Regular Manuscript

GenAI-Based Jamming and Spoofing Attacks on UAVs

Submission ID c8ebf4ca-4c33-4a27-9969-bc4828be2992

Submission Version Initial Submission

PDF Generation 01 Feb 2025 11:29:58 EST by Atypon ReX

Authors

Mrs. BURCU SÖNMEZ SARIKAYA
Corresponding Author
Submitting Author

ORCiD
https://orcid.org/0000-0002-5385-9949

Affiliations

- Cyber Security and Privacy Research Lab, Department of Computer Engineering, Istanbul Technical University, Istanbul, Maslak, 34469, Türkiye

Prof. ŞERİF BAHTİYAR

Affiliations

- Cyber Security and Privacy Research Lab, Department of Computer Engineering, Istanbul Technical University, Istanbul, Maslak, 34469, Türkiye

Additional Information

Keywords

Communication system security

Computer security

Network security

Subject Category

Communications technology

Vehicular and wireless technologies

Files for peer review

All files submitted by the author for peer review are listed below. Files that could not be converted to PDF are indicated; reviewers are able to access them online.

Name	Type of File	Size	Page
IEEE_Access_GAN_IDS_last.pdf	Main Document - PDF	4.5 MB	Page 3
framework-gan-ids.pdf	Graphical Abstract Image	602.4 KB	Page 24
ga_text.txt	Graphical Abstract Caption	382 B	Page 25

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version XXX.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

GenAI-Based Jamming and Spoofing Attacks on UAVs

BURCU SÖNMEZ SARIKAYA[1], ŞERİF BAHTİYAR[1]
[1]Cyber Security and Privacy Research Lab, Department of Computer Engineering, Istanbul Technical University, Istanbul, Maslak, 34469, Türkiye
Corresponding author: Burcu Sönmez Sarıkaya (e-mail: sonmezb18@itu.edu.tr).

This work was supported by Turkcell Iletişim Hizmetleri A.Ş. and Research Fund of Istanbul Technical University, Project Number: 45654

ABSTRACT Recently, areal vehicles have been more connected than ever, where there are many types of the vehicles. Unmanned areal vehicles (UAVs) operate on various environments with different technologies that are subject of many attacks. Creating effective intrusion detection systems against such attacks have been a significant challenge since there is a lack of sufficient attack data that can be used to design an intrusion detection system with advanced computing algorithms. In this research, we propose a novel framework to create attacks data for UAVs by using generative artificial intelligence algorithms. We use Variational Autoencoder, Gaussian Copula, Denoising Diffusion Probabilistic Model (DDPM), and Conditional Tabular Generative Adversarial Network to create synthetic attack data. Specifically, jamming and spoofing attacks on UAVs are generated to fool intrusion detection systems that may be implemented on UAVs. Experimental evaluations show that synthetically generated attack data reduces the accuracy of intrusion detections if the system was trained with inadequate attack data. Additionally, analyses results show that DDPM emerged as the most effective model to create more effective attack data that lead to lowest intrusion detection accuracies with reductions of 49% for jamming and 44% for spoofing attacks. This research highlights the need for more robust and adaptive intrusion detection systems that can be created with synthetic data. Thus, sustainable computing systems on UAVs will be achieved.

INDEX TERMS Cyber security, generative artificial intelligence, intrusion detection system, synthetic data, unmanned aerial vehicles

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are aerial vehicles that operate autonomously under remote human control or via onboard computing systems. The rapid advancement and the commercialization of communication networks and artificial intelligence have made UAVs increasingly versatile with their ability to operate in difficult and inaccessible environments. UAVs have become an integral part of a wide range of civil and military applications since they provide high adaptability and cost-effectiveness. The vehicles use many technologies that require interconnected networks of sensors, complex software, data collection, flight control, mission execution, and etc. Moreover, the operation environment is evolving dynamically. These conditions introduce different vulnerabilities on UAVs, which expands the threat surface considerably. Specifically, UAVs communication systems are the most vulnerable part of UAVs against cyber attacks. Adversaries often attempt to disrupt UAV operations by intercepting their communications, compromising the confidentiality, integrity, and availability of data. For instance, jamming attacks like denial of service(DoS) and Global Positioning System (GPS) spoofing, replay, routing, eavesdropping, and false data injection are such examples, which disrupt UAV availability by affecting communications [1]–[3].

Jamming and GPS spoofing attacks have been used to disrupt wireless sensor networks, such as MavLink protocol and severely affect UAV communications. Jamming attacks are intentionally used to disrupt or block communication signals. Such attacks directly compromise system availability and pose serious challenges to UAV security [4]. GPS spoofing attacks, which alter the trajectory of autonomous UAVs, represent another critical threat to UAV systems [5], [6]. The most critical vulnerabilities include GPS spoofing and GPS jamming attacks, which disrupt communications and undermine the availability of UAVs. In this research, we focus on GPS jamming and GPS spoofing attacks on UAVs.

Security measures have been proposed to mitigate jamming and GPS spoofing threats, such as encryption, authentication protocols, and intrusion detection systems (IDS). Specifically, IDS play an important role to protect UAV systems by

monitoring data traffic and identifying unauthorized or malicious activities. Machine learning-based IDSs have shown promising results in addressing signal spoofing and jamming attacks by analyzing data from UAV flight and sensor readings [7].

Existing IDS solutions for UAVs have several limitations, which prevent to have secure UAV systems. For example, many IDS systems produce excessive false alarms, leading to unnecessary operational disruptions. IDS systems may struggle to provide a real-time threat detection. On the other hand, UAV threats evolve rapidly and existing IDS systems often fail to adapt to new and unknown attack vectors. UAVs typically operate with a limited power supply and a communication bandwidth, which may restrict the deployment of advanced IDS algorithms. Moreover, UAVs have constrained computational and storage capabilities that limit IDSs to implement complex algorithms onboard.

Training IDS systems, particularly those based on machine learning, requires extensive data that include diverse attack scenarios. Such data are often unavailable or incomplete for UAV environments. Training on imbalanced datasets with significantly fewer attack samples than normal operations leads to a poor detection accuracy for rare threats. These highlight the need for more robust, adaptable, and resource-efficient systems that may address the diverse and evolving threat landscape. The challenge in this research is the lack of attack data about UAVs for IDSs. We propose a novel framework to generate synthetic attack data about UAVs by using real data using generative AI. Specifically, we introduce a Generative Artificial Intelligence (GenAI) framework to generate attack data that are used to enhance an IDS performance to protect UAV security. Unlike traditional approaches, the proposed framework may learn different data distributions within UAV systems and produce both realistic attacks and more malicious data. The main contributions of this research are as follows.

- Used many models to generate synthetic data, such as Variational Autoencoder (VAE), Gaussian Copula, Conditional Tabular GAN (CTGAN) and Denoising Diffusion Probabilistic Model (DDPM) models.
- Designed and evaluated an IDS using synthetic datasets generated by GenAI models, such as VAE, Gaussian Copula, DDPM, and CTGAN.
- Highlighted DDPM and Gaussian Copula as the highly effective algorithms for generating realistic synthetic attack patterns about jamming and spoofing attacks.
- Showed that generative models may significantly degrade IDS performance, exposing vulnerabilities in current detection frameworks and emphasizing the need for adaptive solutions.

The rest of the paper is organized as follows. Section II contains literature review about attacks on UAVs, existing security mechanisms. Section III introduces the proposed framework. Section IV describes attack scenarios. Section V is about experimental setup. Section VI contains experimental results. The last section is devoted to conclusion.

II. UAV AND SECURITY

UAV systems contains significant security threats primarily due to their dependence on wireless communication channels, uncontrolled environments, limited resources, dynamic topologies, and their collaboration with other UAVs [8]. This section overviews the state of the art about UAV security.

A. UAV THREATS

UAV networks are vulnerable to a range of attacks that target their communication and routing systems. Attackers may target network traffic, disrupt routing operations, or introduce malicious nodes into the network. The literature extensively documents various threats to UAV networks, emphasizing their impact on the privacy, integrity, and availability of these systems. For example, [3] provides an overview of cyber-attacks on UAVs, while [9] highlights the critical importance of securing communication links between ground control stations and UAVs.

Numerous researches focus on identifying security threats in UAV communications and creating countermeasures. In [10], the research provides a foundation for understanding key challenges in unmanned systems, including privacy and security issues. Common examples of attacks include the interception of critical communications by adversaries and GPS spoofing, which manipulates the flight paths of autonomous UAVs [6], [11], [12]. Attackers can exploit vulnerabilities in communication protocols through various strategies [13].

Jamming attacks pose a critical threat to UAV operations by disrupting the communication signals between an UAV and its Ground Control Station (GCS). These attacks typically involve the use of high-power Radio Frequency (RF) signals to overpower legitimate communication channels that causes significant disruptions to UAV operations. Such attacks may result in the loss of control, forcing the UAV to either land prematurely or crash. Additionally, jamming affects navigation systems by reducing the Signal-to-Noise Ratio (SNR), increasing packet loss, and impairing performance [14], [15].

Researchers highlight various countermeasures to mitigate jamming attacks, such as Frequency Hopping Spread Spectrum (FHSS), adaptive filtering, and machine learning models for reliable jamming identification [14], [15]. The integration of Deep Neural Networks (DNNs), eXtreme Gradient Boosting (XGB), and distributed detection techniques has demonstrated effectiveness to improve jamming detections and responses [14]–[16].

Spoofing attacks are another significant threat where adversaries manipulate GPS signals to mislead UAVs about their locations and trajectories. This attack may cause UAVs to deviate from their intended flight paths, potentially leading to catastrophic consequences, such as collisions or diversion to hostile territories.

Recent researches use machine learning and ensemble methods, such as AdaBoost-Convolutional Neural Network (CNN) and stacked ensemble models, to enhance spoofing

detection accuracy even with limited samples [16], [17]. The integration of artificial intelligence and machine learning for real-time threat detections and responses remain underexplored [14], [15]. Furthermore, the scalability and the cost-effectiveness of existing countermeasures require validation in real-world scenarios. The complexity of GPS spoofing and jamming techniques highlights the need for adaptive and robust counter-strategies [15]–[17]. In the literature, there is a lack of research that addresses UAV threats and their countermeasures with machine learning to ensure their safe and reliable operation.

The use of machine learning-based IDS has recently become a popular approach for securing UAV networks. For example, a machine learning-based IDS was proposed in [18] to address GPS spoofing attacks. This approach utilizes one-class support vector machines and ML autoencoder algorithms to identify anomalous behavior. Similarly, a detection method for eavesdropping attacks on UAV communication is presented in [19]. This approach employs a combination of K-means clustering and support vector machines for the analysis of data and the prediction of future threats. The method operates in two phases. Initially, both communicating parties send signals to the UAV. Subsequently, the UAV relays these signals to a third party, which identifies any deviations. Subsequently, the dataset is classified using machine learning algorithms with the objective of detecting potential attacks.

Machine learning-based IDSs are used to mitigate signal spoofing and jamming attacks. A self-learning approach with multi-class support vector machines ensures a high true positive rate in IDS as in [20]. The system incorporates a deep-Q network, a deep reinforcement learning algorithm, for dynamic route learning as a self-healing mechanism during the IDS recovery phase. The solution collects data from various sources within the UAV, including flight logs and sensor readings. A CNN approach for detecting jamming signals is proposed in [21]. This method considers weights and values of GCS and it selects a relay power element based on the bit error rate. While this algorithm effectively protects communications against jamming attacks, the random selection of relay power may increase the error rate that may result in high computational costs. In addition to these methods, a variety of machine learning security frameworks have been developed to address a range of security challenges, including the detection and prevention of malicious drone activity and DoS attacks [22]. Recent developments indicate that federated learning methodologies may prove more effective than traditional machine learning algorithms. For example, radio frequency-based UAV authentication models utilizing IoT networks have been constructed using federated learning techniques [23].

UAVs rely on wireless communication, making them vulnerable to attacks, such as GPS spoofing, data injection, and DoS attacks. Traditional detection methods rely heavily on pre-defined parameters and they often struggle with emerging threats [24], [25]. For instance, MUVIDS system emphasizes network-level IDS but it faces scalability challenges due to its reliance on MAVLink protocol analysis [25].

Each type of attack has specific targets on UAVs. An attack compromises distinct security requirement. A comprehensive security strategy requires the integration of multiple defense techniques. Additionally, a power consumption and a battery usage must be carefully assessed to identify suitable countermeasures against such attacks. Furthermore, defense mechanisms should be continuously updated and tested to ensure their effectiveness against emerging security threats. The big challenge of having effective ML based IDSs is the lack of attack data.

B. GENAI IN UAV SECURITY

UAVs are increasingly targeted by adversarial attacks, including spoofing, jamming, and data injection. Various methods have been developed to detect and mitigate these attacks that are based on GANs to enhance IDS. GUIDE [26] and G-IDS [27] systems utilize GANs to generate synthetic data, addressing the challenges of imbalanced datasets and improving IDS robustness. GAN-based models like SeqGAN and LeakGAN have demonstrated significant improvements in attack detection, which increase accuracy up to 37% for known attacks and 30% for unknown attacks.

