Enhancing UAV Network Security: A Human-in-the-Loop and GAN-Based Approach to Intrusion Detection

Qingli Zeng, *Graduate Student Member, IEEE*, and Farid Nait-Abdesselam, *Senior Member, IEEE*

***Abstract*—Uncrewed aerial vehicles (UAVs) are becoming essential in various sectors, such as commercial delivery, agricultural monitoring, and disaster response. Despite their benefits, the rapid adoption of UAVs poses substantial security challenges, especially in drone network intrusion detection. Traditional intrusion detection datasets often suffer from limitations like small sample sizes and uneven distribution, undermining the effectiveness of intrusion detection systems. Moreover, conventional machine learning (ML) approaches generally require extensive, well-labeled datasets that are expensive and labor intensive to produce. To overcome these challenges, we introduce a generative adversarial network (GAN) model designed to enhance and balance the limited datasets available for drone networks. This model significantly improves data quality and quantity, thus optimizing the training process for intrusion detection models. Furthermore, we propose a Human-in-the-Loop (HITL) ML framework that integrates human expertise to guide the learning process and mitigate the costs of labeling. Our comprehensive evaluation demonstrates that the combined application of the GAN model and the HITL framework significantly outperforms traditional baseline models. This approach not only achieves an intrusion detection accuracy of up to 99% across various experimental datasets but also dramatically reduces the requirement for large amounts of labeled data by up to 98%, providing a cost-effective solution for enhancing UAV network security.**

***Index Terms*—Generative adversarial networks (GANs), Human-in-the-Loop (HITL), intrusion detection, uncrewed aerial vehicle (UAV) networks.**

I. Introduction

UNCREWED aerial vehicles (UAVs), commonly known as drones, have garnered significant attention due to their versatile applications across various domains [1], [2]. Widely adopted for surveillance, disaster management, agriculture, and industrial inspections, drones have transformed these fields [3]. The advent of drone swarms and networks, which allow multiple drones to collaborate, has further expanded their capabilities. These networks enable drones to communicate and coordinate their actions, significantly enhancing their efficiency and effectiveness in executing complex missions [4].

Received 5 August 2024; revised 1 September 2024, 16 October 2024, and 16 February 2025; accepted 20 February 2025. Date of publication 24 February 2025; date of current version 9 June 2025. *(Corresponding author: Qingli Zeng.)*

Qingli Zeng is with the School of Science and Engineering, University of Missouri-Kansas City, Kansas City, MO 64110 USA (e-mail: zq6mw@umsystem.edu).

Farid Nait-Abdesselam is with Université Paris Cité, 75006 Paris, France (e-mail: farid.nait-abdesselam@u-paris.fr)

Digital Object Identifier 10.1109/JIOT.2025.3545389

However, the growing reliance on drone networks has heightened their vulnerability to a variety of security threats [5], [6], [7]. Unlike traditional computer networks, UAV networks operate in highly dynamic and resource-constrained environments. The mobile nature of drones leads to constantly evolving network topologies and frequent link interruptions, creating unique challenges for network security [8]. In addition, UAVs are limited in computational power and energy reserves, which requires security solutions that are both lightweight and effective. These factors collectively highlight the need for specialized intrusion detection approaches tailored to the specific operational context of UAV networks. Security breaches in drone networks can have serious consequences, such as data leakage, privacy violations, physical damage, and disruption of critical operations [9], [10]. To counter these threats, the implementation of robust intrusion detection systems (IDS) is crucial [11]. IDS are essential for monitoring and detecting malicious activities within drone networks, enabling timely responses to prevent security incidents [12].

Machine learning (ML) techniques have emerged as a promising approach to developing robust and effective IDS in drone networks [13]. These data-driven algorithms capitalize on historical data to identify patterns and anomalies, enabling the detection of known and unknown intrusions [14]. However, deploying ML for IDS in drone networks is fraught with challenges [15]. A primary hurdle is the scarcity and imbalance of available intrusion datasets, which are crucial for training [16]. Collecting comprehensive and diverse datasets is a complex and labor intensive process [17]. Additionally, the limited availability of labeled data restricts the training and evaluation of ML models, as they depend extensively on annotated samples to distinguish between normal and malicious behaviors [18].

To tackle the challenges of data scarcity and imbalance, generative adversarial networks (GAN) have proven highly effective [19]. GAN involve two competing neural networks: 1) a generator that creates synthetic data mimicking real distributions and 2) a discriminator that differentiates between real and generated samples. Through their adversarial training process, GAN can produce realistic and diverse data, thus enhancing and balancing limited intrusion datasets [20]. This enhancement significantly improves the performance of ML models for IDS in drone networks, by enriching the training set with more representative data [21].

Moreover, traditional ML models are mainly based on labeled data, and supervised learning requires a substantial volume of annotated samples to perform effectively [22]. However, the availability of labeled intrusion datasets for drone networks is notably scarce in real-world scenarios [23]. The process of annotating large volumes of network traffic data is not only time-consuming but also costly, demanding significant human expertise and effort [24]. This scarcity of labeled samples, coupled with the high costs of annotation, presents considerable challenges for deploying effective IDS in drone networks [25].

To address these challenges, we propose a Human-in-the-Loop (HITL) ML framework for intrusion detection in drone networks. HITL is an iterative method that uses human experience to guide the learning process, thus minimizing labeling costs [26]. By involving human experts in the loop, our framework focuses on selectively labeling the most informative and representative data samples [27].

The main contributions of this article are as follows.

1) We propose a GAN-based data augmentation technique to address the scarcity and imbalance of drone network intrusion datasets. This approach generates synthetic samples that enhance the diversity and representativeness of training data, thereby improving the robustness and generalization of ML models for IDS.
2) We introduce an HITL ML framework that effectively integrates human expertise to guide the learning process while minimizing labeling costs. This framework facilitates the strategic selection of the most informative samples for labeling, reducing the annotation burden, and enhancing the model's performance.
3) We conduct extensive experiments to evaluate the effectiveness of our proposed methods compared to state-of-the-art ML models. The results demonstrate superior performance of our framework in terms of intrusion detection accuracy, with significantly fewer labeled samples required than traditional supervised learning approaches.

The remainder of this article is organized as follows. Section II provides an overview of related work on intrusion detection in drone networks. Section III presents the system model for intrusion detection in drone networks. Section IV introduces the conditional tabular GAN (CTGAN) and its application in augmenting our dataset. Section V details our proposed HITL ML framework, explaining its key components and functionalities. Section VI describes the experimental setup, including the datasets, evaluation metrics, and simulation parameters. Section VII discusses the results of our experiments, analyzes the performance of our proposed approaches, and compares them with state-of-the-art methods. Finally, Section VIII concludes this article and exploring potential directions for future research.

II. Related Work

The development and deployment of UAV networks have added new complexities to network security, underscoring the need for advanced IDS. This section reviews the pertinent literature, focusing on the evolution of intrusion detection system (IDS) within UAV networks, the application of ML techniques to these systems, and innovative solutions to challenges, such as data scarcity and imbalance. Particularly, we explore the use of GANs and HITL approaches in addressing these issues.

A. Intrusion Detection in Drone Networks

Intrusion detection in drone networks has attracted considerable attention from the research community due to increasing security concerns associated with the widespread adoption of drones [28]. Various methodologies have been proposed to address the challenges of detecting and mitigating cyber threats within these networks. Mitchell and Chen [29] developed an adaptive, specification-based IDS, termed behavior-rule-based UAV IDS (BRUIDS), which models normal UAV behaviors using rules and monitors deviations through peer or ground station surveillance. Sedjelmaci et al. [30] introduced a hierarchical framework that merges anomaly detection with signature-based techniques, enhancing detection accuracy and reducing false positives through drone-ground station cooperation. However, their reliance on predefined signatures or behavior rules limits their ability to detect novel attacks.

Recent advances have focused on more sophisticated approaches. Mughal et al. [31] proposed a deep learning-based framework specifically designed for UAV swarm networks, achieving higher detection accuracy for zero-day attacks. Wang et al. [32] introduced a federated learning approach for distributed intrusion detection in UAV networks, addressing privacy concerns while maintaining detection effectiveness. Karmaka et al. [33] developed an adaptive IDS that combines blockchain technology with ML for secure UAV communications.

B. Machine Learning for Intrusion Detection

ML techniques have been extensively applied to intrusion detection across various domains, including drone networks [34]. Commonly used supervised learning algorithms, such as support vector machines (SVM), Decision Trees, and NNs are pivotal in training ML models for IDS [35]. For example, Almseidin et al. [36] evaluated the performance of different ML algorithms, including SVM, Random Forest, and Decision Tree, utilizing the KDD Cup 99 dataset. Their findings indicated that Random Forest outperformed other algorithms in terms of accuracy. Moreover, deep learning approaches have recently drawn significant attention for their ability to discern complex patterns directly from raw data. Yin et al. [37] implemented a deep-learning model using recurring NNs (RNNs) for intrusion detection, achieving high precision with the NSL-KDD dataset. Additionally, Moustafa and Slay [38] developed a deep-learning-based IDS that combines stacked autoencoders with a decision tree classifier, demonstrating its effectiveness on the UNSW-NB15 dataset in detecting diverse types of attacks. However, the success of these models is highly dependent on the availability and quality of labeled data, which remains scarce and expensive

to acquire in practical settings, particularly for drone networks where data collection and labeling present substantial challenges [39]. In addition, reliance on benchmark datasets in most studies may not fully capture the unique characteristics and threats relevant to drone networks.

