Graph Neural Network-Based Intrusion Detection System for a Swarm of UAVs

Umair Ahmad Mughal*, Rachad Atat†, and Muhammad Ismail‡
*School of Computer Science and Information Systems, Northwest Missouri State University, MO, USA
†Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon
‡Cybersecurity Education, Research, and Outreach Center (CEROC) and Department of Computer Science, Tennessee Technological University, Cookeville, TN, USA
Emails: umughal@nwmissouri.edu, rachad.atat@lau.edu.lb, and mismail@tntech.edu

***Abstract*—Unmanned aerial vehicle (UAV) swarms present significant potential in both civil and military applications, yet the security of swarm communications remains a critical challenge. While machine learning-based intrusion detection systems (IDS) have advanced, their effectiveness is often hindered by the reliance on simulated or irrelevant datasets that do not adequately capture the unique characteristics of UAV swarm communications. Furthermore, existing IDS have predominantly focused on temporal information, overlooking the potential of spatial relationships within the UAV network. To address these limitations, this research establishes a testbed of six UAVs forming an hexagonal graph where each UAV acts as a node and communicates with its immediate neighbors. We then execute various cyber-attacks such as false data injection, evil twin, replay, and denial-of-service attacks on each of the UAVs in the swarm. This allows us to collect spatial and temporal data under normal operations and attack conditions. We propose a graph neural network (GNN)-based IDS that exploits spatial and temporal information patterns. This research seeks to answer the following question: Can leveraging the spatial relationships within a UAV swarm improve detection performance compared to IDS relying solely on temporal information? Through extensive experiments and comparison with traditional deep neural network models, we evaluate the effectiveness of this topology-aware GNN approach in securing UAV swarm communications.**

***Index Terms*—Graph Neural Networks, UAVs, cyber-physical systems, intrusion detection systems, and machine learning.**

I. Introduction

SWARM of unmanned aerial vehicles (UAVs) have gained significant attention due to their diverse civilian and military applications. The coordinated operation of UAV swarms enables novel functionalities and facilitates tasks such as surveillance, coverage, and disaster management, which would be unattainable if UAVs operated individually [1]. However, the security of these networked systems is paramount, as vulnerabilities could lead to catastrophic consequences, including physical harm and data breaches [2].

Previous research has investigated intrusion detection systems (IDS) to protect UAV swarm networks against a range of cyber-attacks, including denial-of-service, false data injection, replay, hijacking, and spoofing [2]. Yet, these existing solutions often overlook the unique characteristics of swarm communication, such as coordination patterns, ground station control, and local data processing. Additionally, the propagation of cyber-attacks within the swarm and the potential impact on other UAVs remain under-explored.

Critically, current IDS primarily rely on temporal data, neglecting the valuable insights that spatial relationships within the swarm can offer for intrusion detection. To address these research gaps, this paper investigates the following questions:

- Will exploiting temporal and spatial correlations within the data improve the detection performance?
- How do the detection capabilities of topology-aware IDS, which consider the spatial structure of the swarm, compare to those of topology-unaware IDS?

Answering these questions presents challenges. First, there is a lack of a dataset that captures the spatial and temporal correlation patterns inherent within the swarm communication. Second, existing datasets do not exhibit normal and malicious swarm behavior examples, which are required for training effective models. Moreover, training an IDS model requires careful feature engineering to identify optimal features, as irrelevant features can hinder model performance. To tackle these challenges, we carry out the following:

- We developed a testbed comprising of six UAVs (UAV1-UAV6) arranged in an hexagonal topology, along with an access point, controller, data collection tools, and an attacker execution environment. This testbed enables the controlled execution of cyber-attacks and the collection of data under both normal and attack conditions.
- We created a comprehensive dataset capturing both normal and malicious UAV swarm communication patterns. Specifically, four distinct attacks such as false data injection, evil twin, replay, and denial-of-service were executed on each of the UAVs within the swarm. The dataset includes both spatial and temporal features of swarm communications.
- We proposed a GNN-based topology aware IDS that incorporates spatial and temporal relationships between UAVs. We evaluated the performance against several benchmark topology-unaware deep learning models, including feedforward neural networks (FNN), recurrent neural networks (RNNs) with long short-term memory (LSTM) cells, and 1D-convolutional neural networks.