While GANs have been used to enhance IDS, they are also leveraged to generate adversarial attacks. IDSGAN [28] is a framework designed to create malicious traffic that deceives IDS systems. By dynamically adapting to IDS models, IDSGAN may effectively bypass detection and challenge system robustness. Similarly, GIDS employs GANs to create synthetic data for in-vehicle network IDS, achieving high detection accuracy for known and unknown attacks [29]. Generative models are employed to address the scarcity of security-related data for UAVs, such as the use of Conditional Tabular GANs (CTGAN), Variational Autoencoders (VAE), and Gaussian Copulas to generate realistic jamming attack scenarios. These models enable the training of IDS on diverse and enriched datasets by improving detection capabilities [30]. Moreover, GANs are combined with active learning to dynamically adapt IDS models, improving their effectiveness against evolving threats [29], [30].

Despite these advancements, the grand challenge, the lack of enough attack data, remains. Generative models often struggle with time-series data augmentation, as highlighted in MUVIDS and GUIDE implementations [25], [26]. While GAN-based IDS may detect novel attacks, they require extensive training data and computational resources, which limit their scalability for real-time applications [28], [31].

III. GENAI-BASED JAMMING AND SPOOFING ATTACKS

Generative AI has emerged as a powerful tool for generating synthetic data. In this research, we propose a framework to generate synthetic attack data for UAVs. The framework consists of two parts. Synthetic attack data is generated with generative models in the first part with models that are VAE, Gaussian Copula, CTGAN and DDPM. In the second part, an intrusion detection algorithm is applied to datasets generated

in the first part. XGBoost classification is used to distinguish attack data and benign data. The proposed framework is shown in Figure 1.

A. GENAI MODELS

1) Variational Autoencoders

An autoencoder (VAE) consists of two principal components: the encoder and the decoder. The encoder compresses input data x into a latent representation z. The output of the encoder is a pair of parameters, namely the mean (μ) and the standard deviation (σ), which together define a Gaussian distribution. The latent vector z is sampled from this distribution, ensuring that the latent variables follow a continuous, typically normal, distribution. This case allows the model to capture essential information in a lower-dimensional space.

The latent vector z represents the compressed knowledge of the input, holding its critical features in a reduced form. It acts as the input for the decoder, enabling data reconstruction. The decoder reconstructs the input data from the latent vector z. Its objective is to minimize the reconstruction error, ensuring that the reconstructed data is as close as possible to the original input. This demonstrates the effectiveness of compressed representation in terms of accuracy.

2) Gaussian Copula

We use Gaussian copulas to generate synthetic attacks by identifying the dependency between variables in the dataset. A copula can be defined as a multivariate distribution, and the use of a Gaussian copula enables the accurate modeling of the correlations between different variables. In the initial phase, the conditional probability approach is employed to integrate metadata with the features in question. This phase employs contextual insights derived from the metadata to modify or enhance the data potentially. The result of this process is an "extended table," which is subsequently transformed using a Gaussian copula, whish is a statistical method designed to model dependencies among multiple variables, serves to prepare the data for analysis in a more sophisticated model. This advanced model is designed to examine the data distribution and covariance, thereby facilitating a more profound understanding of the subject matter [32].

3) Conditional Tabular GAN (CTGAN)

We investigated the potential of CTGAN as a way to generate synthetic attack data for training models. The conditional generator is an effective tool for the generation of synthetic rows, whereby conditioning is applied to a specific discrete column. By employing training-by-sampling, both the conditioning values and the training data are sampled based on the logarithmic frequency of each category [33]. This ensures that CTGAN effectively explores all potential discrete values, allowing the modelling of complex relationships in the data that are necessary to simulate realistic attack scenarios. CTGANs are particularly effective in handling a variety of conditions found in real-world data [34].

The process begins by sampling a condition, which is then passed to the conditional generator G along with a noise vector z drawn from a standard normal distribution $N(0, 1)$ The generator creates a synthetic sample, which is compared against a randomly selected dataset sample that satisfies the same condition. This comparison is evaluated by the conditional discriminator D. To capture all potential correlations between the columns, both the generator and discriminator use fully connected neural networks with two hidden layers each. The generator employs batch normalization and the ReLU activation function, generating synthetic row representations through a mixture activation function after the hidden layers. The discriminator uses the leaky ReLU activation function and applies dropout in each hidden layer to enhance robustness. CTGAN enhances TableGAN (TGAN) by modifying the loss function and normalization techniques to handle imbalanced data, particularly for continuous data with non-Gaussian or multimodal distributions.

4) Denoising Diffusion Probabilistic Model (DDPM)

Finally, we use a DDPM to generate synthetic attack data. In generally diffusion models are a type of latent variable model in machine learning that leverage Markov chains and variational inference to uncover the hidden structure of a dataset. These models learn the data distribution by progressively adding noise to the data and subsequently denoising it, allowing them to generate high-quality and diverse samples.

We used a diffusion model that operates in two main stages, namely forward process (diffusion) and reverse process (denoising). For the DDPM [35], [36] in the forward process, the original data $X_0 \in \mathbb{R}^d$ is progressively corrupted by adding Gaussian noise ϵ step by step, resulting in a sequence of latent variables $X_1, \ldots, X_K$. This process transforms the data into pure Gaussian noise, where the final state $X_K \sim \mathcal{N}(0, I)$. Each transition in this sequence follows a Markov Chain. The reverse process then reconstructs the original data by iterative denoising the Gaussian noise.

5) Algorithms of GenAI Models

Algorithm 1 outlines the proposed GenAI model for generating synthetic jamming and spoofing attacks in UAVs. The algorithm consists of three main steps. In the first step, training a VAE model to generate synthetic attacks is accomplished. Then, we train a Gaussian Copula model to generate synthetic attacks. Finally, we train a CTGAN model to generate synthetic attack, where x_i represents the i-th feature and N is the total number of features and outputs the final generated synthetic attack data for each model.

In step 1, VAE uses the Kullback–Leibler (KL) divergence metric [37]. This metric measures the difference between the learned latent distribution and a target Gaussian distribution, enforcing the latent space to align with the Gaussian assumption. Unlike standard autoencoders that map the input x to a single deterministic latent vector, VAEs map x to two separate vectors representing the mean (μ) and standard deviation (σ) of a Gaussian distribution. This approach ensures a smooth

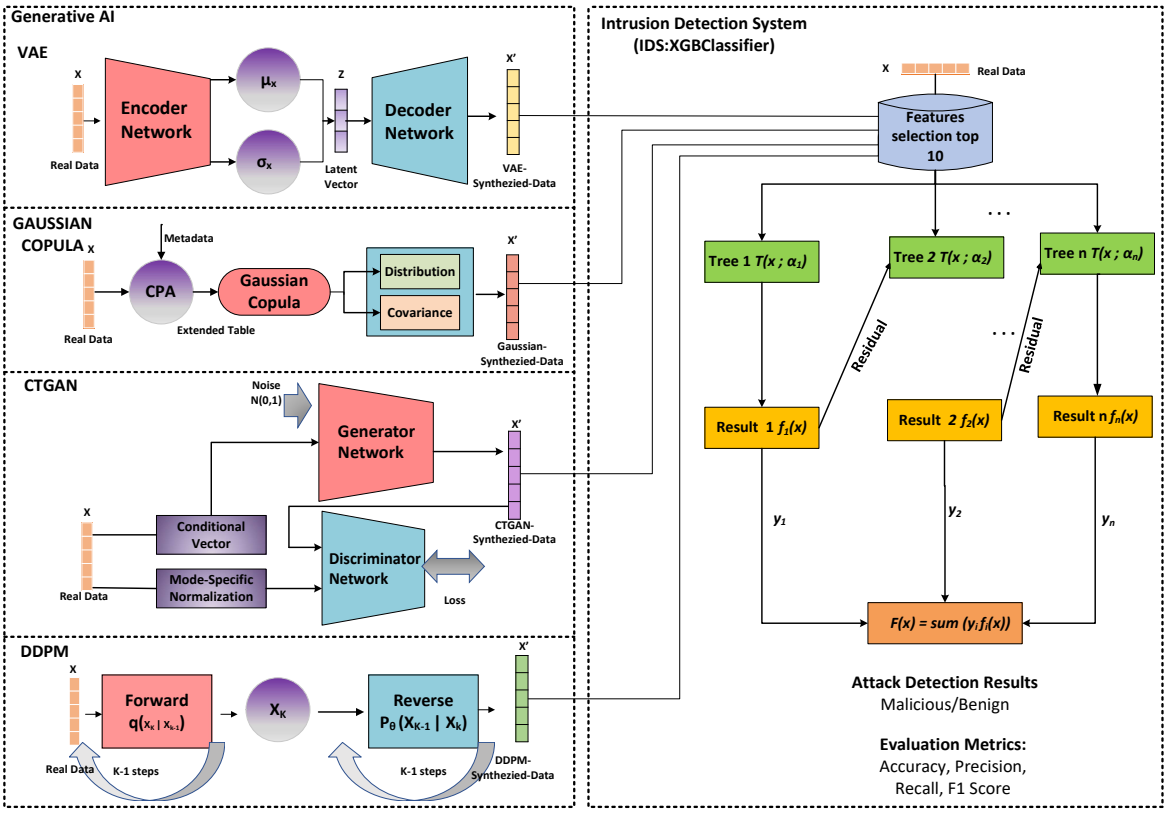


FIGURE 1. The architecture of GenAI-Based attacks against IDS.

and structured latent space, enabling better generative capabilities [38].

$$L_{\mathrm{KL}} = D_{\mathrm{KL}}(q(z|x) \,\|\, \mathcal{N}(0,1)) \tag{1}$$

$$L_{\text{total}} = L_{\mathrm{KL}} + \|x - \tilde{x}\|^2 \tag{2}$$

Variational Autoencoder (VAE) loss function ensures the model to learn meaningful latent representations while accurately reconstructing input data. The 1 equation, calculates the Kullback-Leibler (KL) divergence, which measures how much the learned latent distribution deviates from the desired prior distribution, typically a standard normal distribution $\mathcal{N}(0,1)$. This regularization term enforces structure and continuity in the latent space, enabling smooth interpolation and meaningful sampling. Equation 2 combines the regularization term with the reconstruction loss that are represented as the squared difference between the input x and its reconstruction $\tilde{x}$. This ensures the decoder effectively to reconstruct data while maintaining a well-structured latent space. Together, these losses guide VAE during training, balancing accurate data reconstruction and a robust latent representation [39].

The training of a Gaussian Copula model to generate synthetic attacks is described in step 2. Each feature is transformed into a uniform distribution through the application of cumulative distribution functions (CDFs), and is then further standardized to a normal distribution through the use of the inverse of the standard normal CDF. A correlation matrix is then derived from the transformed variables in order to capture the relationships between them. The matrix is then employed to generate new samples that are drawn from a multivariate normal distribution. Subsequently, the samples are mapped back to the original variable scales through inverse transformations. Initially, they are smoothed with the normal CDF, and then reverted to the original distributions using the inverse CDFs of the fitted models [40], [41].

The training of a Gaussian Copula model to generate synthetic attacks is described in step 3. The CTGAN loss function was optimized using the Adam algorithm. The Wasserstein loss function was employed in the equation 3. In this equation, D represents the discriminator, G denotes the generator, z is a sample from the generator's input noise distribution $p_g 4$, and x is a sample from the real data distribution p_r.

$$\mathcal{L}(D,G) = \mathbb{E}_{x\sim p_r}\left[D(x)\right] - \mathbb{E}_{z\sim p_g}\left[D(G(z))\right] \tag{3}$$

Algorithm 2 outlines the proposed DDPM Model for generating synthetic attacks in UAVs. In our proposed solution algorithm 2 runs after algorithm 1 therefore we continue with step 4 in the second algorithm. The training of a DDPM model to generate synthetic attacks is explained in step 4.

Algorithm 1 Generative Artificial Intelligence for Generated Synthetic Attacks in UAVs

1. **Input:** Training data $x_1, x_2, ..., x_n \in X$
2. **Output:** VAE-synthezied-data, GaussianCopula-synthezied-data, CTGAN-synthezied-data, DDPM-synthezied-data
3. **Step 1:Training a VAE model to generating synthetic attacks**
4. Initialize decoder parameters ϕ, encoder parameters θ

repeat

for $k = 1$ to N **do**

5. Draw samples S from $\epsilon \sim \mathcal{N}(0,1)$
6. Compute latent variable: $z_{(k,s)} = h_\phi(\epsilon^{(k)}, x^{(k)})$

end for

7. Compute reconstruction loss: $E = \|x - \tilde{x}\|^2$
8. Compute KL divergence:

$L_{\text{KL}} = D_{\text{KL}}(q(z|x) \,\|\, \mathcal{N}(0,1))$

9. Combine losses: $L_{\text{total}} = L_{\text{KL}} + E$
10. Update parameters ϕ, θ using Stochastic Gradient Descent

until Parameters ϕ, θ converge

11. **Step 2: Training a Gaussian Copula model to generate synthetic attacks**

for each feature $x_i \in X$ **do**

12. Estimate the marginal distribution of each feature.
13. Transform each feature into a standard normal distribution using CDF.
14. Compute the covariance matrix of the transformed features.
15. Generate synthetic samples from a multivariate normal distribution using the learned covariance matrix.
16. Apply the inverse CDF transformation of each feature to map the synthetic data back to the original space.

end for

17. Combine synthetic features.
18. **Step 3: Training a CTGAN model to generate synthetic attacks**

for each feature $x_i \in X$ **do**

19. Sample noise vector from a standard normal distribution.
20. Sample conditional vector representing feature distributions.
21. Generate synthetic samples using the generator.
22. Discriminate real and synthetic samples using the discriminator.
23. Compute generator loss :

$\mathcal{L}(D,G) = \mathbb{E}_{x\sim p_r}\left[D(x)\right] - \mathbb{E}_{z\sim p_g}\left[D(G(z))\right]$

end for

Algorithm 2 DDPM for Generated Synthetic Attacks

1. **Step 4: Training a DDPM model to generated synthetic attacks:**

for each feature $x_i \in X$ **do**

2. Add Gaussian noise at each step.
3. Train a neural network to predict the noise.
4. Minimize the loss function.

$L_{\text{simple}} = \mathbb{E}_{k,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_k, k)\|^2\right]$

5. Sample
6. Iteratively denoise using the reverse process
7. Denormalize the generated data to match the original data distribution

end for

8. **Step 5: Output:**
9. **Return** VAE-synthezied-data, GaussianCopula-synthezied-data, CTGAN-synthezied-data, DDPM-synthezied-data

Finally, outputs for each model are recorded in last generated synthetic attack data.

