Ensemble Learning-Based Intrusion Detection System for Aerial Base Stations Against Adversarial Evasion Attacks

John Richeson*, Salma Aboelmagd†, Umair Mughal‡, Abdulrahman Takiddin†, and Muhammad Ismail*,
*Cybersecurity Education, Research, and Outreach Center (CEROC), Tennessee Tech University, Cookeville, TN 38505, USA
†Department of Electrical and Computer Engineering, Florida State University, Tallahassee, FL 32310, USA
‡Department of Computer Science, Northwest Missouri State University, Maryville, MO 64468, USA
Email: {jdricheson42, mismail}@tntech.edu; {saboelmagd, a.takiddin}@fsu.edu; umughal@nwmissouri.edu

***Abstract*—Aerial base stations (ABSs) are expected to play a major role in 5G+ wireless networks where unmanned aerial vehicles (UAVs) serve as flying base stations to provide wireless coverage. As the utilization of ABSs continues to expand, ensuring their security and resilience against malicious attacks emerges as a vital concern. In this paper, we demonstrate successful false data injection attacks on a real UAV testbed, leading the UAV off-course, which can impact the ABS coverage. Current research efforts focus on designing intrusion detection systems (IDS) tailored specifically for UAVs. However, the impact of adversarial attacks on UAV IDS is largely overlooked in the literature. Our results herein show that evasion attacks pose a significant threat, capable of deteriorating IDS model detection accuracy by 20%. In this paper, we propose adopting cyber-physical fused datasets to train our proposed unsupervised sequential ensemble learning-based models to improve IDS robustness against evasion attacks. Our results, based on a practical UAV testbed and considering a wide range of evasion attacks, demonstrate that the proposed ensemble of a Transformer autoencoder and long short-term memory recurrent neural network reduces accuracy deterioration to 3% when combined with the physical and cyber fused features.**

***Index Terms*—Adversarial evasion attacks, aerial base station, machine learning, Transformer model, unmanned aerial vehicles.**

I. Introduction

The evolution of 5G networks includes delivering reliable and high-performing connectivity across various domains. Unmanned aerial vehicles (UAVs) as aerial base stations (ABSs) offer the solution by extending quality coverage in emergency and remote scenarios [1]. As networks advance towards 6G, ABSs become more significant with demands for higher performance, enhanced security, and reliable connectivity in rural areas [2]. Thus, the security of the UAV is instrumental to the reliability of the supported networks.

A. Related Work

Existing literature has enhanced the security of UAVs against some attack types. A support vector machine (SVM) model was proposed to detect signal spoofing attacks [3]. Another custom supervised learning model was proposed to detect Sybil attacks [4]. A verification-based autoencoder model was proposed to detect smart attacks using the extended Kalman filter [5]. A model combining convolutional neural network (CNN) and long short-term memory (LSTM) was proposed to detect denial-of-service (DOS) attacks [6]. Multiple studies proposed machine learning-based approaches to detect jamming attacks including spectrogram-based models [7], convolutional attention models [8], reinforcement learning [9], and federated learning [10]. An approach based on a neural network with random forest was proposed to detect attacks, including DOS, brute force, infiltration, botnet, and injection attacks [11].

The aforementioned intrusion detection systems (IDSs) present the following limitations. First, they are specifically designed to detect specific attacks, including spoofing, Sybil, DOS, jamming, hijacking, and false data injection (FDI) attacks, which can compromise data, disrupt operations, or take control of the UAV [12], [13]. Second, existing studies explore the design of IDS based on physical features, such as pitch, roll, and yaw [14] or cyber features collected from the data packets sent to and from the controller, such as frame number, frame length, and MAC addresses [15]. However, the fusion of physical and cyber features was found to improve IDS performance [12]. Third, existing studies overlook the potential of fused-based IDSs to address more vulnerabilities in UAVs, such as evasion attacks, which is a major concern for IDS on ABSs. An evasion attack is a type of adversarial attack that takes place after an IDS model has been trained, aiming to fool and deteriorate the IDS detection performance [16]. To the best of our knowledge, there is no existing literature that explores the robustness of UAV IDSs against evasion attacks.

