Real-time UAV Network Intrusion Detection Using Active Learning and Generative Adversarial Networks

Qingli Zeng, Kailynn Barnt, Luke Ragan, Farid Nait-Abdesselam
School of Science and Engineering
University of Missouri - Kansas City
Kansas City, United States
zq6mw@umsystem.edu,kbrh3@mail.umkc.edu,lorvd9@mail.missouri.edu,naf@umkc.edu

***Abstract*—In recent years, the rapid expansion of Unmanned Aerial Vehicle (UAV) networks has unveiled new security vulnerabilities, making intrusion detection in these networks a top priority. Despite the demonstrated success of various machine learning algorithms in network intrusion detection, these methods often falter under adversarial conditions or when confronted with advanced attack strategies. Further complicating matters, real-world UAV data streams are relentless, vast, and storage-intensive, rendering the storage of all data nearly impossible. Moreover, a notable gap in the research is the absence of universally recognized datasets specifically tailored for UAV network intrusion detection. Considering the above points, this paper presents an innovative approach for real-time intrusion detection. This method employs Generative Adversarial Networks (GANs) to produce synthetic intrusion detection data. Following this, a stream-based active learning mechanism is applied for instantaneous monitoring and action. The hallmark of our approach lies in the novel integration of stream-based modeling with GANs. Our proposed system leverages active learning, working seamlessly alongside domain experts to adapt to ever-changing data scenarios, ensuring superior detection accuracy while substantially reducing false positive rates. When benchmarked against conventional intrusion detection paradigms, our approach demonstrates a marked improvement in detection precision. Future research endeavors will delve into further refining this method, embedding it into prevailing security frameworks, and examining its potential application to other complex network spheres.**

***Index Terms*—Intrusion detection, Active learning, Generative Adversarial Networks**

I. Introduction

Unmanned Aerial Vehicles (UAVs), commonly known as drones, have rapidly evolved from niche military tools to ubiquitous assets spanning diverse applications, from surveillance, delivery, agricultural surveying, to more advanced commercial ventures [1] [2]. This burgeoning adoption, while testifying to the versatility and efficacy of UAVs, has also highlighted potential threats to their operation. In particular, as UAVs become increasingly interconnected, network vulnerabilities arise, potentially jeopardizing the intended functions of these vehicles and posing security threats in their operational environment [3].

UAV networks, despite their immense potential, are beleaguered by a multitude of security vulnerabilities. Common threats encompass spoofing attacks, malicious data injections, and signal jamming, to name a few [4]. The community has taken steps to mitigate these issues, with measures ranging from robust encryption techniques to physical isolation. A significant stratagem among these is the Intrusion Detection System (IDS), Its role in the UAV ecosystem can't be understated. The IDS operates much like a digital immune system. Continuously monitoring the vast volumes of network traffic, it is primed to detect anomalies from the subtlest to the most glaring. By flagging these irregularities in real-time, the IDS not only alerts operators but often initiates predefined countermeasures, ensuring the UAV's mission and the integrity of its data remain uncompromised. The gravitas of a robust and responsive IDS in ensuring the secure operation of UAV networks in an increasingly complex threat landscape is undeniably pivotal.

But what makes UAV networks particularly challenging? Primarily, the inherent intricacies and dynamic nature of UAV operations often lead to patterns of network traffic that are more complex than those found in traditional wired or even wireless networks. The real-time decision making, incessant data streams, and the ever-adaptive nature of these systems make intrusion detection a task that is both critical and complex. However, an even more daunting challenge that researchers face today in UAV network security is the glaring absence of specialized datasets. Unlike many other domains where ample datasets have been curated, standardized, and made available for research, the UAV domain lacks a widely-accepted, specific dataset for intrusion detection [5]. This paucity can cripple the advancement of security measures, as data is the lifeblood of modern machine learning and artificial intelligence algorithms. In the face of these multifarious challenges, an integrative approach that melds the nuanced discernment of human experts with the computational agility of algorithms could emerge as the beacon lighting the path forward in UAV intrusion detection.

Given this challenging scenario, our research aims to provide a robust solution. Instead of waiting for vast real-world datasets to be compiled, which may take years and still lack variety, we propose a novel approach: the synthesis of intrusion detection data using GANs. GANs, with their capability to generate data mirroring real-world complexities, can serve as a formidable tool to bridge this data gap. Combined with a stream-based active learning method, our approach offers real-time monitoring, adaptability, and precision in intrusion detection for UAV networks. By putting the human in the loop, we ensure the system's decisions are not just based on algorithmic outcomes, but also benefit from human expertise. This collaboration between human and machine becomes particularly vital in ambiguous scenarios where algorithmic certainty might be lacking.