C. Generative Adversarial Networks for Data Augmentation

GAN have proven to be a powerful tool for data augmentation and synthesis, significantly addressing the challenges of data scarcity and imbalance [40]. Notable research has explored the utility of GAN in the context of intrusion detection. Lin et al. [41] developed a GAN-based method to generate synthetic network traffic data, thus enhancing the training of intrusion detection models. Usama et al. [42] utilized GAN to create synthetic samples for the minority class in industrial control systems' intrusion detection, effectively mitigating class imbalance and boosting IDS accuracy. Furthermore, Yin et al. [43] introduced a GAN-based framework for botnet detection, employing GAN to mimic the distribution of botnet traffic and generate synthetic data to augment training datasets. Their approach notably enhanced detection capabilities, particularly for minority botnet classes. Despite these promising outcomes, the application of GAN for data augmentation in the specific context of drone networks is still in its infancy. The distinctive characteristics and security challenges associated with drone networks may necessitate tailored adaptations and optimizations of existing GAN-based methods. This targeted approach not only alleviates the annotation burden but also enhances the model's performance by concentrating on the most impactful data points. By integrating the domain knowledge of human experts with advanced ML algorithms, our proposed framework aims to develop highly accurate and efficient IDS tailored for drone networks [44].

D. Human-in-the-Loop Machine Learning

HITL ML is a burgeoning paradigm that synergizes human expertise with ML algorithms to improve the performance and efficiency of learning systems [45]. HITL methodologies actively involve human experts in the learning process, utilizing their domain knowledge to guide the selection of informative samples for labeling [46]. Various studies have explored the application of HITL across different fields. Zhao et al. [47] developed a visual analytics framework integrating human knowledge to train a concept extractor with minimal labeling effort, effectively extracting meaningful visual concepts. Similarly, Huang et al. [48] employed HITL in medical image analysis using an adaptive densely connected convolutional network that incorporates expert feedback to significantly improve diagnostic accuracy and reliability in complex medical scenarios.

Despite these promising implementations, the application of HITL for intrusion detection in drone networks has not been extensively explored. The distinctive dynamics, such as fluctuating network topologies, resource limitations, and the need for real-time detection, call for specialized HITL frameworks tailored to these environments.

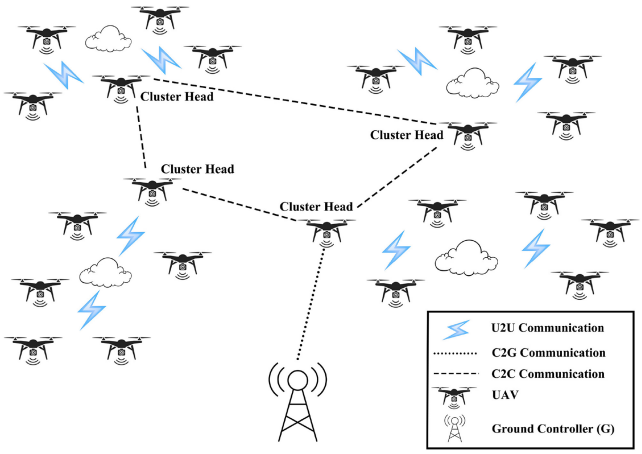


Fig. 1. Drone network architecture.

To overcome these challenges, we propose a novel framework that integrates GAN-based data augmentation with a collaborative HITL approach for intrusion detection in drone networks. Our framework utilizes the generative power of GAN to enrich and balance the limited intrusion datasets, thereby enhancing the diversity and representativeness of the training data. Furthermore, we introduce a collaborative HITL strategy that engages multiple human experts in the learning process. This allows for strategic selection of the most informative samples for labeling, significantly reducing annotation costs. By combining data augmentation with expert-guided learning, our framework aims to develop precise and efficient ML models suitable for drone networks, even under conditions of scarce and imbalanced data.

III. System Model for Intrusion Detection in Drone Networks

In this section, we introduce our system model for detecting intrusions in drone networks. Our model features a hierarchical structure composed of multiple drones, ground stations, and a central command center. Each drone is equipped with a range of sensors and communication devices to collect, exchange, and collaborate on data. The ground stations act as intermediaries, connecting drones to the command center and enabling data aggregation, processing, and dissemination. The command center oversees mission planning, coordination, and decision making for the entire network.

A. Drone Network Architecture

Our drone network architecture employs a hierarchical structure, as illustrated in Fig. 1. This design is optimized for the efficient management and security of a large-scale UAV network. The system is composed of three main components: 1) individual UAVs; 2) cluster heads; and 3) a ground controller.

UAVs: The lowest tier comprises individual UAVs that form an ad hoc network for local communication and coordination. These UAVs use UAV-to-UAV (U2U) communication to exchange information and maintain situational awareness within their local clusters.

Cluster Heads: The middle tier includes selected UAVs that act as cluster heads. These cluster heads serve as regional controllers for groups of UAVs, aggregating data from the drones within their clusters and facilitating communication between clusters. They use Cluster-to-Cluster (C2C) communication to enable broader coordination and data sharing across the network.

Ground Controller: The highest tier is the ground controller (G), which oversees the entire drone network and makes high-level decisions based on information received from the cluster heads.

Each UAV, including the cluster heads, is equipped with an onboard intrusion detection module that continuously monitors the drone's behavior and communication patterns to detect any anomalies or suspicious activities. Cluster heads host more advanced intrusion detection capabilities, aggregating and analyzing data from multiple UAVs within their clusters to identify potential threats.

B. Real-World Deployment Strategies for IDS

To deploy our HITL-ML framework in real-world UAV networks, we propose a two-tier implementation strategy aimed at balancing security effectiveness with operational constraints. This approach leverages the computational advantages of both UAVs and ground stations.

In our drone network model, the intrusion detection workflow involves collaboration between ML models across various levels of the hierarchy. At the UAV level, we will implement a lightweight, optimized version of our model, specifically designed for the constrained computational environment of drone systems. This onboard model will perform rapid, preliminary threat assessments, enabling real-time processing of network traffic data without overburdening the UAV's limited resources.

Complementing the UAV-level detection, ground control stations will host the full HITL-ML model, which is capable of conducting comprehensive analyses on aggregated data from multiple UAVs. This centralized processing approach allows for the use of more complex threat detection algorithms and deeper analysis of network behaviors, mitigating the computational limitations of individual UAVs.

To address potential network latency issues, our system architecture prioritizes essential data transmission between UAVs and ground stations. The onboard lightweight model provides immediate threat detection capabilities, reducing reliance on constant communication with the ground station and minimizing the impact of network delays on system performance.

While our current framework uses offline-trained models, we plan to implement a mechanism for periodic model updates. These updates, incorporating new threat data and insights from human expert interactions at the ground station level, will be deployed to both UAVs and ground stations during scheduled maintenance, ensuring the system remains adaptive to evolving threat landscapes.

Future research will focus on enhancing the real-time processing capabilities of our system, particularly in dynamic UAV network environments. This includes exploring adaptive thresholding techniques and edge computing solutions to further optimize performance and reduce latency. By adopting this multitiered approach, we aim to develop a robust, efficient, and scalable intrusion detection system capable of meeting the unique challenges posed by real-world UAV network security applications.

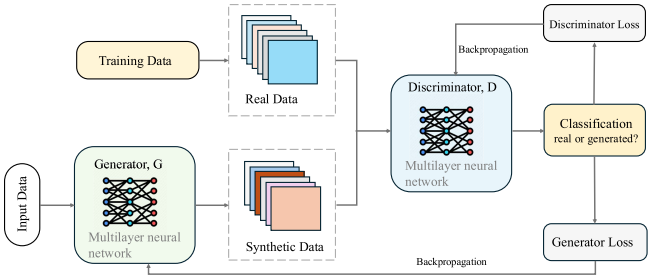


Fig. 2. Architecture of CTGAN model.

IV. Conditional Tabular Generative Adversarial Network

In this section, we introduce the CTGAN, which we employ to balance and augment our intrusion detection dataset for drone networks. CTGAN is a specialized variant of GANs designed specifically to handle tabular data. At its core, a CTGAN consists of two competing NNs: 1) a generator and 2) a discriminator. To understand CTGAN, imagine a scenario where an art forger (the generator) attempts to create fake paintings, while an art expert (the discriminator) tries to distinguish between real and forged artworks. Through this "competition," both the forger and the expert continuously improve their skills, ultimately leading to the creation of highly realistic forged paintings (our synthetic data). For those interested in the detailed structure and full mathematical description of the CTGAN, please refer to [19] and [49].

A. CTGAN Components

As shown in Fig. 2, the CTGAN consists of the following components.

1) Generator: This component is responsible for producing the data. Like our hypothetical art forger, the generator takes random input (which can be thought of as random inspiration) and attempts to create synthetic data that closely resembles the real data. The generator's process can be simplified as follows:

$$\text{Synthetic Data} = G(\text{Random Noise}, \text{Conditional Info}) \quad (1)$$

where G represents the generator function, which transforms random noise into synthetic data based on certain conditional information.