B. Contributions

This paper makes the following contributions to the field:

- We develop a realistic UAV testbed and demonstrate successful FDI attacks with a practical dataset collection process.
- We study the impact of adversarial evasion attacks, including the fast gradient sign method (FGSM), basic iterative method (BIM), and Carlini & Wagner (C&W) which lead to an average deterioration of 20% in accuracy against benchmark IDSs.

This work was supported by the NSF CyberCorps SFS grant #2043324.

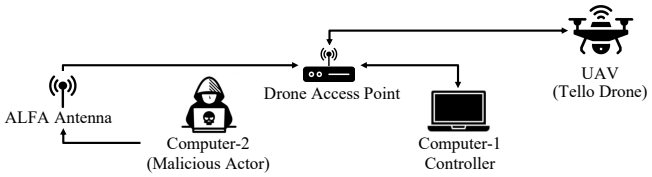

Fig. 1. UAV testbed for dataset collection.

- We develop an ensemble learning-based cyber-physical IDS. The proposed ensemble of the LSTM and Transformer reduces the impact of evasion attacks to 3% deterioration in accuracy when using a fused cyber and physical feature dataset.

The rest of this paper is organized as follows. Section II presents the practical UAV testbed, demonstrates successful FDI attacks on the UAV, and outlines the data collection process. Section III introduces the benchmark IDS models, evasion attacks, and proposed ensemble learning-based IDS. Section IV evaluates the performance of the models against evasion attacks. Section V concludes the paper.

II. Testbed Development

This section presents the construction of the testbed and data collection, including a demonstration of successful FDI attacks.

A. Testbed

The equipment needed to set up the testbed includes the following: DJI Tello EDU drone [17], DJI Tello Mission Pad [18], Computer-1 as the controller or base station, Computer-2 with Kali Linux distribution [19], and ALFA AWUS036ACH antenna [20]. An illustration of the equipment is shown in Fig. 1. The location of the testbed was chosen based on safety and space requirements. All flights are conducted on the DJI Tello Mission Pad, which measures 10 feet by 10 feet. The UAV connects to the Mission Pad through a Python script, enabling coordinate tracking. The center of the pad is marked as the origin (0,0), allowing the UAV to move up to 150 units in any direction on the x and y axes.

Computer-1 serves as the controller for operating and monitoring the UAV. It runs a Python script that allows the operator to connect to the UAV, send coordinate commands, and receive the status of the UAV. This script stores each command sent and the status received from the UAV. The Wi-Fi access point establishes a connection between Computer-1 and the UAV for two-way communication. The ALFA antenna is connected to Computer-2, running Kali Linux, to provide Wi-Fi monitoring and attacking capabilities. Computer-2 runs Wireshark [21] software to capture packets between the access point and the controller. Computer-2, also acting as the adversary, conducts Wi-Fi attacks using Aircrack-ng [22] software. Both operations require the antenna to be in monitor mode.

B. Practical FDI Attack

To collect malicious data, a stealthy FDI [23] attack that diverts the UAV off its intended course is conducted. This attack involves establishing a man-in-the-middle setup between the UAV and its controller. Using Computer-2 and the ALFA antenna, the attacker intercepts the two-way communication broadcast over the WLAN. By eavesdropping on both the forward and feedback channels, the attacker generates an estimate of the UAV's current state. The attack begins by targeting the feedback channel, slightly altering the UAV's sensor readings to cause incorrect state estimations to be sent to the controller. Then, the attack manipulates the forward channel, modifying the coordinates sent to reflect the incorrect UAV state.

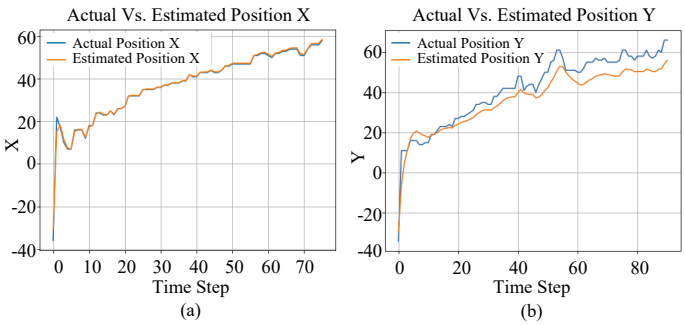

Fig. 2. Successful FDI attack. (a): x-coordinate of UAV position. (b) y-coordinate with a deviation between the actual and estimated values.

