Machine Learning-Based Intrusion Detection for Swarm of Unmanned Aerial Vehicles

Umair Ahmad Mughal, Samuel Chase Hassler, and Muhammad Ismail
Department of Computer Science, Tennessee Technological University, Cookeville, TN, USA
Emails: {uamughal42, schassler42, mismail}@tntech.edu

***Abstract*—Swarms of unmanned aerial vehicles (UAVs) are widely adopted in civilian and military applications. However, this cyber-physical system is threatened by cyber-attacks. Recently, machine learning-based intrusion detection systems have been successfully adopted to detect cyber-attacks. Yet, the following questions remain unanswered: (a) Can the fusion of cyber and physical features collected from the attacked UAV improve the detection performance? (b) Can the fusion of cyber and physical features collected from unattacked UAVs in the swarm help to detect the attack? (c) Can the fusion of cyber and physical features collected from all UAVs in the swarm (attacked and unattacked) improve the detection performance? To answer the aforementioned questions, and due to the absence of practical datasets, we develop a preliminary testbed of two UAVs flying in coordination. We launch a range of cyber-attacks on one of the UAVs including false data injection (FDI), denial-of-service (DoS), replay, and evil twin attacks. Then, we collect cyber and physical features from the UAVs under normal operation and attack conditions. Next, we develop a set of intrusion detection systems based on shallow and deep machine learning models including support vector machine (SVM), feedforward neural networks (FNN), recurrent neural networks (RNN), and convolutional neural networks (CNN). The developed models are trained using cyber-only, physical-only, and cyber-physical features collected from the attacked UAV, the unattacked UAV, and both UAVs in the swarm. The extensive studies carried out herein provide answers to the aforementioned questions and pave the way toward effective intrusion detection systems in UAV swarms.**

***Index Terms*—UAV swarms, cyber-attacks, machine learning, and intrusion detection systems.**

I. Introduction

Swarms of unmanned aerial vehicles (UAVs) have gained tremendous attention in recent years due to their diverse civilian and military applications. However, their wide adoption presents a considerable challenge due to the required systematic approach to balancing operational needs and safety/security concerns. For example, an adversary could potentially target UAV swarms causing them to malfunction and crash in urban cities. As such, several ongoing efforts have been taken to address UAV operational and security challenges. For instance, NASA has developed an unmanned traffic management (UTM) research platform that focuses on UAV operation within and beyond visual line-of-sight. Furthermore, the U.S. Department of Defense (DoD) is developing strategies to secure the operation of multiple/swarms of autonomous UAVs in hostile environments. Moreover, several research works focused on developing intrusion detection systems (IDSs) that ensure secure operations of UAV swarms against cyber-attacks. In the literature, several IDSs have been developed to detect cyber-attacks such as denial-of-service (DoS) [1], false data injection (FDI) [2], replay [3], hijacking [4], and spoofing attacks [5]. However, one common aspect among existing works is that they do not treat UAVs as cyber-physical systems, and hence, base the IDS on either physical/behavioral features (e.g., coordinates, speed, etc.) or cyber features (e.g., frame and packet information). Specifically, UAVs are cyber-physical systems that are equipped with sensors to collect measurements, radio to receive command signals from a ground control unit, and actuators to enforce physical behaviors. Hence, the existing strategies do not portray a complete picture that captures the cyber and physical features of the UAV system, and hence, the existing IDSs offer limited performance. Moreover, when considering UAV swarms, the existing IDSs do not exploit the coordination patterns within the swarm's cyber and physical data to enhance attack detection. In this context, the following open questions require further investigation:

- Can the fusion of cyber and physical features collected from attacked UAV improve the detection performance?
- Can cyber-physical fusion of features collected from unattacked UAVs in the swarm help to detect the attack?
- Can the fusion of cyber and physical features collected from all UAVs in the swarm (attacked and unattacked) improve the detection performance?

Answering the aforementioned questions is challenged by the fact that developing IDSs require datasets that reflect the system operation in normal and attack conditions. However, existing works do not provide access to cyber and physical datasets for a single or swarm of UAVs. This calls for developing a testbed and designing a data collection methodology. In addition, providing the IDS with numerous cyber and physical features may confuse the model, and hence, deteriorates the detection performance. This calls for a methodology to identify the most effective cyber and physical features that can improve detection performance. Finally, reaching a balance between computational complexity and detection performance requires exploring a wide range of detection strategies, optimizing their structure, and comparing their performance.

Consequently, we have carried out the following:

- We developed a testbed that consists of two UAVs (UAV1 and UAV2), an access point, a controller, a network

This work was supported by NSF award #2220346.

adapter, and data collection tools. To imitate the attacker's behavior, we launched four types of attacks on UAV1, namely, FDI, de-authentication DoS, replay, and evil twin attacks. We collected cyber and physical features using our testbed under both normal and attack conditions.

- We have investigated a range of IDSs using shallow and deep machine learning models that include support vector machines (SVM), feedforward neural networks (FNN), recurrent neural networks (RNN) with long-short term memory (LSTM) cells, and one-dimensional convolutional neural networks (1D-CNN). To provide answers to the aforementioned research questions, we explored nine different cases for each model. Specifically, for each model, we compare the development of IDS using cyber-only, physical-only, and cyber-physical data from UAV1 only, UAV2 only, and both UAVs. These cases are referred to as UAV1-cyber, UAV1-physical, UAV1-fused, UAV2-cyber, UAV2-physical, UAV2-fused, central-cyber, central-physical, and central-fused.
- We performed Shapley additive explanations (SHAP) analysis on the aforementioned models to identify the most effective features in each case. Moreover, we optimized the structure of each model by tuning their hyper-parameters using a random grid search.

Our extensive studies carried out in this paper provide answers to the aforementioned research questions and pave the way toward effective IDSs in swarms of UAVs.

The rest of this paper is organized as follows. Section II summarizes the related works and discusses their limitations. Section III presents the testbed, cyber-attacks, and data collection methodology. Section IV discusses the data pre-processing, machine learning models, and feature selection using SHAP analysis. Section V summarizes the experimental results and provides answers to the research questions. Conclusions are made in Section VI.