The loss function (in equation 4) L_{simple} is a fundamental component of DDPM models, designed to optimize the model's ability to reconstruct data by accurately predicting the noise introduced during the forward diffusion process. It is defined as the expected mean-squared error (MSE) between the true noise ϵ and the predicted noise $\epsilon_\theta(x_k, k)$, where the prediction is performed by a neural network parameterized by θ [35], [42].

$$L_{\text{simple}} = \mathbb{E}_{k,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_k, k)\|^2\right] \tag{4}$$

The expectation $\mathbb{E}_{k,x_0,\epsilon}$ is computed over the timestamps k, real data samples x_0, and Gaussian noise ϵ. By minimizing this loss, the model learns to predict the noise for any noisy input x_k at a given timestep k. This is crucial for the reverse diffusion process, where the model progressively removes noise to reconstruct the original data.

B. INTRUSION DETECTION SYSTEMS

We tested generated synthetic attack data by using intrusion detection systems for UAVs. We use XGBoost algorithm for the intrusion detection system. XGBoost, a highly scalable tree boosting system, was introduced by Chen et. al. [43] as an extended version of the Gradient Boosting Decision Tree (GBDT) algorithm [44].

The process of gradient boosting consists of three principal stages. The initial step is to identify a differentiable loss function that is appropriate for the problem at this stage of the process. One advantage of gradient boosting is its flexibility, in that new algorithms do not need to be derived for different loss functions. Instead, an appropriate loss function can be selected and integrated into the framework. Secondly, a weak learner is utilized to make predictions, with decision trees being the optimal choice. In particular, regression trees are utilized as weak learners, as they generate real-valued outputs

for splits, thereby allowing their outputs to be aggregated. This additive property enables the model to iteratively enhance the accuracy of residuals. The trees are constructed in a greedy manner, frequently with constraints that guarantee they remain weak learners while maintaining computational efficiency. Thirdly, an additive model is constructed by sequentially combining the outputs of the weak learners in order to minimize the loss function. New trees are added to the sequence one at a time, with each tree's output refining the predictions of the previous ones. This process continues until the loss function is sufficiently optimized. XGBoost builds upon gradient boosting but introduces a significant enhancement. In contrast to traditional gradient boosting, where weak learners are added sequentially, XGBoost employs a multi-threaded approach.

A framework for evaluating attack detection using XGBoost with both real and synthesized datasets is shown in Figure 1. From each dataset, the top 10 most important features are selected for training purposes. The XGBoost is then used to model the data. Subsequently, XGBoost, a gradient boosting algorithm, is employed for modeling the data. The algorithm employs a sequence of decision trees, wherein each tree refines the residual errors of the previous one. The trees are then combined into an additive model, whereby the final prediction is the weighted sum of the outputs of all trees. The framework is designed to predict whether an input sample is "malicious" or "benign." Its performance is evaluated using a range of metrics, including accuracy, precision, recall and F1 score. This configuration enables a comparison of the quality of synthetic data to support effective attack detection relative to the real dataset.

Algorithm 3 IDS for Synthetic and Real Attacks on UAVs

Input: Training top 10 important features $x_1, x_2, ..., x_i \in X$ extracted from synthetic and real datasets C: Number of trees, P: Set of other parameters of XGBoost classifier

Output: M: Trained XGBoost model

1. Set initial values for the parameters of the XGBoost classifier 2.

for each instance f_i in F **do**

3. $x_1 \leftarrow \text{Sort}(x_i)$

4. $p_1 \leftarrow \text{Split-best}(\text{lowest-gain}(f))$

5. Optimize objective target function to compute the tree depth and choose the best split point

6. $M \leftarrow \text{Optimizing}(\text{Choosing descriptor-point}(p_i))$ 7.

8. Tree-leaves $\leftarrow$ Prediction-score$(M, \text{Tree-leaves})$

9. Bottom-up-Prune-negative-nodes $(M, \text{Tree-leaves})$

10. Repeat

11. Repeat steps 3 and 4 until $\text{Length}(M) = \text{Max-tree-depth}$

12. Repeat steps 3 to 10 until cumulative training includes all trees in C 13.

Until 14. Return M

15. **Step 4: Evaluation:**

16. Evaluate final M model using test features.

A method for training an intrusion detection system based on the XGBoost classifier, specifically designed to address synthetic and real attacks on UAVs is described in Algorithm 3. It takes the top 10 important features as inputs that are extracted from both synthetic and real datasets, along with hyperparameters such as the number of trees and other classifier settings. Initially, the parameters of XGBoost classifier are set. For each feature in the dataset, the algorithm sorts the feature values, selects the optimal split point using a lowest-gain-first approach, and optimizes the objective target function to determine the tree depth and split point. Subsequently, the tree structure is updated by optimizing and pruning unnecessary nodes based on their contribution to the prediction task. This process iterates until the maximum tree depth is reached. The cumulative training is repeated for all trees in the dataset, refining the model. Finally, the trained model is returned and evaluated using test data, completing the IDS training process. This approach ensures an efficient and optimized classifier that effectively detects synthetic and real attacks.

IV. ATTACKS AND SCENARIOS

UAVs use a combination of communication networks, sensors, electronic receivers and transmitters, and software systems. UAVs collect and process data, share data over communication networks, control their flight, and perform specific tasks. These systems produce logs containing large amounts of sensitive information, such as location data, video footage, sensor data, and control commands. Attackers can exploit vulnerabilities in these systems to gain unauthorized access, manipulate data, disrupt communications, take control of systems, or cause physical damage.

Dataset contains the UAV's log records during the attack and the flights where no attack occurred. In the scenario implemented by Whelan et al., [45] the UAV must first perform a flight where no attack occurred. Benign flight can be performed in a controlled environment or in a non-hostile area such as a military base. After this flight is completed, it is recorded as benign flight logs. Their scenario used the Global Positioning System (GPS) for navigation of Unmanned Aerial Vehicles (UAVs) and the MAVLink protocol for communication with the Ground Control Station (GCS) [7]. These systems are vulnerable to certain threats, especially jamming and GPS spoofing. Spoofing disrupts communication or GPS signals, while GPS spoofing involves broadcasting false GPS signals to deceive the UAV's navigation system. Figure 3 illustrates a typical scenario involving a Ground Control Station (GCS) and an Unmanned Aerial Vehicle (UAV) communicating via the MAVLink protocol. The scenario involves a UAV that communicates with a GCS using the MAVLink protocol, a widely used messaging protocol for UAVs.

MAVLink is a lightweight protocol designed for communication links with limited bandwidth and systems with constrained resources, which are used to carry telemetry and command and control data for numerous small unmanned

aerial vehicles [46]. MAVLink protocol defines the mechanism on the structure of messages and how to serialize them at the application layer. These messages are then forwarded to lower layers, such as transport layer and physical layer, to be transmitted to the network. The advantage of MAVLink protocol is that it supports different types of transport layers and mediums thanks to its lightweight structure. It can be transmitted through WiFi, Ethernet, or serial telemetry low bandwidth channels.

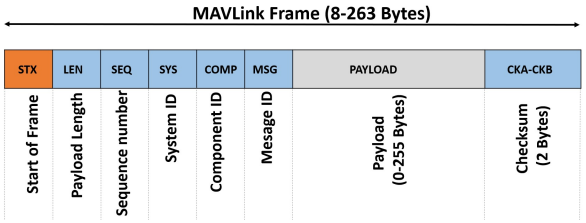


FIGURE 2. MAVLink Protocol Data Frame.

Figure 2 describes the structure and meaning of each component in a MAVLink frame, which is a communication protocol widely used in UAVs. The first byte is the "Packet Start Sign" (STX), which has a fixed value of 0xFE and signals the beginning of a new packet. The "Payload Length" (LEN) indicates the size of the following payload, ranging from 0 to 255 bytes. The "Packet Sequence" (SEQ) is used for tracking the sequence of packets to help detect any packet loss. The "System ID" (SYS) and "Component ID" (COMP) dentify the sending system and its specific component, allowing for differentiation between multiple platforms and components on the same network. The "Message ID" (MSG) denotes the type of message, which determines the payload's meaning and how to interpret it. The "Data" (Payload), ranging from 0 to 255 bytes, starts depends on the message ID. Lastly, the checksum ensures data integrity by computing a hash of the previous bytes, incorporating an extra parameter to enhance error-checking. This structured format helps MAVLink achieve reliable and organized communication in real-time applications [47].

MAVLink protocol has no privacy or authentication mechanism. Therefore, it does not provide any security and can be hacked quite easily. GCS communicates with drones over an unauthenticated and unencrypted channel. Since MAVLink message streams are sent without encryption, anyone with a suitable transmitter can communicate with the drone, intercept communication, eavesdrop, and inject commands. Thus, any attack can be launched easily. In [48] discusses the vulnerabilities of MAVLink protocol and proposes a security-integrated mechanism for MAVLink that ensures the protection of MAVLink messages exchanged between UAVs and GCSs by utilizing the use of encryption algorithms.

A Holybro S500 UAV equipped with a Pixhawk 4 flight controller was used to explore UAV intrusion detection under GPS spoofing and jamming attacks in [7]. A HackRF software-defined radio (SDR) performed the attacks, while

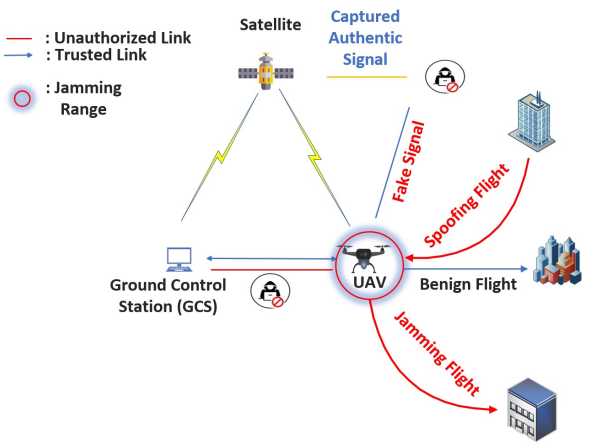


FIGURE 3. An attack scenario.

a Keysight EXG signal generator provided controlled GPS signals simulating a location. The UAV first performed a harmless flight to collect basic data. During the GPS spoofing attack, the GPS-SDR-SIM generated false GPS data streams, causing the UAV to lose its balance and crash. Similarly, in the GPS jamming attack, white Gaussian noise corrupted the UAV's GPS, again leading to a crash. Observations from these harmless and harmless flights were recorded for further analysis. The log records of all flights were shared openly, shedding light on our work. We generated synthetic attack data using generative AI by utilizing the data collected in the attack scenario described above. We compared both the statistical and intrusion detection success between our results data and the real data obtained from [7].

V. EXPERIMENTAL SETUP

In our research, PyTorch v1.13.0 framework was used for training and evaluating models.

A. DATASETS AND FEATURES

UAV attack dataset in [45] provides a comprehensive repository of real communication data between UAVs and ground control stations, providing a valuable basis for simulating various attack scenarios, including GPS spoofing and jamming attacks. Moreover, the dataset includes both simulated and real flight records, and data from benign flights and flights under attack. The dataset typically includes telemetry data, commands, and status messages exchanged between an UAV and a ground control station. Data cover both benign and malicious examples derived from real flight records, reaching a total of 6,446 rows and 84 columns for jamming attacks and 3,623 rows and 84 columns for GPS spoofing attacks. In jamming attack, 22.7 percent of the data is malicious and 77.3 percent is benign. In spoofing attack, 13.7 percent of data is malicious and 86.3 percent of data is benign. In figure 4, malicious-to-benign data ratios of all synthetic data and real data for the spoofing attack are given. Malicious-to-benign

data ratios of all synthetic data and real data for the jamming attack are given in figure 5.

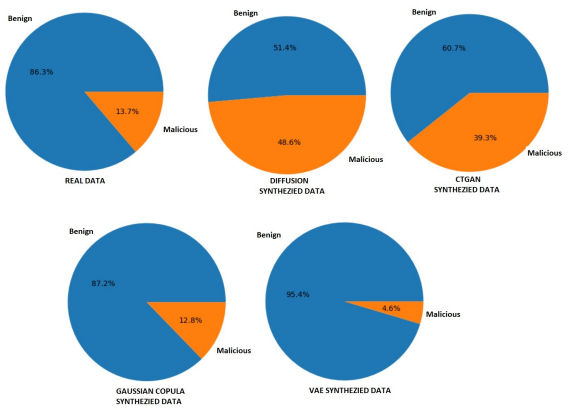


FIGURE 4. Data rates for spoofing attacks.