Although the channels use an anomaly detector to notify when there are deviations from normal status, the attack on both channels allows the attacker to remain stealthy. This FDI attack assumes the adversary has partial knowledge of the UAV's system and can infer the system matrices. This process is repeated through a Python script until the UAV is fooled that it has reached its desired position. The FDI targets the UAV's coordinates to alter its flight path, with deviation measured by comparing the coordinates before and after the attack. As the flight duration increases, so does the extent of the deviation.

Fig. 2 highlights the actual and estimated x and y positions of the UAV recorded by the controller. Notably, while the x-coordinate shows no difference between the actual and estimated positions in Fig. 2(a), Fig. 2(b) reveals a deviation in the y-coordinate, indicating that the UAV was off course by 10 units, equivalent to 1 foot. This deviation is indicative of a successful attack, where the difference between the actual and estimated coordinates of the UAV can severely affect the wireless coverage of the ABS. When the UAV is diverted from its intended path, the coverage area of the ABS becomes uneven, leading to gaps or overlaps. This could result in some areas experiencing weak or unstable connections, ultimately compromising the reliability of the entire network.

C. Data Collection

This section provides an overview of the data collection process, comprising both benign and malicious data. It explains the two sets of experiments conducted, involving normal flights and flights under attack. Then, we introduce the key features of the cyber and physical datasets collected during these experiments.

1) Benign Data: Benign data collection begins with Computer-1 connecting to the UAV's access point and running a Python script for autonomous flight. The UAV will initially

transmit its current coordinates using the pad where the script calculates the necessary flight commands to reach the operator's desired destination. During the flight, the UAV continuously updates its physical measurements after each command. Once the script is stopped, it converts these sensor measurements into a CSV file. Simultaneously, Computer-2 uses Aircrack-ng tools to identify the basic service set identifier (BSSID) of the UAV and access point. The BSSIDs are provided to Wireshark to capture WiFi traffic between the controller and the access point. Once the UAV completes its flight, Wireshark is stopped, and the captured packets are saved.

2) Malicious Data: During the malicious data collection phase, the UAV connects to Computer-1 and is instructed to fly autonomously to a designated destination. The FDI attack is initiated via a Python script that manipulates the coordinates being sent to the UAV before they reach the access point. The script uses these altered commands to generate a new estimated UAV status, which is then sent to Computer-1, replacing the original status. This process continues throughout the flight, with the coordinates being subtly adjusted until the UAV reaches its destination. Consequently, all sensor measurements recorded by the controller are manipulated before being logged into the CSV file. Simultaneously, Computer-2 employs Aircrack-ng to identify BSSIDs, enabling Wireshark to isolate and capture network traffic. Packet capturing is terminated once the flight is complete. The raw data consists of ten benign CSV files and ten malicious PCAP files, which are initially separated into physical and cyber datasets for preprocessing.

3) Features: We present next an overview of the cyber and physical datasets and briefly explain what each feature denotes.

a) Physical Dataset: The physical dataset comprises 31 features overall. For the physical features, mid denotes mission pad detection, x, y, and z represent the UAVs coordinates, pitch, roll, yaw denote the UAV orientation, mpitch, mroll, and myaw represent the UAV orientation on the mission pad, x_speed (vgx), y_speed (vgy), and z_speed (vgz) indicate speed in each axis, templ and temph are low and high temperatures of the main board, tof is the time-of-flight, h is the height relative to take-off, bat is battery percentage, baro represents the height measured by the barometer, flight_time shows the motor running time, agx, agy, and agz denote the acceleration in the x, y, and z axis, est_x and est_y represent the estimated coordinates of the UAV from the controller, cntl_x and cntl_y show the coordinate inputs given by the controller, residual1, residual2, residual3, and residual4 denote Kalman filter results for x and y injection.