2) Discriminator: Working in tandem with the generator, the discriminator tries to differentiate between the real and synthetic datasets. If we continue our art analogy, the discriminator acts as an art expert trying to distinguish between genuine masterpieces and forgeries. It provides feedback to

the generator, directing the latter to refine its data generation process. The discriminator's process can be simplified as

$$\text{Real/Fake Classification} = D(\text{Input Data}) \tag{2}$$

where D represents the discriminator function, which classifies input data as either real or synthetic.

3) Real Data and Synthetic Data: Real data refers to the authentic datasets utilized during the training process. On the other hand, synthetic data is the output generated by the GAN model itself. In our intrusion detection context, real data corresponds to actual network traffic logs, while synthetic data consists of artificially generated logs that mimic real traffic patterns.

4) Classification: In the depicted GAN schematic, the classification component corresponds to the discriminative function of the Discriminator. As shown in Fig. 2, the Discriminator receives both real data from the training set and synthetic data generated by the Generator as input. It then performs a binary classification to determine whether each input is real or generated. The classification result directly contributes to the Discriminator Loss, which is subsequently used to update the Discriminator via backpropagation. Simultaneously, this process informs the Generator Loss, which is used to update the Generator, also through backpropagation. The Discriminator's role is twofold: it evaluates whether the input data is real, or synthetic, produced by the Generator. Through this "competition," both networks improve, leading to the generation of high-quality synthetic data.

B. Enhancing CTGAN Robustness and Performance

GANs offer significant potential for data generation, yet they remain vulnerable to issues like overfitting and mode collapse [19]. To mitigate these risks in our application of CTGAN, we implemented several strategies aimed at improving stability and preventing overfitting.

1) Training Optimizations: CTGAN leverages a conditional generator and a training-by-sampling technique to model complex distributions in tabular data. These mechanisms help capture data diversity and mitigate the risk of mode collapse by conditioning on different values of discrete variables. To further optimize the training process, we reduced the learning rates for both the generator and discriminator and extended the number of training epochs. This gradual learning approach allows the model components to reach equilibrium over time, leading to a more accurate and stable representation of the underlying data distribution.

2) L2 Regularization: To mitigate potential overfitting, we incorporated L2 regularization into the model [50]. This technique adds a penalty term to the loss function, proportional to the sum of the squared weight values. By discouraging excessively large weights, L2 regularization helps prevent the model from overfitting to the training data, promoting better generalization to unseen data

$$L_{\text{total}} = L_{\text{original}} + \lambda \sum_i w_i^2. \tag{3}$$

Here, L_{original} denotes the original loss, w_i represents the network weights, and λ is the regularization parameter that controls the strength of the penalty. By constraining the magnitude of the weight values, L2 regularization discourages the model from relying too heavily on any individual feature, thereby reducing the risk of overfitting. This penalty term promotes learning more robust and generalizable patterns from the training data, which ultimately leads to better performance on unseen data.

3) WGAN-GP Gradient Penalty: To further stabilize the training process and mitigate mode collapse, we incorporated the gradient penalty term from the Wasserstein GAN with Gradient Penalty (WGAN-GP) [51]. This method enforces Lipschitz continuity on the discriminator, resulting in smoother gradients and more reliable feedback for the generator. The gradient penalty is defined as

$$L_{\text{GP}} = \lambda_{\text{GP}}\, \mathbb{E}_{\hat{x}}\left[\left(\left\|\nabla_{\hat{x}} D(\hat{x})\right\|_2 - 1\right)^2\right] \tag{4}$$

where $\hat{x}$ represents the interpolated samples, $D(\hat{x})$ is the discriminator's output, and λ_{GP} is the gradient penalty coefficient controlling the strength of the Lipschitz constraint. By incorporating L_{GP} into the overall WGAN loss, we reduce training instabilities often encountered in vanilla GANs, thereby achieving smoother convergence.

These strategies collectively enhanced the robustness of our CTGAN model, allowing for the generation of diverse and representative synthetic data. This augmentation significantly improved the balance and richness of our intrusion detection dataset, ultimately enabling the development of more accurate models for drone network security.

V. HITL Machine-Learning Framework

In this section, we introduce our HITL ML framework for intrusion detection in drone networks. This framework incorporates human expertise to guide the learning process, enhancing the performance of the intrusion detection model while minimizing the need for extensive data annotation. Fig. 3 provides a detailed illustration of the overall workflow of the HITL framework.

This framework comprises three main phases: 1) the data preprocessing phase; 2) the ML phase; and 3) the HITL annotation phase. In the data preprocessing phase, we clean, normalize, and augment the intrusion detection dataset using the CTGAN method, as detailed in Section IV. During the ML phase, we train the intrusion detection model with the augmented data and evaluate its performance on a separate test dataset. Finally, in the HITL annotation phase, human experts are engaged to annotate selected samples and refine the model's predictions.

A. Data Preprocessing Phase

In the data preprocessing phase, we transform the intrusion detection dataset into a format suitable for training and testing our ML model. We begin by cleaning the data, removing incomplete, inconsistent, or irrelevant entries, including null values, NaN, and Infinity. Next, we address non-numeric features, such as attack and protocol types, by applying one-hot encoding to convert them into numerical representations. To ensure that no single feature dominates the learning process,

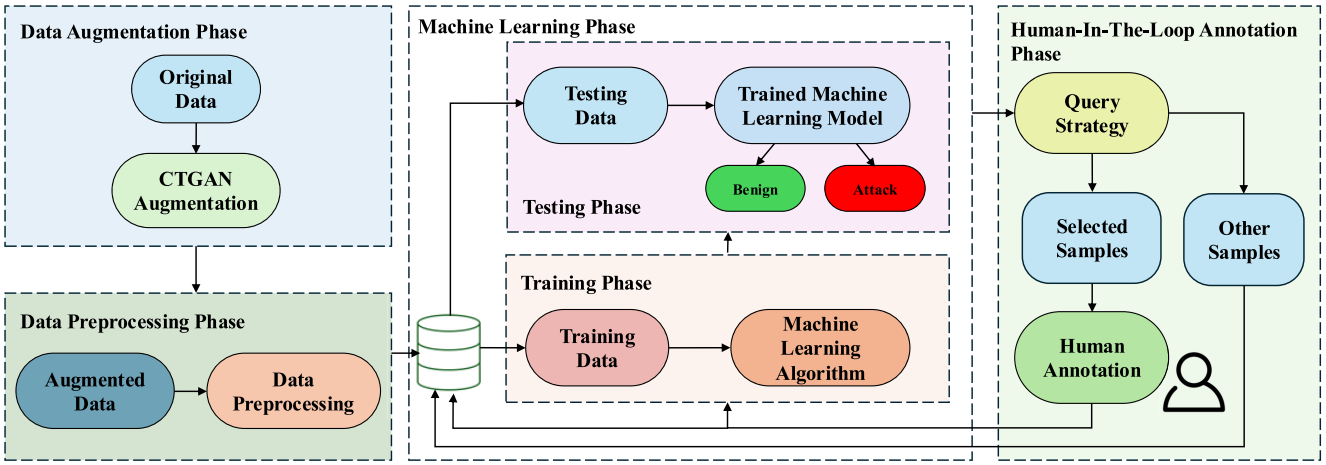


Fig. 3. CTGAN-based HITL ML system architecture.

we normalize the dataset by scaling all features to a common range, allowing them to contribute equally to the model's decision-making. After cleaning, encoding, and normalizing, we use CTGAN to augment the dataset, generating synthetic samples that closely resemble the real data distribution. This helps balance the class distribution and tackle the challenges of data scarcity and imbalance. Finally, we split the augmented dataset into training and testing sets, with the training set used to build the model and the testing set reserved for evaluating its performance on unseen data.

B. Machine-Learning Phase

The ML phase in our framework consists of two main components: 1) Training Phase and 2) Testing Phase.

Training Phase: In this phase, we use the preprocessed and augmented data as our training dataset. The ML model is trained on this data to learn the patterns of both benign network traffic and various types of attacks. The output of this phase is a trained ML model capable of classifying network traffic into either benign or attack categories.

Testing Phase: We evaluate the model's performance using a separate testing dataset that was not involved in training. The model classifies each instance in the testing data, allowing us to assess its generalization to unseen data.

As shown in Fig. 3, a feedback loop connects the testing phase to the training phase, illustrating our iterative approach to model improvement. Based on the model's performance on the test data, we refine the training process, adjust model parameters, or incorporate newly annotated data from the HITL annotation phase to enhance accuracy and robustness.

By iteratively refining the model, we continuously improve its ability to detect intrusions, adapting to new types of network traffic and emerging threats. This data-driven approach, combined with human expertise from the HITL annotation phase, enables the development of a more effective and adaptable intrusion detection system for drone networks.

C. HITL Annotation Phase

In the HITL annotation phase, we leverage human expertise to improve the performance of the intrusion detection model. To efficiently utilize expert knowledge, we employ uncertainty sampling to select the most informative samples for human review.