GANs stand as a cuttingedge facet within the broad spectrum of artificial intelligence models [6]. The uniqueness of GANs can be distilled into their remarkable ability to craft synthetic data, a capability especially pertinent given the data scarcity issues in UAV networks. The operational brilliance of GANs can be attributed to their twin neural network composition:

a) Generator: This entity, akin to a master forger, crafts synthetic data with the ambition of making it as indistinguishable from genuine data as possible. In the context of UAVs, the generator aims to produce data that mirrors genuine network traffic or behaviors [7].

b) Discriminator: Serving as the vigilant gatekeeper, the discriminator's role is to discern the authentic from the fabricated. It distinguishing real data from the synthetic samples crafted by the generator. When it identifies a deception, it guides the generator in refining its creation process [8].

This dynamic interplay, resembling a high-stakes chess match, ensures a constant evolutionary process. As the generator refines its synthetic production techniques, the discriminator simultaneously hones its discernment skills. The iterative cycle continues until the synthetic data resonates so closely with real-world UAV network behaviors that distinguishing between the two becomes a formidable task. The strength of GANs lies not just in their ability to create, but in their adaptability and iterative learning. They continuously evolve, ensuring the generated data is not static but dynamic, reflecting the nuanced changes real-world datasets often undergo.

Active learning(AL) emerges as an innovative paradigm within the machine learning sphere, characterized by its dynamic approach to model training. Instead of passively consuming vast datasets, models in active learning proactively seek out data points that, once labeled by human experts, provide the maximum informational benefit [9]. This strategy not only curtails the exhaustive efforts associated with data labeling but often results in constructing more accurate models with a fraction of the data conventionally required.

Building upon this foundational concept, stream-based active learning comes into play when handling the incessant flow of data, emblematic of systems like UAV networks [10]. With continuous data streams inundating the system, the onus is on the model to discern which data merits the attention and annotation of domain experts [11]. This real-time filtration process ensures efficient use of resources while optimizing the learning process.

A salient feature of this approach is the integral role of the human in the loop [12]. By intertwining human expertise with algorithmic processing, the framework achieves a harmonious balance [13]. It leverages computational prowess for high-speed data handling while drawing upon human intuition for nuanced decision-making. Given the volatile and dynamic nature of UAV networks, where anomalies might not always fit a predefined mold, this amalgamation of human discernment and machine efficiency becomes indispensable. It ensures that the intrusion detection mechanism remains both robust and adaptive, encapsulating the evolving challenges and intricacies specific to UAV environments [14].

The main contributions of this research are: (1) We harness the potential of Generative Adversarial Networks to create synthetic intrusion detection data for UAVs networkings, a solution that robustly addresses the challenges of data scarcity in UAV networks. Furthermore, our research pioneers the integration of GANs with a stream-based active learning framework, ensuring real-time, flexible, and accurate intrusion detection. (2) In a distinctive maneuver, our methodology explicitly integrates human expertise with machine learning, marrying human intuition with algorithmic prowess, enhancing both the fidelity and responsiveness of the intrusion detection process.

The remainder of this paper is as follows. Section 2 provides an overview of a related work in the fields of UAS network intrusion detection, active learning, and GANs model methodologies. Section 3 presents the proposed GANs+AL framework, while Section 4 describes the experimental setup and dataset used for the evaluation. Section 5 discusses the results, and Section 6 concludes the paper and outlines future research directions.

II. Related work

The evolving landscape of UAV network security has seen diverse methodologies striving to address the inherent vulnerabilities of such systems [15]. As UAV networks have risen in popularity, they've found pivotal roles in various sectors including agriculture [16], logistics [17], emergency response [18] [19], environmental monitoring [20], and entertainment [21]. With their widening applications, the imperative to safeguard these networks from potential threats has intensified. A broad categorization of existing methods in UAV intrusion detection provides a scaffold on which specific, contemporary techniques can be evaluated [22]. This section attempts to traverse this evolving research landscape, and endeavors to summarize the progression of research in this domain.

Intrusion detection systems for UAV networks have largely been bifurcated into:

a) Signature-Based Methods: These techniques, explored extensively by Sun [23], rely on known patterns of malicious behavior. While effective against recognized threats, they often falter when confronted with novel attack vectors.

b) Anomaly-Based Methods: In contrast to signature-based approaches, these methodologies, as discussed by Leandro [24], detect intrusions by identifying deviations from established behavioral norms. Though potentially powerful against new threats, they suffer from higher false-positive rates.

c) Hybrid Methods: Merging the strengths of both the aforementioned categories, hybrid systems, as demonstrated by Jean Philippe [25], aim to provide a balanced intrusion detection strategy. However, they inherit the complexity and computational costs from both parents.

Machine learning, with its rich tapestry of algorithms and techniques, has deeply penetrated the realm of UAV intrusion detection. Traditional machine learning techniques, such as Support Vector Machines (SVM) [26], Random Forests [27], and K-Nearest Neighbors (KNN) [28], have historically laid the groundwork in this domain. These methods have proven efficient in capturing linear and some non-linear patterns within UAV network traffic, with SVM and Random Forests often praised for their robustness against overfitting and KNN for its simplicity.

