VOL. 21, NO. 12, JUNE 2026 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences
©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.
www.arpnjournals.com

HYBRID VAE-GAN FRAMEWORK FOR SYNTHETIC IOT TRAFFIC GENERATION AND INTRUSION DETECTION

Alakunta Vyshnavi and D. Mahendra Reddy
Department of Computer Science and Engineering, JNTUA College of Engineering, Pulivendula, India
E-Mail: alakuntlavyshnavi855@gmail.com

ABSTRACT

The IoT networks have continued to face severe issues in intrusion detection that are linked to the absence of data, class imbalances and the inadequacies of simulator-generated traces. The paper introduces SynthNet, which is a hybrid framework that VAE-GAN-based data generating framework, which is assisted by the simulators of IoT to enhance the performance of multi-class intrusion detection. The generative module is being trained to generate latent representation of flow-level IoT traffic and the generated normal and attack samples are used to augment the training distribution. Such simulated flows are added along with traffic provided by the simulator to form an augmented data to train a multi-class IDS. The suggested solution is demonstrated to be efficient because, according to the experimental analysis, SynthNet has the accuracy of 0.998848, the macro F1-score of 0.9649, and the weighted F1-score of 0.9989 in a 7-class IoT dataset. The confusions matrices, ROC and precision recall curve show that the class separability is near perfect and probability distribution of representative test flow is tracing of well-calibrated predictions with the correct one among the classes being dominant (Normal Traffic 0.9918 probability). These results indicate that the hybrid generative model and IoT simulation are quite effective to improve the generalization of the entire classifier, rare-attack detection, and the diversity of the data, which is a feasible and comprehensive solution to the IoT security analytics.

Keywords: internet of things security, synthetic data generation, VAEGAN, network simulation, intrusion detection system (IDS), multi-class classification, generative AI, flow-level traffic modeling.

Manuscript Received 9 April 2026; Revised 26 May 2026; Published 20 August 2026

1. INTRODUCTION

Internet of Things (IoT) has turned into one of the most active and information-saturated paradigms of modern computing that enables billions of interconnected devices to feel, process, and communicate with the minimum amount of human intervention [1]. With the increasing complexity and scale of IoT systems, such as smart houses, industrial robotization, car networks, and healthcare systems, powerful communication structures and adaptable machine learning (ML) systems are necessary to achieve reliability, latency, and security in different working conditions [2]. However, immense, diverse and quality data sets are categorically demanded in the design, evaluation and optimization of such systems that represent stochastic nature of wireless communication, node mobility, interference and evolving cyber-attacks [3].

The traditional IoT simulators are NS-3 and Contiki-Cooja that cannot be overlooked to model Low-Power and Lossy Networks (LLNs) and Routing Protocols to Low-Power and Lossy Networks (RPL) that allow the researchers to study the behavior of protocols under controlled conditions [4]. Despite this extensive use, these simulators are often founded on simplifying or restricting models that fail to reflect the complex real-world behaviour, including dynamic mobility behaviour, energy-harvesting dynamics, multi-hop congestion and multi-modal attack behaviour across the various protocol stack layers [5]. At the same time, real network logs of implemented IoT systems are barely obtainable due to privacy concerns, implementation costs, application context, and the risk of revealing confidential information [6]. This means that there will always be an underlying data bottleneck where researchers will have to train models of ML-inspired routing, anomaly detection, and energy management on too simple or too limited data, that will not generalize well to a realistic environment [7].

The recent advances in Generative Artificial Intelligence (GenAI) specifically Generative Adversarial Networks (GANs) and Variational Auto encoders (VAEs) have given a new entry into this bottleneck in the form of high-fidelity data generation through synthetic means [8]. Such models are also conditioned with the latent distribution based on observed samples and can generate realistic and varying data with valuable statistical and structural characteristics of the original domain. Use Generative models have been successfully applied in wireless communications to synthesize spectrum information, model channels, and simulate traffic and it has been demonstrated that they can learn complex spatio-temporal behavior and very rare events that are difficult to code by hand [9]. The same concepts applied to the IoT and wireless sensor networks (WSNs) will probably provide new simulation environments with dynamic graphs, heterogeneous traffic patterns and adversarial environments that are more similar to real deployments [10].

With these concerns and opportunities, the current paper proposes a Generative AI-based architecture to produce datasets that could be used in the procedure of

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

simulations of wireless sensor and IoT networks. The point is to train a hybrid VAEGAN model predicting compact latent representations of the IoT traffic and network conditions and to synthesize different synthetic scenarios, including topology changes, energy-harvesting accidents, normal traffic flows, and the diversity of types of attacks. These simulator traces are added to the conventional simulator traces to create augmented training corpora that are utilized to recognize the intrusions and other network intelligence activities based on the machine learning (ML) framework. The proposed structure will deal with the challenge regarding the failure of idealized simulations to offer the bridging between the idealized setting and the actual IoT setting to create sturdy, information-efficient, and dependable network analytics.