Uncertainty sampling targets instances where the model's prediction confidence is the lowest. These samples often lie near decision boundaries and, when correctly labeled, can significantly enhance the model's accuracy. In a multiclass setting, we quantify uncertainty using information entropy

$$\text{Entropy} = -\sum_{i}(P_i \cdot \log(P_i)) \tag{5}$$

where P_i represents the model's predicted probability for the ith class. Information entropy measures the average amount of uncertainty in the model's prediction. A higher entropy value indicates greater uncertainty, suggesting that the model's predictions are spread more evenly across multiple classes rather than concentrated on a single class.

We set an entropy threshold τ to identify uncertain samples. Any sample exceeding τ is considered "most informative" and prioritized for human review. In our experiments, setting τ around the top 10% of the entropy distribution struck a good balance between annotation efficiency and model improvement. Selected samples are presented to human experts through a custom interface displaying key features (e.g., packet-level details) to ensure accurate labeling.

Annotation experts were chosen for their background in network security and intrusion detection. A structured onboarding program ensured consistency across the team, starting with training on common attack patterns and ambiguous cases. Periodic feedback further refined annotation accuracy, and labeled samples were fed directly back into the training pipeline for iterative improvement.

By selecting high-entropy samples for expert review, we focus human effort on the most challenging and informative cases. This targeted approach efficiently refines the model's decision boundaries and enhances overall performance.

VI. Experiments Setup

A. Dataset

To evaluate the effectiveness of our proposed HITL ML framework for intrusion detection in drone networks, we

TABLE I
SUMMARY OF DATASETS

Dataset	Original labels	New labels	Number for each class
CIC-IDS2017	Benign	Normal	2,271,320
	Dos Hulk, DDoS, Dos GoldenEye, Dos Slowloris, Dos Slowhttptest	DoS	379,737
	PortScan	Probe	158,804
	Bot	Botnet	1,956
	Infiltration	Infiltration	36
	Web Attack: Brute Force, Web Attack: XSS, Web Attack: Sql Injection	Web Attack	2,180
	FTP-Patator, SSH-Patator	Brute Force	13,832
	Heartbleed	Heartbleed	11
	Sample Size of Original Data: 2,830,743 Sample Size after Preprocessing: 2,827,876		
NUSW-NB15	Backdoor	Backdoor	2,329
	DoS	DoS	16,353
	Exploits, Generic, Fuzzers	Exploits	127,642
	Normal	Normal	93,000
	Reconnaissance, Analysis	Reconnaissance	16,664
	Shellcode	Shellcode	1,511
	Worms	Worms	174
	Sample Size of Original Data: 257,673 Sample Size after Preprocessing: 257,673		

utilize two well-known intrusion detection datasets: 1) CIC-IDS2017 [52] and 2) UNSW-NB15 [38].

It is important to recognize that, as of now, there are no widely recognized datasets specifically designed for the detection of UAV network intrusion within the research community. Given this limitation, we have adopted an approach similar to that of other researchers in the field by selecting established cybersecurity datasets that can be adapted to UAV contexts. The CIC-IDS2017 and UNSW-NB15 datasets were chosen for their extensive coverage of network attacks and their potential relevance to UAV network scenarios [53], [54].

These datasets possess several characteristics that make them suitable for UAV network environments. First, they cover a broad spectrum of cyberattacks, reflecting the diverse threat landscape that UAV networks may encounter. The CIC-IDS2017 dataset contains more than 2.8 million labeled samples and 15 imbalanced attack classes, including DDoS, DoS, Brute Force, SQL Injection, Infiltration, and Botnet. The UNSW-NB15 dataset comprises over 250 000 data samples and nine imbalanced attack classes, such as Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. These datasets provide a comprehensive representation of various intrusion scenarios and are widely used for evaluating IDS.

Furthermore, these datasets encompass traffic from various network protocols and exhibit time-sensitive attack patterns, which align with the heterogeneous and real-time communication requirements of UAV swarms. The scale and diversity of these datasets enable us to rigorously evaluate our framework's robustness against a wide range of potential threats under different network conditions, which is crucial for ensuring reliable UAV operations.

B. Data Preprocessing

To preprocess the datasets, we first remove any null values and invalid data points, such as NaN and Infinity, to ensure data consistency and reliability. Additionally, we merge attack classes with similar characteristics to reduce the complexity of the classification task and improve the model's generalization ability. Table I presents the characteristics of the preprocessed CIC-IDS2017 and UNSW-NB15 datasets after removing anomalies and merging similar attack classes. Based on Table I, it is evident that both the CIC-IDS2017 and UNSW-NB15 datasets suffer from class imbalance issues. In the CIC-IDS2017 dataset, the number of samples in the "Benign" class (2 271 320) is significantly higher than the number of samples in attack classes, such as "Heartbleed" (11) and "Infiltration" (36). Similarly, in the UNSW-NB15 dataset, the "Normal" class has 93 000 samples, while attack classes like "Worms" (174) and "Shellcode" (1511) have substantially fewer samples. This imbalance in class distribution can bias ML models toward the majority class, leading to poor performance in detecting minority class instances.

To address the class imbalance issue and augment the datasets, we employ the CTGAN as described in Section IV. Specifically, CTGAN is used to synthesize samples for attack classes with lower sample sizes, thus achieving a more balanced distribution of samples across various intrusion categories [55]. Table II shows the characteristics of the original CIC-IDS2017 and UNSW-NB15 datasets compared to the datasets after applying CTGAN for data augmentation.

C. Simulation Parameters and Environment

All experiments were conducted using Python 3.10 on Google Colab's TPU v2-8 platform. As outlined in Table III, the CTGAN models were trained for 3 000 epochs with a batch size of 1 500 on both datasets. Learning rates and decay parameters were fine-tuned to optimize performance, with slight adjustments made between the two datasets to account for their unique characteristics.

Table IV details the parameters used during the HITL annotation phase. We began with an initial set of 500 labeled samples and incrementally added 50 samples per iteration. The uncertainty sampling strategy used information entropy, with probability values constrained to the range $[1 \times 10^{-10}, 1.0]$ to ensure numerical stability.

D. Performance Metrics

To evaluate the performance of our proposed HITL ML framework for intrusion detection in drone networks, we employ four widely used metrics: 1) Accuracy (Acc); 2) F1 score (F1); 3) precision (Prc); and 4) recall (Rc). For clarity, we define the standard confusion matrix outcomes used in these metrics: true positives (TP), where intrusions are correctly identified; false positives (FP), where normal samples are incorrectly flagged as intrusions; true negatives

TABLE II
COMPARISON OF ORIGINAL AND AUGMENTED DATA FOR CIC-IDS2017 AND UNSW-NB15

Dataset	Classes	Original Dataset			Augmented Dataset		
		Total Number	Train	Test	Total Number	Train	Test
CIC-IDS2017	Normal	2,271,320	1,817,231	454,089	2,271,320	1,817,231	454,089
	DoS	379,737	303,585	76,152	379,737	303,585	76,152
	Probe	158,804	127,092	31,712	158,804	127,092	31,712
	Botnet	1,956	1,570	386	6,956	5,000	1,391
	Infiltration	36	29	7	5,036	5,000	36
	Web Attack	2,180	1,762	418	7,180	5,000	1,436
	Brute Force	13,832	11,020	2,812	13,832	11,020	2,812
	Heartbleed	11	8	3	5,011	5,000	11
UNSW-NB15	Backdoor	2,329	1,863	466	7,329	5,000	1,465
	Dos	16,353	13,083	3,270	16,353	13,083	3,270
	Exploits	127,642	102,114	25,528	127,642	102,114	25,528
	Normal	93,000	74,400	18,600	93,000	74,400	18,600
	Reconnaissance	16,664	13,331	3,333	16,664	13,331	3,333
	Shellcode	1,511	1,209	302	6,511	5,000	1209
	Worms	174	139	35	5,174	5,000	174

TABLE III
CTGAN TRAINING PARAMETERS

Parameter	CIC-IDS2017	UNSW-NB15
Epochs	3000	3000
Generator Learning Rate	0.00001	0.0006
Discriminator Learning Rate	0.00002	0.0008
Generator Decay(λ_G)	1×10^{-6}	1×10^{-6}
Discriminator Decay(λ_D)	1×10^{-6}	1×10^{-6}
Batch Size	1500	1500
Gradient Penalty Coefficient (λ_{GP})	1	1

TABLE IV
HITL ANNOTATION PARAMETERS

Parameter	Value
Datasets	CIC-IDS2017, UNSW-NB15
Initial Labeled Samples	500
Samples per Iteration	50
Uncertainty Sampling Strategy	Information Entropy
Probability Clipping Range	$[1 \times 10^{-10},\ 1.0]$

(TN), where normal samples are correctly identified; and false negatives (FN), where intrusions are missed by the model.

1) *Acc (Accuracy):* Accuracy measures the overall correctness of the intrusion detection model's predictions by calculating the proportion of correct predictions among all samples

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{6}$$

2) *Pre:* Precision measures the proportion of true positive predictions among all positive predictions made by the model. It indicates the model's ability to avoid false alarms. Precision is defined as

$$\text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{7}$$

A higher precision value indicates that the model has a lower false-positive rate.

