipulative attack for the trajectory shown in Fig. 19(e).

Handling Missing Data and Duplicates: To address missing data, we employ interpolation techniques between t_1 and t_{1+x}, amalgamating data from all seven flights into a unified dataset. Upon further interpolation, duplicate removal, and discarding zero-value columns, our dataset encompasses 225,921 samples spanning 636 feature columns. We perform feature selection and select 33 pivotal ones. These essential features capture the heart of the UAV's control process: control inputs, positional accuracy, GPS data, altitude, orientation metrics, and core motion angles. The methodology considers the real-time nature of the system by selecting features (Appendix. E) that can be acquired in real-time from the UAV via MAVlink. This is important for developing a real-time IDS to detect attacks and anomalies during the UAV's flight, allowing immediate corrective actions. In contrast, data obtained from flight logs are only available after the mission is complete and may not be suitable for real-time IDS. Therefore, selecting features that can be obtained in real-time is necessary for developing an effective real-time IDS.

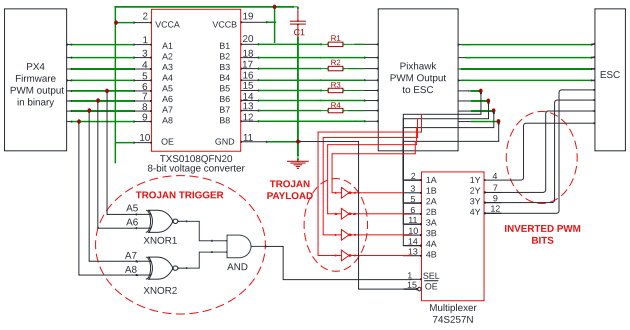
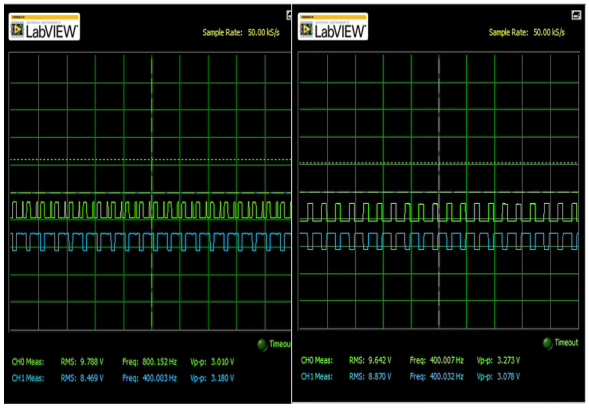


Figure 21: Implemented hardware Trojan circuit



(a) (b)

Figure 22: Duty cycle alteration in PWM due to hardware Trojan attack.

D FLIGHT TRAJECTORIES FOR DATA COLLECTION

We conducted various flight experiments using a Pixhawk 2.4.8 UAV, closely simulating real-world mission scenarios. These flights included circular, polygonal, multi-waypoint paths with varying speeds and altitudes and survey missions where the UAV navigated various obstacles. These trajectories are detailed in Fig. 19(a) through 19(f).

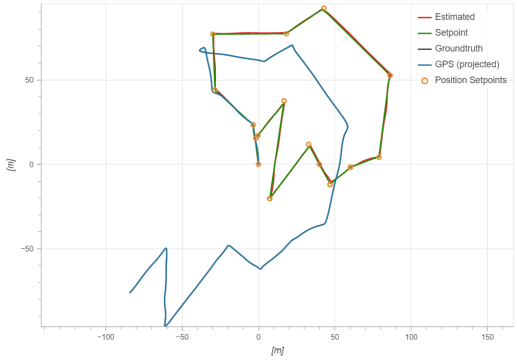


Figure 23: GPS Spoofing Attack

Table 4: List of Features and Their Descriptions for Confide

No.	Feature Name	Description
1	control[0]	Control signal for channel 1
2	control[1]	Control signal for channel 2
3	control[2]	Control signal for channel 3
4	control[3]	Control signal for channel 4
5	pos_horiz_accuracy	Horizontal position accuracy
6	pos_vert_accuracy	Vertical position accuracy
7	mag_test_ratio	Magnetic data test ratio
8	vel_test_ratio	Velocity data test ratio
9	pos_test_ratio	Positional data test ratio
10	hgt_test_ratio	Height data test ratio
11	lat	Latitude position
12	lon	Longitude position
13	alt	Altitude
14	x	Linear x-position
15	y	Linear y-position
16	z	Linear z-position
17	yaw	Drone's orientation
18	pwm[0]	PWM signal for channel 1
19	pwm[1]	PWM signal for channel 2
20	pwm[2]	PWM signal for channel 3
21	pwm[3]	PWM signal for channel 4
22	yawspeed	Yaw speed of drone
23	q[0]	Quaternion component 1
24	q[1]	Quaternion component 2
25	q[2]	Quaternion component 3
26	q[3]	Quaternion component 4
27	roll_body	Drone's body roll orientation
28	pitch_body	Drone's body pitch orientation
29	yaw_body	Drone's body yaw orientation
30	thrust_body[2]	Thrust related to body frame
31	roll	Global roll orientation
32	pitch	Global pitch orientation
33	yaw	Global yaw orientation
34	eph	Positional error in horizontal
35	epv	Positional error in vertical
36	output[0]	Output signal 1
37	output[1]	Output signal 2
38	output[2]	Output signal 3
39	output[3]	Output signal 4

E FEATURE SELECTION FOR CONFIDE

The features selected for *Confide* encompass a comprehensive set of parameters critical for assessing the drone's flight dynamics, orientation, and control. Features such as `control` signals provide insights into the immediate commands dispatched to the drone, ensuring that real-time decisions are made based on authentic and unaltered signals. Positional metrics, including `lat`, `lon`, `alt`, and linear positions (`x, y, z`), are essential to accurately track the drone's location and movement in 3D space. Quaternion orientation and body metrics give a nuanced perspective on the drone's orientation in three-dimensional space, which is crucial for maintaining stability during flight. Additionally, test ratios, such as `mag_test_ratio`, ensure the authenticity of various data streams, guarding against potential anomalies or intrusions. Lastly, `PWM` signals and output signals reveal the drone's motor control dynamics, a vital component for flight control and maneuvering. These features were selected to ensure a robust and holistic view of the drone's operation, making *ConFIDe* an effective tool for detecting and mitigating possible anomalies. The features are enlisted in Table. 4. ConFIDe's detailed feature selection, including quaternion components, yaw speed, and thrust_body[2], enhances its monitoring capabilities, vital for drone security in the rapidly evolving cyber threat landscape. These features critically track spatial orientation, aiding in detecting unauthorized intrusions or malfunctions. Additionally, error metrics like *eph* and *epv* bolster ConFIDe's precision, flagging even minor positional deviations. Such meticulous attention to detail is crucial for UAVs, as small errors can lead to significant navigational issues over time. ConFIDe thus plays a pivotal role in protecting operational integrity, ensuring airspace safety, and safeguarding ground assets. Lastly, incorporating MAVlink communication metrics, ConFIDe effectively interprets signal integrity and timing, critical for verifying command execution fidelity. Signal-to-noise ratio (SNR) measurements of GPS signals are also utilized, enhancing the detection of spoofing attempts by analyzing deviations from expected transmission profiles. These specific metrics further enhance ConFIDe's diagnostic capabilities, ensuring comprehensive surveillance over the UAV's communication and control systems.