II. Related Works

This section summarizes the related works in attack detection in UAVs. We categorize the existing strategies as model-based and data-driven. We further classify data-driven IDSs as cyber-based or physical-based. Finally, we highlight the limitations of existing works to motivate our research.

A. Model-based IDSs

Zhao *et al.* [2] designed an FDI attack by solving an optimization problem with a constraint on UAV's energy and developed a detection strategy using the subspace coding theory. Xiao *et al.* [6] designed a sliding innovation sequences detector to identify the attacks on the UAV's sensor and actuator. The detector calculates the norm of the normalized innovation (residual) sequence within a time window and subsequently initiates an alarm when the computed value exceeds a predefined threshold. Mousavinejad *et al.* [7] designed two ellipsoid sets namely, the prediction set and estimation set, to detect replay and FDI attacks. Ye *et al.* [8] designed a summation detector to detect cyber attacks on a UAV by utilizing the summation of the innovations. Sedjelmaci *et al.* [9] presented a threat estimation model that decides, using estimated beliefs, the existence of threats within the system.

A common limitation with the aforementioned detectors is that they require accurate models to design the alarm threshold, which cannot be developed when certain system parameters are unknown. As a result, data-driven/machine learning-based attack detection methods have been adopted in recent years to overcome this limitation.

B. Data-Driven IDSs

Lee *et al.* [10] trained a support vector regression model and principle component analysis for anomaly detection in UAVs. Abbaspour *et al.* [11] designed an adaptive neural network to identify faults and detect FDI attacks on the UAV's sensor readings. Bozkut *et al.* [12] proposed a stochastic game framework between the attacker and the controller interaction and adopted linear temporal logic (LTL) to solve the planning problem using a model-free reinforcement learning strategy. Park *et al.* [13] designed stacked autoencoder to detect faults in UAV states. Praveena *et al.* [14] used Black Widow optimization to optimize reinforcement learning and designed a deep belief network to detect cyber attacks. The designed IDS in [14] is validated using the NSL-KDD Cup dataset, which incorporates cyber features irrelevant to UAVs' operation. All these works have examined an operation of a single UAV to develop an IDS model.

In a closely related work, Ahn *et al.* [15] presented a machine learning-based framework to detect abnormal behavior in UAV swarms. The author utilized unsupervised learning techniques and a deep neural network classifier, i.e., 1D-CNN, to differentiate anomalies from normal behavior. However, the author only used the physical data, i.e., features associated with the UAV's physical behavior, to train the IDS. Also, [15] do not consider attacks conducted on the UAV swarm. Khanapuri *et al.* [16] designed IDS using machine and deep learning techniques to detect FDI attacks in a multi-UAVs network. However, simulated datasets are used in [16] to train the IDS. Ouiazzane *et al.* [17] proposed an IDS for a swarm of UAVs operating in an ad-hoc communication network. A decision tree algorithm is used in [17] following a supervised learning approach. However, the IDS model in [17] was trained using the CICIDS2017 Intrusion Detection Evaluation Dataset, which includes only cyber features that are irrelevant to UAVs.

C. Summary and Limitations

Table I summarizes the related works. None of the existing works investigated cyber-physical fusion to detect attacks on UAVs. Instead, the existing works relied on either cyber or physical data. Moreover, none of the existing works that relied on cyber data involved real communication with an actual UAV to obtain benign and malicious data samples. Instead, all of the utilized datasets have been simulated or obtained from publicly available datasets that are irrelevant to UAVs. In addition, existing works have examined IDSs

TABLE I
SUMMARY OF THE RELATED WORKS

Ref.	Year	Attack Type	UAVs	Detection Mechanism	Dataset
[2]	2020	FDI	Single	Subspace coding theory	Model-based IDS
[6]	2022	FDI/DoS	Single	Sliding Innovation-Based Technique	Model-based IDS
[7]	2018	Replay/FDI	Single	Ellipsoidal filtering	Model-based IDS
[8]	2019	FDI	Single	Summation Detector	Model-based IDS
[9]	2016	FDI/DoS	Single	Estimation Belief approach	Model-based IDS
[10]	2019	Faults	Single	SVR with PCN	Actual Physical
[11]	2016	Faults	Single	Adaptive Neural Network	Simulated Physical
[12]	2021	FDI	Single	Model-free RL	Simulated Physical
[13]	2021	Faults	Single	Stacked Autoencoder	Actual Physical
[14]	2022	DoS	Single	Reinforcement Learning	Irrelevant Cyber
[15]	2019	Faults	Swarm	Convolutional neural network	Actual Physical
[16]	2022	FDI	Swarm	Fully Connected DNN	Simulated Physical
[17]	2020	Faults	Swarm	Decision Tree	Irrelevant Cyber

against either FDI or DoS attacks [10]–[12], [16]. Furthermore, the existing works that investigate the swarm of UAVs have not investigated whether using cyber and/or physical data from attacked and/or unattacked UAVs would help in detecting cyber-attacks. Hence, the existing literature lacks a comprehensive investigation on the resilience of machine learning-based attack detection strategies against a diverse set of cyber-attacks considering both cyber and physical features within a multi-UAV network (a swarm of UAVs).

III. Testbed and Data Collection

This section discusses the data generation methodology. As aforementioned, there is no publicly available dataset that includes both cyber and physical features collected from a swarm of UAVs under normal operation and attack conditions. Hence, we discuss herein the development of a testbed and its usage to collect benign and malicious datasets.