The remainder of this paper is organized as follows. Section

This work was supported by NSF Award 2220346.

II delves into relevant prior work and its limitations. Section III details the testbed setup, data collection, and the specific cyber-attacks considered. Section IV explores data preprocessing and feature extraction techniques. Section V outlines the proposed GNN-based IDS. Section VI presents the experimental results, and Section VII concludes the paper.

II. Related Work

Existing research on IDS for UAV swarm communications has predominantly focused on three main approaches:

1) Cyber-only IDS: These IDS models are trained exclusively on cyber features, such as packet information, frame numbers, and payload characteristics. For instance, Mehmood et al. [3] simulated UAV communications within the COOJA simulator and generated traffic data to train support vector machine (SVM), random forest, and K-nearest neighbor-based IDS. Kou et al. [4] proposed a combined deep autoencoder and convolutional neural network (CNN) model for anomaly detection in UAV communications using the InSDN dataset. Han et al. [5] developed an LSTM-based IDS to detect anomalies in packets from the CICIDS-2017 dataset.

2) Physical-only IDS: Conversely, physical-only IDS models are trained solely on physical behavioral features of UAVs, such as speed, velocity, and temperature. Park et al. [6] designed a stacked autoencoder for fault detection in UAV states. Ahn et al. [7] utilized a 1D-CNN to develop an IDS for identifying anomalies, while Khanapuri et al. [8] employed a fully connected deep neural network trained on simulated physical data.

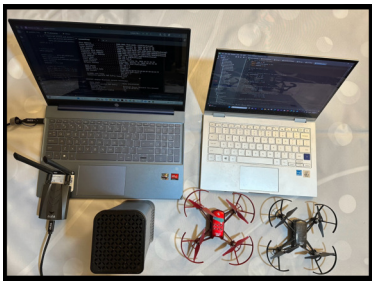
3) Cyber-Physical Fused IDS: A more recent approach involves fusing both cyber and physical features to enhance IDS performance. Hassler et al. [9] developed an IDS that combines both types of features using various deep learning models, including SVM, FNN, LSTM, and 1D-CNN.

A. Limitations of Existing Work

While these existing approaches have made significant contributions to the field, they all share a common limitation: a reliance on temporal information alone. None of the existing research has explored the potential of leveraging spatial relationships within the UAV swarm for intrusion detection. This oversight leaves a gap in the understanding of how spatial information, such as the relative positions and orientations of UAVs, can be exploited to identify anomalies in a swarm. Furthermore, there is a lack of publicly available datasets that capture both the spatial and temporal aspects of UAV swarm communication under normal and attacked conditions.

III. Testbed and Dataset Collection

This section details the equipment, experimental setup, and methodology used for data collection during both normal and attacked UAV swarm operations.



(a)

(b)

Fig. 1. Illustration of the testbed. (a) Tested setup and equipment. The computer on the right acts as a ground control station, the computer on the left acts as an attacker equipped with a network adapter, and the black tower is the access point; (b) Cartesian 2D Flight Map for the decentralized positional coordinate system.

A. Equipment

The equipment used includes six DJI Tello EDU drones, a flight map, a Sagemcom SAC2V2s WiFi access point, an ALFA AWUS036ACH network adapter with antenna, and two computers as shown in Fig. 1.

- The first computer (Computer 1) acts as the ground control station, connecting to the UAVs and running Python scripts to send control commands and receive status telemetry reports from each drone. These reports contain behavioral data such as barometer readings and IMU measurements. This data is termed as physical data.
- The flight map provides positional coordinates for enabling decentralized UAV-to-UAV communication.
- The access point facilitates the connection between the UAVs and the ground control station, ensuring seamless control and coordination of the swarm.
- The second computer (Computer 2) mimics an attacker, equipped with the ALFA adapter and running Kali Linux along with various tools like Aircrack-ng, Airgeddon, TCPdump, Scapy, and Wireshark. This computer serves two primary functions: capturing cyber data exchanged between the UAVs and the ground station, and executing cyber-attacks on the swarm.