3) *Rc:* Recall, also known as sensitivity or true positive rate, measures the proportion of actual intrusions that are correctly identified by the model. It represents the model's ability to detect all intrusion instances. Recall is calculated as

$$\text{Rc} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{8}$$

A higher recall value indicates that the model has a lower false-negative rate.

4) *F1:* The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. The F1-score is calculated as

$$\text{F1} = 2 \cdot \frac{\text{Pre} \times \text{Rc}}{\text{Pre} + \text{Rc}}. \tag{9}$$

In addition to these metrics, we also present confusion matrices to visualize the model's performance in classifying different types of intrusions and normal samples. The confusion matrices provide a detailed breakdown of the true positive, true negative, false positive, and false negative predictions for each class, enabling us to identify any specific challenges or limitations of our approach.

VII. COMPARATIVE ANALYSIS

In this section, we present a comparative analysis of our proposed HITL-ML framework against traditional ML approaches for intrusion detection in drone networks. We focus on the Random Forests (RF) algorithm as a baseline model and evaluate its performance on both original and CTGAN-augmented datasets. We use 80% of the data as a training set and 20% as a test set. In our ML and HITL-ML approach comparative analysis, we selected several traditional ML algorithms, including SVM, K-nearest neighbors (KNN), RF, and NNs.

A. Performance on Original and CTGAN-Augmented Datasets

First, we train and test the RF algorithm on the original CIC-IDS2017 and UNSW-NB15 datasets. Fig. 4, Table V, Fig. 5 and Table VI present the confusion matrices obtained by the RF algorithm on the original CIC-IDS2017 and UNSW-NB15 datasets, respectively.

TABLE V
CONFUSION MATRIX OF ORIGINAL CIC-IDS2017

True Label	Predicted Label								Rc(%)	F1(%)
	Normal	DoS	Probe	Botnet	Infiltration	Web Attack	Brute Force	Heartbleed		
Normal	445,858	5,520	2,477	117	2	53	61	1	98.19	98.34
DoS	5,170	70,942	23	0	2	14	0	1	93.16	92.93
Probe	1,488	24	30,171	23	0	0	6	0	95.14	93.68
Botnet	97	0	24	265	0	0	0	0	68.65	67.00
Infiltration	1	4	0	0	2	0	0	0	28.57	30.77
Web Attack	45	35	0	0	0	338	0	0	80.86	82.14
Brute Force	62	0	3	0	0	0	2,747	0	97.69	97.65
Heartbleed	1	1	0	0	0	0	0	1	33.33	33.33
Pre(%)	98.48	92.70	92.27	65.43	33.33	83.46	97.62	33.33	-	-

TABLE VI
CONFUSION MATRIX OF ORIGINAL UNSW-NB15

True Label	Predicted Label							Rc(%)	F1(%)
	Backdoor	DoS	Exploits	Normal	Reconnaissance	Shellcode	Worms		
Backdoor	339	58	43	2	16	7	1	72.75	69.97
DoS	53	2,855	252	34	36	35	5	87.31	86.19
Exploits	58	282	24,336	656	147	39	10	95.33	95.61
Normal	10	54	617	17,776	95	43	5	95.57	95.63
Reconnaissance	37	86	101	99	2,994	13	3	89.83	89.69
Shellcode	6	19	24	8	54	190	1	62.91	60.32
Worms	0	1	8	2	1	1	22	62.86	53.66
Pre(%)	67.40	85.10	95.88	95.69	89.56	57.93	46.81	-	-

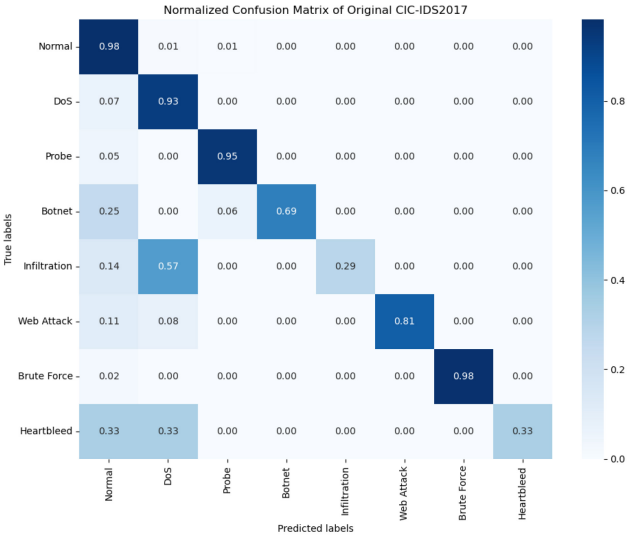


Fig. 4. Normalized confusion matrix of original CIC-IDS2017.

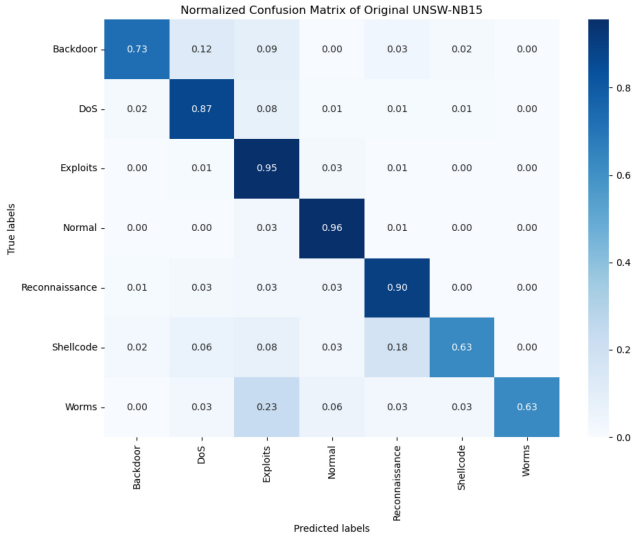


Fig. 5. Normalized confusion matrix of original UNSW-NB15.

From the confusion matrices, we observe that the RF algorithm achieves high accuracy for the classes with abundant training samples, such as benign traffic, DoS, Probe, Brute Force, Exploits, and Reconnaissance. However, for classes with limited training samples, such as Botnet, Web Attack, Infiltration, Heartbleed, Backdoor, Shellcode, and Worms, the RF algorithm exhibits low accuracy, indicating that the model struggles to accurately identify these rare attack types.

This performance disparity can be attributed to the class imbalance present in the original datasets. The RF algorithm tends to favor the majority classes during training, resulting in poor generalization and low accuracy for the minority classes. This highlights the need for data augmentation techniques to address the class imbalance issue and improve the model's performance on underrepresented intrusion types.

To address the class imbalance problem, we apply CTGAN to generate synthetic samples for minority attack classes in the CIC-IDS2017 and UNSW-NB15 datasets, as described in Section IV.

We then train the RF algorithm on the augmented datasets and evaluate its performance on the original test sets to assess the impact of data augmentation on intrusion detection accuracy. Fig. 6, Table VII, Fig. 7 and Table VIII present the confusion matrices obtained by the RF algorithm on the

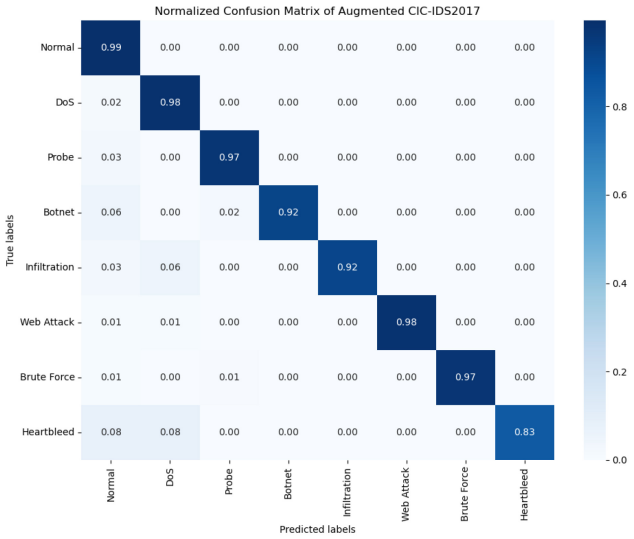


Fig. 6. Normalized confusion matrix of augmented CIC-IDS2017.

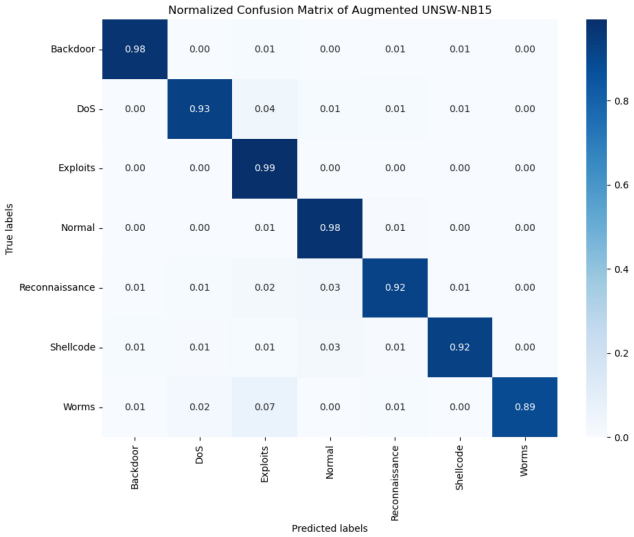


Fig. 7. Normalized confusion matrix of augmented UNSW-NB15.