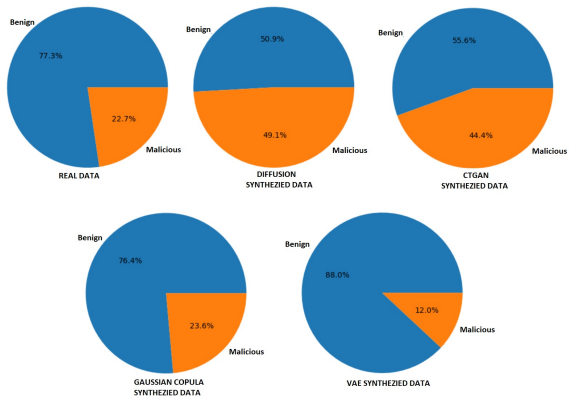


FIGURE 5. Data rates for jamming attacks.

We used XGBOOST classifiers to extract important features from the real dataset. The provided feature importance data explains how certain features play important roles in identifying jamming and spoofing attacks in GPS systems. These two types of attacks have different impacts on GPS signal quality and navigation data, and the extracted features provide insights into the detection of these attacks and the generation of attacks with similar features to generative models. For example, jamming attack focuses on degrading GPS signal quality, which leads to a noise, a reduced satellite utilization, and a poor positional accuracy. Spoofing attack actively manipulates GPS data to mislead users, which results in impacts on location, speed, and orientation.

Important features of GPS jamming attacks identified by the XGBOOST classifier are given in figure 6. In the figure, *eph_x* represents the horizontal position accuracy of the GPS signal. *epv* indicates the vertical position accuracy. These features achieve the highest feature importance scores, highlighting their strong predictive power in identifying jamming. They measure the deviations in GPS position accuracy that jamming significantly increases.

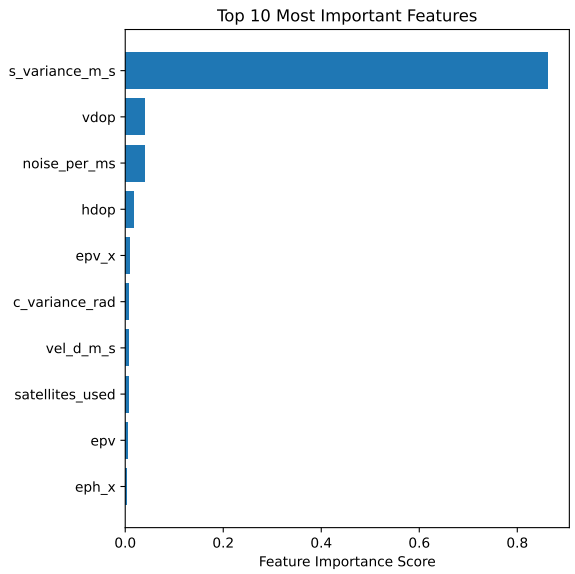


FIGURE 6. Important features for jamming attacks.

Feature *satellites_used* denotes the number of satellites utilized in the GPS positioning system. The feature importance score of this feature indicates its importance in detecting a loss of usable satellites, which is sign of jamming. A reduction in the number of active satellites has a significant impact on the reliability of the GPS system. *vel_d_m_s* feature measures vertical velocity in meters per second. The high feature importance value of this feature suggests that jamming disrupts vertical velocity calculations, creating noisy data.

Feature *c_variance_rad* represents the variance of the path over the ground. The predictive power of this feature reflects the effect of obstructions on the stability of the computed trajectory. *hdop* and *vdop* features indicate the quality of the horizontal and vertical position solution. With significant feature importance values, these features show how jamming undermines the accuracy of GPS positioning by affecting satellite geometry. *noise_per_ms* feature measures the noise in the GPS signal. The high feature importance value of this feature indicates its effectiveness in capturing the increased signal noise associated with jamming. *s_variance_m_s* feature is the variance of the calculated speed. A high feature importance score for this feature indicates that it is useful for identifying irregular speed fluctuations caused by jamming.

In summary, if we consider effects of jamming features, jamming attacks degrade GPS signal quality by creating noise and reducing the number of effective satellites. Features such as *hdop*, *vdop*, and *noise_per_ms* effectively capture this degradation. Positional and velocity inaccuracies (*epv_x*, *epv*, *vel_d_m_s*) provide additional evidence of jamming attack.

Important features of GPS spoofing attacks identified by the XGBOOST classifier are given in Figure 7. In the figure, *eph* represents the horizontal position error similar to *eph_x* in jamming. This feature has a high feature importance score indicating its effectiveness in detecting positional inaccura-

cies.

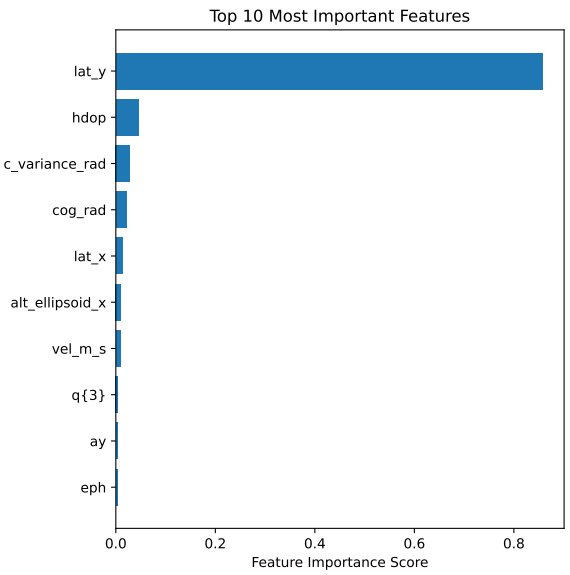


FIGURE 7. Important features for spoofing attacks.

The feature *ay* represents the acceleration along the y-axis. $q\{3\}$ represents the 3D orientation in space. These features achieve high feature importance scores, demonstrating their ability to detect the inconsistencies in motion and orientation caused by spoofed signals. *vel_m_s* feature is the overall velocity in meters per second. The feature importance score here indicates its sensitivity to altered velocity data typical of spoofing attacks.

alt_ellipsoid_x feature represents the GPS altitude relative to a reference ellipsoid. This feature's feature importance score reflects its usefulness in detecting spoofed altitude values that differ from actual measurements. *lat_x* and *lat_y* are latitude coordinates. With remarkable feature importance scores, these features demonstrate their effectiveness in capturing false positional information injected by spoofing. *cog_rad* represents the direction of travel in radians. A high feature importance score for this feature emphasizes its role in detecting spoofing changes to the directional data. *c_variance_rad* feature measures course consistency. This feature captures inconsistency in course calculations caused by spoofing based on the feature importance score. *hdop*, like jamming, measures horizontal accuracy. Unlike jamming, spoofing can artificially reduce this value to create a false sense of precision that the feature successfully identifies.

In summary, when considering the impact of spoofing features on spoofing detection, features related to a position, a motion, and an orientation are the most important. Their high feature importance values confirm their ability to detect the manipulated data injected during spoofing attacks.

Analysis of feature importance scores reveals distinct patterns in the importance of features for detecting jamming and spoofing attacks. Jamming features emphasize the signal degradation and the noise, while spoofing features emphasize the positional and the navigational inconsistencies. This distinction allows us to create targeted detection strategies for these types of GPS attacks. Thus, extracted features are proven to be defining features of the attacks according to the feature importance score. These defining features helped us both in attack detection and in similarity tests of the generated synthetic data similar to original data.

B. GENERATIVE AI MODELS CONFIGURATION

In this section, we discuss the architectural settings, hyperparameters, and baselines used for each generative model.

- VAE, we employ TVAE Synthesizer framework that uses a variational autoencoder [49]. It has different hyperparameters related to neural network like *batch_size*, *compress_dims*, and *decompress_dims*. These settings are specific to the neural network and are intended for optimizing the technical architecture and modeling process. *batch_size* parameter defines the number of data samples processed in each step with a default value of 10. Each model is trained for up to 300 *epochs*. *compress_dims* parameter specifies the size of each hidden layer in the encoder, with a default value of (128, 128). *decompress_dims* determines the size of each hidden layer in the decoder with default values (128, 128). *embedding_dim* parameter sets the embedding dimension size used by both the encoder and decoder, with a default value of 128. Regularization is controlled by *l2scale* parameter, which defaults to 1×10^{-5}. Lastly, *loss_factor* parameter acts as a multiplier for the reconstruction error, with a default value of 2.
- Gaussian Copula, we used Gaussian Copula Synthesizer, synthetic data vault (SDV), framework for the implementation [50]. First, the synthesizer learns the marginal distribution for each individual column, such as a beta distribution with parameters $\alpha = 2$ and $\beta = 5$. It uses this learned distribution to normalize the column values, transforming them into standard normal distributions with a mean of $\mu = 0$ and a standard deviation of $\sigma = 1$. Next, the synthesizer learns the covariance between each pair of normalized columns. These covariances are stored in an $n \times n$ matrix, where n represents the number of columns in the dataset.
- CTGAN, we employ fully-connected networks in both the generator and discriminator to capture all possible correlations among the columns. The generator and discriminator each utilize *two fully-connected hidden layers*. In the generator, batch normalization and *ReLU activation functions* are applied. Following the two hidden layers, the generator produces synthetic row representations using a combination of activation functions. Scalar values are generated with a *tanh activation*, while mode indicators and discrete values are generated using a *Gumbel softmax function*. In the discriminator, each hidden layer employs leaky *ReLU activation functions* and incorporates *dropout* for regularization [51].

- DDPM, models are configured with different number of neurons [256, 512, 1024, 2048, 4096, 8192] and hidden layers [4, 6, 8, 10, 12]. Each model is trained for up to 3000 epochs with a minibatch size of 512. Adam optimizer is employed for training, with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, alongside a cosine learning rate scheduler. The model weights are initialized randomly following the method described in [52].

C. IDS CONFIGURATION

We use XGBoost version 2.1.0 as XGBClassifier. The parameters of the classifier are set to default to provide a balance between the performance and the generalization. By default, *booster type* is set to $'gbtree'$, which means the model uses tree-based methods. Since our classifier performs binary classification as malicious/benign, the objective function is $'binary : logistic'$. *verbosity* level is set to 1, thereby ensuring that any relevant warning messages are displayed and that *use_label_encoder* parameter is *enabled*, which applies label encoding to the target variable.

learning rate(*eta*) is set to 0.3, which serves to regulate the step size during the optimization process. *maximumtreedepth* is 6, thereby allowing for moderately complex models. Additional tree-specific parameters include a *minimum child weight* of 1, which guarantees sufficient instance weight in child nodes, and a *gamma* value of 0, indicating that no minimum loss reduction is necessary to facilitate further splits.

The model uses all training instances with a *subsample* of 1 and considers all features per tree with a *colsample_bytree* of 1. Regularization is applied with an $L1$ *penalty* of 0 and an $L2$ *penalty* of 1. The default number of estimators is 100, and the *scale_pos_weight* is set to 1, thereby maintaining balance in binary classification tasks. The default tree construction method is "*auto*" which selects the optimal method based on the characteristics of the dataset.

D. EVALUATION METRICS

We use the following metrics to evaluate the proposed solution.

- **Kullback–Leibler (KL) Divergence:** KL divergence is a measure of the discrepancy between two probability distributions, quantifying the extent to which the synthetic data generated deviates from the actual data set.

$$D_{KL}(P\|Q) = \sum_{x\in\mathcal{X}} P(x)\cdot\log\frac{P(x)}{Q(x)} \tag{5}$$

- **Wasserstein Distance:** Wasserstein distance quantifies the difference between two probability distributions. In the case of one-dimensional distributions, it is formally defined as follows.

$$W(P,Q) = \int_{x\in\mathbb{R}} \left|P(x) - Q(x)\right| dx \tag{6}$$

- **Kolmogorov-Smirnov Statistic (KS):** KS is a measure used to determine whether two samples originate from the same distribution. It is calculated as the largest absolute difference between the cumulative distribution functions (CDFs) of the two samples, denoted as F_1 and F_2.

$$D = \max_x \left|F_1(x) - F_2(x)\right| \tag{7}$$

D quantifies the disparity between two CDFs. A p-value is then derived from KS statistic to assess the statistical significance of the observed difference.

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{8}$$

The evaluation of classification models often employs a range of metrics, including accuracy, precision, recall, and F1 score. The accuracy metric assesses the overall accuracy of the model by calculating the ratio of correctly predicted examples (true positives and true negatives) to the total number of examples. However, accuracy can be misleading in imbalanced datasets, so it is necessary to evaluate precision and recall as well.

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{9}$$

Precision is defined as the ratio of true positive predictions among all positive predictions. It indicates the accuracy of a model in identifying the positive class.

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{10}$$

Recall quantifies the proportion of true positive examples that are correctly identified by the model, thereby highlighting its capacity to detect positive cases.

$$\text{F1 Score} = 2\cdot\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}} \tag{11}$$

The F1-score is a single metric that combines the precision and recall values through a harmonic mean calculation.

$$\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{12}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}} \tag{13}$$

In combination, these metrics provide a comprehensive assessment of model performance, highlighting the various aspects of predictive accuracy and robustness.

VI. EXPERIMENTAL ANALYSIS

In this section, statistical results of synthetic attacks generated with GenAI models are analyzed. IDS performance results are given for both spoofing and jamming attacks with the generated synthetic attack data.

A. GENAI STATISTICAL RESULTS

1) GPS Jamming Attack Generation Results

Table 1 contains a detailed comparison of synthetic data generation methods CTGAN, VAE-GAN, Gaussian Copula, and DDPM for features in jamming scenarios. Metrics used for the evaluation include KL Divergence, Wasserstein Distance, and KS Statistic.