A. Testbed Setup

The testbed setup includes the following equipment: Two DJI Tello EDU drones [18], Sagemcom SAC2V2s WiFi access point [19], ALFA AWUS036ACH network adapter (antenna) [20], and two computers. The architecture of the testbed is shown in Fig. 1 and its components are detailed next:

- Computer-1 mimics a ground station that is used to monitor and control the swarm of UAVs. This computer has a number of Python scripts that are used to connect the legitimate operator to the UAVs, pass commands, and receive telemetry physical/behavioral data of the UAVs.
- The WiFi access point establishes a connection between the ground controller (Computer-1) and the two UAVs to provide smooth communication and control.
- Computer-2 along with the ALFA AWUS036ACH antenna have dual roles. This computer is running Kali Linux that runs software such as Aircrack-ng, Tcpdump, and Wireshark. First, Computer-2 and the antenna are used to collect the cyber features when operating in the monitoring mode. Also, Computer-2 and the antenna can mimic the attacker and launch cyber-attacks on UAV1.

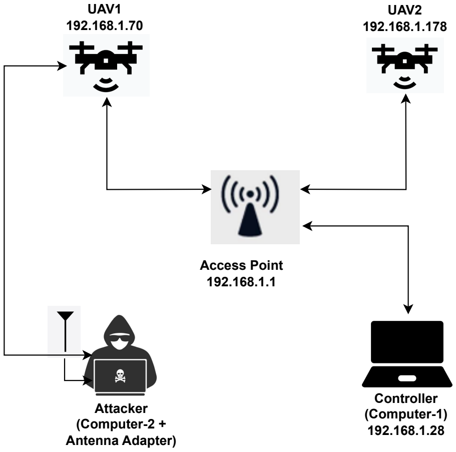


Fig. 1. Illustration of the testbed under consideration.

The data collection process was executed in two phases. The first involved UAV flights under normal operation, which constructed the benign cyber-physical dataset. The second phase included UAV flights when UAV1 is under attack, hence, constructing the malicious cyber-physical dataset. The following subsections describe the data collection methodology.

B. Benign Data Collection

The benign data collection phase starts with the initiation of flight by the Tello EDU drones. The controller (Computer-1) begins to establish communication with the UAVs through the access point, thereby receiving physical measurement data. This data includes a range of sensor readings, such as Inertial Measurement Units (IMU), thermometers, Time-of-Flight (ToF) sensors, and barometers, among others. These readings constitute the physical features of the UAVs.

Simultaneously, a second computer (Computer-2) equipped with Kali Linux and an Alfa adapter operates in the monitoring mode. This configuration enables the capture and analysis of the WiFi traffic between the Tello EDU drones, access points, and the controller (Computer-1), using the Airodump-ng tool. This tool captures the WiFi traffic and identifies the access points and their Basic Service Set Identifier (BSSID). From each BSSID, various characteristics are collected such as the channel, authentication protocol, encryption algorithms, cipher, and the Extended Service Set Identification (ESSID). Also, numerous cyber features are collected from the UAVs.

To collect the cyber and physical datasets, several flight missions of varying complexity were conducted. These include back-and-forth missions, square missions, search missions, spiral missions, and rectangle missions. The data collection exercise involved 20 benign flights, in addition to 20 flights where UAV1 was attacked, totaling 40 flights. Hence, 8 flights were conducted for each mission, half under normal operating conditions and half under attack.

C. Malicious Data Collection

In this subsection, we discuss various cyber-attacks that were launched on UAV1. The collected cyber and physical datasets from UAV1 and UAV2 during this stage constitute the malicious dataset.

1) De-authentication Attack: We used the Aircrack-ng software suite and Wireshark to execute the attack. To initiate the attack, we first put the ALFA network adapter into monitor mode using the command `airmon-ng start wlan0`, where wlan0 represents the wireless interface. In the monitor mode, the network adapter captures all the traffic over the wireless network. Subsequently, we execute `sudo airodump-ng wlan0` command to list all the wireless networks, including their BSSIDs (MAC addresses), SSID, signal strength, channel numbers, and the number of sent data packets. Once we identified the network of the target UAV (UAV1), we used the command `aireplay-ng --deauth 100 -a [AP mac] -c [target mac] wlan0` to send de-authentication packets to the target (UAV1). In this command, 100 is the number of de-authentication packets, AP mac is the MAC address of the access point that the Tello is connected to, and target mac is the MAC address of the target UAV. Executing this attack, the target Tello (UAV1) does not receive commands from the controller (Computer-1), causing it to hover in place until the connection is re-established.

2) Replay attack: We use the same software suite to launch this attack. The first step is to capture the packets transmitted between the target (UAV1) and the controller (Computer-1). So, we put the ALFA network adapter into monitor mode using the command `airmon-ng start wlan0` and initiate the capturing process with the command `airodump-ng wlan0`. During this phase, we ensure that the target UAV is active and receiving commands from the controller. The goal is to capture the legitimate commands between the controller and the target UAV for future replay. We save the captured packets into a PCAP file and use the Wireshark filter to examine and separate the target UAV1 command packets. Using the Aireplay-ng, we executed the command `aireplay-ng --inject replay -r capture.pcap wlan0`, where replay refers to the replay mode and capture.pcap is the PCAP file containing the captured packets. This command replays the captured packets, fooling the target UAV into executing the commands again. By replaying command packets, the target UAV showed abnormal behavior and was not able to pursue its normal mission.

3) Evil Twin Attack: We used the Aircrack-ng and Airgeddon tool kits to execute the attack. The first step in this attack is to establish an Evil Twin access point by creating a rogue WiFi network that mimics the legitimate Service Set Identifier (SSID) of the target UAV network. To obtain the SSID, we executed the command `sudo airodump-ng wlan0`, where wlan0 is the network interface in monitor mode. This command lists the BSSIDs (MAC addresses), SSID, etc. After acquiring the legitimate SSID, we create an Evil Twin access point using the Airgeddon interface. Airgeddon is a suite of tools that automates the process by setting up a new rogue access point with the same SSID as the targeted UAV network. To ensure that the targeted UAV (UAV1) connects to our rogue network, we need to make our signal stronger than the legitimate network's signal. For this purpose, we employed the command `iwconfig wlan0 txpower 30`, which sets the transmission power of our network adapter to 30 dBm. Additionally, we carried out a de-authentication attack on the target UAV for a short duration to disconnect it from the legitimate access point, which results in a connection to our rogue network. Once the target UAV connects to our rogue access point, it becomes a victim of a man-in-the-middle (MitM) attack. It allows monitoring, intercepting, stealing, or manipulating the legitimate data.