In more recent times, deep learning, exemplified by Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [29] [30] [31], has emerged as a game-changer [32]. These networks have displayed a particular aptitude for handling complex patterns and high-dimensional data intrinsic to UAV networks.

However, the journey is riddled with challenges [33]. Machine learning models, while powerful, often require diligent parameter tuning. Additionally, susceptibility to adversarial attacks across these models remains a concern, potentially undermining UAV network security [34].

A consistent limitation observed across these studies is the choice of datasets. Despite employing cutting-edge methodologies, many researchers tend to lean on generic datasets rather than using authentic, real-world UAV network traffic. The problem of intrusion detection in drones is still new, and to our knowledge, there is no specific dataset related to drone attacks [35]. This is often compounded by the fact that a significant portion of genuine UAV data is classified or proprietary, making it inaccessible for broad research endeavors.

Given these challenges, by leveraging the power of Generative Adversarial Networks (GANs), we aim to produce authentic-seeming UAV network data. Moreover, it harmonizes human expertise with our diverse machine learning palette, while ensuring transparency and reducing susceptibility to adversarial attacks.

III. Methodology

In our quest to address UAV intrusion detection effectively, the methodology we devised converges on two primary pivots: a robust synthetic data generation using Conditional Tabular Generative Adversarial Networks (CTGANs) and an amalgamation of various machine learning techniques to ensure accurate intrusion detection. Through these strategies, complemented by the integration of human expertise, we aim to tackle the complex and dynamic challenges inherent to UAV network security. Herein, we detail each segment of our approach.

A. Dataset Generation using GANs

The pressing need for realistic UAV network datasets compelled us to employ CTGANs for synthetic data generation [36]. Unlike traditional GANs, CTGANs are specifically designed to generate synthetic tabular data. They model the data distribution conditionally, allowing for better handling of discrete and continuous variables often found in network traffic data.

As show in Fig. 1, the CTGAN consists of the following components:

a) Generator: This component is responsible for producing data. Taking random noise as input, it generates synthetic tabular data, attempting to match the real data distribution. The conditional generator $G(z, cond)$ can be formally described as below.

$$\begin{cases} h_0 = z \oplus cond \\ h_1 = h_0 \oplus ReLU(BN(FC_{|cond|+|z| \rightarrow 256}(h_0))) \\ h_2 = h_1 \oplus ReLU(BN(FC_{|cond|+|z|+256 \rightarrow 256}(h_1))) \\ \hat{\alpha}_1 = tanh(FC_{|cond|+|z|+512 \rightarrow 1}(h_2)) \\ \hat{\beta}_i = gumbel_{0.2}(FC_{|cond|+|z|+512 \rightarrow m_i}(h_2)) \\ \hat{d}_i = gumbel_{0.2}(FC_{|cond|+|z|+512 \rightarrow |D_i|}(h_2)) \end{cases}$$

where z represents the random noise, and cond represents the conditional probability.

b) Discriminator: Working in tandem with the generator, the discriminator tries to differentiate between the real and synthetic datasets. It provides feedback to the generator, directing the latter to refine its data generation process. We use the PacGAN [37] framework with 10 samples in each pac to prevent mode collapse. The structure of the discriminator

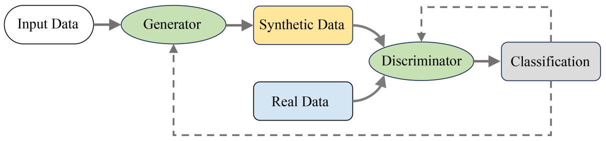

Figure 1. The architecture of GANs model

network $C(r_1, \ldots, r_{10}, cond_1, \ldots, cond_{10})$ can be formally described as below.

$$\begin{cases} h_0 = r_1 \oplus \ldots \oplus r_{10} \oplus cond_1 \oplus \ldots \oplus cond_{10} \\ h_1 = drop(leaky_{0.2}(FC_{10|r|+10|cond| \to 256}(h_0))) \\ h_2 = drop(leaky_{0.2}(FC_{256 \to 256}(h_1))) \\ C(\cdot) = FC_{256 \to 1}(h_2) \end{cases}$$

where r_i denotes an example.

c) Conditional Vectors: CTGANs also incorporate conditional vectors that guide the synthetic data generation based on given conditions or categories, ensuring that the generated data remains coherent and contextually relevant to specific UAV scenarios.

Consider a unique vector, termed as *cond*, constructed to encapsulate specific conditions associated with discrete data columns. When representing discrete columns $D_1, D_2, \ldots, D_{Nd}$, we utilize the one-hot encoding method, resulting in a set of vectors $d_1, d_2, \ldots, d_{Nd}$. Specifically, the i^{th} column's corresponding one-hot vector, di, is articulated as (1).

$$d_i = [d_i^{(k)}] \tag{1}$$

where k iterates from 1 through the cardinality of D_i, represented as $|D_i|$.