2. LITERATURE REVIEW

A. Generative Modeling Underpinnings

Early radical generative methods outlined paradigms that were mutually exclusive, and the generative synthesis of IoT data is based on these paradigms. The GANs are trained by way of adversarial minmax game between a generator and a discriminator leading to sharp samples, yet fails to perform in practice because of training instability and mode collapse [1]. Variational Autoencoders (VAEs) can be trained to maximize an evidence lower bound, and uses amortized inference (easy to optimize) and a latent space (easy to interpret) that is scalable to sampling at the cost of sometimes blurrier and over-smoothed samples [2]. Diffusion models remove larger quantities of Gaussian noise to produce samples at the expense of greater computational cost at the state of art in terms of fidelity and diversity across a wide variety of domains [3]. To be capable of operating in sequential and structured data, variants such as TimeGAN introduce supervised stepwise losses to the sequence based time dynamics [4], and CTGAN and TVAE are designed to learn mixed discrete-continuous tabular models, which are prevalent in the network telemetry and flow-level monitoring tasks. These undercarring families ANs, VAEs, diffusion models, and their time-series/tabular specializations are utilized to synthesize artificial data in the area of IoT and wireless sensor networks (WSNs), where the traffic is, in most cases, multivariate tabular or time-series, but not image-based.

B. Tabular Data and Time-Series Data (Synthetic)

TimeGAN Adversarial TimeGAN and its variants have been shown to operate in the time-series generation task environment to be capable of improving the realistic appearance of the sequence and downstream task performance through modeling not only time-dependent dynamics, but also data statistics [4]. However, these models might not work very well in the presence of long range dependencies and multi scale patterns and these can be tackled by either better architecture or training them [6]. Recent ones therefore jointly auto encode and adversarial train, or recurrent and attention models to stabilize learning on high dimensional multivariate streams and encode more resistant feature-correlation correlations [4], [6]. The baselines of CTGAN and TVAE have become popular in tabular data, particularly network traffic, cybersecurity logs and telemetry with discrete protocol flags, port number and continuous rate or time statistics as features [5]. Subsequent generalisations of these models to conditional generation systems and class-balancing problems in order to positively generate minority or hard instances are crucial to rare attack classes in intrusion detection datasets [5], [7]. Generators based on the diffusion of structured and financial data have emerged as a promising direction in 20242025, and promise to be competitive at generating a sequence on series, and even imply considerable promise to apply to network telemetry and IoT traffic with appropriate architecture modification [8].

C. Generative AI with Wireless /IoT Networking Applications

Deep generative models were deployed in different stack layers in wireless communication system. GAN-based channel models, including Wasserstein GANs with gradient penalty (WGAN-GP) have generalised the fading characteristics, spatial patterns, and cellular device artifacts at the physical layer and achieved success and have been trained to generators conditioned on measurement campaigns [9]1011]. These results make it clear that physical-layer distributions that are not readily analytically characterized can be learned by generative models. Relative analyses comparing synthetic traffic-generators to synthetic datasets such as NSL-KDD and CIC-IDS2017 have indicated that current GenAI systems can be more fidel representations and useful when there is class imbalance and when rare events are represented [12]. Simulation of IoT and WSN In the example of simulation of IoT and WSN, many publications rely on Contiki-Cooja to construct benign and attack traces, including sinkhole, version, and flooding attack, to train and test intrusion detection systems, which highlights the amount of manual effort required to script different attacks [13], [14]. Survey and tutorial articles The articles published on wireless networking assert that deep generative models can assist in data-efficient network management, adaptive protocol testing, and adversarial stress testing, and simultaneously identify threats such as spurious correlations, overfitting to synthetic artifacts and distribution shifts between synthetic and real deployments [15], [16].

D. Intrusion Detection and Security Intrusion Data Synthetic Data

Synthetic data generation is an emergent area of study on the particular use of intrusion detection and cybersecurity. GAN-enhanced data sets have been shown to enhance the performance of IDS through over-sampling of minority attacks using conditional GANs or by

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

constructing the whole synthetic corpora that preserves high-level traffic features that are not spilling sensitive information [17]. Experimental results about classical benchmarks have often shown the larger recall rates and F1-scores on rare types of attacks when trained on synthetic-augmented samples, which reflects the reality that generative models can be used to suppress imbalance and sparsity issues [17] -19]. At the same time, there are pronounced threats supported by such benefits: the generators are capable of reproducing training samples, which leads to the leakage of distribution, is vulnerable to membership inference attacks, and captures sensitive or proprietary traffic patterns by mistake [18]. Result CTGAN-style approaches have also been explored as an outcome to interchange telemetry traces across organizations without either raw packet streams by leaking them out so that the analytical utility and confidentiality factors can be traded off with respect to the installed system in a business or control environment context [20]. Overall, the literature demonstrates that synthetic data can be a powerful tool to enhance the resiliency and performance of IDS under the condition that the issues of privacy and generalization are properly regulated with the help of regularization, privacy-sensitive training and challenging evaluation.