B. Testbed Setup

Our experimental setup employed six UAVs configured to establish an hexagonal communication graph through a consensus-based decentralized control protocol. The testbed environment consists of the following components:

1) Flight Map: A 2D Cartesian coordinate flight map with dimensions of 3×3 meters was utilized. As illustrated in Fig. 1b, this flight map serves as a reference for UAV

positioning and enables decentralized communication. This implementation is essential because Tello EDU drones, primarily designed for indoor use, lack global positioning system (GPS) or other positional sensors. Instead, they rely on a vision positioning system comprising a 720p camera and two 3D infrared sensors for stable hovering. By detecting the flight map, the drones transmit their positional coordinates (x-axis, y-axis, and z-axis) and additional telemetry data (roll, pitch, yaw, etc.) to the ground control station.

2) Onboard PID Controller: Each UAV was equipped with an independent proportional-integral-derivative (PID) controller to govern its flight dynamics. This controller employs feedback mechanisms to maintain desired performance metrics. The control output $u(t)$ is given as

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{\mathrm{d_e(t)}}{\mathrm{dt}}, \tag{1}$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively. The term $e(\tau)$ represents the error between the measured output and the desired setpoint at time t. The proportional term adjusts the control signal in proportion to the current error. The integral term accounts for the accumulated past errors, mitigating steady-state deviations. The derivative term responds to the rate of change of the error. Tuning these gains ensures stable flight for each UAV.

3) Consensus-based Decentralized Control Protocol: In addition to the onboard PID controllers, we implemented a consensus-based decentralized control protocol to coordinate movement within a swarm. This protocol leverages local communication and cooperation among the UAVs to reach a consensus on their collective behavior. The inter-UAV communication network is modeled as an undirected graph as follows:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \tag{2}$$

where $\mathcal{V}$ represents the set of nodes (UAVs), $\mathcal{E}$ represents the edges (communication links) between the adjacent UAVs to exchange information. The interaction between UAVs are encoded into the adjacency matrix $\mathcal{A}$ as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3}$$

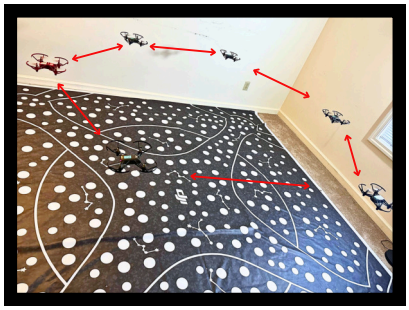
The adjacency matrix A represents the hexagonal communication topology among the six UAVs, where a non-zero entry a_{ij} indicates a communication link between UAV i and UAV j. The consensus algorithm employed is a weighted average consensus protocol, detailed in [10]. At each time step k, each UAV updates its state $x_i(k)$ based on the states of its neighbors, weighted by the elements of a weight matrix. In our implementation, all weights are set to 1 for uniform interaction. The state update rule is given by:

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j \in N_i} a_{ij}(x_j(k) - x_i(k)), \tag{4}$$

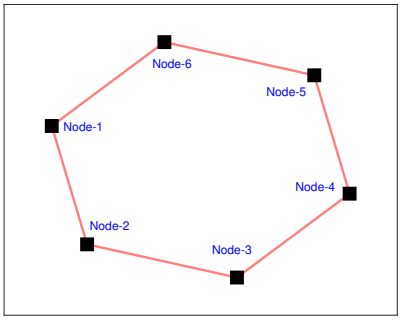
where ε is a step-size parameter or consensus gain controlling the convergence rate, N_i denotes the set of neighbors of UAV i, defined by the communication topology represented by the adjacency matrix A. Through the iterative execution of this protocol, the swarm gradually converges to a common state, enabling coordinated movement. More information about the consensus control algorithm can be found in [10], [11].

C. Data Collection Methodology

The dataset was collected in two distinct phases: normal flights and flights under cyber-attacks. The normal phase involved UAVs performing standard flight operations to gather benign data. In the second phase, the UAV swarm was subjected to four different cyber-attacks, with data collected during this phase labeled as malicious. To collect benign data, we conducted 20 flights under normal operating conditions. For the malicious dataset, we subjected each of the six UAVs to four distinct cyber-attack types, resulting in a total of 24 attack flights.



(a)



(b)

Fig. 2. Illustration of UAVs making a hexagonal graph.

D. Cyber-Attacks Execution

This subsection outlines the cyberattacks executed on each of the UAV in the swarm to collect the malicious dataset. The attacks are explained next.