b) Cyber Dataset: For the cyber dataset, we have 34 features. The frame.number represents the sequential number of each frame, frame.len denotes the frame length, wlan.ta, wlan.sa, wlan.ra, wlan.da show the MAC addresses for transmitter, source, receiver, and destination, wlan.bssid is the BSSID of the access point, wlan.frag is the fragment number, wlan.seq is the sequence number, wlan.fc.type and wlan.fc.subtype refers to the frame control type, wlan.flags represents the status flags, wlan.fcs indicates the frame-checking sequence, wlan.qos, wlan.qos.priority, wlan.qos.ack indicates quality-of-service, priority, and acknowledgment, wlan.ccmp.extiv is a counter mode, wlan.wep.key denotes a wired equivalent privacy (wep) security protocol, data.len is the length of the data, radiotap.hdr_length is the header length, radiotap.antenna_signal is the signal strength, radiotap.signal_quality is signal quality, radiotap.channel.flags.ofdm is for orthogonal frequency-division multiplexing, radiotap.channel.flags.cck is a complementary code keying, wlan_radio.datarate and wlan_radio.frequency are the rate and frequency at which data is transmitted, wlan.radio.signal_strength (dbm) is the signal strength, wlan_radio.Noise level (dbm), wlan.radio.SNR (db), and wlan.radio.preamble are all linked with radio data and indicate noise, Signal-to-Noise Ratio (SNR), and the preamble of the radio frame, and timestamp_c denotes timestamps.

III. Benchmark and Proposed Robust UAV IDS

This section presents the benchmark models that we adopt to study the impact of adversarial evasion attacks on UAV IDSs. Next, we present the evasion attacks under consideration. Finally, we propose an ensemble Transformer-LSTM based IDS designed to limit the detection deterioration impact of the evasion attacks, also evaluated using the three datasets.

A. Benchmark IDSs

Each IDS is trained and tested on all three datasets (physical-only, cyber-only, and fusion of physical and cyber) of labeled benign and malicious operations in a supervised manner. Feature selection is performed using the Shapley Additive Explanations (SHAP) approach [24], which quantifies the contribution of each feature to the model's predictions, providing insight into the factors that influence the model's decision. Adopting the SHAP approach helps identify the most important features to be used in the model during training. SVM is a shallow static model that separates classes using a hyperplane. Feed-forward neural network (FNN) is a static deep neural network model with a one-directional flow of information. Long short-term memory (LSTM) recurrent neural network (RNN) is a dynamic approach suited for time-series data that captures the sequential patterns in the data while addressing the vanishing gradient problem found in prolonged back-propagation, which is a limitation with FNN models. Transformer is a deep dynamic approach that handles large datasets and captures temporal patterns. The Transformer model uses self-attention to process sequences in parallel to capture short and long-range dependencies. The encoder part was used rather than an encoder-decoder approach to reduce complexity while retaining its ability to identify complex patterns. We adopt a randomized grid search to find the optimal parameters, which are reported in Table I.

B. Evasion Attacks

In this section, we describe three evasion attacks adopted to evaluate both the machine learning-based IDS models selected as benchmarks and our proposed ensemble learning-based IDS.

TABLE I
OPTIMAL HYPER-PARAMETERS

Model	Parameter	Features		
		Physical	Cyber	Cyber-Physical
SVM	Kernel	Poly	Poly	Poly
	Regularization	100	100	1
	Gamma	0.3	0.5	0.3
FNN	Activation	ReLU	ReLU	ReLU
	Layers	5	4	3
	Neurons	1024	1024	1024
	Dropout rate	0.5	0.4	0.6
LSTM	Activation	ReLU	ReLU	ReLU
	Layers	2	4	3
	Neurons	128	512	512
	Dropout Rate	0.4	0.4	0.6
Transformer	Embedding Size	256	128	128
	Attention Heads	4	4	4
	Attention Blocks	4	4	1
	Layers	4	4	4
	Neurons	128	128	128
	Dropout Rate	0.4	0.5	0.4
Proposed Transformer	Embedding Size	64	128	128
	Attention Heads	2	4	6
	Attention Blocks	2	4	6
	Layers	6	6	6
	Neurons	64	128	256
	LSTM Dropout	0.4	0.5	0.3
	Transformer Dropout	0.3	0.5	0.4

1) The Fast Gradient Sign Method Attack: The FGSM attack [25] is an adversarial attack that generates perturbations by computing the gradients of the loss function in machine learning models. We express the FGSM attack function as

$$X_A = X_B - \epsilon \,\text{sign}\left(\nabla_{X_B} J(\varphi, X_B, y)\right), \tag{1}$$

where X_A is the adversarial example, X_B is the benign sample, ϵ is the perturbation value, $sign$ indicates the use of the signum function, ∇_{X_B} refers to the model gradients, J is the model's loss function, φ is the model's parameters, and y is the true label. The gradients identify the direction in which changes to input values will increase loss. Perturbations are made in this direction with a magnitude ϵ value of 0.5 for this attack based on its ability to generate adversarial examples with significant deterioration while remaining undetectable.