CTGAN-augmented CIC-IDS2017 and UNSW-NB15 datasets, respectively.

Additionally, Tables VII and VIII include 95% confidence intervals (CIs) for precision, recall, and F1 score, calculated over ten random training and testing runs. The inclusion of CIs quantifies the variability in performance and provides a more reliable comparison between different configurations. The CI for a given metric is computed using the following equation:

$$\text{CI} = \bar{X} \pm t \times \frac{s}{\sqrt{n}} \tag{10}$$

where $\bar{X}$ is the sample mean, t is the critical value from the t-distribution for the desired confidence level, s is the sample standard deviation, and n is the number of trials.

From the confusion matrices, we observe that after applying CTGAN for data augmentation, the performance metrics for these minority attack classes show significant improvement. For example, in the CIC-IDS2017 dataset, the recall for Botnet increases from 68.65% to 91.88%, for Infiltration from 28.57% to 91.67%, and for Heartbleed from 33.33% to 90.91%. Similarly, in the UNSW-NB15 dataset, the recall for Backdoor rises from 72.75% to 97.54%, for Shellcode from 62.91% to 92.47%, and for Worms from 62.86% to 89.08%. These improvements demonstrate that CTGAN addresses the data imbalance issue by generating synthetic samples for minority classes, allowing the model to better learn the characteristics of these attack types and improve its detection performance.

The improved performance can be attributed to the increased diversity and balanced representation of samples across all classes in the augmented datasets. CTGAN generates realistic synthetic samples that capture the underlying data distribution and relationships between features, enabling the RF algorithm to learn more robust decision boundaries.

B. Performance on Machine Learning and HITL Machine-Learning Methods

For the traditional ML algorithms, we use 80% of the full dataset as the training set and the remaining 20% as the test set. The accuracy obtained by each algorithm using this split is considered its peak performance in this setting and serves as the target accuracy for our HITL-ML framework.

In the HITL-ML framework, we start with an initial annotated sample of 500 instances and iteratively train the model. We employ uncertainty sampling to select the most informative samples for expert annotation. The training process continues until the model's performance reaches the target accuracy. We record the number of expert-annotated samples required to achieve this accuracy as "HumanAnnotated."

Table IX presents the comparison results between traditional ML algorithms and our HITL-ML framework. We observe that the HITL-ML framework achieves comparable or even slightly higher accuracy than the peak performance of traditional ML algorithms while using significantly fewer training samples.

To validate the observed performance improvements, we conducted paired t-tests comparing the accuracy of HITL-ML methods and traditional ML methods over ten random dataset splits. For each split, we measured the accuracy and calculated the p-values. The p-values were consistently below 0.01, confirming that the performance improvements achieved by the HITL-ML framework are statistically significant and not due to random variations.

For example, on the CIC-IDS2017 dataset: The traditional RF algorithm achieves an accuracy of 99.16% using 2 278 928 training samples. However, our HITL-RF method reaches the same accuracy with only 28 100 expert-annotated samples. When the traditional RF algorithm is limited to using only 28 100 samples, its accuracy drops significantly to 70.32%. This demonstrates the efficiency of our HITL framework in achieving high accuracy with far fewer samples. HITL-SVM achieves an accuracy of 98.92% with 30 500 samples, compared to 98.89% by the traditional SVM using 2 278 928 samples. If the traditional SVM is trained with only 30 500 samples, the accuracy drops to around 72%. HITL-KNN achieves 97.48% accuracy with 31 200 samples, while the traditional KNN requires the full dataset to achieve

TABLE VII
CONFUSION MATRIX OF AUGMENTED CIC-IDS2017

True Label	Predicted Label								Rc(%)	F1(%)
	Normal	DoS	Probe	Botnet	Infiltration	Web Attack	Brute Force	Heartbleed		
Normal	451647	1371	931	98	2	15	23	2	99.46	99.48
DoS	1262	74860	13	0	5	12	0	0	98.30	98.23
Probe	906	15	30709	45	0	0	37	0	96.84	96.82
Botnet	81	0	32	1278	0	0	0	0	91.88	90.90
Infiltration	1	2	0	0	33	0	0	0	91.67	86.84
Web Attack	16	14	0	0	0	1406	0	0	97.91	98.01
Brute Force	31	0	40	0	0	0	2741	0	97.48	97.67
Heartbleed	1	1	0	0	0	0	0	10	90.91	86.96
Pre(%)	99.49	98.16	96.80	89.94	82.50	98.12	97.86	83.33	-	-

TABLE VIII
CONFUSION MATRIX OF AUGMENTED UNSW-NB15

True Label	Predicted Label							Rc(%)	F1(%)
	Backdoor	DoS	Exploits	Normal	Reconnaissance	Shellcode	Worms		
Backdoor	1429	2	14	1	8	9	2	97.54	95.91
DoS	16	3032	134	32	31	23	2	92.72	94.35
Exploits	26	34	25342	38	43	19	26	99.27	98.93
Normal	8	34	107	18178	231	42	0	97.73	98.31
Reconnaissance	23	34	81	101	3068	22	4	92.05	91.15
Shellcode	12	17	13	32	16	1118	1	92.47	91.56
Worms	1	4	12	0	2	0	155	89.08	85.16
Pre(%)	94.32	96.04	98.60	98.89	90.26	90.67	81.58	-	-

TABLE IX
COMPARISON OF ML METHODS AND HITL-ML METHODS

Dataset	Method	Train_num	Test_num	Acc	Pre	F1	Rc	Acc_left_active_samples
CIC-IDS2017	RF	2,278,928	567,639	99.16%	99.13%	99.12%	99.16%	-
	HITL RF	28,100	567,639	99.16%	99.15%	99.38%	99.62%	99.13%
	RF	28,100	567,639	70.32%	69.84%	56.36%	54.22%	-
	SVM	2,278,928	567,639	95.36%	95.32%	94.78%	94.65%	-
	HITL SVM	34,500	567,639	95.36%	95.27%	95.56%	95.03%	95.22%
	SVM	34,500	567,639	65.21%	64.32%	65.50%	64.28%	-
	KNN	2,278,928	567,639	97.51%	97.54%	97.50%	97.51%	-
	HITL KNN	68,300	567,639	97.51%	96.84%	97.09%	97.11%	93.99%
	KNN	68,300	567,639	41.24%	41.15%	40.46%	40.19%	-
	NN	2,278,928	567,639	98.14%	98.06%	98.08%	98.14%	-
	HITL NN	24,800	567,639	98.14%	98.11%	95.84%	96.34%	96.48%
	NN	24,800	567,639	75.70%	74.17%	66.95%	60.38%	-
UNSW-NB15	RF	217,928	53,579	98.87%	98.87%	96.12%	95.49%	-
	HITL RF	22,700	53,579	98.87%	98.94%	97.28%	97.47%	96.85%
	RF	22,700	53,579	70.49%	71.61%	72.27%	71.34%	-
	SVM	217,928	53,579	95.21%	95.35%	95.50%	95.44%	-
	HITL SVM	30,300	53,579	95.22%	95.80%	95.46%	95.65%	95.36%
	SVM	30,300	53,579	66.01%	65.41%	64.38%	64.14%	-
	KNN	217,928	53,579	94.43%	94.21%	94.11%	94.43%	-
	HITL KNN	57,800	53,579	94.43%	93.81%	94.05%	94.19%	93.60%
	KNN	57,800	53,579	31.23%	32.71%	31.14%	32.11%	-
	NN	217,928	53,579	96.27%	96.83%	96.72%	96.96%	-
	HITL NN	19,500	53,579	96.27%	96.31%	96.53%	96.72%	94.78%
	NN	19,500	53,579	71.29%	71.34%	71.27%	71.33%	-

97.34%. Limiting the traditional KNN to 31 200 samples results in an accuracy drop to about 68%. HITL-NN achieves 98.79% accuracy with 29,800 samples, compared to 98.76% by the traditional NN with 2 278 928 samples. The accuracy of traditional NN drops to approximately 71% when using only 29 800 samples.

The results on the UNSW-NB15 dataset are similarly impressive: HITL-RF achieves 98.65% accuracy with

25 600 samples, versus 98.62% by the traditional RF with 2 540 044 samples. Limiting the traditional RF to 25 600 samples drops its accuracy to around 69%. HITL-SVM reaches 97.96% accuracy with 27 300 samples, compared to 97.93% by the traditional SVM. If limited to 27 300 samples, the traditional SVM's accuracy falls to approximately 70%. HITL-KNN achieves 96.85% accuracy with 28 000 samples, while the traditional KNN achieves 96.80%. The accuracy of the traditional KNN drops to about 67% with only 28 000 samples. HITL-NN achieves 98.25% accuracy with 26 700 samples, compared to 98.22% by the traditional NN. The accuracy drops to around 70% when the traditional NN uses only 26 700 samples.