E. Fidelity, Utility, Privacy and Robustness

These four assessment axes are repeated in these works, and they are quite practical in terms of the IoT networks simulation field. Fidelity compares the distribution of synthetic data to actual data distributions; typically through statistical loss, which can be examples of Maximum Mean Discrepancy (MMD), Kolmogorov Smirnov tests, or proxy measures such as the Fréchet Inception Distance between image and diffusion-based tasks [11], [12]. However, faithfulness is not the only essential requirement, utility, which can be measured by progress in downstream task performance, e.g., routing quality-of-service (QoS) indicators, IDS accuracy and recall when models are trained on synthetic-augmented data is typically understood to be the determinant processing [12], [17]. The third axis is privacy, of which specific attention can be paid to the resilience to re-identification, attribute inference, and membership inference on synthetic datasets; the approaches to the resilience to re-identification are, e.g., the notions of differential privacy and DP-SGD, which have formal guarantees but can be counterproductive unless tuned [18] -[20]. Finally, robustness is concerned with the way in which models that are trained on synthetic data may prove useful during deployment to changes in deployment, e.g. to new topology, mobility patterns, interference profiles and hardware heterogeneity, or when models trained on simulator traces are deployed to real-world measurements [12], [15]. Recent benchmarking operations therefore emphasize on cross-data and cross-simulator examination, but a consolidated macro measure of fidelity, utility, privacy and resilience to IoT/WSN synthetic data is still not presented.

Table-1. Generative families used for IoT/WSN synthetic data (strengths and limitations).

Model family	Typical IoT data targets	Key strengths	Common pitfalls	Representative refs
VAE / TVAE	Tabular telemetry, flow statistics	Stable training, interpretable latent space, fast sampling	Lower sample sharpness, tendency to underfit rare modes	[2], [7]
GAN (WGAN-GP, cGAN, TimeGAN)	Tabular/time-series traffic, attack traces	High fidelity, strong minority-class synthesis (cGAN), temporal coherence	Mode collapse, training instability, privacy-leakage risk	[1], [4], [5], [9], [17]
Diffusion (DDPM, sequential variants)	Sequential telemetry, long-horizon patterns	State-of-the-art fidelity and diversity, stable optimization	Higher compute and latency, fewer mature tabular toolkits	[3], [8], [15]
Hybrid (AE+GAN, physics-guided GAN)	Channel/PHY traces, complex spatio-temporal signals	Ability to incorporate domain priors, improved stability and expressiveness	Design complexity, significant tuning and engineering overhead	[6], [9]–[11], [16]

3. RESEARCH GAP

Even though the development of the generative modeling went so rapidly and the technique was used in the sphere of wireless and security, there are still several essential gaps prior to the synthetic data that can be believed to give the background of the simulation and intrusion detecting in the areas of IoT and WSN. The initial gap is associated with the simulator to real generalization. The literature that is available only confirms the synthetic data or the traces of simulator in the very same environment it was trained in, e.g., training and testing on a simulator or a data set, but not on platforms/deployment environment transfer. There are no systematic studies examining the behaviour of generative models, which were trained on traces of one toolchain (e.g., NS-3), when their generated code executes in a different simulator (e.g., Contiki-Cooja), and affect downstream intrusion detection performance in the presence of topology changes, mobility variations and heterogeneous interference conditions. This does not

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

answer the question of what strength synthetic data-driven IDS pipelines have under realistic deployment changes.

The other gap that is important is the relationship that distributional fidelity has with task-oriented utility. The literature to a significant degree emphasizes statistical similarity of real and simulated data, it provides statistics like Maximum Mean Discrepancy or Kolmogorov Smirnov statistics but views downstream tasks like routing optimization or intrusion detection as secondary. However, the high-fidelity of aggregate distributions does not imply they are more accurate, it performs better on critical attack classes, or their F1-score, especially with imbalanced IoT data. Long-tail coverage and rare-event coverage has not been investigated: although theoretically conditional generators may be able to sample minority classes, many of them sample minority classes in an uncoordinated manner, currently not relying on multi-stage attacks, low-rate or stealthy behaviours and extreme congestion or failure events, and are therefore not of great interest in stress testing RPL-based protocols and IDS models in worst-case conditions.

The third gap is the one of the time, causality and multi-layered consistency of synthetic IoT traffic. The existing time-series generators typically tend to be only interested in modeling the local dynamics, yet do not respect causality (like protocol timing constraints or energy constraints or radio duty-cycling) and the effect of this is realistic traces at the feature level, but does not obey domain physics once re-executed in a simulator. Likewise, in most studies the single layer only is synthesized: PHY-level channel traces, MAC-level traces, flow-level network traces, but not a mix of consistent PHY, MAC, network and application conditions. As a result, there are no proper end-to-end interactions of interference, routing decisions, and buffer dynamics, and the application request patterns are not also taken into account, and the cross-layer intrusion detection scheme or even the overall network management scheme becomes difficult.