4) False Data Injection Attack: We used the Aircrack-ng suite, Scapy, and custom Python scripts in the execution of this attack. To inject stealthy readings and commands, we calculated the state-space matrices (A, B, C) of UAV1 using Dynamic Mode Decomposition (DMD) and Matlab's System Identification Toolbox. We focused on three states of the UAV, namely, roll, pitch, and yaw. Next, we implemented a custom Python script to design stealthy attack vectors. Specifically, we modified the roll, pitch, and yaw measurements according to the equation $y = y + \gamma$, where y represents the original sensor measurement and γ is the attack vector for sensor measurements. Similarly, we modified the control signal using $u = u + \eta$, where u represents the original control signal and η represents the attack vector for a control signal. We calculated stealthy attack vectors γ and η and the state-space matrices (A, B, C) such that the residual error is minimized. To inject the false data, we created a Python script using the Scapy library to craft and send the necessary packets to the UAV's IP address and controller port, namely, 192.168.1.70 and 8890, respectively. As a result, the target UAV received false readings, leading to incorrect state estimations, and consequently, improper control actions. The drone's behavior became irregular, and it failed to maintain its position within the swarm.

IV. Designing Intrusion Detection system

In this section, we develop IDSs for a swarm of UAVs. First, we discuss the pre-processing of the data. Then, we discuss the machine learning models, hyper-parameter optimization, and feature selection using SHAP analysis.

A. Data Pre-processing

As aforementioned, we collected cyber and physical features for various missions under normal operations and cyber-attacks. The recorded data samples were asynchronous as both features were received at different rates, making fusing them challenging. To address this issue, we interpolate the lower-rate data points according to the timestamp of the higher-rate data points. Initially, we recorded the cyber data in PCAP files, filter the data between each UAV, access point, and controller, and export them to JSON format. These JSON data files are then linked to their respective UAV's physical data files via the UAV's IP address. The physical data files were recorded at the controller and stored in CSV files. Next, we extracted the cyber features from the JSON files using a custom Python script that integrates them into pandas' data frames. Concurrently, physical features are also incorporated into the pandas' data frames. We extracted a total of 40 cyber features and 20 physical features, which are listed in Table II.

In the following, we provide a brief explanation for each feature listed in Table II. For the 40 cyber features: frame.number represents the sequential number of each captured frame, frame.len length of the captured frame in bytes, frame.protocols indicate the protocols of the transmission of the frame, wlan.duration is the time to be transmitted, whereas wlan.ra, wlan.ta, wlan.da, and wlan.sa represent the receiver, transmitter, destination, and source MAC addresses, respectively, wlan.bssid is the Basic Service Set Identifier, i.e., MAC address of the Access Point, wlan.frag is the fragment number, wlan.seq is the sequence number for packets, wlan.fc.type and wlan.fc.subtype refer to the type and subtype of the frame control field, wlan.flags indicates the status flags, wlan.fcs stands for the frame-checking sequence, ip.len is a IP packet's length, whereas udp.length is the length of the User Datagram Protocol data, data.len is the length of the actual data being transmitted, signal_strength (dbm) is the signal strength, ip.id is the identification field of the IP header, wlan_radio.noise (dbm), wlan_radio.SNR (db), and wlan_radio.preamble are all linked with radio data and indicate noise in decibels milliwatts, Signal-to-Noise Ratio, and the preamble of the radio frame, respectively, ip.src and ip.dst indicate the source and destination IP addresses, respectively, wlan.fcs indicates frame checking sequence, wlan.fcs.status shows status indicating if errors detected, wlan.qos, wlan.qos.priority, and wlan.qos.ack all concern the quality-of-service and its priority and acknowledgment, wlan.ccmp.extiv is a counter mode with cipher block chaining message authentication code protocol (ccmp) extended initialization vector, wlan.wep.key is a wired equivalent privacy (wep) security protocol, radiotap.hdr_length is header length and radiotap.antenna_signal is the signal strength, radiotap.signal_quality shows signal quality, radiotap.channel.flags.ofdm is for the Orthogonal Frequency-Division Multiplexing, radiotap.channel.flags.cck is a complementary code keying (a modulation scheme), wlan_radio.datarate and wlan_radio.frequency are the rate and frequency at which data is being transmitted. For the 20 physical features: x, y, and z represent the UAV's position coordinates, pitch, roll, and yaw denote the UAV's orientation, x_speed (vgx), y_speed (vgy), and z_speed (vgz) indicate the speed along the x, y, and z axes, respectively, templ and temph are the low and high temperatures of the main board in celsius, tof is the time-of-flight, h is the height relative to the take-off position, the barometer shows the height measured by the barometer, agx, agy, and agz show the acceleration along x, y, z axis, respectively. The timestamp_c and timestamp_p are the collected cyber and physical data timestamps, respectively.

TABLE II
RAW EXTRACTED CYBER AND PHYSICAL FEATURES

Cyber	Cyber	Physical
frame.number	ip.id	timestamp_p
frame.len	wlan_radio.noise (dbm)	x
frame.protocols	wlan_radio.SNR (db)	y
wlan.duration	wlan_radio.preamble	z
wlan.ra	ip.src	pitch
wlan.ta	ip.dst	roll
wlan.da	wlan.fcs	yaw
wlan.sa	wlan.fcs.status	x_speed (vgx)
wlan.bssid	wlan.qos	y_speed (vgy)
wlan.frag	wlan.qos.priority	z_speed (vgz)
wlan.seq	wlan.qos.ack	templ
wlan.fc.type	wlan.ccmp.extiv	temph
wlan.fc.subtype	wlan.wep.key	tof
wlan.flags	radiotap.hdr_length	height
wlan.fcs_len	radiotap.antenna_signal	battery
ip.len	radiotap.signal_quality	barometer
udp.length	radiotap.channel.flags.ofdm	flight_time
data.len	radiotap.channel.flags.cck	agx (cm/s^2)
timestamp_c	wlan_radio.datarate	agy (cm/s^2)
signal_strength (dbm)	wlan_radio.frequency	agz (cm/s^2)