2) The Basic Iterative Method Attack: The BIM attack [26] is an extension of the FGSM attack that introduces iterative refinement of the perturbations. The BIM attack produces stronger and more effective adversarial examples using two additional parameters, I and α, which represent the number of iterations and the step size of each iteration respectively, with the same ϵ value of 0.5. Additionally, the BIM attack offers a clipping mechanism to ensure perturbations remain within a defined range of ϵ making the attack stealthier than the FGSM attack. We represent the BIM attack function as

$$X_A^{(i)} = \text{Clip}_{X_B,\epsilon}\{X_A^{(i-1)} - \alpha \text{sign}\left(\nabla_{X_B^{(i-1)}} J\left(\varphi, X_A^{(i-1)}, \boldsymbol{y}\right)\right)\}, \tag{2}$$

where $X_A^{(i)}$ is an adversarial example at the i-th iteration, $X_A^{(i-1)}$ is the adversarial example from the previous iteration, and $\nabla_{X_B^{(i-1)}}$ represents the model gradients evaluated at the benign sample from the previous iteration.

3) The Carlini & Wagner Method Attack: The C&W [27] attack optimizes perturbations to maximize their influence on the model while simultaneously minimizing their detectability. The C&W [27] attack uses the variables ϵ, I and α along with iterative refinements. The C&W attack introduces the $L2$ loss normalization to keep track of how much the adjusted example has deviated from the original sample in terms of Euclidean distance. Due to its design, C&W cannot be implemented on the SVM model because the SVM model does not rely on gradient-based calculations. We express the C&W attack function as

$$X_A = \min_{\epsilon} \omega(X_B, X_B + \epsilon), \tag{3}$$

where ω is the Euclidean distance between X_B and $X_B + \epsilon$.

C. Proposed Robust UAV IDS

Our proposed IDS is based on an ensemble learning method that involves sequentially combining multiple high-performing models to enhance the robustness against evasion attacks. Using an unsupervised approach, our proposed ensemble learning-based model is trained only on benign data to detect anomalies. This approach allows the model to identify both unknown and known attack patterns referred to as zero-day attacks. Due to the offered advantages of the Transformer and LSTM models, we propose adopting them in an ensemble learning manner (shown in Algorithm 1) to boost their detection performance.

Algorithm 1: Training of Proposed Ensemble Model

Input Data: X_{B}
Hyperparameters: Initial hyperparameters are randomly assigned
Initialization: Input layers with the shape input shape, weights W and biases b for each Transformer and LSTM layer for each Transformer and LSTM layer
Transformer_output ← output from Transformer model after processing X_{B} with W and b applied
LSTM_output ← output from LSTM model applied to Transformer_output with its own W and b
Compile: Ensemble model with Adam optimizer and mean squared error loss
while *not converged* **do**
  for *each epoch* **do**
    for *each batch b in* X_B **do**
      Forward Propagation: Generate predictions by passing b through Transformer and LSTM
      Compute Loss: Mean squared error between predictions and true values
      Back-propagation and Update Parameters:
      Compute gradients with respect to W and b within each layer
      Update W and b within each layer using computed gradients and Adam optimization
    end
    Evaluate model on validation set
    Early Stopping: If validation loss increases, terminate training
  end
end
Output: Trained Ensemble Model parameters optimized for benign samples

As shown in Algorithm 1, the first block is based on a Transformer due to its unique ability to handle large datasets and capture temporal patterns. Using an attention mechanism,