1) De-authentication Attack: A de-authentication attack aims to disconnect legitimate users from a network. We executed this attack using the Aircrack-ng suite and Wireshark. Initially, the ALFA wireless network adapter was placed into monitor mode via the command `airmon-ng start wlan0`, enabling passive monitoring of Wi-Fi traffic. The `sudo airodump-ng wlan0` command then identified the target UAV's network. Finally, 100 de-authentication

packets were transmitted using `aireplay-ng --deauth 100 -a [AP mac] -c [target mac] wlan0`, where `AP mac` and `target mac` are the respective MAC addresses of the access point and the targeted UAV. This attack forcibly disconnected the victim UAV, triggering its 15-second inactivity landing protocol and resulting in a forced landing.

2) Replay attack: In a replay attack, the attacker captures legitimate communication packets and retransmits them. We utilized the Aircrack-ng suite for this as well. The steps involved placing the ALFA adapter into monitor mode (`airmon-ng start wlan0`) and initializing packet capture (`airodump-ng wlan0`). Legitimate command traffic between the target UAV and Computer 1 was captured and saved to a PCAP file. Using Wireshark, we filtered for command packets specifically targeting the victim UAV. The command `aireplay-ng --inject replay -r capture.pcap wlan0` was then used to replay the captured packets, causing the UAV to repeatedly execute the same commands. This attack led to abnormal behavior, preventing the UAV from performing its intended mission and responding to new commands.

3) Evil Twin Attack: An evil twin attack creates a rogue access point mimicking the legitimate network. In our experiment, the target UAV's SSID was identified using the command `sudo airodump-ng wlan0`. A rogue AP with the same SSID was then established using Airgeddon, and its signal strength was boosted using `iwconfig wlan0 txpower 30`. A brief de-authentication attack forced the UAV to disconnect from the legitimate AP and connect to the evil twin, enabling a man-in-the-middle attack scenario where the attacker could monitor, intercept, or manipulate the UAV's communication.

4) False Data Injection Attack: In a false data injection attack, malicious data is introduced to disrupt system operations. Utilizing the Aircrack-ng suite, Scapy, and custom Python scripts, we derived the state-space matrices of the target UAV's roll, pitch, and yaw using dynamic mode decomposition (DMD). Stealthy attack vectors were then crafted, modifying these measurements and control signals according to the equations $y = y+\gamma$ and $u = u+\eta$, respectively, where y and u represent the original values, and γ and η are the attack vectors. These falsified data packets were transmitted to the target UAV, causing it to receive incorrect sensor readings and execute erroneous control actions.

IV. Data Preprocessing and Feature Extraction

This section outlines the preprocessing steps applied to the collected cyber and physical data. Cyber data means communication data between the UAVs and the ground control station, while physical data means the physical behavioral characteristics of the UAV in space. We have extracted each data's spatial and temporal features, which are explained next.

A. Data Preprocessing and Fusion

Cyber data was initially stored in PCAP files. These files were processed in Wireshark to filter data based on the MAC addresses of each UAV. The filtered data was then converted to JSON format for further analysis. A custom Python script was developed to extract relevant cyber features from the JSON files and integrate them into Pandas DataFrames.

On the other hand, physical data was collected in .csv format. The data was received at a microsecond rate, at which it exhibited repetitions and missing values. A Python script was used to clean and restructure this data and finally store it in a .csv file.

To enable data fusion, the asynchronous cyber and physical datasets were aligned using interpolation based on the timestamps of the higher-frequency data. The fused dataset was then standardized using StandardScaler, which preserves the distribution of data while handling outliers effectively. This is in contrast to MinMaxScaler which scales all the negative values to zero, leading to the distortion of important coordination patterns of the swarm communication. In total, 88 data files were compiled containing cyber and physical data, including 40 benign files and 48 malicious files. The number of data samples for cyber, physical, and cyber-physical (fused) features are summarized in Table I.

TABLE I
SIZE OF DATA POINTS/SAMPLES FOR EACH DATASET

Datasets	Cyber Samples	Physical Samples	Fused Samples
UAV1	22,440	19,012	21,036
UAV2	21,552	18,132	20,152
UAV3	20,269	18,072	19,041
UAV4	19,357	18,341	18,607
UAV5	20,244	17,936	19,042
UAV6	20,053	18,098	19,504

B. Feature Extraction

From the preprocessed data, we extracted spatial and temporal features, crucial for constructing a graph representation of the UAV swarm and training GNN-based IDS.