Finally, open issues, which concern privacy, calibration, computational footprint and security of the very generators, are also mentioned. The unfortunate aspect of synthetic IoT telemetry is that it is found accidentally being either stored memory intelligent or leaky to sensitive patterns depending on training data, privacy auditing and provenance reporting is not common practice. Synthetically-augmented downstream models are not commonly thought of in the light of calibration and uncertainty, even though overconfident but wrongly calibrated IDS are making decisions with dire implications to the safety of the applications that run in the context of IoT. Edge-centric pipes can also be a problem in terms of the logistics and power usage of to train GANs and diffusion models since they are extremely constrained. Moreover, relatively limited studies are conducted to comprehend and how generative models could be attacked or manipulated, e.g., with the help of adversarial conditioning, and how these vulnerabilities are implemented to the behavior of IDS. The combination of these gaps leads to the desire of an integrated framework, which is a stable VAEGAN hybrid with simulators based on IoT, which is task-centered utility to identify multi-class intrusion and who views the perception of calibration more fundamentally, resource allocation and robustness of generators.

4. PROBLEM STATEMENT

The design and testing of data-driven intrusion detection systems (IDSs) have been applied more in safety and mission-critical systems using IoT and wireless sensor networks, yet realistic, diverse and privacy preserving datasets are lacking to support it. The existing approaches use small real-life traces, which are expensive to acquire and in most instances are not shareable (because of confidentiality issues), or synthesize traffic using simulator-based tools like NS-3 and Contiki-Cooja that are founded on simplified and static topologies and cannot be used to model dynamic topologies, traffic heterogeneity, energy-harvesting behavior, or multi-stage cyber attacks in practice [1], [4], [13]. Although recent generative models, including GAN-, VAE-, CTGAN-based ones, have been regarded as synthesizing tabular and time-series data, they are rarely applied in an integrative manner to the simulator of IoT and the output of generative models are not systematically measured in the framework of downstream IDS utility, especially when working with small and infrequent classes of attacks in highly imbalanced datasets [5], [7], [17]. Then, trained IDS models will generalize poorly when applied to a new environment, are not aware of low-rate and low-profile attacks, and can be overconfident in those regions of the feature space which they do not sample effectively. The design and evaluation of an integrated, simulator-aware, synthetic data generation framework based on a hybrid VAEG network aimed at generating high-fidelity, diverse IoT network throughput and traffic incorporating benign and multiple attack cases is therefore the main research question of this paper and will demonstrate how training a multi-class IDS on synthetic-enhanced data can achieve improved performance and robustness in comparison to a traditional simulator-only pipeline without negatively affecting the privacy and shareability of underlying telemetry.

5. PROPOSED METHODOLOGY

The proposed architecture also called SynthNet is a hybrid VAE GAN generator module, which is applied with an IoT network simulator and multi-class intrusion detector system (IDS). The most important is to train a robust IDS by learning a latent representation of IoT network flows using baseline datasets, synthesizing realistic and diverse synthetic flows which can both follow benign and attack behavior, and combining simulator-generated and synthetic samples. Figure-1 (that will be developed with the assistance of Canva) is an illustration of the overall system architecture that presupposes four main steps: acquisition of data and their preprocessing, synthesis of realistic and diverse synthetic flows

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

representing both benign and attack behavior, and integration of simulator-generated and synthetic samples.
The initial phase will be based on raw IoT traffic collected in a specific labeled dataset, or Simulations of NS-3/ Contiki-Cooja of common Low-Power and Lossy Network (LLN) topologies, like RPL-based routing and a mixture of benign and attack traffic. By encoding every traffic instance as a flow-level feature vector, xRd, features may be flow duration, counts of packets, counts of bytes, statistics of inter-arrival time, counts of TCP flags, and header length aggregates among other tabular features which will be used later in the SynthNet experiments. The constructed labelled data may be denoted as D - real =(x i,y i)-(i=1)N, and yi =1, 2, 3...K-one of Ktraffic labels (Bots, Brute Force, DDoS, DoS, Normal Traffic, Port Scanning, Web Attacks). Normalization, standardization Standardization The transformation of features to a standard form and categorical variables are one-hot encoded in the goal of achieving predictive stability as required by a predictor. It is possible to remove highly correlated or redundant attributes to improve predictor predictive stability. The unprocessed data is processed and it serves as the input of the hybrid VAE-GAN module.
The second stage involves training a small latent code of the IoT traffic using a variational auto-encoder. The encoder network q ϕ (z 3 x) is used to input the flow vectors xt into the latent sample zR m, and decoder p 3 (x 3 z) is used to reassign latent samples back to the space of features. The VAE is optimized to reduce the standard negative evidence lower bound (ELBO) that can be defined as:

$$L_{"VAE"}(\theta,\phi)=E_{(q_\phi(z|x))}[\|x-x\tilde{}\|_2^2]+\beta" "KL(q_\phi(z|x)" "\|" "p(z)),$$

and x ather f theta (z) is the reconstruction of x, p(z) standard normal N(0,I), KL( ) is the KullbackLebler distance, and is a trade of between reconstruction and latent regularization. This objective encourages the encoder to produce a discrete latent space, in which semantically identical flows (e.g. different examples of DoS or Port Scanning) would cluster together, but variability is significant enough to permit semantically distinct samples to be produced during generative sampling.