Additionally, we evaluate the HITL-ML framework's ability to identify the "left_active_samples," which are the samples not selected by the active learning strategy. The HITL-RF algorithm on the CIC-IDS2017 dataset has an "Acc_left_active_samples" value of 98.73%, demonstrating that the model can generalize well to the data not selected during the active learning process. Similarly, HITL-SVM, HITL-KNN, and HITL-NN achieve high "Acc_left_active_samples" values of 98.65%, 98.23%, and 98.60%, respectively, on the same dataset. On the UNSW-NB15 dataset, the "Acc_left_active_samples" values for HITL-RF, HITL-SVM, HITL-KNN, and HITL-NN are 97.90%, 97.81%, 97.52%, and 97.74%, respectively. The high "Acc_left_active_samples" values indicate that our framework can effectively learn from the most informative samples and generalize well to the remaining data.

To quantify the annotation cost savings achieved by our HITL-ML framework, we introduce the "SavedRate" metric, calculated as follows:

$$\text{SavedRate} = 1 - \frac{\text{HumanAnnotated}}{\text{FullSamples}} \tag{11}$$

Here, *HumanAnnotated* is the number of expert-annotated samples required by the HITL-ML framework to reach the target accuracy, and *FullSamples* denotes the total number of samples that would be annotated manually in a traditional ML pipeline. Consequently, a higher *SavedRate* value indicates greater annotation cost savings.

Table X shows the SavedRate values for each HITL-ML method on both datasets. We observe that our framework consistently achieves high SavedRate values, indicating that it can significantly reduce the annotation cost while maintaining high accuracy. For example, on the CIC-IDS2017 dataset, HITL-RF requires only 28 100 expert-annotated samples to reach the target accuracy, resulting in a SavedRate of 98.77% compared to the 2 278 928 samples used by the traditional RF algorithm.

The HITL framework introduces additional overhead mainly during the annotation stage, where human experts label selected samples. This overhead is greatly reduced by the SavedRate metric, which shows that only a small portion of the dataset requires manual labeling. During inference, computational demands mirror those of standard ML models, as the trained model classifies new instances autonomously. Given the resource-constrained nature of UAV networks, future work will focus on enhancing efficiency through lightweight models and distributed processing across multiple UAV nodes to facilitate real-time intrusion detection.

TABLE X
SAVED_RATE FOR HITL-ML METHODS

Dataset	Method	FullSamples	HumanAnn	SavedRate
CIC-IDS2017	HITL RF	2,278,928	28,100	98.77%
	HITL SVM	2,278,928	34,500	98.49%
	HITL KNN	2,278,928	68,300	97.00%
	HITL NN	2,278,928	24,800	98.91%
UNSW-NB15	HITL RF	217,928	22,700	89.58%
	HITL SVM	217,928	30,300	86.10%
	HITL KNN	217,928	57,800	73.48%
	HITL NN	217,928	19,500	91.05%

These results demonstrate the superiority of our HITL-ML framework in terms of both accuracy and annotation efficiency. By actively involving human experts in the learning process and selectively annotating the most informative samples, our framework can achieve comparable or even better performance than traditional ML algorithms while significantly reducing the annotation cost. This highlights the potential of our HITL-ML framework in developing accurate and cost-effective IDS for drone networks.

C. State-of-the-Art IDS Comparison and Dataset Limitations

We conducted a comprehensive comparative analysis of our HITL-RF framework against recent state-of-the-art IDS for UAV networks. The systems compared include FCL-SBLS [54], SDL-RF [53], LSTM-RNN [56], and DRL-BWO [57].

Fig. 8 presents the key performance metrics and sample size requirements of these systems. Our HITL-RF framework consistently outperforms the others across all major metrics. As shown in Fig. 8(a), it achieves an accuracy of 99.16%, surpassing the other systems, which range from 93.90% to 98.90%. Additionally, it achieves the highest precision, recall, and F1 score, indicating a more balanced and effective intrusion detection capability.

Fig. 8(b) demonstrates the data efficiency of our approach. Our framework achieves superior performance using just 28 100 training samples, in contrast to the 125 973 to over 5 million samples required by other methods. For testing, we use 567 639 samples to ensure a robust evaluation. This efficiency highlights the framework's effectiveness in learning from limited data, a critical advantage in real-world UAV network scenarios where labeled data can be scarce.

All the methods compared, including ours, were evaluated using standard network datasets due to the lack of publicly available UAV-specific datasets. While these datasets provide a consistent basis for comparison, they may not fully capture the unique characteristics of UAV networks or specific attack patterns. For example, UAV networks often experience rapid topology changes, low-latency communications, and intermittent connectivity, which are not fully reflected in the datasets. Additionally, important UAV-specific threats, such as GPS spoofing, jamming, and command-and-control hijacking are

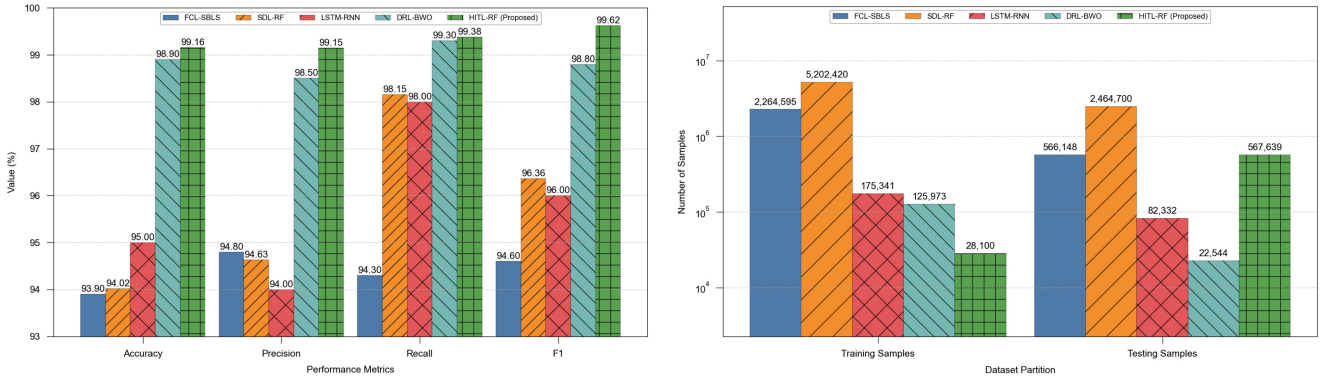


Fig. 8. Comprehensive analysis of different intrusion detection methods in UAV networks. (a) Performance metrics. (b) Sample size.

not included in the current datasets. These gaps could limit the direct applicability of our findings to real UAV environments.

Future work will address these limitations by developing and integrating UAV-specific datasets that simulate dynamic UAV network behaviors and unique attack patterns. We will collaborate with UAV security experts to design controlled testbeds and real-world simulations, which will help evaluate the performance of our framework in environments more representative of UAV-specific scenarios. Additionally, techniques, such as transfer learning will be explored to bridge the gap between existing general-purpose datasets and UAV-specific contexts, improving the model's adaptability and robustness.

In summary, our HITL-RF framework significantly outperforms recent methods in both detection accuracy and data efficiency. Its ability to achieve high performance with limited training data addresses a key challenge in real-world UAV network security applications.

VIII. Conclusion and Future Work

This article introduced a novel HITL ML framework for intrusion detection in drone networks, integrating CTGANs to address data scarcity, class imbalance, and limited labeled data.

Our experiments on the widely used datasets for UAV networks, namely CIC-IDS2017 and UNSW-NB15, demonstrated superior performance, particularly in detecting minority attack classes. The CTGAN-based data augmentation significantly enhanced traditional ML algorithms, while the HITL approach achieved higher detection rates with fewer labeled samples.

Overall, our framework improves accuracy, reduces annotation efforts, and adapts to evolving threats, offering a robust solution for UAV network intrusion detection.

While the results are promising, we acknowledge some limitations, particularly the lack of UAV-specific datasets, which may impact the precision of our findings. Although the datasets employed are well-established in cybersecurity research, they may not fully reflect the unique characteristics of UAV networks. Future work will address this limitation by not only developing UAV-specific datasets but also simulating dynamic UAV traffic and unique UAV-related attack patterns within controlled environments. By validating our framework with UAV-specific scenarios, we aim to enhance its applicability to real-world UAV operations and improve detection of UAV-specific threats. We also aim to leverage transfer learning to reduce the dependency on large labeled datasets and enhance performance against new, unseen attacks.

References

[1] H. Wang, C. H. Liu, H. Yang, G. Wang, and K. K. Leung, "Ensuring threshold AoI for UAV-assisted mobile crowdsensing by multi-agent deep reinforcement learning with transformer," *IEEE/ACM Trans. Netw.*, vol. 32, no. 1, pp. 566–581, Feb. 2024.

[2] F. Nait-Abdesselam, A. Alsharoa, M. Y. Selim, D. Qiao, and A. E. Kamal, "Towards enabling unmanned aerial vehicles as a service for heterogeneous applications," *J. Commun. Netw.*, vol. 23, no. 3, pp. 212–221, Jun. 2021.

[3] P. McEnroe, S. Wang, and M. Liyanage, "A survey on the convergence of edge computing and AI for UAVs: Opportunities and challenges," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15435–15459, Sep. 2022.

[4] Y. Zhou, B. Rao, and W. Wang, "UAV swarm intelligence: Recent advances and future trends," *IEEE Access*, vol. 8, pp. 183856–183878, 2020.