1) Spatial Features: Spatial features, such as positions, orientations, and relative distances, define the geometric structure of the graph, representing the location of each drone and its spatial relationship to others. These features, listed in Table II, are categorized as physical and cyber features.

a) Spatial Physical Features: The brief explanation of the spatial physical features is as follows: The x-axis (map), y-axis (map), and z-axis (map) represent the spatial position coordinates of the UAVs according to flight map; roll_(map), pitch_(map), and yaw_(map) denote the angular orientation of the UAV according to the flight map; relative_dx, relative_dy, and relative_dz are the relative distance between the UAVs; x-axis, y-axis, and z-axis are the absolute spatial position of the UAVs in space; relative_px, relative_py, and relative_pz correspond to the relative positioning; and pitch, roll, and yaw denote the absolute angular orientation in a global frame.

b) Spatial Cyber Features: The following wireless communication data features are derived to be used as proxies for spatial information: signal_strength (dBm) is the strength of the received signal; wlan_radio.SNR (dB) is a radio signal-to-noise ratio that indicates the quality of the received signal; radiotap.signal_quality represents the overall signal and link quality; wlan.sa and wlan.ra are the source and receiver MAC

TABLE II
RAW SPATIAL FEATURES

Spatial Features		
Spatial Physical Features		**Spatial Cyber Features**
Physical	**Physical**	**Cyber**
x-axis (map)	x-axis	signal_strength (dBm)
y-axis (map)	y-axis	wlan radio.SNR (dB)
z-axis (map)	z-axis	radiotap.signal_quality
roll_angle_(map)	relative_px	wlan.sa
pitch_angle_(map)	relative_py	wlan.ra
yaw_angle_(map)	relative_pz	
relative_dx	roll_angle	
relative_dy	pitch_angle	
relative_dz	yaw_angle	

TABLE III
RAW TEMPORAL FEATURES

Temporal Features		
Temporal Physical Features	**Temporal Cyber Features**	
timestamp_p	frame.number	ip.id
x_speed	frame.len	wlan_radio.noise (dBm)
y_speed	frame.protocols	timestamp_c
z_speed	wlan.duration	wlan_radio.preamble
templ	wlan_radio.frequency	ip.src
temph	wlan_radio.datarate	ip.dst
tof	wlan.da	wlan.fcs
height	wlan.ta	wlan.fcs.status
battery	wlan.bssid	wlan.qos
barometer	wlan.frag	wlan.qos.priority
flight time	wlan.seq	wlan.qos.ack
x_acceleration	wlan.fc.type	wlan.ccmp.extiv
y_acceleration	wlan.fc.subtype	wlan.wep.key
z_acceleration	wlan.flags	radiotap.hdr_length
cntrl_x	wlan.fcs_len	radiotap.antenna_signal
cntrl_y	vip.len	radiotap.channel.flags.cck
cntrl_z	udp.length	radiotap.channel.flags.ofdm
	data.len	

addresses used to identify individual UAVs and infer relative distances corresponding to their signal strengths. Note that the signal strength and SNR can be correlated with distance in space because higher SNR values are typically observed at shorter distances.

2) Temporal Features: Temporal features capture changes over time in individual and swarm behaviors, including variations in velocity, acceleration, communication patterns, and environmental conditions. Table III lists the physical and cyber spatial features.