VAE struggles to match the real data distribution, showing significantly higher KL Divergence for variables like *vel_d_m_s* and *c_variance_rad*, suggesting its synthetic data is less aligned with the real data. VAE also shows higher Wasserstein Distances across most variables, with notable gaps for *vel_d_m_s* and *c_variance_rad*, confirm its weaker alignment with the real data. VAE consistently performs poorly, with higher KS Statistic scores across all variables, showing significant distributional differences between its synthetic data and the real data.

Gaussian Copula performs well for variable *eph_x* achieving competitive KL Divergence values close to CTGAN. It provides competitive Wasserstein Distance values, particularly excelling for variables *noise_er_ms*. Gaussian Copula achieves reasonable KS Statistic values for important features. The results of Gaussian Copula suggest its ability to accurately replicate certain variable distributions. Furthermore, Gaussian Copula is the model that produces the closest data to reality after DDPM.

CTGAN consistently achieves the lowest KL Divergence for most variables, such as *eph_x*, *satellites_used*, and *noise_per_ms* indicating its strength in closely approximating the distribution of real data.

DDPM demonstrates strong performance with the lowest KL Divergence, Wasserstein Distance and, KS Statistic for all important features. This highlights DDPM's capability to minimize the distributional difference between real and synthetic data. It indicats a closer match to the cumulative distribution of real data.

The evaluation of synthetic data generation methods for jamming scenarios highlights DDPM as the most reliable and consistent performer across all metrics, particularly excelling in KL Divergence and Wasserstein Distance. Its ability to closely replicate real data distributions for critical variables. Gaussian Copula provides competitive results for selected variables, such as *c_variance_rad* and *vel_d_m_s*, which indicats its suitability for these distributions. However, VAE consistently under-performs, with higher KL Divergence and Wasserstein Distance values across most variables, underscoring its limitations in accurately capturing real data characteristics. These findings emphasize the importance of selecting the appropriate synthetic data generation method based on the specific application, with DDPM emerging as the most robust choice, while CTGAN and Gaussian Copula offer valuable alternatives for variable-specific requirements.

CTGAN emerges as the most reliable model, consistently producing synthetic data with mean and standard deviation values closest to the real data across most variables. For instance, variables like *eph_x*, *satellites_used*, and *hdop* show minimal deviations, reflecting CTGAN's robust ability to replicate the central tendency and variability of real data distributions. This makes CTGAN particularly suitable for tasks that demand precise statistical replication.

DDPM performs competitively, especially for variables such as *vel_d_m_s* and *s_variance_m_s*, where it closely approximates the real data in terms of both mean and standard deviation. However, for more complex variables like *c_variance_rad* and *noise_per_ms*, DDPM demonstrates slightly larger deviations, suggesting limitations in modeling distributions with higher variability.

Gaussian Copula provides moderate accuracy and performs well for variables such as *epv* and *vdop*, achieving mean and standard deviation values relatively close to real data. However, its performance is inconsistent, as evident in variables like *epv_x* and *satellites_used*, where it shows significant deviations, indicating challenges in handling complex or multimodal distributions.

In VAE generating plausible data for some variables such as *noise_per_ms*, exhibits significant deviations in both mean and standard deviation for variables like *c_variance_rad* and *hdop*. This underperformance highlights limitations in its architecture, making it less effective for accurately replicating statistical properties of real data in jamming scenarios. Table 2 summarizes the mean and standard deviation of the specified variables for real data and synthetic data generated by CTGAN, DDPM, Gaussian Copula, and VAE has been calculated.

In general, Gaussian Copula demonstrates the most consistent and reliable performance, excelling at reproducing real data distributions across most variables. DDPM shows promise for certain features but lacks the overall consistency of Gaussian Copula. CTGAN delivers competitive results, especially for features like *eph_x* and *satellites_used*, but slightly under-performs on others. VAE exhibits strong performance for some variables but requires further optimization to achieve the consistency observed in Gaussian Copula.

The distributions of three selected important features are shown as density plots on the graphs for each model, VAE, Gaussian Copula, CTGAN, DDPM, in the jamming attack. The density plot demonstrates the distribution of the *s_variance_m_s* feature across the datasets comprising the real data, CTGAN, Gaussian Copula, DDPM, and VAE-GAN models in the context of a GPS jamming attack in Figure 8. The real data, depicted by the blue curve, represents the ground truth and exhibits an uni-modal distribution with a peak occurring at approximately 0.8. Among the generative models, Gaussian Copula, green curve, most closely resembles the real data, exhibiting a peak at nearly the same location and a similar spread.

VAE, purple curve, shows a broader distribution, deviating from the compactness of the real data while slightly overestimating the density at higher values. CTGAN, orange curve, demonstrates a reasonable alignment with the real data, though it shifts the peak slightly to the right and broadens the spread. DDPM, red curve, captures the general shape but

TABLE 1. Similarity of important features between real and synthetic data for GPS jamming attack.

Metric	Important Features	CTGAN	VAE-GAN	Gaussian Copula	DDPM
KL Divergence	*eph_x*	**1.234**	2.345	1.567	1.789
	epv	0.987	1.234	0.876	**0.765**
	satellites_used	**0.567**	2.345	1.234	0.987
	vel_d_m_s	2.345	3.456	2.123	**1.890**
	c_variance_rad	1.456	2.678	1.345	**0.987**
	epv_x	0.890	1.234	0.765	**0.654**
	hdop	0.543	0.678	0.432	**0.321**
	noise_per_ms	**1.234**	2.345	1.678	**1.234**
	vdop	0.987	1.456	0.876	**0.654**
	s_variance_m_s	0.678	1.234	0.543	**0.432**
Wasserstein Distance	*eph_x*	0.234	0.345	0.123	**0.098**
	epv	0.123	0.234	0.098	**0.076**
	satellites_used	0.456	0.678	0.432	**0.321**
	vel_d_m_s	0.567	0.789	0.543	**0.432**
	c_variance_rad	0.234	0.456	0.321	**0.210**
	epv_x	0.098	0.123	0.076	**0.054**
	hdop	0.321	0.432	0.210	**0.098**
	noise_per_ms	0.654	0.789	**0.543**	0.432
	vdop	0.210	0.321	0.098	**0.054**
	s_variance_m_s	0.098	0.123	0.076	**0.054**
KS Statistic	*eph_x*	0.543	0.654	0.432	**0.321**
	epv	0.321	0.543	0.234	**0.123**
	satellites_used	0.789	0.890	0.678	**0.543**
	vel_d_m_s	0.432	0.543	0.321	**0.234**
	c_variance_rad	0.543	0.678	0.432	**0.321**
	epv_x	0.321	0.432	0.234	**0.123**
	hdop	0.654	0.789	0.543	**0.432**
	noise_per_ms	0.543	0.654	0.432	**0.321**
	vdop	0.432	0.543	0.321	**0.234**
	s_variance_m_s	0.321	0.432	0.234	**0.123**

underestimates the density at the peak while overestimating it in the tails. This analysis suggests that Gaussian Copula is the most effective model for replicating this feature, while VAE and DDPM require further refinement to improve their fidelity. CTGAN offers a moderate balance between fidelity and diversity.

The density plot demonstrates the distribution of *epv_x* feature across the datasets in GPS jamming attacks in Figure 9. The real data, blue curve, exhibits a sharp, concentrated peak at approximately 4.375. Among the generative models, DDPM, red curve, aligns most closely with the real data, accurately replicating the peak density and shape.

Gaussian Copula, green curve, also performs well but slightly broadens the distribution, reducing its fidelity. CTGAN, orange curve, demonstrates a significant shift in the peak and a broader spread, resulting in a less accurate replication. VAE, purple curve, further deviates, exhibiting inconsistencies in both the peak density and the general shape of the distribution. These results indicate that DDPM is the most effective model for replicating this feature, with Gaussian Copula as a close second. Both CTGAN and VAE demonstrate notable deviations, highlighting areas for further optimization.

The density plot for *noise_per_ms* feature highlights the distribution modeling challenge, with real data, blue curve, that shows an uni-modal distribution centered around 100 in Figure 10. Among the generative models, Gaussian Copula, green curve, replicates the general shape of the real data but introduces slightly broader variability and subtle secondary peaks, indicating moderate fidelity.

VAE, purple curve, captures the peak but exhibits sharper curves and slightly reduced spread, which might suggest overfitting and limited diversity. CTGAN, orange curve, produces a broader distribution with a peak shifted slightly to the right, showing a moderate deviation from the real data. DDPM, red curve, also shows broader variability and slightly underrepresents the peak density.

Analyses results suggest that Gaussian Copula provides the

TABLE 2. Synthetic data comparison for GPS jamming attack.

Feature	Real Data	CTGAN	DDPM	Gaussian Copula	VAE-GAN
eph_x	1.23 ± 0.34	1.25 ± 0.36	1.21 ± 0.33	1.22 ± 0.35	1.27 ± 0.37
epv	0.98 ± 0.12	0.97 ± 0.11	0.96 ± 0.10	0.99 ± 0.13	0.95 ± 0.14
satellites_used	5.12 ± 1.34	5.15 ± 1.35	5.10 ± 1.33	5.08 ± 1.32	5.13 ± 1.36
vel_d_m_s	2.45 ± 0.54	2.50 ± 0.55	2.42 ± 0.52	2.46 ± 0.53	2.48 ± 0.56
c_variance_rad	0.23 ± 0.06	0.24 ± 0.07	0.22 ± 0.05	0.25 ± 0.08	0.21 ± 0.07
epv_x	0.34 ± 0.09	0.33 ± 0.08	0.35 ± 0.10	0.32 ± 0.09	0.36 ± 0.10
hdop	1.12 ± 0.31	1.13 ± 0.32	1.10 ± 0.30	1.15 ± 0.33	1.14 ± 0.34
noise_per_ms	0.45 ± 0.12	0.46 ± 0.13	0.44 ± 0.11	0.47 ± 0.14	0.43 ± 0.12
vdop	0.67 ± 0.18	0.66 ± 0.17	0.68 ± 0.19	0.69 ± 0.20	0.65 ± 0.18
s_variance_m_s	0.78 ± 0.22	0.80 ± 0.23	0.76 ± 0.21	0.79 ± 0.24	0.77 ± 0.22

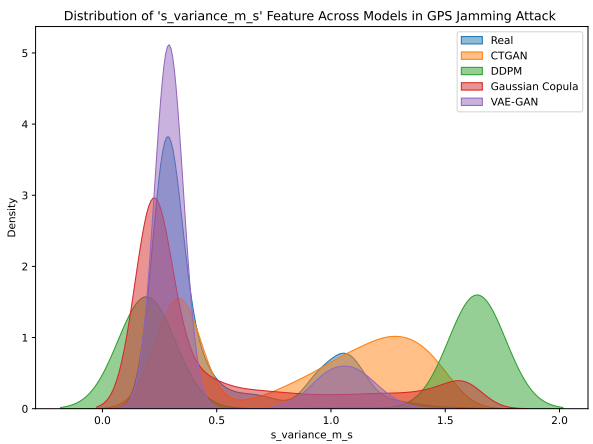


FIGURE 8. Distributions of 's_variance_m_s' feature across models in GPS jamming attacks.

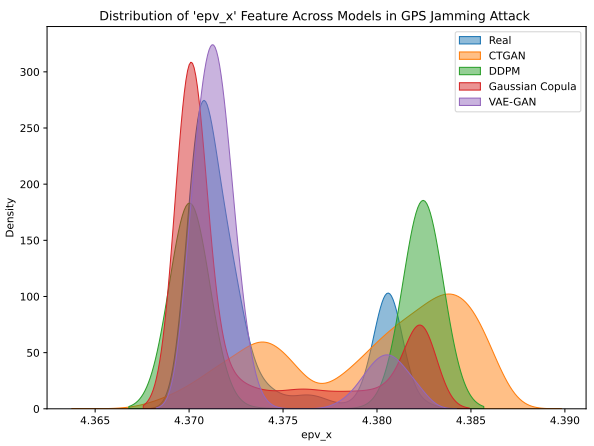


FIGURE 9. Distributions of 'epv_x' feature across models in GPS jamming attacks.

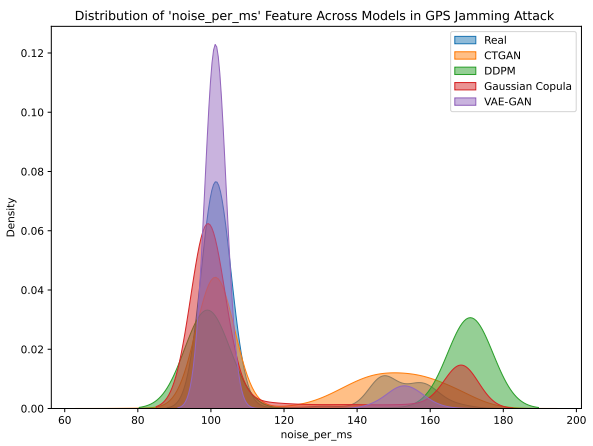


FIGURE 10. Distributions of 'noise_per_ms' feature across models in GPS jamming attacks.

best overall balance between fidelity and diversity for this feature, while VAE offers high fidelity but limited diversity. CTGAN and DDPM demonstrate moderate performance but deviate in peak location and spread, which indicates areas for improvement.

2) GPS Spoofing Attack Generation Results

VAE struggles with consistently higher KL Divergence values, particularly for $q\{3\}$ and *cog_rad*, which indicates a poor alignment with the real data distributions. VAE underperforms, with higher Wasserstein Distance values for most

variables, especially *cog_rad* and *lat_x*. Furthermore, it indicates significant distributional differences. VAE consistently shows higher KS Statistic values, particularly for *cog_rad* and *lat_x*, reflecting significant discrepancies in distribution alignment.