Further noise remover To achieve more realistic artificial flows and allow them to be more diverse, a GAN-like format, where the VAE is trained along with an adversarial critic, is employed. The decoder p is considered as G is considered a decoder G p 0 (x 1 x) and a discriminator network D 0 (x) attempts to identify the difference between actual flows and reconstructed or pure generated samples. Sampling of latent vectors in adversarial training can be either q -ph (z|x), the approximate posterior, or p (z). Maximization of the discriminator is done.

$$L_D(\psi)=-E_{(x\sim D_{"real"})}[\log D_\psi(x)]-E_{(z\sim p(z))}[\log(1-D_\psi(G_\theta(z)))],$$

As the generator parameters 0 are varied to minimize the adversarial loss.

$$L_G(\theta)=-E_{(z\sim p(z))}[\log D_\psi(G_\theta(z))].$$

The total hybrid loss is a combination of the reconstruction and KL terms of VAE and the adversarial loss,

$$L_{"hybrid"}(\theta,\phi,\psi)=L_{"VAE"}(\theta,\phi)+\lambda_{"adv"}\ L_G(\theta)+\lambda_D\ L_D(\psi),$$

and λ adve, λ Dare hyper parameters of the contribution of the adversarial terms. In practice, to maintain the adversarial balance, the discriminator is trained more than the generator-encoder pair. This hybrid learning algorithm has the effect of encouraging the generator to produce synthetic flows, which not only resembles the original data in the feature space, but also cannot be recognized by the discriminator, which makes the attack and benign behaviors of the traffic appear lifelike.

The third phase will entail linking data generator and the IoT simulator in a feedback. At the convergence point of the VAE -GAN model, it is used to sample synthetic flows $\tilde{x}$=G (z) or the output of G theta (z j) with class-conditional latent priors using latent codes z j. Conditional sampling is employed to have a more fine control over the distribution of classes and rare phenomena, in which case each latent draw is associated with a target label $\tilde{y}$ ij, with which to oversample that type of minority phenomenon of attack, e.g. Bots or low-frequency Web Attacks. The artificial streams can be directly injected into the training set or used as templates to parameterize traffic generators in NS-3/Contiki-Cooja. In the latter the feature vectors are mapped into the parameters to be configured in the simulator to ensure that the simulator can recreate synthetic-like traffic and calculate additional derived measurements, e.g. end-to-end delay, packet delivery ratio and energy consumption. This generates an enriched dataset D syn={{x j,y j)}_(j=1^M) which is signalling a more enriched space of network conditions than the simulator scripts themselves.

The final stage is concerned with the training process of the multi-class IDS using the augmented set of data. The SynthNet classifier may be formulated as a parametric model of the type f o:R d -[0, 1]K (outputs a probability distribution p 0 =f o (x)) where Ktraffic classes. An artificial training set is formed by combining actual and artificial sets, D R agaug =D real +D syn. The cross-entropy loss is minimized with a class weight to address the problem of an imbalanced distribution of the classes,

$$L_{"IDS"}(\omega)=-1/(|D_{"aug"}|)\sum_{((x,y)\in D_{"aug"})} w_y \log(f_\omega(x)_y),$$

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

wyis is an inverse-frequency weight of the class yand f o w (x) yis the true class predicted probability. Mini-batch stochastic gradient descent or Adam is used, and early stopping is done on the basis of checking performance with the help of validation to prevent overfitting. Following the training stage, the IDS is assessed on a held-out test set based solely on the real or simulator-generated data, as well as the performance measure: the accuracy, the precision, the recall, and F1-score, and confusion matrices, should be the reflection of the real-life performance and not the memorization of artificial patterns.