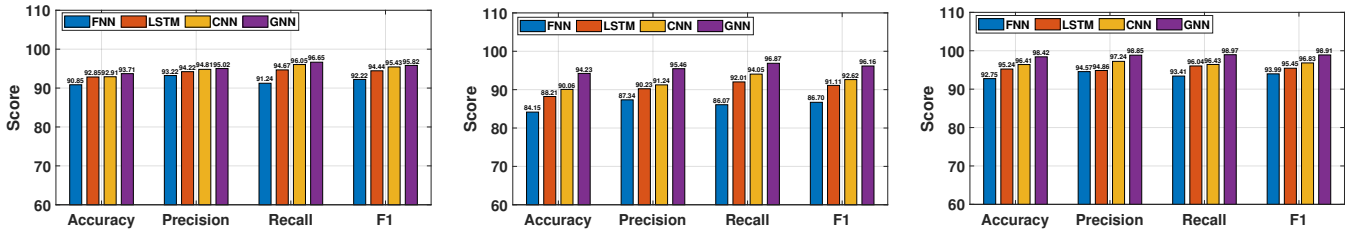
a) Temporal Physical Features: The brief explanation of the temporal physical features is as follows: timestamp_p is the timestep; x_speed, y_speed, z_speed represent the speed measurement across each axis; temph and templ represent the high and low temperature of the UAV in celsius, respectively; tof denotes the time-of-flight distance to measure the depth; height indicates altitude above a reference point; battery shows the remaining power; barometer measures the air pressure and use for altitude measurements; flight time is the elapsed operating duration; x_acceleration, y_acceleration, z_acceleration represent the acceleration along each axis; and cntrl_x, cntrl_y, cntrl_z correspond to the control inputs from the PID controller applied along each axis.

b) Temporal Cyber Features: A brief explanation of each temporal cyber feature listed is as follows: frame.number is the sequential identifier for each captured frame; frame.len indicates frame length in bytes; frame.protocols specify the encapsulated network protocols like TCP or UDP; wlan.duration is the reservation time in microseconds to transmit the frame, while wlan.ta and wlan.sa are the receiver and source MAC addresses, respectively; wlan.bssid is the MAC address of the access point; wlan.frag is the fragment number; wlan.seq is the sequence number; wlan.fc.type and wlan.fc.subtype categorize the frame type and subtype, respectively; wlan.flags are the control indicators; wlan.fcs is the frame check sequence; ip.len is the IP packet length; udp.length and data.len are the UDP segment length and packet data length, respectively; ip.id is the IP header ID; wlan_radio.noise (dBm) and wlan_radio.preamble indicate radio noise and preamble, respectively; ip.src and ip.dst are the source and destination IP addresses, respectively; wlan.fcs.status shows the integrity check result, wlan.qos, wlan.qos.priority, and wlan.qos.ack relate to quality-of-service settings; wlan.ccmp.extiv is the extended initialization vector; wlan.wep.key is the WEP encryption key; radiotap.hdr_length is the header size; radiotap.antenna_signal is the antenna signal strength; radiotap channel flags indicate modulation types; and wlan_radio.datarate and wlan_radio.frequency show the transmission rate and frequency, respectively. Features adding complexity in model learning were removed.

V. GNN-BASED INTRUSION DETECTION SYSTEM

The proposed IDS employs a GNN to model the interactions and behaviors of a swarm of six UAVs arranged in an hexagonal communication topology. Each UAV is represented as a node in the graph, characterized by both spatial features (position, orientation, relative distances) and temporal features (velocity, acceleration, control signals). Edges connect neighboring nodes as defined by the adjacency matrix of the hexagon, weighted by the communication links. Given the link connections within the hexagonal structure, a neighborhood order of 1 is sufficient to capture relevant spatial dependencies. The architecture begins by inputting the UAVs' raw spatial and temporal features, which are concatenated into a comprehensive feature vector for each node.

The GNN architecture consists of two Chebyshev graph convolutional layers with 64 and 128 output channels that aggregate information from the immediate neighbors of each node using the Chebyshev polynomial filters to capture the localized structural patterns of the graph. These node embeddings then undergo non-linear transformations using Relu activation function, followed by a batch normalization layer for stable training. Next, the max pooling layer aggregates the final node embeddings into a single graph-level representation to capture the overall behavioral pattern of the UAV network. This graph-level representation is fed into the fully connected dense layer with a sigmoid activation function to detect whether the UAV behavior is normal or malicious in a network. The model is trained using the Adam optimizer with a learning rate of 0.001 and binary cross-entropy as the loss function. A batch size of 32 is used during training.



(a) Cyber-only dataset (b) Physical-only dataset (c) Cyber-physical fused dataset

Fig. 3. Detection Performance of IDS Models.