For each one-hot vector, there's a corresponding mask vector, denoted by m_i. This mask vector is defined as (2).

$$m_i = [m_i^{(k)}] \tag{2}$$

for k in the range 1 to $|D_i|$. The primary purpose of this mask vector is to embody the condition where a specific column, D_i^*, is equivalent to a specific value, k^*. Mathematically, this is expressed as (3).

$$m_i^{(k)} = \begin{cases} 1 & \text{if } i = i^* \text{ and } k = k^* \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Subsequently, our *cond* vector is formed by the bitwise XOR operation (denoted as $\oplus$) over all individual mask vectors shows in (4).

$$cond = m_1 \oplus m_2 \oplus \ldots \oplus m_{Nd} \tag{4}$$

As an illustrative example, consider two discrete columns: D_1 with values {1, 2, 3} and D_2 with values {1, 2}. To represent the condition $D_2 = 1$, the mask vectors derived are $m_1 = [0, 0, 0]$ and $m_2 = [1, 0]$. Concatenating these vectors yields the conditional vector $cond = [0, 0, 0, 1, 0]$.

Throughout the iterative process, the generator refines its outputs to generate data that closely resembles genuine UAV network traffic. By the end of the training phase, the discriminator's inability to distinguish between the real and synthetic data indicates the model's success. This synthetic dataset's inherent advantage lies in its capability to model UAV-specific intricacies, providing a comprehensive playground for testing and refining our subsequent machine learning models.

B. Machine Learning-based Intrusion Detection

The application of machine learning in UAV intrusion detection serves as an essential bridge between raw data interpretation and actionable insights for safeguarding UAV networks. Our architectural representation elucidates this detection process, guiding the reader through the various stages. Flow of the Intrusion Detection Process is illustrated in Fig. 2.

a) Data Preprocessings: Every insightful machine learning journey starts with clean, structured data. Raw UAV network traffic, including our CTGAN-synthesized entries, undergoes preprocessing. This stage involves normalization, feature extraction, and selection, ensuring the data's format is conducive to efficient learning.

b) Data Segmentation: Once we refine the dataset, it's imperative to segment it into training, validation, and testing subsets. This arrangement sets the foundation—train the model, optimize it, and finally, validate its prowess on untouched data.

c) Modeling and Training: With our training set in hand, we engage various machine learning algorithms to discern and memorize the data patterns. Their ultimate goal: distinguishing between benign and malicious traffic in the UAV network.

d) Validation and Refinement: The validation subset serves a pivotal role, it helps us refine our model. By adjusting hyperparameters and tuning certain settings during this phase, we sculpt our model, ensuring it's both adaptive and accurate.

e) Intrusion Assessment: Lastly, the test set steps into the spotlight. Here, we measure our models' performances, extracting metrics like accuracy, precision, and recall. It's the moment of truth, how well does the model classify data points as benign or malicious.

The stream based selective sampling is utilized when we do not have static data and the learner has to process a continuous stream of data. It was tailored specifically for real-time data evaluation, is indispensable in scenarios like UAV network intrusion detection, where data is incessantly lowing, and swift, accurate responses are pivotal. Fig. 3 depicted workflow of Stream-Based Active Learning.

f) Continuous Data Stream Acquisition: The machine learning algorithm constantly processes the dynamic inflow of UAV network traffic data. Each incoming data instance undergoes an initial assessment to determine its classification.

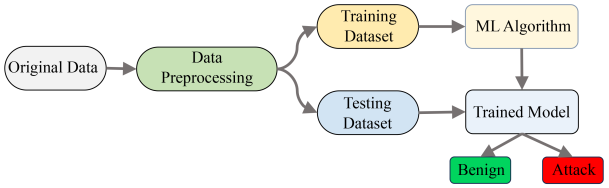

Figure 2. Machine learning based IDS framework

C. Stream-based active learning

a) Human-in-the-loop Annotation: Each classification is accompanied by a confidence score. When the model's confidence dips below a predefined threshold, indicating uncertainty, the specific data instance is flagged for further review. Using entropy as a measure of this confidence, we can quantify the certainty of these predictions. For multiclass classification, entropy gauges the uncertainty or disorder in the predictions. When given a data point with a set of predicted probabilities $P = \{p_1, p_2, ..., p_n\}$ across n classes, its entropy H is calculated as (5). Instances with notably high uncertainty, where entropic values surpass a given threshold are presented to human experts for closer examination.

$$H(P) = -\sum_{i=1}^{n} p_i \cdot \log_2(p_i) \tag{5}$$

The crux of entropy sampling lies in pinpointing those data instances for which the model's prediction carries maximum uncertainty, these are instances where the entropy values are maximal. By channeling human expertise specifically towards these high-entropy instances, we ensure that the most ambiguous and potentially enlightening instances are meticulously examined.

b) Model Update: After human experts provide their annotations, these labeled instances are funneled back into the learning algorithm. This feedback mechanism serves a dual purpose: it refines the model's understanding and enhances its future prediction capabilities.