Gaussian Copula KL divergence values are close to DDPM in capturing some features, on the other hand, it falls behind CTGAN in capturing some features (*eph*, *vel_m_s*,*cog_rad*, *c_variance_rad*). Gaussian Copula achieves notable KS Statistic values for key features, demonstrating its capability to accurately replicate certain variable distributions. Additionally, Gaussian Copula is the second most effective model, producing synthetic data closely aligned with real data, following DDPM.

CTGAN exhibits strong performance across most variables, achieving the lowest KL Divergence values for variables such as *eph*, *vel_m_s* and *c_variance_rad*. This demonstrates its capability to approximate real data distributions closely.

DDPM exhibits outstanding performance, achieving the lowest KL Divergence, Wasserstein Distance, and KS Statistic in almost all key features. This underscores DDPM's ability to effectively minimize distributional differences between real and synthetic data, indicating a closer alignment with the cumulative distribution of the real data. Table 3 contains a detailed comparison of synthetic data generation methods, CTGAN, VAE, Gaussian Copula, and DDPM for important features in jamming scenarios.

DDPM emerges as the most reliable synthetic data generation method for spoofing scenarios, consistently outperforming other methods in replicating distributions of real data. DDPM provides strong results for select variables, showcasing its potential for variable-specific applications. VAE requires substantial improvement, as it consistently fails to align its synthetic data with real data distributions. These findings reinforce the importance of selecting the appropriate synthetic data generation method based on the specific variables and metrics critical to the application.

CTGAN demonstrates moderate performance but it does not consistently achieve the closest approximation to real data. While it performs reasonably well for variables like *eph* and *lat_x*, it exhibits noticeable deviations for variables such as *hdop* and *c_variance_rad*. These deviations suggest that while CTGAN can capture central tendencies, it struggles with higher accuracy in complex variable distributions.

DDPM performs strongly overall with closely matching the real data for variables like *vel_m_s* and *alt_ellipsoid_x*. However, its performance slightly lags for variables like *cog_rad*, where deviations are small but more noticeable compared to Gaussian Copula. DDPM consistently minimizes errors in both mean and variance, making it a strong contender, especially for features with moderate complexity.

Gaussian Copula demonstrates moderate accuracy, particularly excelling in variables like *c_variance_rad* and *q{3}*. However, its higher variability in key metrics such as *lat_x* and *hdop* indicates limitations in its ability to model distributions with higher complexity. Gaussian Copula demonstrates solid performance, particularly excelling in variables like *c_variance_rad* and $q\{3\}$. However, it shows higher variability for features like *lat_x* and *hdop*, which slightly reduces its overall accuracy. Despite this, Gaussian Copula remains effective for simpler variable distributions and demonstrates reasonable fidelity to real data.

VAE shows mixed performance with significant deviations in variables like *lat_y* and *cog_rad*. While it performs adequately for variables like *vel_m_s*, its inability to consistently replicate both the mean and standard deviation of the real data highlights areas requiring optimization. These results suggest that VAE is less suited for replicating complex data distributions. Table 4 summarizes the mean and standard deviation of specified variables for real data and synthetic data generated by CTGAN, DDPM, Gaussian Copula, and VAE has been calculated.

Overall, DDPM emerges as the most reliable model, excelling in closely replicating the real data for most features, especially those with moderate complexity. Gaussian Copula performs well overall, particularly for simpler features, but exhibits slightly higher variability for complex variables. CTGAN demonstrates moderate performance but struggles with certain variables, resulting in less accurate replication compared to DDPM and Gaussian Copula. VAE shows the most significant deviations, indicating the need for further refinement to improve its ability to replicate real data distributions accurately.

The distribution of the three most important features is shown as density plots on the graphs for each model (VAE, Gaussian Copula, CTGAN, DDPM)in the spoofing attack. The distribution of *lat_y* feature compares real data, CTGAN, Gaussian Copula, and VAE-GAN in GPS spoofing attacks as in Figure 11. Real data exhibits a sharp and concentrated unimodal peak, indicating low variability in this feature. Among generative models, DDPM emerges as the most effective in replicating the distribution, closely matching the real data's peak and spread. VAE-GAN also performs well but it shows signs of over-fitting, with a sharper and overestimated peak. CTGAN and Gaussian Copula under-perform, with CTGAN producing a flatter distribution and Gaussian Copula failing to capture the compactness of the real data. Overall, DDPM is the best model for this feature, while VAE-GAN may be considered for applications where slight over-fitting is acceptable

The distribution of *cog_rad* feature observed during GPS spoofing attacks shows that CTGAN again closely matches the real data distribution as in Figure 12. The real data exhibits a bimodal distribution with distinct peaks. Among the models, DDPM most effectively replicates the bimodal nature, though it slightly underestimates the density of the secondary peak. Gaussian Copula captures the general shape but produces broader peaks with reduced sharpness. VAE-GAN struggles to replicate the secondary peak, focusing excessively on the primary peak. CTGAN fails to reproduce the bimodal structure, instead generating a flatter, less accurate distribution.

TABLE 3. Similarity of features between real and synthetic data for GPS spoofing attacks.

Metric	Important Features	CTGAN	VAE	Gaussian Copula	DDPM
KL Divergence	*eph*	**1.123**	2.456	1.789	1.987
	ay	0.876	1.234	0.765	**0.654**
	q{3}	2.456	3.789	2.123	**1.890**
	vel_m_s	**0.765**	1.234	0.987	0.876
	alt_ellipsoid_x	1.345	2.456	1.234	**0.987**
	lat_x	0.543	0.678	0.432	**0.321**
	cog_rad	1.678	2.345	1.890	**1.234**
	c_variance_rad	**1.234**	2.345	1.567	1.432
	hdop	0.987	1.234	0.876	**0.765**
	lat_y	0.678	1.234	0.543	**0.432**
Wasserstein Distance	*eph*	0.234	0.345	0.123	**0.098**
	ay	0.123	0.234	0.098	**0.076**
	q{3}	0.567	0.789	0.543	**0.432**
	vel_m_s	0.321	0.432	0.210	**0.098**
	alt_ellipsoid_x	0.654	0.789	0.543	**0.432**
	lat_x	0.098	0.123	0.076	**0.054**
	cog_rad	0.210	0.321	0.098	**0.054**
	c_variance_rad	0.098	0.123	0.076	**0.054**
	hdop	0.234	0.345	0.123	**0.098**
	lat_y	0.210	0.321	0.098	**0.054**
KS Statistic	*eph*	0.543	0.654	0.432	**0.321**
	ay	0.321	0.543	0.234	**0.123**
	q{3}	0.789	0.890	0.678	**0.543**
	vel_m_s	0.432	0.543	0.321	**0.234**
	alt_ellipsoid_x	0.543	0.678	0.432	**0.321**
	lat_x	0.321	0.432	0.234	**0.123**
	cog_rad	0.654	0.789	0.543	**0.432**
	c_variance_rad	0.543	0.654	0.432	**0.321**
	hdop	0.432	0.543	0.321	**0.234**
	lat_y	0.321	0.432	0.234	**0.123**

DDPM is the best model for this feature, while Gaussian Copula provides a reasonable alternative for applications tolerating reduced peak definition.

The distribution of *c_variance_rad* feature compares real data, CTGAN, Gaussian Copula, and VAE-GAN in the context of a GPS spoofing attack as in figure 13. Real data displays a sharp uni-modal distribution with a peak around 0.5. DDPM emerges as the most accurate model, closely replicating the peak and spread of the real data. VAE-GAN performs well but slightly overestimates the peak density, indicating possible over-fitting. CTGAN generates a broader peak, achieving a balance between fidelity and diversity. Gaussian Copula fails to capture the sharpness of the real data, producing a flatter and less accurate distribution. Overall, DDPM is the most effective model for this feature, while VAE-GAN offers high fidelity with the risk of overfitting, and CTGAN provides a generalized alternative with increased variability.

B. IDS PERFORMANCE RESULTS

The performance analyses of the intrusion detection system is conducted by using both real and GenAI-synthesized data for jamming and spoofing attacks. Figures 14 and 15 present the classification accuracy of XGBoost on datasets generated by models VAE, Gaussian Copula, CTGAN, and DDPM alongside real data. The primary objective is to achieve lower IDS accuracy, indicating a more effective attack simulation.

In this context, the aim is to achieve lower accuracy that indicates a more successful attack. When looking at the accuracy and F1-score, it is more difficult to detect spoofing attacks and jamming attacks produced with CTGAN. Real data reaches the highest final accuracy of 100% for both attacks, indicating an almost perfect classification. VAE also shows high accuracy and an F1-score of 98% for jamming attacks

TABLE 4. Synthetic data comparison for GPS spoofing attack.

Feature	Real Data	CTGAN	DDPM	Gaussian Copula	VAE-GAN
eph	1.23 ± 0.34	1.21 ± 0.35	1.22 ± 0.33	1.24 ± 0.37	1.25 ± 0.36
ay	0.45 ± 0.12	0.44 ± 0.13	0.46 ± 0.11	0.47 ± 0.14	0.43 ± 0.12
q{3}	2.45 ± 0.54	2.50 ± 0.55	2.48 ± 0.52	2.46 ± 0.56	2.42 ± 0.53
vel_m_s	3.12 ± 0.78	3.10 ± 0.79	3.11 ± 0.77	3.13 ± 0.80	3.15 ± 0.78
alt_ellipsoid_x	4.23 ± 1.12	4.25 ± 1.13	4.21 ± 1.10	4.22 ± 1.14	4.24 ± 1.12
lat_x	5.78 ± 1.45	5.75 ± 1.46	5.80 ± 1.44	5.76 ± 1.47	5.79 ± 1.45
cog_rad	1.89 ± 0.47	1.88 ± 0.48	1.90 ± 0.46	1.87 ± 0.49	1.85 ± 0.48
c_variance_rad	0.67 ± 0.18	0.66 ± 0.19	0.65 ± 0.17	0.68 ± 0.20	0.69 ± 0.18
hdop	1.12 ± 0.31	1.11 ± 0.32	1.13 ± 0.30	1.14 ± 0.33	1.10 ± 0.31
lat_y	6.23 ± 1.54	6.22 ± 1.55	6.21 ± 1.53	6.24 ± 1.56	6.25 ± 1.54

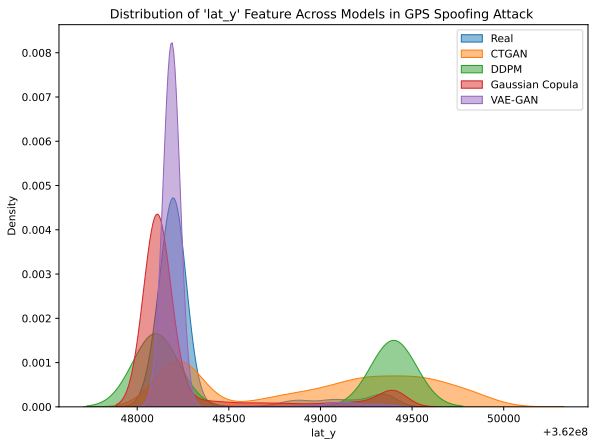


FIGURE 11. Distributions of 'lat_y' feature across models in GPS spoofing attacks.

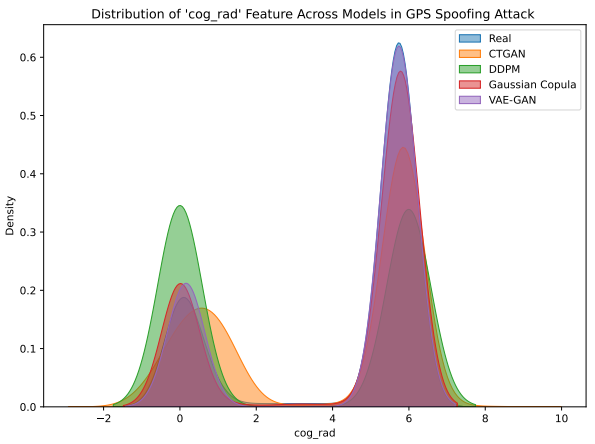


FIGURE 12. Distributions of 'cog_rad' feature across models in GPS spoofing attacks.

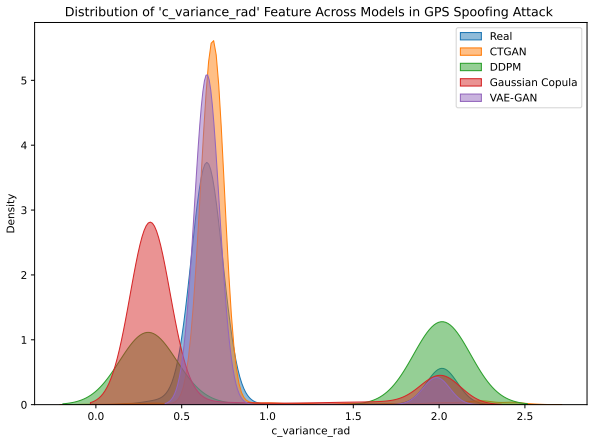


FIGURE 13. Distributions of 'c_variance_rad' feature across models in GPS spoofing attacks.

and 98% for spoofing attacks, suggesting that the synthetic data produced with VAE cannot significantly increase the real jamming attack data. Malicious jamming data produced with VAE is limited to 4.6%. In addition, malicious spoofing data produced with VAE is limited to 12%. This indicates that although VAE can produce synthetic data similar to real attack data in certain aspects, it cannot adequately capture the typical complexities or variations of complex cyber-attacks. As a result, IDS can still detect synthetic attacks generated with VAE relatively easily.