TABLE IV
IMPROVEMENT OF DIFFERENT IDS MODELS WITH RESPECT TO GNN ON CYBER-PHYSICAL (CP) FUSED DATASET

Model	Dataset	Accuracy	Precision	Recall	F1
FNN	CP	6%	4.2%	5.56%	4.92%
LSTM	CP	3.18%	3.99%	2.93%	3.46%
1D-CNN	CP	2.01%	1.61%	2.54%	2.08%

VI. EXPERIMENTAL RESULTS

This section presents the experimental results of the proposed GNN model. We have compared the GNN model performance with the conventional deep learning models such as FNN, recurrent LSTM networks, and 1D-CNN. We compared the performance of these models in terms of accuracy, precision, recall, and F1-score metrics across three distinct datasets: cyber-only, physical-only, and a fused cyber-physical dataset. Grid search is used to find optimal hyperparameters.

- Fig. 3 represents the detection performance of the GNN model compared to conventional deep learning models (FNN, LSTM, and 1D-CNN). It can be observed that GNN-IDS outperforms the benchmark models across cyber-only, physical-only, and cyber-physical fused datasets. In addition, the GNN model achieved superior detection on a cyber-physical fused dataset compared to the cyber and physical dataset alone. Notably, we can see the GNN model achieved the highest detection performance with an F1-score of 98.91%.
- In specific, Table IV shows the performance enhancement of the GNN model over other deep learning IDS models on the cyber-physical dataset, which exhibited the highest overall detection rates. The GNN model demonstrates a substantial improvement, with a 4.92% increase in F1-score over FNN, a 3.46% increase over LSTM, and a 2.08% increase over 1D-CNN.

Why GNN Performs Better?

GNN outperforms traditional deep learning models due to its ability to exploit both spatial and temporal information within the UAV swarm communication. Unlike traditional deep learning models (FNN, LSTM, 1D-CNN) that focus solely on temporal patterns, the GNN incorporates the swarm's graph structure, capturing spatial relationships between UAVs. This topological awareness enables the GNN to identify anomalies based on deviations from both temporal and spatial patterns, thereby enhancing its detection capabilities compared to topology-unaware models.

VII. CONCLUSIONS

In this paper, we developed a testbed of six UAVs communicating in an hexagonal graph topology. We collected a dataset encompassing both normal swarm operation and swarm operation under various cyber-attacks (false data injection, evil twin, replay, and denial-of-service). We developed a GNN-based IDS and compared its performance against benchmark IDS models such as FNN, LSTM, and 1D-CNN. Our investigations revealed that the GNN-IDS outperforms the other models, achieving a 4.92%, 3.46%, and 2.08% higher detection rate on unseen data compared to FNN, LSTM, and 1D-CNN, respectively. This highlights the effectiveness of topology-aware IDS in UAV swarm communication security.

In future work, we will incorporate more swarm topologies into the network and evaluate the robustness of the GNN-IDS on unseen topology data.

REFERENCES

[1] A. Khan, E. Yanmaz, and B. Rinner, "Information exchange and decision making in micro aerial vehicle networks for cooperative search," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 4, pp. 335–347, 2015.

[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] R. T. Mehmood, G. Ahmed, and S. Siddiqui, "Simulating ml-based intrusion detection system for unmanned aerial vehicles (uavs) using cooja simulator," in *2022 16th International Conference on Open Source Systems and Technologies (ICOSST)*. IEEE, 2022, pp. 1–10.

[4] L. Kou, S. Ding, T. Wu, W. Dong, and Y. Yin, "An intrusion detection model for drone communication network in sdn environment," *Drones*, vol. 6, no. 11, p. 342, 2022.

[5] J. Han and W. Pak, "Hierarchical lstm-based network intrusion detection system using hybrid classification," *Applied Sciences*, vol. 13, no. 5, p. 3089, 2023.

[6] 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.

[7] 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.

[8] 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.

[9] S. C. Hassler, U. A. Mughal, and M. Ismail, "Cyber-physical intrusion detection system for unmanned aerial vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[10] J. Wang, Z. Zhou, C. Wang, and J. Shan, "Multiple quadrotors formation flying control design and experimental verification," *Unmanned Systems*, vol. 7, no. 01, pp. 47–54, 2019.

[11] M. A. Toksöz, S. Oğuz, and V. Gazi, "Decentralized formation control of a swarm of quadrotor helicopters," in *2019 IEEE 15th International Conference on Control and Automation (ICCA)*. IEEE, 2019, pp. 1006–1013.