The model then continues to the next data point in the stream, repeating the aforementioned steps. Over time, as it encounters and learns from more uncertain instances, the model's overall performance and confidence should see progressive enhancement.

IV. Implementation

Our proposed methodology harmoniously integrates the power of stream-based active learning with a selection of classical machine learning models. In this section, we detail the actual implementation processes from dataset generation to the comparative analysis of the performance of machine learning models within and outside the active learning framework.

A. Data Generation using CTGan

CTGAN, or Conditional Tabular Generative Adversarial Network, stands as a refined extension of traditional GANs. It's tailored explicitly for generating synthetic tabular data, making it a fitting choice for our objectives.

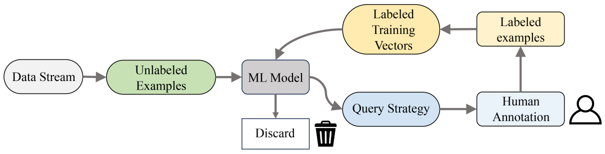

Figure 3. Stream-based active learning framework

For the foundation of our synthetic data generation, we utilized the CIC-IDS2017 dataset [38]. CIC-IDS2017, developed by the Canadian Institute for Cybersecurity, is a reputable dataset in intrusion detection research. It's designed to mimic real-world network traffic, encapsulating both benign activities and a wide spectrum of malicious attacks. This broad coverage, reflecting actual attack vectors like DDoS, Brute Force, XSS, and SQL Injection. The dataset includes several features, such as packet size, timestamp, source and destination IP addresses, protocol information, and flow statistics.

Fig. 4 compares the distribution shapes of individual columns in the synthetic data to their corresponding columns in the real data. It quantifies how closely the synthetic data captures the distribution of real data for every attribute or feature. It can be calculated as (6). Where $Similarity$ is a function that measures the similarity between two distributions, and N is the number of columns or features. S means synthetic data, R means real data.

$$Score = \frac{1}{N}\sum_{i=1}^{N} similarity(R_i, S_i) \tag{6}$$

Generally, a score nearing 1 suggests near-perfect replication, while a score around 0.5 points towards an arbitrary match akin to random guessing. With an average score of 0.85, it suggests that the CTGAN has performed commendably in replicating the distribution of most columns.

Fig. 5 compares the central tendency (mean) and spread (standard deviation) of the log-transformed numeric data columns between the real and synthetic datasets. It can be calculated as Equation (7). Where $Value_i$ represents each data point in the numeric column and N is the number of data points.

$$\begin{cases} LogM = \frac{1}{N}\sum_{i=1}^{N}\log(Value_i) \\ LogSTD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(Value_i) - LogM)^2} \end{cases} \tag{7}$$

Similar to the mean plot, an ideal situation here would have all data points lie on the diagonal, indicating an exact match between the variability in the real and synthetic datasets. Points below the diagonal indicate features where the synthetic data has less variability (smaller standard deviation) than the real data. Conversely, points above the diagonal represent features where the synthetic data exhibits more variability than the real data. It's evident that our synthetic data hews closely to the real dataset's mean and variability, ensuring congruence in fundamental statistical properties.

There are some example features shown in Fig. 6, this visual metric evaluates the cumulative distribution of values for each feature in both synthetic and real datasets. It can be calculated as (8).

$$CumSum_i = \sum_{j=1}^{i} FeatureValue_j \tag{8}$$

By comparing the cumulative sums of the real vs. synthetic datasets graphically, one can gauge how well the synthetic data replicates the incremental value distribution of each feature from the real dataset. Graphs should closely mirror each other for both synthetic and real datasets.

Conclusively, through the confluence of visual plots and these analytical metrics, we can confidently affirm the high fidelity and representational quality of our generated synthetic dataset. The harmonious integration of this data with the real CIC-IDS2017 dataset serves as a bedrock for efficacious intrusion detection modeling.

B. Model Implementation

We incorporated several classical machine learning models into our active learning framework: Random Forest (RF); decision trees; Support Vector Machine (SVM); K-Nearest Neighbors (KNN); Deep Learning and XGBoost. Within the stream-based active learning paradigm, these models adapt in real-time using expert-labeled instances, refining their decision-making efficacy.

We subjected each machine learning model to two scenarios:

a) Within the active learning loop: Here, models exploit both their intrinsic prediction capability and the guidance from human expert-labeled instances.

b) Classic machine learning based method: Here, models rely solely on their intrinsic predictive capacities without any external guidance.

Comparisons were made to discern the efficiency, accuracy, and real-time adaptability of models in both scenarios. This systematic comparison aims to showcase the advantages and potential enhancements brought about by the integration of human expertise via active learning.