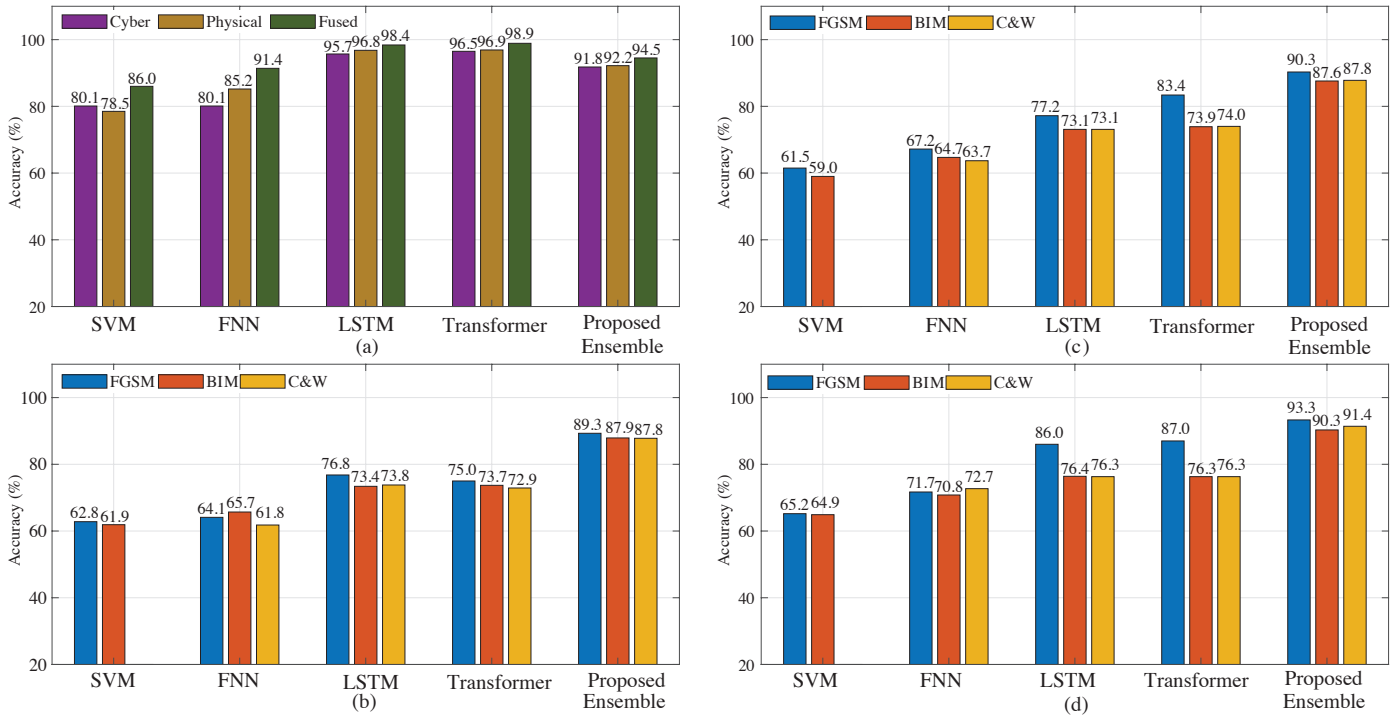

Fig. 3. Accuracy (a) of traditional attacks without evasion, (b) with evasion attacks using cyber-feature only IDS, (c) with evasion attacks using physical-feature only IDS, and (d) with evasion attacks using fused IDS.

it processes sequences in parallel, allowing it to capture long-range dependencies efficiently. Benign input X_B is passed through the Transformer layer with the encoder and decoder sections aiming to minimize reconstruction error. The encoder uses multi-head attention and FNN layers to capture a compressed representation of the input, while the decoder applies attention over the encoder's output to focus on relevant parts of the input data for reconstruction. The reconstructed samples are then passed through a convolution layer, where the input is represented in a higher-dimensional space to extract complex features. These features are then passed into the LSTM layer as shown in lines 4 and 5.

The LSTM layer processes the input data in sequence using hidden and cell states to keep track of previous time steps. This allows the model to retain information from earlier steps while introducing the current input. After passing through the LSTM layer, the ensemble's output represents the reconstructed input with learned temporal features. The reconstruction error is measured by comparing the ensemble's output with the original input, using mean squared error (MSE).

To determine whether a sample is anomalous or not, we set a threshold value based on statistical analysis of the reconstruction errors. The threshold is calculated as the median of the reconstruction errors plus 1.5 times the interquartile range (IQR) [28]. This calculation ensures that the threshold adapts to the distribution of benign samples, providing a robust method for detecting anomalies [29]. Using the IQR method helps avoid the influence of outliers, making the threshold less sensitive to extreme values. By combining the Transformer and LSTM architecture, the ensemble model provides the ability to identify anomalous behavior in spatial and temporal domains.