To synchronize the cyber and physical features, a ratio is calculated between the cyber and physical data frames. This enables us to assign an equivalent interval to interpolate the data within the low-rate data points (physical data) to synchronize them with the high-rate data points (cyber data). As a result of this synchronization process, some samples in the high-rate data (cyber) towards the end of the flight were discarded as no corresponding physical data were recorded. The fused cyber-physical data was then standardized utilizing a Min-Max scaler, labeled as either benign or attacked, stored in a data frame list, and the process is repeated for the next consecutive data files. We compiled a total of 160 data files, 80 benign and 80 malicious data files. These are cyber and physical data files for both UAVs under normal and attacked conditions. The number of data samples for cyber, physical, and cyber-physical (fused) features for each UAV and the combined (central) features are summarized in Table III.

We would like to highlight that some of the collected features were not contributing to the learning process of the machine learning models, and instead, were increasing their complexity. Hence, we removed these features while training the models, which include the battery, barometer, frame.number, wlan.bssid, and the cyber and physical timestamps. We used a $3:1$ train to test split ratio for the datasets and 5-fold cross-validation within the train set.

TABLE III
SIZE OF DATA POINTS/SAMPLES FOR EACH DATASET

Datasets	Cyber Samples	Physical Samples	Fused Samples
UAV1	$71,352$	$31,895$	$65,802$
UAV2	$34,382$	$22,035$	$30,242$
Central	$105,734$	$53,930$	$96,044$

TABLE IV
INVESTIGATED IDS STRATEGIES

Datasets	Cyber	Physical	Fused
UAV1	UAV1_cyber	UAV1_physical	UAV1_fused
UAV2	UAV2_cyber	UAV2_physical	UAV2_fused
Central	Central_cyber	Central_physical	Central_fused

B. Investigated Models for IDSs

To develop an effective IDS, a range of machine-learning models have been investigated including shallow and deep models and temporal and non-temporal models. The investigated models are summarized next.

1) Shallow Models: An SVM model is adopted herein. It represents a supervised model that is trained and tested on benign and malicious data. Also, it is a non-temporal model that does not exploit the temporal correlation within the collected cyber and physical time-series data.

2) Deep Models: These are based on stacked neural networks that are capable of learning complex patterns embedded in the data. They represent supervised models that are trained and tested on benign and malicious data.

a) FNN: This model stacks dense hidden layers that process data in a feedforward manner, i.e., it does not leverage the temporal correlation present in the time-series data.

b) LSTM-RNN: This model is based on the LSTM variant of the RNN. It is enriched with feedback connections that enable the model to capture temporal correlations within the time-series data. The LSTM cells overcome the vanishing-exploding gradient problem of the RNN by retaining values over specific time intervals, controlled by input, output, and forget gates that manage the information flow.

c) 1D-CNN: This model also captures the temporal correlation within the time-series data using convolutional filters traversing through the sequential data.

Overall, we investigated 9 IDS strategies for each of the aforementioned models. Each strategy relies on specific datasets, which are summarized in Table IV. Specifically, cyber-only models are trained and tested using only the collected cyber features. These include UAV1_cyber models trained and tested using cyber features from the attacked UAV1, UAV2_cyber models trained and tested using cyber features from the unattacked UAV2, and central_cyber models trained and tested using cyber features from the attacked UAV1 and unattacked UAV2. Similarly, the physical-only models are trained and tested using only the collected physical features. Finally, the fused (cyber-physical) models are trained and tested using both the cyber and physical features. One can think of the UAV1 and UAV2 models as IDSs deployed on the UAV while the central models as IDSs deployed at the ground controller (Computer-1).

TABLE V
OPTIMAL HYPER-PARAMETERS

Detector	$\mathcal{H}$	**UAV1 Cyber**	**UAV1 Physical**	**UAV1 Fused**	**Central Fused**
SVM	C	5	8	10	10
	G	0.1	0.25	0.4	0.2
	K	Linear	Linear	Linear	Poly
FNN	L	4	4	5	5
	N	128	128	256	256
	D	0.2	0	0.5	0.5
	O	Adam	Adam	Adam	Adam
	A	ReLu	tanh	tanh	tanh
LSTM	L	3	3	5	5
	N	256	256	256	512
	D	0.5	0.7	0.7	0.8
	O	Adam	Adam	Adam	Adam
	A	ReLu	ReLu	ReLu	ReLu
	W	glorot	glorot	glorot	he-uniform
1D-CNN	L	2	2	2	2
	N	64	64	128	128
	D	0.5	0.5	0.8	0.5
	O	Adam	Adam	Adam	Adam
	A	tanh	tanh	tanh	ReLu
	S	3	3	3	5

C. Optimal Hyper-parameters

In order to optimize the structure of the aforementioned models, we executed a sequential grid search to fine-tune each model's hyper-parameters. The optimal hyper-parameters $\mathcal{H}$ offering the highest detection rate during the validation phase were selected. These optimal hyper-parameter values were selected from a predefined search space $\mathcal{P}$ as follows: Number of layer $\mathcal{L} = \{2, 3, 4, 5, 6, 7, 8\}$, number of neurons/cells $\mathcal{N} = \{64, 128, 256, 512\}$, dropout rate $\mathcal{D} = \{0, 0.4, 0.6, 0.8\}$, optimizer $\mathcal{O}$ = {adam, SGD, Adamax}, and activation function $\mathcal{A}$ = {ReLu, tanh, Sigmoid, softmax}, and weight initialization $\mathcal{W}$ = {he_uniform, glorot_uniform, glorot_normal}. Table IV-C lists the optimal hyper-parameters of some detection strategies. For the rest of the strategies, the hyper-parameters remain the same with slightly different weight initialization. For 1D-CNN, S is the kernel size of the convolutional layers, with a max-pooling layer with a pool size of 2. In the case of LSTM, we also included the l_1 and l_2 regularization layer with an $l_1 = 0.0001$ and $l_2 = 0.001$. Exponential learning with a rate of 0.0001 is adopted for the Adam optimizer. For SVM, C is the regularization, G is gamma, and K is the kernel.