C. Stream-based Active Learning Framework

Stream-based Active Learning Algorithm is illustrated in Algorithm 1, our active learning framework operates in a streamed fashion, sequentially processing each data point from the input stream. The general workflow is: Model processes an incoming data point. Based on the model's prediction uncertainty, calculated via Entropy Sampling, a decision is made: If uncertainty surpasses a threshold, the human expert is queried for a label. This newly labeled data point is subsequently utilized to update the model on-the-fly.

Algorithm 1 Stream-based Active Learning Loop

```
Initialize model $M$
for each data_point in data_stream do
    P = $M$.P(data_point)
    entropy = $-\sum_i \mathrm{P}_i \times \log(\mathrm{P}_i)$
    if entropy > threshold then
        true_label = request_label_from_human(data_point)
        update_training_set(data_point, true_label)
        $M$ = retrain_model($M$, updated_training_set)
    end if
end for
```

D. Performance Metrics

To evaluate the effectiveness of our intrusion detection models, we've selected a suite of performance metrics. These metrics will provide a comprehensive understanding of the model's capabilities across various facets. Here, we define each metric using standard notations: TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives).

Accuracy is the proportion of correctly classified instances out of the total instances in the dataset. As per (9), accuracy gives a holistic overview of the model's prediction capabilities.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

Precision gauges the proportion of positive identifications that are truly positive. It's especially vital in contexts where False Positives bear significant costs. Equation (10) denotes the precision of our model, highlighting its capability to avoid false positives.

$$P = \frac{TP}{TP + FP} \tag{10}$$

Recall, or Sensitivity, calculates the fraction of actual positives that the model correctly identified, making it vital in contexts where missing a True Positive (False Negatives) is costly. Referencing (11), Recall indicates how many of the actual positives the model managed to capture.

$$Rc = \frac{TP}{TP + FN} \tag{11}$$

The F1-Score is the harmonic mean of Precision and Recall. It becomes crucial when there's an uneven class distribution in the dataset. By (12), the F1-Score ensures we don't lean too heavily towards Recall or Precision, giving a balanced measure.

$$F1 = \frac{2}{\frac{1}{Pr} + \frac{1}{Rc}} \tag{12}$$

AUC-ROC reflects the model's ability to discern between the classes. A perfect model scores 1, while a random model gets 0.5. For AUC-ROC, plotting true positive rate against the false positive rate across thresholds yields the ROC curve. The AUC is then the area under this curve.

V. SIMULATION RESULTS AND EVALUATION

We evaluate the efficacy of integrating stream-based active learning of intrusion detection framework compared to using classical machine learning models alone. We carried out extensive simulations, and table 1 juxtaposes the performance metrics of both approaches.

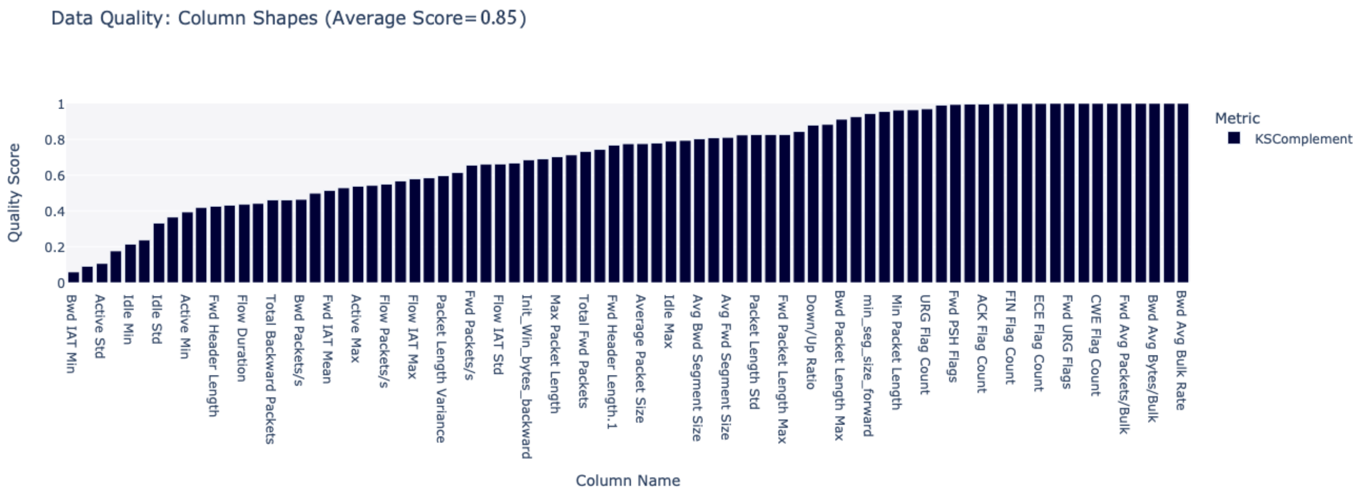

Figure 4. Comparison of the distribution shapes in the synthetic data to the real data