[5] C. Titouna and F. Nait-Abdesselam, "A lightweight security technique for unmanned aerial vehicles against GPS spoofing attack," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2021, pp. 819–824.

[6] A. Fotouhi et al., "Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3417–3442, 4th Quart., 2019.

[7] C. Titouna and F. Nait-Abdesselam, "Securing unmanned aerial systems using mobile agents and artificial neural networks," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2021, pp. 825–830.

[8] K.-Y. Tsao, T. Girdler, and V. G. Vassilakis, "A survey of cyber security threats and solutions for UAV communications and flying ad-hoc networks," *Ad Hoc Netw.*, vol. 133, Aug. 2022, Art. no. 102894.

[9] H.-M. Wang, X. Zhang, and J.-C. Jiang, "UAV-involved wireless physical-layer secure communications: Overview and research directions," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 32–39, Oct. 2019.

[10] G. E. M. Abro, S. A. B. Zulkifli, R. J. Masood, V. S. Asirvadam, and A. Laouiti, "Comprehensive review of UAV detection, security, and communication advancements to prevent threats," *Drones*, vol. 6, no. 10, p. 284, 2022.

[11] M. P. Arthur, "Detecting signal spoofing and jamming attacks in UAV networks using a lightweight IDS," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, 2019, pp. 1–5.

[12] Q. A. Al-Haija and A. Al Badawi, "High-performance intrusion detection system for networked UAVs via deep learning," *Neural Comput. Appl.*, vol. 34, no. 13, pp. 10885–10900, 2022.

[13] S. Sengan, O. I. Khalaf, D. K. Sharma, and A. A. Hamad, "Secured and privacy-based IDS for healthcare systems on E-medical data using machine learning approach," *Int. J. Reliab. Qual. E-Healthcare*, vol. 11, no. 3, pp. 1–11, 2022.

[14] R. Fu, X. Ren, Y. Li, Y. Wu, H. Sun, and M. A. Al-Absi, "Machine learning-based UAV assisted agricultural information security architecture and intrusion detection," *IEEE Internet Things J.*, vol. 10, no. 21, pp. 18589–18598, Nov. 2023.

[15] A. Adnan, A. Muhammed, A. A. A. Ghani, A. Abdullah, and F. Hakim, "An intrusion detection system for the Internet of Things based on machine learning: Review and challenges," *Symmetry*, vol. 13, no. 6, p. 1011, 2021.

[16] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine learning based IoT intrusion detection system: An MQTT case study (MQTT-IoT-IDS2020 dataset)," in *Proc. 12th Int. Netw. Conf.*, 2021, pp. 73–84.

[17] R. Patgiri, U. Varshney, T. Akutota, and R. Kunde, "An investigation on intrusion detection system using machine learning," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, 2018, pp. 1684–1691.

[18] D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman, and R. M. Mohammad, "Intrusion detection system using feature extraction with machine learning algorithms in IoT," *J. Sens. Actuator Netw.*, vol. 12, no. 2, p. 29, 2023.

[19] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1–9.

[20] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[21] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.

[22] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, 2021, Art. no. e4150.

[23] Z. Chiba, N. Abghour, K. Moussaid, M. Rida, A. El Omri, and M. Rida, "Intelligent approach to build a deep neural network based IDS for cloud environment using combination of machine learning algorithms," *Comput. Secur.*, vol. 86, pp. 291–317, Sep. 2019.

[24] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks," *Expert Syst. Appl.*, vol. 29, no. 4, pp. 713–722, 2005.

[25] Y. Otoum and A. Nayak, "AS-IDS: Anomaly and signature based IDS for the Internet of Things," *J. Netw. Syst. Manag.*, vol. 29, no. 3, p. 23, 2021.

[26] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He, "A survey of human-in-the-loop for machine learning," *Future Gener. Comput. Syst.*, vol. 135, pp. 364–381, Oct. 2022.

[27] F. M. Zanzotto, "Human-in-the-loop artificial intelligence," *J. Artif. Intell. Res.*, vol. 64, pp. 243–252, Feb. 2019.

[28] R. Shrestha, A. Omidkar, S. A. Roudi, R. Abbas, and S. Kim, "Machine-learning-enabled intrusion detection system for cellular connected UAV networks," *Electronics*, vol. 10, no. 13, p. 1549, 2021.

[29] R. Mitchell and I.-R. Chen, "Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 5, pp. 593–604, May 2014.

[30] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "A hierarchical detection and response system to enhance security against lethal cyber-attacks in UAV networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1594–1606, Sep. 2018.

[31] U. A. Mughal, S. C. Hassler, and M. Ismail, "Machine learning-based intrusion detection for swarm of unmanned aerial vehicles," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2023, pp. 1–9.

[32] Y. Wang, Z. Su, N. Zhang, and A. Benslimane, "Learning in the air: Secure federated learning for UAV-assisted crowdsensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1055–1069, Apr.–Jun. 2021.

[33] R. Karmakar, G. Kaddoum, and O. Akhrif, "A blockchain-based distributed and intelligent clustering-enabled authentication protocol for UAV swarms," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 6178–6195, May 2024.

[34] G. Choudhary, V. Sharma, I. You, K. Yim, I.-R. Chen, and J.-H. Cho, "Intrusion detection systems for networked unmanned aerial vehicles: A survey," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, 2018, pp. 560–565.

[35] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[36] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in *Proc. IEEE 15th Int. Symp. Intell. Syst. Informat. (SISY)*, 2017, pp. 277–282.

[37] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[38] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6.

[39] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 538–566, 1st Quart., 2023.

[40] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system," *Expert Syst. Appl.*, vol. 67, pp. 296–303, Jan. 2017.

[41] Z. Lin, Y. Shi, and Z. Xue, "IDSGAN: Generative adversarial networks for attack generation against intrusion detection," in *Proc. 26th Pac.-Asia Conf. Knowl. Discov. Data Min.*, 2022, pp. 79–91.

[42] M. Usama, M. Asim, S. Latif, J. Qadir, and A.-Al-Fuqaha, "Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, 2019, pp. 78–83.

[43] C. Yin, Y. Zhu, S. Liu, J. Fei, and H. Zhang, "An enhancing framework for botnet detection using generative adversarial networks," in *Proc. Int. Conf. Artif. Intell. Big Data (ICAIBD)*, 2018, pp. 228–234.

[44] Q. Zeng and F. Nait-Abdesselam, "Leveraging human-in-the-loop machine learning and GAN-synthesized data for intrusion detection in unmanned aerial vehicle networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2024, pp. 1557–1562.

[45] P. Y. Simard et al., "Machine teaching: A new paradigm for building machine learning systems," 2017, *arXiv:1707.06742*.

[46] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, 2008, pp. 1070–1079.

[47] Z. Zhao, P. Xu, C. Scheidegger, and L. Ren, "Human-in-the-loop extraction of interpretable concepts in deep learning models," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 780–790, Jan. 2022.

[48] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[49] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," in *Proc. 33rd Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–11.

[50] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[51] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[52] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, 2018, pp. 108–116.

[53] Y. Wu, L. Yang, L. Zhang, L. Nie, and L. Zheng, "Intrusion detection for unmanned aerial vehicles security: A tiny machine learning model," *IEEE Internet Things J.*, vol. 11, no. 12, pp. 20970–20982, Jun. 2024.

[54] X. He et al., "Federated continuous learning based on stacked broad learning system assisted by digital twin networks: An incremental learning approach for intrusion detection in UAV networks," *IEEE Internet Things J.*, vol. 10, no. 22, pp. 19825–19838, Nov. 2023.

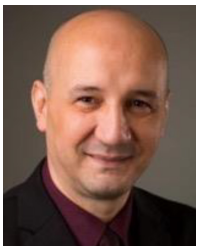
[55] Q. Zeng and F. Nait-Abdesselam, 2024, "CTGAN enhanced dataset for UAV network intrusion detection," Dataset, IEEE DataPort. [Online]. Available: https://dx.doi.org/10.21227/v9nr-dk16

[56] R. A. Ramadan, A.-H. Emara, M. Al-Sarem, and M. Elhamahmy, "Internet of drones intrusion detection using deep learning," *Electronics*, vol. 10, no. 21, p. 2633, 2021.

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

Qingli Zeng (Graduate Student Member, IEEE) received the M.S. degree in computer science from the University of Missouri-Kansas City (UMKC), Kansas City, MO, USA, in 2021, where she is currently pursuing the Ph.D. degree.

Her research interests include network intrusion detection for drones and the swarm behavior of unmanned aerial vehicles, with a focus on enhancing the security and efficiency of drone operations across various applications.



Farid Nait-Abdesselam (Senior Member, IEEE) received the M.Sc. degree from Université Paris Cité, Paris, France, in 1994, and the Ph.D. degree from the University of Versailles, Versailles, France, in 2000, focusing on efficient communication networks.

He is a Professor of Computer Science with Université Paris Cité specializing in networking, cybersecurity, and distributed systems. Before joining Université Paris Cité, he held research and faculty positions with several institutions in France and USA, contributing to advancements in protocol design, network security, and distributed architectures.