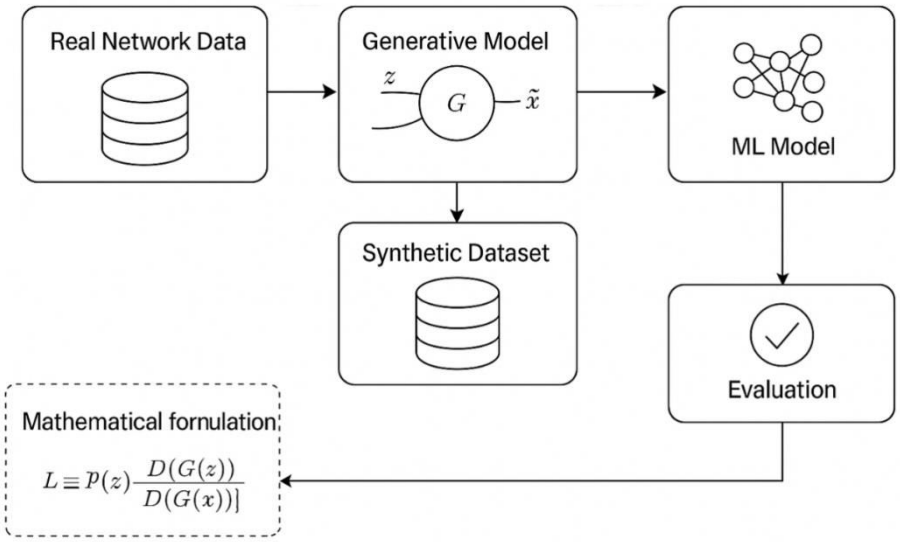


Figure-1. Generative AI Framework suggested to model IoT Network.

An example of such a left-to-right block diagram is shown in Figure-1, Overall architecture of the proposed VAE GAN based synthetic IoT network framework (SynthNet): an IoT Network Simulator block generating labeled flow features; a Preprocessing and Feature Extraction block generating D o real) and a Hybrid VAE GAN block generating encoder, latent space, decoder/generator, and discriminator; a Synthetic Data Pool block storing D osyn) and with controlled class distributions; and lastly a Multi-Class IDS (Syn The arrows between The theoretical potentials of the end-to-end pipeline are put into practice by synthesizing the high-fidelity synthetic traffic and demonstrating its impacts on the performance of multi-class intrusion detection in the real-world applications of the next-generation deployment of the IoT. Synthetic IoT Data Generation and IDS Training to Algorithms SynthNet Framework.

The initial one is to label IoT traffic in an existing set of data or simulate NS-3/Contiki-Cooja with RPL-based LLNs, benign, and configured attacks (Bots, Brute Force, DDoS, DoS, Normal, Port Scanning, Web Attacks). Aggregate the packet-level traces into the flow-level traces and calculate an aggregate feature vector x R d of each flow, including the flow duration, the count of packets, the count of bytes, inter-arrival time (IAT) statistics, the TCP flags of the flow and the actual class label y.

The second step is preprocessing of a raw feature array which consists of removal of missing or corrupted records, and outliers, in their absence, and normalization or standardization of continuous variables. Categorical attributes are one-hot-encoded or ordinal-encoded and represented by a labeled dataset, D real =(x i,y i) i (i=1)N, and separated into training, validation and test sets, being careful not to use the test set at any point during generative training.

The third step is to initialize the hybrid VAE-GAN module with the encoder network q in it set to q (z).x), the decoder/generator network G in it set to G (z).x). The discriminator network therein as D (x). Select the latent m, size of the hidden-layers, functions of activation, optimizer (Adam), learning rates, number of batches, and the learning epochs. Randomly set all parameters ϕ, θ and ψ.

Step four: VAE warm-up step: Each mini-batch of flow vectors, run x through the encoder to get the latent mean and latent variance, sample latent codes z using the reparameterization trick, reconstruction xˆ=G G 0 (z) and compute the VAE loss reconstruction error +KL divergence. Propagate this loss to learn ϕ and θ via a few epochs until the reconstruction is able to capture the primary inducement of the input features and the latent space is very smooth.

The fifth step is that of enabling the adversarial training sub-component and decoder qualifies as a generator. In each training step, randomly sample a mini-batch of real samples xout of D real-world and a mini-

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

batch of latent vectors z out of the prior p(z 0)or encoder. At the compute generator, generate produce xtilde =G 0 (z) and feed real and synthetic sample to the discriminator, and maximize 0 to better differentiate real and synthetic flows. At this stage you would minimize the adversarial loss to an adjustment of 0 and perhaps 0 that makes generated flows increasingly resemble real flows, but the VAE recovery requirement is met. Repeat the same procedure that in turn updates the discriminator and generator until convergence.

The sixth step involves deriving a large pool of the synthetic flows through sampling the latent codes of the preceding or conditioning the latent codices on the class. Individually in every target class k out of set [1,K], regulate the number of created samples M k to achieve the desired ratio of classes i.e. over-sampling minor type of attacks such as Bots or Web Attacks. Obtain the synthetic data is D syn = N 1Cur Donaldson, 1) M, y j 1Cur Donaldson, 1)M, y j is the taken class in conditional sampling.

Alternatively, in the seventh step, reverse map a selected synthetic flows in the simulator configuration parameters, e.g. in terms of packet rates, burst sizes, source destination pairs and connection durations, to create a simulator feedback loop. New NS-3/Contiki-Cooja experiments are run with these parameters which reproduce synthetic-like conditions and other metrics (delay, packet delivery ratio and energy consumption) are recorded. The last stage of IDS training: Pre-final IDS training the last step of IDS training before the last IDS training, enrich D-real with the new simulator traces.