Table I
COMPARISON OF MACHINE LEARNING METHODS AND ACTIVE LEARNING

Methods	**Baseline**					**Active learning**				
	Acc	**P**	**Rc**	**F1**	**ROC AUC**	**Acc**	**P**	**Rc**	**F1**	**ROC AUC**
RF	85.54%	88.21%	86.37%	86.64%	0.8648	96.76%	96.47%	96.95%	96.07%	0.9679
SVM	85.38%	85.07%	85.45%	86.13%	0.8630	95.88%	95.55%	95.74%	95.95%	0.9584
KNN	83.47%	83.02%	83.30%	83.15%	0.8312	93.92%	94.52%	94.30%	93.78%	0.9415
Deep Learning	90.12%	89.37%	90.58%	90.53%	0.025	98.55%	98.73%	97.96%	98.48%	0.9856
XGBoost	92.74%	91.89%	92.68%	92.80%	0.9259	99.79%	99.87%	99.69%	99.74%	0.9980

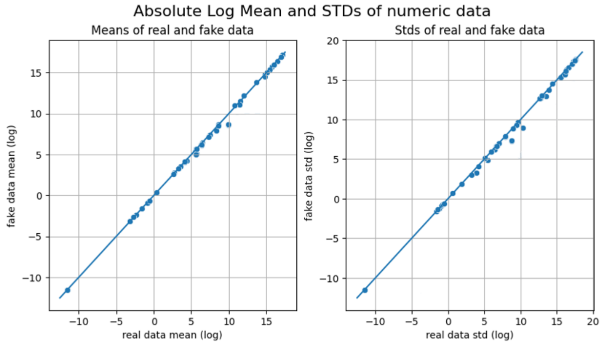

Figure 5. Comparison of Mean and Standard Deviation for Log-Transformed Data: Real vs. Synthetic Datasets

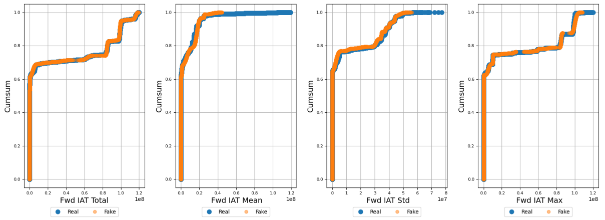

Figure 6. Cumulative distribution of values in both synthetic and real data

Table I clearly illustrate the consistent improvements in performance metrics, including accuracy, precision, recall, and F1-score, when human-in-the-loop is employed. The active learning framework consistently outperformed or, at the very least, matched the performance of the classical models. This is indicative of the benefits that arise from selectively querying the human expert for labels, focusing on instances where the model is most uncertain. The results emphasize the potential advantages of integrating active learning into intrusion detection systems, especially in scenarios where labeling data is costly, time-consuming, or requires domain expertise. By focusing on uncertain data points and reducing the labeling burden, active learning not only improves accuracy but also ensures efficient use of resources.

VI. CONCLUSION

Throughout our research, we have delved deeply into enhancing intrusion detection mechanisms by integrating the merits of stream-based active learning with classical machine learning algorithms. Our primary objective was to discern whether such an amalgamation could produce a system that surpasses traditional models in terms of accuracy while optimizing the label querying process.

The simulation results brought forth compelling evidence of the advantages of our proposed approach. Across multiple algorithms, including RF, SVM, KNN, Deep Learning, and XGBoost, we consistently observed that active learning variants either matched or outperformed their non-active counterparts. This underscores the capability of active learning in efficiently harnessing human expertise, especially when navigating through vast streams of uncertain data points.

In the vast expanse of cybersecurity, where threats evolve rapidly, and systems need to adapt swiftly, the importance of such a flexible and efficient systems cannot be understated. By seamlessly combining human intuition with algorithmic precision, stream-based active learning presents a formidable tool in the arsenal against cyber threats.

REFERENCES

[1] Q. Zeng and Z. Chen, "Scalable and probabilistic point-cloud generation for uas-based structural assessment," in *Experimental Vibration Analysis for Civil Engineering Structures: Select Proceedings of the EVACES 2021*. Springer, 2022, pp. 595–604.

[2] L. Negash, H.-Y. Kim, and H.-L. Choi, "Emerging uav applications in agriculture," in *2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA)*. IEEE, 2019, pp. 254–257.

[3] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," *IEEE communications surveys & tutorials*, vol. 18, no. 2, pp. 1123–1152, 2015.

[4] N. M. Rodday, R. d. O. Schmidt, and A. Pras, "Exploring security vulnerabilities of unmanned aerial vehicles," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016, pp. 993–994.

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

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[7] B. Poole, A. A. Alemi, J. Sohl-Dickstein, and A. Angelova, "Improved generator objectives for gans," *arXiv preprint arXiv:1612.02780*, 2016.