D. SHAP Analysis

In addition to the grid search, we perform SHAP analysis for each detection strategy. SHAP analysis uses game-theoretically optimal Shapley values to interpret the decision-making process of various machine learning models. We use SHAP as a feature selection mechanism, to select the most significant features and neglect the less important features for each developed detection model.

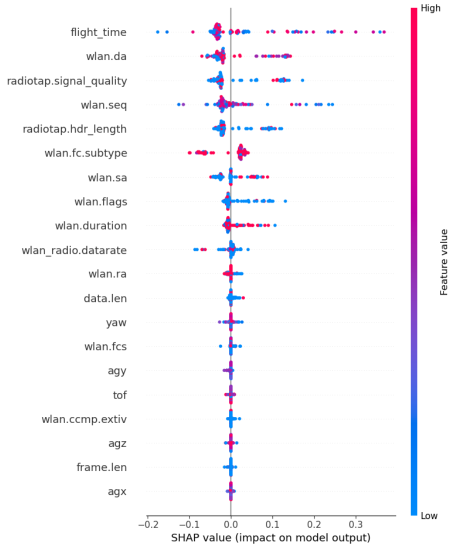


Fig. 2. Top SHAP values for SVM model trained on central_fused dataset

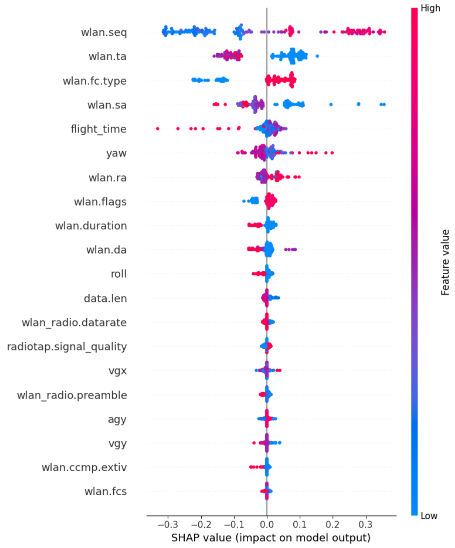


Fig. 3. Top SHAP values for CNN model trained on central_fused dataset

V. Experimental Results

This section outlines the detection performance of the IDSs. For each machine learning model (SVM, FNN, LSTM, and CNN), nine versions are trained and tested. These versions include models trained on cyber-only, physical-only, and cyber-physical (fused) datasets for each UAV and central node, as described in Table IV. First, we show sample results of the SHAP analysis to specify the most effective features. Next, we present sample learning curves to study the learning process. Finally, we compare the detection performance metrics across the developed models to answer the research questions.

A. Evaluation Metrics

The IDS performance is described in terms of accuracy, precision, recall, F1 score, and area under the curve (AUC) of the receiver operating characteristics (ROC). Accuracy describes the percentage of samples that have been classified correctly, which is given by

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{1}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. On the other hand, precision specifies the proportion of correctly identified positive samples out of the total predicted positives, which is given by

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2}$$

Furthermore, recall shows the proportion of accurately identified positive samples out of all actual positive samples, which is expressed as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3}$$

In scenarios with imbalanced data distributions, the precision and recall metrics provide more valuable insights compared to accuracy. The F1 score harmonizes precision and recall metrics, providing an effective analytical measure. It is calculated as the harmonic mean of Precision and Recall, as follows

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{4}$$

B. Result of the SHAP Analysis

We perform the SHAP analysis for each model using 100 random samples across all datasets and calculated their respective SHAP values. Due to space limitation, we are showing the SHAP values for the shallow model (SVM) and one of the best deep models (CNN) across the central_fused dataset in Fig. 2 and Fig. 3. We show only the top 20 cyber and physical features which affected the models' performance the most. The most effective cyber and physical features obtained from the SHAP analysis are those that are used in the final training and testing of the IDS models, whose performance results are summarized next.

C. Learning Curves

The learning curves show the F1-score for each model evaluated on training and validation datasets. These are used to ensure that models are not under or over-fitted. In total, we have 36 learning curves, and due to space limitations, we show only the learning curves for one of the best deep models, namely, the LSTM-RNN trained on cyber-only, physical-only, and cyber-physical (fused) data of the central access point (i.e., combining datasets from both attacked UAV1 and unattacked UAV2). Using the features specified by SHAP analysis and the optimal hyper-parameters specified by the grid search, we can see that the behavior of the F1 score on the training and validation data shows the model's learning and generalization abilities. It should be noted that we adopted the patience 10 strategy, i.e., the model training was stopped if the validation loss stay less than 0.01 for the consecutive 10 epochs.

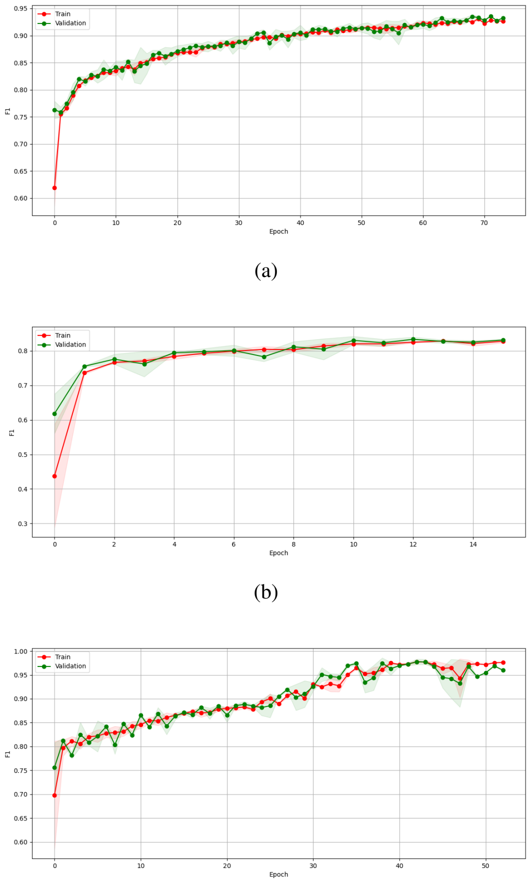
D. Performance Results and Discussions

The performance results are shown in Fig. 5, Fig. 6, and Fig. 7. The following observations can be made:

- Overall, deep learning-based detection models (i.e., FNN, LSTM-RNN, and CNN) offer better detection performance compared with the shallow detection model (i.e., SVM). Moreover, IDS models that exploit the correlation within the data (i.e., LSTM-RNN and CNN) offer better detection results compared with the other models (i.e., SVM and FNN). The improvement in detection performance is significant, up to $5-13\%$ in the F1 score when cyber features are considered (Fig. 5), up to $12-16\%$ in the F1 score when physical features are considered (Fig. 6), and up to $6-9\%$ in the F1 score when cyber-physical features are considered (Fig. 7).
- The considered attacks can be detected using cyber data or physical data. However, better detection results are attained when cyber data are considered. Comparing Fig. 5 with Fig. 6, we can see that physical features can detect attacks with F1 scores up to the level of 83% while cyber features can detect attacks with F1 score up to the level of 94%. Hence, cyber features offer an improvement in the F1 score up to 11% compared with the physical features.
- Observing features from the unattacked UAV (UAV2) is useful in detecting the attack on UAV1. For instance, the F1 score for detecting an attack on UAV1 by observing cyber and physical features from UAV2 is up to 85% and 79%, respectively. While these values are 7% and 2% less than observing corresponding features from the attacked UAV1, the detection performance based on features from UAV2 is high. This detection is possible because the two UAVs achieve their missions in coordination, and hence when one UAV is attacked (UAV1), this reflects on the physical and cyber behavior of the other unattacked UAV (UAV2). This is useful in case the attacker tampers the IDS on UAV1 so it does not flag an alarm, the unattacked UAV2 still can detect and report this attack on UAV1.

From Fig. 5 and Fig. 6, we observe that cyber features outperform physical features in detection performance. Also, we observe that features collected from UAV1 outperform features collected from UAV2 in detection performance. Hence, if no fusion is considered, cyber features from UAV1 offer the best detection performance. So, we compare in Table VI the percentage improvement in the detection performance of IDSs based on fused models compared with IDS models based on cyber-only features from UAV1. The considered fused models are cyber-physical fusion from UAV1, fusing cyber features from UAV1 and UAV2 in the central node (ground controller), and fusing cyber-physical features from UAV1 and UAV2 in the central node. The following observations can be made:

- Fusing the cyber features from UAV1 and UAV2 at the central node offers 2.6% improvement in the F1 score. This is because the IDS considers not only the patterns from individual UAVs but also the coordination carried out between the two UAVs while making its decision.



(a)

(b)

(c)

Fig. 4. Learning curves for the LSTM model trained on (a) Central_cyber, (b) Central_physical, and (b) Central_fused datasets.

- Cyber-physical fusion helps to improve detection performance. This is because UAVs are cyber-physical systems, and hence, observing features from both domains (cyber and physical) provides a complete picture of the system and results in improved detection results. Considering cyber-physical fusion only from UAV1 leads to improvement up to 2% in the F1 score (with absolute detection performance up to 94.5% in the F1 score in Fig. 7). The improvement is up to 4% when fusing cyber and physical features at the central node (with absolute detection performance up to 96.3% in the F1 score in Fig. 7).

VI. Conclusions

In this paper, we investigated the problem of designing IDS for a UAV swarm that can detect FDI, DoS, replay, and evil twin attacks. The considered system involved two UAVs with only one UAV under attack. The following conclusions can be made: (a) the fusion of cyber and physical features collected from the attacked UAV improves the detection performance up to 2% in the F1 score, (b) the fusion of cyber and physical features collected from the unattacked UAV offers a high

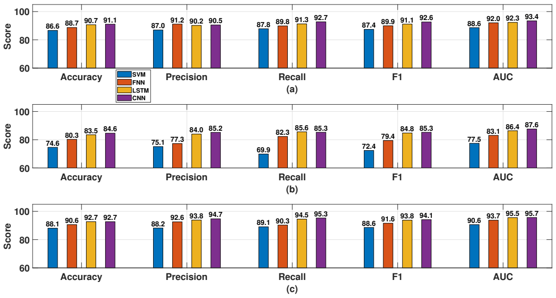


Fig. 5. Detection Performance of IDS (a) UAV1_cyber, (b) UAV2_cyber, (c) Central_cyber.

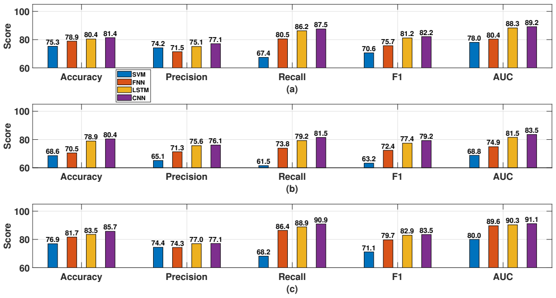


Fig. 6. Detection Performance of IDS (a) UAV1_physical, (b) UAV2_physical, (c) Central_physical.