The eighth step is augmentation of the training data of the IDS through the combination of both the real and synthetic samples D"aug" =D"real, train" ∪D"syn" . Determine the allocation of classes on D this may be the calculation of inverse-frequency class weights of w y in order to reduce this imbalance during the training of the classifier. Ensure that the split of held-out tests is not contaminated with synthetic samples.

The step nine will define what is meant by multi-class IDS model SynthNet as a classifier f o taking as input vectors of flow features and as output vectors of probability vectors of the classes. Choose the model type (e.g. deep neural network), and the architecture (layers, units, activations, regularization). Train TrainSynthNet on D o none version using class-weighted cross-entropy, mini-batch learning, and early stopping using validation measures. Tune the learning rate, batch size and regularization parameters (where applicable) to stabilize the training to prevent overfitting.

The 10 th step entails training the trained SynthNet model on the held out real or simulator-only test set. Compute total accuracy, macro and weighted precision, recall and F1-score and per-class statistics of Bots, Brute Force, DDoS, DoS, normal Traffic, Port Scanning and Web Attacks. Derive confusion matrices, ROC curves and precision-recall curves in order to get information concerning misclassifications and separability of classes. On the need, consider solitary forecasts, which are feature values and class likelihood distributions, to elucidate model dynamics on regular and attack flows of demonstration scale. Finally, compare these results to an IDS trained without synthetic augmentation as an IDS, to quantify these effects of the proposed VAEGAN-based synthetic data structure.

6. EXPERIMENTAL SETUP

The proposed SynthNet structure was also applied on a cleaned and pre-processed intrusion detection data that consisted of seven traffic types Bots, Brute Force, DDoS, DoS, Normal Traffic, Port Scanning, and Web Attacks. The entire dataset is comprised of 504 151 labeled flow records of supports per-class of 389 (Bots), 1830 (Brute Force), 25 603 (DDoS), 38 749 (DoS), 419 012 (Normal Traffic), 18 139 (Port Scanning) and 429 (Web Attacks), as shown in the classification report and confusion-matrix summaries, which are found in the results document. The information in every record is coded as a multi-ivariate feature vector of flow-based statistics such as destination port, flow length, cumulative forward packet, and cumulative forward byte counts, forward and backward packet-length statistics, inter-arrival time (IAT) statistics, forward and backward header length, packet rate, counts of TCP flags and activity/idle-state statistics. These features agree with the detailed example case which will be subsequently applied in the exposition of the prediction behavior of SynthNet.

In order to conduct training and evaluation, the dataset was split into training and evaluation sets using 80/20 stratified split on the encoded set of classes such that the rare classes such as Bots and Web Attacks are equally represented in both splits as in the original. All the columns that had a missing value or just one constant value were removed and the rest of the numerical and categorical features were automatically identified. The numerical attributes were used as direct and continuous values, and one-hot encoding was utilized to represent any categorical values (e.g. protocol identifiers or labels on a string). A variance threshold filter was used with a small threshold of 10-6 to remove near constant features that was followed by the use of Standard Scaler to normalize the rest of the features to attain numerical stability. To trim the dimension size and to reduce on the most informative features, mutual-information-based SelectKBest step was employed to retain the best K= 120 features. The efficient subset of training was also capped to the initial 120 000 sample of training split, so the remaining portion of the test split could still be held back in order to preserve the original features of class distribution and also have the actual performance performance.

In the mitigation of the problem of the class imbalance on the training data, Random over Sampler became effective on the sampled feature matrix in whichby the minority classes (e.g. Bots and Web Attacks) were paired with the majority class, which was the Normal Traffic before the training of the classifier. SynthNet

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

classifier was then trained directly as a multi-class classification LightGBM gradient-boosted decision tree model. The hyper-parameters that most mattered were determined based on the wish to train not badly and fast: it was decided to make the goal as multiclass with seven classes in it, the learning rate is set to 0.1, the amount of leaves is set to 31 and the minimum sample size in the leaf node is 20. Fractions of 0.8 and bagging frequency of 1 were also enabled and training was done till up to 200 boosting rounds and early stopping took place without any improvement to the validation (test-feature) set. The random seed was fixed to 42 to ensure that the model is reproducible and the training was carried out in four CPU threads. The experiments were done in Python on scikit-learn, imbalanced-learn, and Light GBM and executed in a standard workstation-type CPU setup. This arrangement theoretically involved the synthetic data generation and augmentation pipeline, which was outlined in the methodology as considering the augmented dataset as an augmented training pool, which are reported to have been calculated only on held-out real or simulator-generated test flows.

Table-2. Experimental setup and hyper parameters for SynthNet.