[8] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[9] M. Prince, "Does active learning work? a review of the research," *Journal of engineering education*, vol. 93, no. 3, pp. 223–231, 2004.

[10] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič, "Stream-based active learning for sentiment analysis in the financial domain," *Information sciences*, vol. 285, pp. 181–203, 2014.

[11] B. Settles, "Active learning literature survey," 2009.

[12] Z. CHEN and Q. ZENG, "Human-in-the-loop robotic inspection-framework and point cloud assessment," *AI*, vol. 3, no. 2, pp. 35–46, 2022.

[13] A. Holzinger, "Interactive machine learning for health informatics: when do we need the human-in-the-loop?" *Brain Informatics*, vol. 3, no. 2, pp. 119–131, 2016.

[14] K. Yang, J. Ren, Y. Zhu, and W. Zhang, "Active learning for wireless iot intrusion detection," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 19–25, 2018.

[15] Y. Zhi, Z. Fu, X. Sun, and J. Yu, "Security and privacy issues of uav: a survey," *Mobile Networks and Applications*, vol. 25, pp. 95–101, 2020.

[16] 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 of Things Journal*, 2023.

[17] J. Tao, T. Han, and R. Li, "Deep-reinforcement-learning-based intrusion detection in aerial computing networks," *IEEE Network*, vol. 35, no. 4, pp. 66–72, 2021.

[18] M. Erdelj, M. Król, and E. Natalizio, "Wireless sensor networks and multi-uav systems for natural disaster management," *Computer Networks*, vol. 124, pp. 72–86, 2017.

[19] M. Erdelj and E. Natalizio, "Uav-assisted disaster management: Applications and open issues," in *2016 international conference on computing, networking and communications (ICNC)*. IEEE, 2016, pp. 1–5.

[20] D. Gallacher, "Drone applications for environmental management in urban spaces: A review," *International Journal of Sustainable Land Use and Urban Planning*, vol. 3, no. 4, 2016.

[21] A. Kumari, R. Gupta, S. Tanwar, and N. Kumar, "A taxonomy of blockchain-enabled softwarization for secure uav network," *Computer Communications*, vol. 161, pp. 304–323, 2020.

[22] G. Choudhary, V. Sharma, I. You, K. Yim, R. Chen, and J.-H. Cho, "Intrusion detection systems for networked unmanned aerial vehicles: a survey," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 560–565.

[23] Y. Sun, S. Abeywickrama, L. Jayasinghe, C. Yuen, J. Chen, and M. Zhang, "Micro-doppler signature-based detection, classification, and localization of small uav with long short-term memory neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 6285–6300, 2020.

[24] L. M. Da Silva, I. G. Ferrão, C. Dezan, D. Espes, and K. R. Branco, "Anomaly-based intrusion detection system for in-flight and network security in uav swarm," in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2023, pp. 812–819.

[25] J.-P. Condomines, R. Zhang, and N. Larrieu, "Network intrusion detection system for uav ad-hoc communication: From methodology design to real test validation," *Ad Hoc Networks*, vol. 90, p. 101759, 2019.

[26] X. Tan, S. Su, Z. Zuo, X. Guo, and X. Sun, "Intrusion detection of uavs based on the deep belief network optimized by pso," *Sensors*, vol. 19, no. 24, p. 5529, 2019.

[27] C. Liu, Z. Gu, and J. Wang, "A hybrid intrusion detection system based on scalable k-means+ random forest and deep learning," *Ieee Access*, vol. 9, pp. 75 729–75 740, 2021.

[28] Y. Alharbi, A. Alferaidi, K. Yadav, G. Dhiman, and S. Kautish, "Denial-of-service attack detection over ipv6 network based on knn algorithm," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–6, 2021.

[29] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, "A survey of cnn-based network intrusion detection," *Applied Sciences*, vol. 12, no. 16, p. 8162, 2022.

[30] M. Sheikhan, Z. Jadidi, and A. Farrokhi, "Intrusion detection using reduced-size rnn based on feature grouping," *Neural Computing and Applications*, vol. 21, pp. 1185–1190, 2012.

[31] A. Chawla, B. Lee, S. Fallon, and P. Jacob, "Host based intrusion detection system with combined cnn/rnn model," in *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18*. Springer, 2019, pp. 149–158.

[32] R. Vinayakumar, K. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2017, pp. 1222–1228.

[33] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature machine intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[34] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv preprint arXiv:1712.04248*, 2017.

[35] 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.

[36] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," *Advances in neural information processing systems*, vol. 32, 2019.

[37] Z. Lin, A. Khetan, G. Fanti, and S. Oh, "Pacgan: The power of two samples in generative adversarial networks," *Advances in neural information processing systems*, vol. 31, 2018.

[38] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, pp. 108–116, 2018.