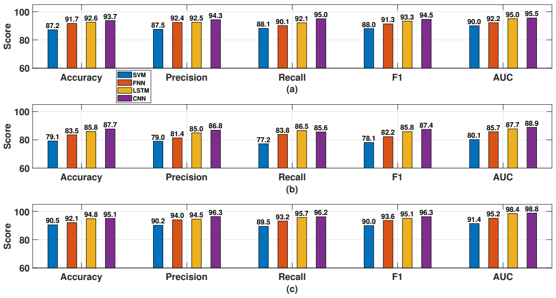


Fig. 7. Detection Performance of IDS (a) UAV1_fused, (b) UAV2_fused, (c) Central_fused.

detection performance with an F1 score up to 87.4%, (c) the fusion of cyber and physical features from both UAVs at the ground control station offers the best detection performance with an F1 score of 96.3%.

TABLE VI
IMPROVEMENT OF DIFFERENT IDS MODELS WITH RESPECT TO UAV1 CYBER-ONLY

Model	Dataset	Accuracy	Precision	Recall	F1	AUC
SVM	UAV1_fused	0.52%	0.56%	0.28%	0.63%	1.40%
SVM	Central_cyber	1.43%	1.26%	1.22%	1.22%	1.95%
SVM	Central_fused	3.87%	3.26%	1.61%	2.62%	2.80%
FNN	UAV1_fused	2.95%	1.24%	0.29%	1.37%	0.17%
FNN	Central_cyber	1.88%	1.43%	0.48%	1.64%	1.69%
FNN	Central_fused	3.35%	2.90%	3.40%	3.69%	3.15%
LSTM	UAV1_fused	1.87%	2.33%	0.85%	2.15%	2.71%
LSTM	Central_cyber	1.99%	3.67%	3.24%	2.61%	3.22%
LSTM	Central_fused	4.14%	4.36%	4.48%	3.99%	6.11%
1D-CNN	UAV1_fused	2.64%	3.83%	2.27%	1.90%	2.09%
1D-CNN	Central_cyber	1.59%	4.21%	2.57%	1.50%	2.26%
1D-CNN	Central_fused	3.98%	5.85%	3.47%	3.65%	5.42%

REFERENCES

[1] Z. Lian, P. Shi, C.-C. Lim, and X. Yuan, "Fuzzy-model-based lateral control for networked autonomous vehicle systems under hybrid cyber-attacks," *IEEE Transactions on Cybernetics*, 2022.

[2] Z. Zhao, Y. Huang, Z. Zhen, and Y. Li, "Data-driven false data-injection attack design and detection in cyber-physical systems," *IEEE Transactions on Cybernetics*, vol. 51, no. 12, pp. 6179–6187, 2020.

[3] B. Chen, D. W. Ho, G. Hu, and L. Yu, "Secure fusion estimation for bandwidth constrained cyber-physical systems under replay attacks," *IEEE transactions on cybernetics*, vol. 48, no. 6, pp. 1862–1876, 2017.

[4] O. Westerlund and R. Asif, "Drone hacking with raspberry-pi 3 and wifi pineapple: Security and privacy threats for the internet-of-things," in *2019 1st International Conference on UVS*. IEEE, 2019, pp. 1–10.

[5] J. Noh and et al., "Tractor beam: Safe-hijacking of consumer drones with adaptive GPS spoofing," *ACM Transactions on Privacy and Security (TOPS)*, vol. 22, no. 2, pp. 1–26, 2019.

[6] J. Xiao and M. Feroskhan, "Cyber attack detection and isolation for a quadrotor UAV with modified sliding innovation sequences," *IEEE Trans Vehicular Technology*, vol. 71, no. 7, pp. 7202–7214, 2022.

[7] E. Mousavinejad, F. Yang, Q.-L. Han, and L. Vlacic, "A novel cyber attack detection method in networked control systems," *IEEE transactions on cybernetics*, vol. 48, no. 11, pp. 3254–3264, 2018.

[8] D. Ye and T.-Y. Zhang, "Summation detector for false data-injection attack in cyber-physical systems," *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2338–2345, 2019.

[9] H. Sedjelmaci, S. M. Senouci, and M.-A. Messous, "How to detect cyber-attacks in unmanned aerial vehicles network?" in *2016 IEEE GLOBECOM*. IEEE, 2016, pp. 1–6.

[10] H. Lee and et al., "Anomaly detection of aircraft system using kernel-based learning algorithm," in *AIAA Scitech 2019 Forum*, 2019, p. 1224.

[11] A. Abbaspour, K. K. Yen, S. Noei, and A. Sargolzaei, "Detection of fault data injection attack on UAV using adaptive neural network," *Procedia computer science*, vol. 95, pp. 193–200, 2016.

[12] A. K. Bozkurt, Y. Wang, and M. Pajic, "Secure planning against stealthy attacks via model-free reinforcement learning," in *2021 ICRA*. IEEE, 2021, pp. 10 656–10 662.

[13] K. H. Park, E. Park, and H. K. Kim, "Unsupervised fault detection on unmanned aerial vehicles: Encoding and thresholding approach," *Sensors*, vol. 21, no. 6, p. 2208, 2021.

[14] V. Praveena and et al., "Optimal deep reinforcement learning for intrusion detection in UAVs," *Cmc-Computers Materials & Continua*, vol. 70, no. 2, pp. 2639–2653, 2022.

[15] H. Ahn, H.-L. Choi, M. Kang, and S. Moon, "Learning-based anomaly detection and monitoring for swarm drone flights," *Applied Sciences*, vol. 9, no. 24, p. 5477, 2019.

[16] E. M. Khanapuri, R. Sharma, and K. Brink, "Learning-based detection of stealthy false data injection attack applied to cooperative localization problem," in *AIAA SCITECH 2022 Forum*, 2022, p. 2543.

[17] S. Ouiazzane and et al., "Towards a multi-agent based network intrusion detection system for a fleet of drones," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 10, 2020.

[18] DJI, "Tello edu." [Online]. Available: https://www.ryzerobotics.com/tello-edu

[19] Sagemcom, "Model sac2v2s." [Online]. Available: https://www.amazon.com/Sagemcom-802-11ac-Wireless-Connectivity-Required/dp/B082F2PVYT

[20] ALFA Network, "Awus036ach." [Online]. Available: https://www.alfa.com.tw/products/awus036ach?variant=36473965871176