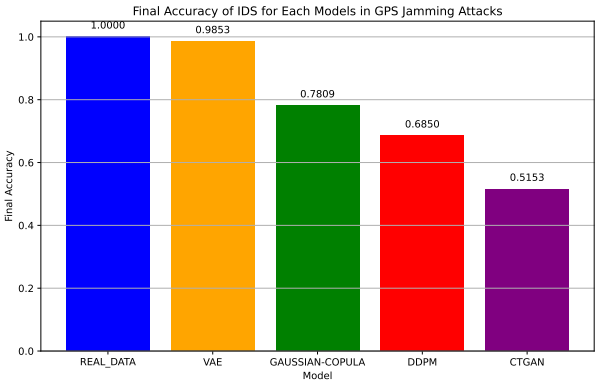


FIGURE 14. Jamming attack detection accuracy.

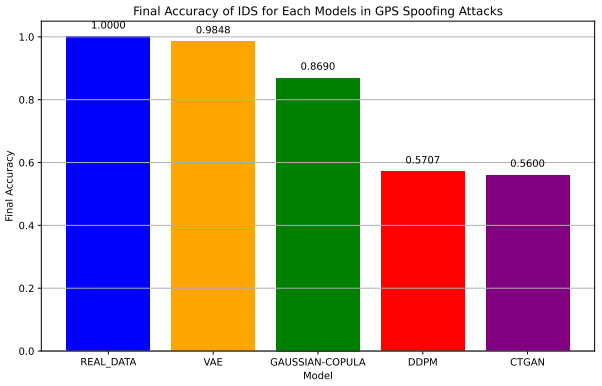


FIGURE 15. Spoofing attack detection accuracy.

Gaussian Copula achieved a lower accuracy of 78% and an F1-score of 76%, which indicates that synthetic data generated by this method are more difficult to distinguish from real compression attacks. Low accuracy and F1-score suggest that the IDS has greater difficulty detecting jamming attacks generated by synthetic data compared to real jamming attacks. Additionally, the IDS achieved an accuracy of 86% and an F1-score of 84% for spoofing attacks. Detecting synthetic spoofing attacks generated using the Gaussian copula is more challenging compared to real spoofing attacks, with a 14% decrease in IDS accuracy observed for spoofing attacks and a 22% decrease for jamming attacks. These results highlight that Gaussian copula is more effective in capturing the subtle features of attack data compared to VAE. Furthermore, the proportion of malicious data generated using the Gaussian Copula is 23.6% for jamming attacks and 12.8% for spoofing attacks, demonstrating that the Gaussian Copula increases both the quantity and diversity of generated data more effectively than the VAE.

DDPM achieved a lower accuracy and F1-score of 68%, indicating that synthetic jamming attack data generated with CTGAN challenges IDS more effectively. This decreased accuracy and F1-score suggest that IDS has more difficulty detecting synthetic jamming attacks generated with DDPM than Gaussian copula and VAE models. Additionally, IDS achieved an accuracy and F1-score of 57% for spoofing attacks. Synthetic spoofing attacks generated using DDPM were more difficult to detect than real spoofing attacks and VAE, Gaussian Copula models.

CTGAN has the lowest accuracy and F1-score of 51%, indicating that synthetic jamming attack data generated with CTGAN challenges IDS more effectively. This decreased accuracy and F1-score suggest that IDS has more difficulty detecting synthetic compression attacks generated with CTGAN than other models compared to real jamming attacks. Additionally, IDS achieved an accuracy of 56% and F1-score of 54% for spoofing attacks. Synthetic spoofing attacks generated using CTGAN were more difficult to detect than real spoofing attacks and other generative models.

IDS accuracy for GPS jamming attacks with the baseline accuracy was obtained with real data of all models. A 49% decrease with CTGAN-generated data, a 32% decrease with DDPM-generated data, a 22% decrease with Gaussian copula-generated data, and only a 2% decrease when using VAE-generated data were observed. The results demonstrate that DDPM is particularly effective at generating synthetic data that mimics real attack behaviors and patterns. The significant 32% decrease in accuracy compared to real data underscores the challenge that DDPM-generated attacks pose to IDS, highlighting the potential for training more robust, adaptive intrusion detection systems.

The accuracy of IDS for GPS spoofing attacks with the baseline accuracy with real data across all models. A 44% decrease was observed for CTGAN-generated data, 43% for DDPM-generated data, 14% for Gaussian copula-generated data, and only 2% for VAE-generated data was observed . These results demonstrate that DDPM is particularly effective at generating synthetic data that mimics real attack behaviors and patterns. The significant 43% and 44% decrease in accuracy compared to real data underscores the challenge that CTGAN and DDPM-generated attacks pose for IDS, highlighting the potential for training more robust, adaptive intrusion detection systems. While CTGAN significantly challenges IDS accuracy for GPS jamming attacks, CTGAN and DDPM achieve similar IDS results for GPS spoofing attacks.

TABLE 5. Results of GPS jamming attack on IDS.

Model	Accuracy	f1 Score	Precision	Recall
REAL	100%	100%	100%	100%
VAE	98%	98%	98%	98%
GAUSSIAN COPULA	78%	76%	76%	78%
DDPM	68%	68%	68%	68%
CTGAN	51%	51%	50%	51%

C. DISCUSSION

Analyses results demonstrate that the effectiveness of synthetic attack data in challenging IDS performance is strongly tied to the fidelity and the diversity of the generated data.

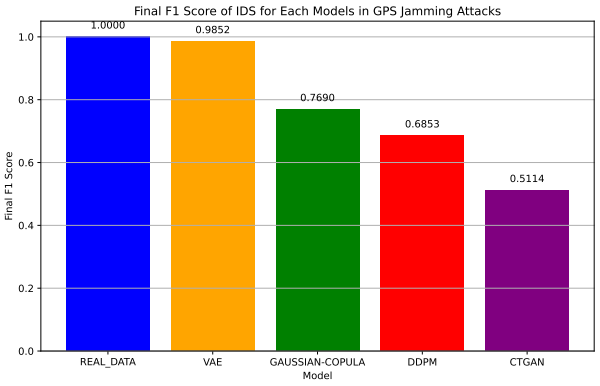


FIGURE 16. Jamming attack detection F1-score.

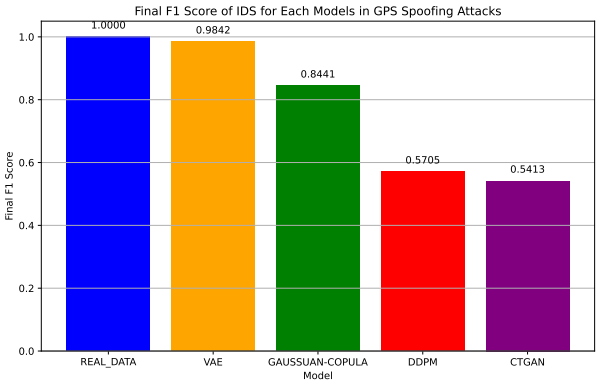


FIGURE 17. Spoofing attack detection F1-score.

CTGAN and DDPM outperform other GenAI models in reducing IDS accuracy, particularly for jamming and spoofing attacks, due to their superior ability to capture the intricate patterns and variability present in real attack data. This observation is reinforced by low KL divergence, Wasserstein distance, and KS statistic values for critical features, indicating high similarity between synthetic and real data.

Gaussian Copula model, while moderately effective, falls short of CTGAN and DDPM in achieving high-fidelity data generation. However, it is the second strongest model after DDPM in producing realistic synthetic data. In contrast, VAE performed the worst among all models, with minimal reduction in IDS accuracy and relatively high similarity scores only for basic features. This suggests that VAE struggles to replicate the complexities and variability inherent in real-world attack scenarios.

TABLE 6. Results of GPS spoofing attacks on IDS.

Model	Accuracy	f1 Score	Precision	Recall
REAL	100%	100%	100%	100%
VAE	98%	98%	98%	98%
GAUSSIAN COPULA	86%	84%	83%	86%
DDPM	57%	57%	57%	57%
CTGAN	56%	54%	53%	56%

The combined analysis of IDS performance and data similarity metrics underscores the importance of selecting high-fidelity generative models like CTGAN and DDPM for simulating realistic adversarial conditions. These models not only degrade IDS performance but also exhibit greater fidelity in replicating real attack patterns, as validated through similarity metrics. This combined evaluation is critical for training more robust IDS frameworks capable of detecting sophisticated cyber-attacks.

VII. CONCLUSION

The primary aim of this research is to show that the performance of IDS can be reduced by leveraging synthetic attack data produced by generative artificial intelligence models. An IDS framework was designed and evaluated using synthetic datasets created by VAE, Gaussian Copula, DDPM, and CTGAN. Analyses results show that real data consistently outperformed generative models in terms of attack detection accuracy, supporting our hypothesis that current IDS systems remain more effective at identifying real-world attacks.

Analyses results highlight the potential of GenAI models, particularly DDPM and Gaussian Copula, in emulating real attack patterns. DDPM significantly degraded IDS accuracy for both jamming and spoofing attacks, achieving high-fidelity synthetic data as validated through similarity metrics such as KL divergence, Wasserstein distance, and KS statistic. These findings confirm that generative models may effectively imitate real datasets and challenge IDS frameworks.

In this research, we showed that synthetic data may closely mimic real attack behavior, emphasizing the need for more robust and adaptive attack detection systems. By integrating high-fidelity synthetic data during IDS training, future systems may better address evolving cyber attacks and improve detection capabilities of IDSs. Further research should explore hybrid approaches that combine real and synthetic data to enhance IDS resilience against complex attack vectors. This circumstance will help researchers to create more accurate intrusion detection algorithms that make computing systems more sustainable.

REFERENCES

[1] Naveen Kumar and Ankit Chaudhary. Surveying cybersecurity vulnerabilities and countermeasures for enhancing uav security. *Computer Networks*, 252:110695, 2024.

[2] Nils Miro Rodday, Ricardo de O. Schmidt, and Aiko Pras. Exploring security vulnerabilities of unmanned aerial vehicles. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pages 993–994, 2016.

[3] C. G. Leela Krishna and Robin R. Murphy. A review on cybersecurity vulnerabilities for unmanned aerial vehicles. In *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pages 194–199, 2017.

[4] Aristides Mpitziopoulos, Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou. A survey on jamming attacks and countermeasures in wsns. *IEEE Communications Surveys Tutorials*, 11(4):42–56, 2009.

[5] Jie Su, Jianping He, Peng Cheng, and Jiming Chen. A stealthy gps spoofing strategy for manipulating the trajectory of an unmanned aerial ve-

hicle**this work is supported by national science foundation of china under grant u1401253 and national key rd program under grant 2016yfb0800204. *IFAC-PapersOnLine*, 49(22):291–296, 2016. 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2016.

[6] Andrew J. Kerns, Daniel P. Shepard, Jahshan A. Bhatti, and Todd E. Humphreys. Unmanned aircraft capture and control via gps spoofing. *J. Field Robot.*, 31(4):617–636, jul 2014.

[7] Jason Whelan, Thanigajan Sangarapillai, Omar Minawi, Abdulaziz Almehmadi, and Khalil El-Khatib. Novelty-based intrusion detection of sensor attacks on unmanned aerial vehicles. In *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Q2SWinet '20, page 23–28, New York, NY, USA, 2020. Association for Computing Machinery.

[8] Burcu Sönmez Sarıkaya and Şerif Bahtiyar. A survey on security of uav and deep reinforcement learning. *Ad Hoc Networks*, 164:103642, 2024.

[9] Katrina Mansfield, Timothy Eveleigh, Thomas H. Holzer, and Shahryar Sarkani. Unmanned aerial vehicle smart device ground control station cyber security threat model. In *2013 IEEE International Conference on Technologies for Homeland Security (HST)*, pages 722–728, 2013.

[10] Pietro Boccadoro, Domenico Striccoli, and Luigi Alfredo Grieco. An extensive survey on the internet of drones. *Ad Hoc Networks*, 122:102600, 2021.

[11] Jie Su, Jianping He, Peng Cheng, and Jiming Chen. A stealthy gps spoofing strategy for manipulating the trajectory of an unmanned aerial vehicle. *IFAC-PapersOnLine*, 49(22):291–296, 2016. 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2016.

[12] Liang Xiao, Caixia Xie, Minghui Min, and Weihua Zhuang. User-centric view of unmanned aerial vehicle transmission against smart attacks. *IEEE Transactions on Vehicular Technology*, 67:3420–3430, 2018.

[13] Arslan Shafique, Abid Mehmood, and Mourad Elhadef. Survey of security protocols and vulnerabilities in unmanned aerial vehicles. *IEEE Access*, 9:46927–46948, 2021.

[14] Waleed Aldosari. Received power based unmanned aerial vehicles (uavs) jamming detection and nodes classification using machine learning. *Computers, Materials and Continua*, 75(1):1253–1269, 2023.

[15] Princess Joeaneke, Onyinye Obioha Val, Oluwaseun Oladeji Olaniyi, Olumide Samuel Ogungbemi, Anthony Obulor Olisa, and Oluwaseun Ibrahim Akinola. Protecting autonomous uavs from gps spoofing and jamming: A comparative analysis of detection and mitigation techniques. *Oluwaseun Oladeji and Ogungbemi, Olumide Samuel and Olisa, Anthony Obulor and Akinola, Oluwaseun Ibrahim, Protecting Autonomous UAVs from GPS Spoofing and Jamming: A Comparative Analysis of Detection and Mitigation Techniques (October 03, 2024)*, 2024.