IV. Experimental Results

This section discusses the detection results for the benchmark and proposed IDSs across the physical-only, cyber-only, and cyber-physical fusion datasets. We then evaluate the model performance against the traditional attacks as well as adversarial ones using accuracy ACC $= \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$, where TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative samples.

A. Performance Against Traditional Attacks

The performance of the benchmark IDSs against the traditional attack (discussed in Section II-C) before launching evasion attacks is shown in Fig. 3(a). The static models, which include SVM and FNN, perform the lowest across all datasets due to their inability to handle the time-series aspect of our data. Using the fused cyber and physical features dataset yielded the highest accuracy against traditional attacks, with static models improving by an average of 7.7% compared to using only cyber or only physical features. This increase in performance is due to the fused dataset capturing both cyber and physical behavior, allowing the models to process both aspects, which results in enhanced detection performance. In the case of the dynamic models, which include LSTM, Transformer, and the proposed ensemble IDS, the use of the fused dataset improved the accuracy by 3.1% compared to utilizing cyber-only and physical-only features. On average, the dynamic models outperform the

static ones by 12.4% in accuracy, as the dynamic models are able to capture the temporal features of the datasets.

B. Performance Against Evasion Attacks

The impact of evasion attacks on model accuracy is illustrated in Figs. 3(b), (c), and (d). Evasion attacks significantly reduce the accuracy of both static and dynamic benchmark models due to their stealthy nature. When utilizing the fused dataset, the static models exhibit an average accuracy deterioration of 20.2% compared to traditional attacks. Notably, the C&W attack employs gradient-based methods and is therefore inapplicable to SVM models, which do not provide the necessary gradient information. In the case of the dynamic models, the LSTM and Transformer show the greatest vulnerability, particularly with the physical and cyber datasets, experiencing average accuracy deterioration of 21.0% and 22.2%, respectively, in comparison to traditional attacks. This heightened susceptibility is due to their sensitivity to subtle temporal variations, which adversarial attacks exploit. The use of the fused dataset improved the deterioration of the dynamic benchmark models by 2.8% in accuracy, compared with traditional attacks. The proposed ensemble IDS model consistently outperforms both static and dynamic models across all three datasets against evasion attacks, achieving minimal accuracy deterioration of 3.6%, 4.0%, and 3.1% for the physical, cyber, and fused datasets, respectively, compared to traditional malicious attacks. This enhanced resilience is attributed to the ensemble model's ability to combine the strengths of both the LSTM and Transformer. By leveraging the LSTM's capability to capture temporal and sequential dependencies alongside the Transformer's effectiveness in detecting complex global patterns and long-range relationships, the ensemble model can identify a broader range of anomalies.

V. Conclusion

In this paper, we investigated the impact of evasion attacks against machine learning-based IDSs compared to traditional attacks. Evasion attacks, due to their stealthier nature, resulted in an average accuracy deterioration of 20% for the benchmark IDS models compared to traditional attacks. To enhance the robustness of IDSs against evasion attacks, we proposed an ensemble learning-based IDS that outperforms benchmark dynamic and static machine learning models, with a 4% average decrease in accuracy when detecting evasion attacks. Additionally, by fusing the cyber and physical datasets, we boosted the accuracy of all benchmark models—both static and dynamic—by 8% and 3%, respectively, in detecting traditional malicious attacks. Overall, the fused-based ensemble model outperformed the benchmarks, deteriorating by 3% in accuracy in detecting evasion attacks. Our proposed model presented a promising solution for implementing IDSs for UAVs, particularly within critical infrastructure systems such as 5G networks, where UAVs serve as ABSs.

References

[1] H. Huang *et al.*, "Deployment of heterogeneous uav base stations for optimal quality of coverage," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16 429–16 437, Sep 2022.

[2] S.-Y. Chang *et al.*, "Securing uav flying base station for mobile networking: A review," *Future Internet*, vol. 15, no. 5, May 2023.

[3] A. Shafique *et al.*, "Detecting signal spoofing attack in uavs using machine learning models," *IEEE Access*, vol. 9, pp. 93 803–93 815, Jun. 2021.

[4] D. Chulerttiyawong *et al.*, "Sybil attack detection in internet of flying things-ioft: A machine learning approach," *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12 854–12 866, Jul. 2023.

[5] A. Aladi *et al.*, "Uav attack detection and mitigation using a localization verification-based autoencoder," *IEEE Access*, vol. 11, pp. 117 752–117 764, Oct. 2023.

[6] R. Fu *et al.*, "Machine-learning-based uav-assisted agricultural information security architecture and intrusion detection," *IEEE Internet Things J.*, vol. 10, no. 21, pp. 18 589–18 598, Nov. 2023.

[7] Y. Li *et al.*, "Jamming detection and classification in ofdm-based uavs via feature- and spectrogram-tailored machine learning," *IEEE Access*, vol. 10, pp. 16 859–16 870, Feb. 2022.

[8] J. Viana *et al.*, "Deep attention recognition for attack identification in 5g uav scenarios: Novel architecture and end-to-end evaluation," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 131–146, Jan. 2024.

[9] J. Ghelani *et al.*, "Gradient monitored reinforcement learning for jamming attack detection in fanets," *IEEE Access*, vol. 12, pp. 23 081–23 095, Feb. 2024.

[10] Z. A. E. Houda *et al.*, "A privacy-preserving collaborative jamming attacks detection framework using federated learning," *IEEE Internet Things J.*, vol. 11, no. 7, pp. 12 153–12 164, Apr. 2024.

[11] Y. Wu *et al.*, "Intrusion detection for unmanned aerial vehicles security: A tiny machine learning model," *IEEE Internet Things J.*, vol. 11, no. 12, pp. 20 970–20 982, Jun. 2024.

[12] S. C. Hassler *et al.*, "Cyber-physical intrusion detection system for unmanned aerial vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 6, pp. 6106–6117, Jun 2024.

[13] U. A. Mughal *et al.*, "Machine learning-based intrusion detection for swarm of unmanned aerial vehicles," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*. Orlando, FL, USA, 02–05 Oct. 2023, pp. 1–9.

[14] C. Pu *et al.*, "Defending against flooding attacks in the internet of drones environment," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*. Madrid, Spain, 07–11 Dec. 2021, pp. 1–6.

[15] V. Praveena *et al.*, "Optimal deep reinforcement learning for intrusion detection in uavs," *Comput. Mater. Continua*, vol. 70, no. 2, pp. 2639–2653, Feb 2022.

[16] A. Takiddin, M. Ismail, and E. Serpedin, "Robust detection of electricity theft against evasion attacks in smart grids," in *Proc. IEEE Int. Conf. Commun. (ICC)*. Montreal, QC, Canada, 14–23 Jun 2021, pp. 1–6.

[17] DJI, "Tello edu," March 2024, [Online: accessed Sept. 2024]. [Online]. Available: https://www.ryzerobotics.com/tello-edu

[18] "Bannerbuzz," March 2024, [Online: accessed Sept. 2024]. [Online]. Available: https://www.bannerbuzz.com/

[19] Offensive Security, "Kali linux," [Online: accessed Sept. 2024]. [Online]. Available: https://www.kali.org/

[20] ALFA Network, "Awus036ach," April 2022, [Online: accessed Sept. 2024]. [Online]. Available: https://tinyurl.com/bj92nwm2

[21] Wireshark Foundation, "Wireshark (version 4.2.3)," March 2024, [Online: accessed Sept. 2024]. [Online]. Available: https://www.wireshark.org/

[22] "Aircrack-ng," [Online: accessed Sept. 2024]. [Online]. Available: https://www.aircrack-ng.org/

[23] U. A. Mughal *et al.*, "Stealthy false data injection attack on unmanned aerial vehicles with partial knowledge," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*. Orlando, FL, USA, 02–05 Oct. 2023, pp. 1–9.

[24] Scott Lundberg, "Shapley additive explanations," June 2022, [Online: accessed Sept. 2024]. [Online]. Available: https://shap.readthedocs.io/en/latest/index.html

[25] I. J. Goodfellow *et al.*, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[26] A. Kurakin *et al.*, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[27] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, May 2017, pp. 39–57.

[28] J. Yang *et al.*, "Outlier detection: How to threshold outlier scores?" in *Proc. Int. Conf. Artif. Intell. Inf. Process. Cloud Comput.* Sanya, China, 19–21 Dec. 2019, pp. 1–6.

[29] A. Takiddin *et al.*, "Robust data-driven detection of electricity theft adversarial evasion attacks in smart grids," *IEEE Trans. Smart Grid*, vol. 14, no. 1, pp. 663–676, Jan 2023.