[16] Dingchen She, Wei Wang, Zhisheng Yin, Jiaqi Wang, and Haifeng Shan. Gps spoofing attack recognition for uavs with limited samples. *IEEE Internet of Things Journal*, 2024.

[17] Muhammad Umer, Imran Ashraf, Yongwan Park, et al. Enhanced machine learning ensemble approach for securing small unmanned aerial vehicles from gps spoofing attacks. *IEEE Access*, 2024.

[18] Jason Whelan, Thanigajan Sangarapillai, Omar Minawi, Abdulaziz Almehmadi, and Khalil El-Khatib. Novelty-based intrusion detection of sensor attacks on unmanned aerial vehicles. In *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Q2SWinet '20, page 23–28, New York, NY, USA, 2020. Association for Computing Machinery.

[19] Tiep M. Hoang, Nghia M. Nguyen, and Trung Q. Duong. Detection of eavesdropping attack in uav-aided wireless systems: Unsupervised learning with one-class svm and k-means clustering. *IEEE Wireless Communications Letters*, 9(2):139–142, 2020.

[20] Menaka Pushpa Arthur. Detecting signal spoofing and jamming attacks in uav networks using a lightweight ids. In *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–5, 2019.

[21] Mesay Belete Bejiga, Abdallah Zeggada, Abdelhamid Nouffidj, and Farid Melgani. A convolutional neural network approach for assisting avalanche search and rescue operations with uav imagery. *Remote Sensing*, 9(2), 2017.

[22] Bowon Yang, Eric T. Matson, Anthony H. Smith, J. Eric Dietz, and John C. Gallagher. Uav detection system with multiple acoustic nodes using machine learning models. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 493–498, 2019.

[23] Abbas Yazdinejad, Reza M. Parizi, Ali Dehghantanha, and Hadis Karimipour. Federated learning for drone authentication. *Ad Hoc Networks*, 120:102574, 2021.

[24] Muneeba Asif, Mohammad Ashiqur Rahman, Kemal Akkaya, Hossain Shahriar, and Alfredo Cuzzocrea. Adversarial data-augmented resilient intrusion detection system for unmanned aerial vehicles. In *2023 IEEE International Conference on Big Data (BigData)*, pages 5428–5437. IEEE, 2023.

[25] Seonghoon Jeong, Eunji Park, Kang Uk Seo, Jeong Do Yoo, and Huy Kang Kim. Muvids: false mavlink injection attack detection in communication for unmanned vehicles. In *Workshop on automotive and autonomous vehicle security (AutoSec)*, volume 2021, page 25, 2021.

[26] Jeong Do Yoo, Haerin Kim, and Huy Kang Kim. Guide: Gan-based uav ids enhancement. *Computers & Security*, 147:104073, 2024.

[27] Md Hasan Shahriar, Nur Imtiazul Haque, Mohammad Ashiqur Rahman, and Miguel Alonso. G-ids: Generative adversarial networks assisted intrusion detection system. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 376–385. IEEE, 2020.

[28] Zilong Lin, Yong Shi, and Zhi Xue. Idsgan: Generative adversarial networks for attack generation against intrusion detection. In João Gama, Tianrui Li, Yang Yu, Enhong Chen, Yu Zheng, and Fei Teng, editors, *Advances in Knowledge Discovery and Data Mining*, pages 79–91, Cham, 2022. Springer International Publishing.

[29] Eunbi Seo, Hyun Min Song, and Huy Kang Kim. Gids: Gan based intrusion detection system for in-vehicle network. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–6, 2018.

[30] Burcu Sönmez Sarikaya and Şerif Bahtiyar. Generative adversarial networks for synthetic jamming attacks on uavs. In *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pages 760–765, 2024.

[31] Garima Agrawal, Amardeep Kaur, and Sowmya Myneni. A review of generative models in generating synthetic attack data for cybersecurity. *Electronics*, 13(2):322, 2024.

[32] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE international conference on data science and advanced analytics (DSAA)*, pages 399–410. IEEE, 2016.

[33] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019.

[34] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7499–7519, June 2024.

[35] Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

[36] Jascha Narain Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ArXiv*, abs/1503.03585, 2015.

[37] Zhiqiang Wan, Yazhou Zhang, and Haibo He. Variational autoencoder based synthetic data generation for imbalanced learning. In *2017 IEEE symposium series on computational intelligence (SSCI)*, pages 1–7. IEEE, 2017.

[38] Alvaro Figueira and Bruno Vaz. Survey on synthetic data generation, evaluation methods and gans. *Mathematics*, 10(15):2733, 2022.

[39] Mina Razghandi, Hao Zhou, Melike Erol-Kantarci, and Damla Turgut. Variational autoencoder generative adversarial network for synthetic data generation in smart home. In *ICC 2022-IEEE International Conference on Communications*, pages 4781–4786. IEEE, 2022.

[40] Florian M Hollenbach, Iavor Bojinov, Shahryar Minhas, Nils W Metternich, Michael D Ward, and Alexander Volfovsky. Multiple imputation using gaussian copulas. *Sociological Methods & Research*, 50(3):1259–1283, 2021.

[41] Aryan Pathare, Ramchandra Mangrulkar, Kartik Suvarna, Aryan Parekh, Govind Thakur, and Aruna Gawade. Comparison of tabular synthetic data generation techniques using propensity and cluster log metric. *International Journal of Information Management Data Insights*, 3(2):100177, 2023.

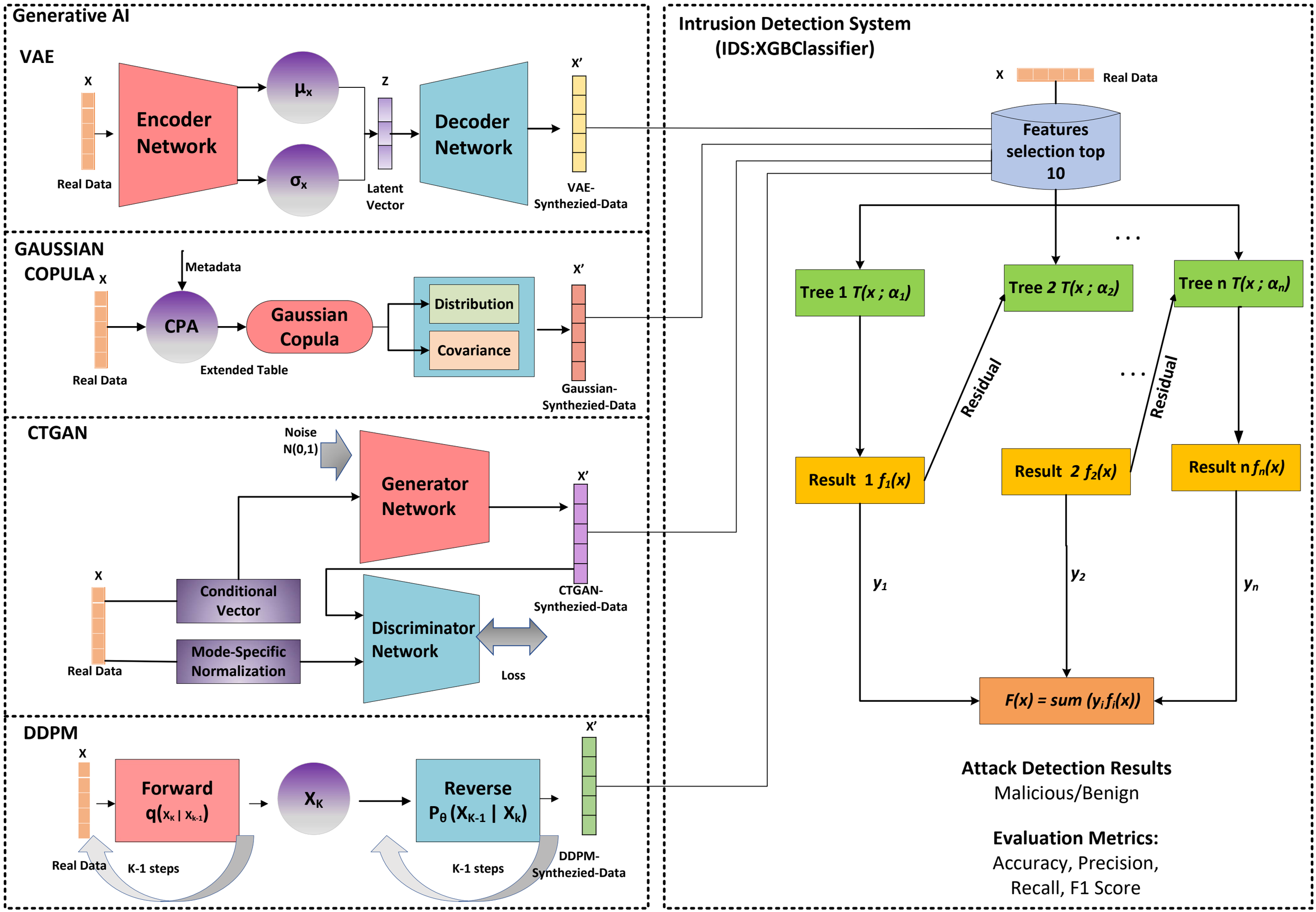
[42] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *ArXiv*, abs/2102.09672, 2021.

[43] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[44] Yonghui Xu, Xi Zhao, Yinsheng Chen, and Zixuan Yang. Research on a mixed gas classification algorithm based on extreme random tree. *Applied Sciences*, 9(9), 2019.
[45] Jason Whelan, Abdulaziz Almehmadi, and Khalil El-Khatib. Artificial intelligence for intrusion detection systems in unmanned aerial vehicles. *Computers and Electrical Engineering*, 99:107784, 2022.
[46] Anis Koubâa, Azza Allouch, Maram Alajlan, Yasir Javed, Abdelfettah Belghith, and Mohamed Khalgui. Micro air vehicle link (mavlink) in a nutshell: A survey. *IEEE Access*, 7:87658–87680, 2019.
[47] Young-Min Kwon, Jaemin Yu, Byeong-Moon Cho, Yongsoon Eun, and Kyung-Joon Park. Empirical analysis of mavlink protocol vulnerability for attacking unmanned aerial vehicles. *IEEE Access*, 6:43203–43212, 2018.
[48] Azza Allouch, Omar Cheikhrouhou, Anis Koubâa, Mohamed Khalgui, and Tarek Abbes. Mavsec: Securing the mavlink protocol for ardupilot/px4 unmanned aerial systems. *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 621–628, 2019.
[49] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems*, 2019.
[50] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, 2016.
[51] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. *Modeling tabular data using conditional GAN*. Curran Associates Inc., Red Hook, NY, USA, 2019.
[52] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

BURCU SÖNMEZ SARIKAYA is a Ph.D. student in the Department of Computer Engineering at Istanbul Technical University. She is a member of Cyber Security and Privacy Research Laboratory, SPF LAB, at Istanbul Technical University. Her research interests include UAVs security, machine learning, deep learning security.

ŞERİF BAHTİYAR is an associate professor in the Department of Computer Engineering at Istanbul Technical University. He received his BS in Control and Computer Engineering and MS in Computer Engineering degrees both from Istanbul Technical University respectively, and his Ph.D. degree in Computer Engineering from Bogaziçi University. Dr. Bahtiyar was with MasterCard, TU-Berlin in Germany, and National Research Institute of Electronics and Cryptology. Dr. Bahtiyar is the founder and the director of Cyber Security and Privacy Research Laboratory, SPF LAB, at Istanbul Technical University. His current research includes cyber security and privacy, mobile systems, trust modeling, machine learning, e-health, UAV, and financial systems.


Generative AI
VAE
X
Real Data
Encoder Network
μ_x
σ_x
Z
Latent Vector
Decoder Network
X'
VAE-Synthezied-Data
GAUSSIAN COPULA
X
Real Data
Metadata
CPA
Extended Table
Gaussian Copula
Distribution
Covariance
X'
Gaussian-Synthezied-Data
CTGAN
Noise N(0,1)
Generator Network
X'
CTGAN-Synthezied-Data
X
Real Data
Conditional Vector
Mode-Specific Normalization
Discriminator Network
Loss
DDPM
X
Real Data
Forward $q(x_k \mid x_{k-1})$
K-1 steps
X_K
Reverse $P_\theta (X_{K-1} \mid X_k)$
K-1 steps
X'
DDPM-Synthezied-Data
Intrusion Detection System (IDS:XGBClassifier)
X
Real Data
Features selection top 10
Tree 1 $T(x ; \alpha_1)$
Tree 2 $T(x ; \alpha_2)$
Tree n $T(x ; \alpha_n)$
Residual
Residual
Result 1 $f_1(x)$
Result 2 $f_2(x)$
Result n $f_n(x)$
y_1
y_2
y_n
$F(x) = sum (y_i f_i(x))$
Attack Detection Results
Malicious/Benign
Evaluation Metrics:
Accuracy, Precision,
Recall, F1 Score

Overview of the proposed framework: Generative AI models (VAE, Gaussian Copula, DDPM, CTGAN) synthesize jamming and spoofing attack data for UAVs. The framework evaluates IDS performance by analyzing detection accuracy and statistical similarity metrics, demonstrating how synthetic attacks degrade IDS effectiveness and highlighting the need for more adaptive security solutions.