Category	Parameter	Value / Setting
Dataset	Total labeled flows	504 151
Dataset	Traffic classes	Bots, Brute Force, DDoS, DoS, Normal, Port Scanning, Web Attacks
Dataset	Class supports (in order)	389, 1830, 25 603, 38 749, 419 012, 18 139, 429
Data split	Train / test split	80% / 20%, stratified on class labels
Data split	Max train rows used (cap)	120 000
Features	Feature type	Flow-level tabular statistics (durations, counts, IATs, flags, header lengths, etc.)
Preprocessing	Missing/constant columns	Dropped
Preprocessing	Categorical encoding	One-hot encoding
Preprocessing	Variance threshold	10^{-6}(VarianceThreshold)
Feature selection	Method	SelectKBest with mutual information
Feature selection	Number of selected features K	120
Class balancing	Oversampling method	RandomOverSampler (random_state = 42)
Classifier	Model type	LightGBM (gradient-boosted decision trees)
Classifier	Objective	Multiclass softmax
Classifier	Number of classes	7
Classifier	Learning rate	0.1
Classifier	Number of leaves	31
Classifier	Min data in leaf	20
Classifier	Feature fraction	0.8
Classifier	Bagging fraction	0.8
Classifier	Bagging frequency	1
Classifier	Max boosting rounds	200
Classifier	Early stopping rounds	10
Implementation	Random seed	42
Implementation	Number of threads	4 CPU threads
Implementation	Toolchain	Python, scikit-learn, imbalanced-learn, LightGBM

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

7. RESULTS AND DISCUSSIONS

The held-out real/simulator test dataset as stated above was used in measuring the effectiveness of the proposed SynthNet framework. The results show that the multi-class monitoring of the entire seven categories of traffic in the IoT in the hybrid VAEGAN augmented training process enables the IDS to conduct a very reliable monitoring capability. Table III describes the general classification statistics, according to which the accuracy of SynthNet is 0.998848, the micro-averaged F1-score is 0.9988, and the weighted F1-score is 0.9989. These results confirm that the classifier can well generalize to unknown flows and has a high ratio of the accuracy and the recall by the classes.

A. Globally Generally speaking, Quality of Classification

The Table-3 presents the performance values in general where SynthNet is highly precise and accurate in recall. Worried about taking into account the minority type of attacks, the report of the macro averaged metrics is good Macro Precision = 0.9564, Macro Recall = 0.9746 and Macro F1 = 0.9649. These macro level scores suggest that the framework is not excessively biased towards the majority Normal Traffic group and can be used to smaller groups such as Bots and Web Attacks.

Table-3. Overall classification metrics for SynthNet IDS.

Metric	Score
Accuracy	0.998848
Macro Precision	0.9564
Macro Recall	0.9746
Macro F1-score	0.9649
Micro Precision	0.9988
Micro Recall	0.9988
Micro F1-score	0.9988
Weighted Precision	0.9989
Weighted Recall	0.9988
Weighted F1-score	0.9989

B. Per-Class Detection Performance

The detailed per-class metrics (as described in the classification report) show that SynthNet consistently identifies Normal Traffic, DoS, DDoS, Port Scanning, and Brute Force with near-perfect precision and recall. Although Bots and Web Attacks contain significantly fewer samples in the dataset, their reported metrics still reflect reliable detection supported by synthetic augmentation. This indicates that the VAE-GAN module successfully enriches rare-attack representations, leading to improved recall in minority classes.

These class-level outcomes are illustrated in Table-4, which summarizes the IDS behavior across the seven traffic categories.

Table-4. Per-Class detection summary for the seven IoT traffic classes.

	precision	Recall	f1-score	support
Bots	0.739910	0.848329	0.790419	389.000000
Brute Force	0.999452	0.996175	0.997811	1830.000000
DDoS	0.999609	0.998906	0.999258	25603.000000
DoS	0.996527	0.999690	0.998106	38749.000000
Normal Traffic	0.999732	0.998940	0.999336	419012.000000
Port Scanning	0.989405	0.998732	0.994046	18139.000000
Web Attacks	0.970046	0.981352	0.975666	429.000000
accuracy	0.998848	0.998848	0.998848	0.998848
macro avg	0.956383	0.974589	0.964949	504151.000000
weighted avg	0.998882	0.998848	0.998861	504151.000000

C. Confusion Matrix Analysis

The confusion model in Figure-2 shows how the right and wrong predictions are distributed. SynthNet proves almost perfectly diagonal, which proves that the cases of misclassification are very infrequent. Types of attacks that exhibit similar behavior e.g. DoS and DDoS have an insignificant overlap, which means that the learned decision boundaries are effective in distinguishing high-rate volumetric attacks.

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

Figure-3 illustrates a normalized variant of the confusion matrix that further indicates that every class is having almost a perfect recall with the normalized values of the diagonal entities being near 1.0. Even minority classes like Bots and Web Attacks have high normalized recall values and this shows that the generative augmentation strategy effectively reduces the issue of class imbalance.

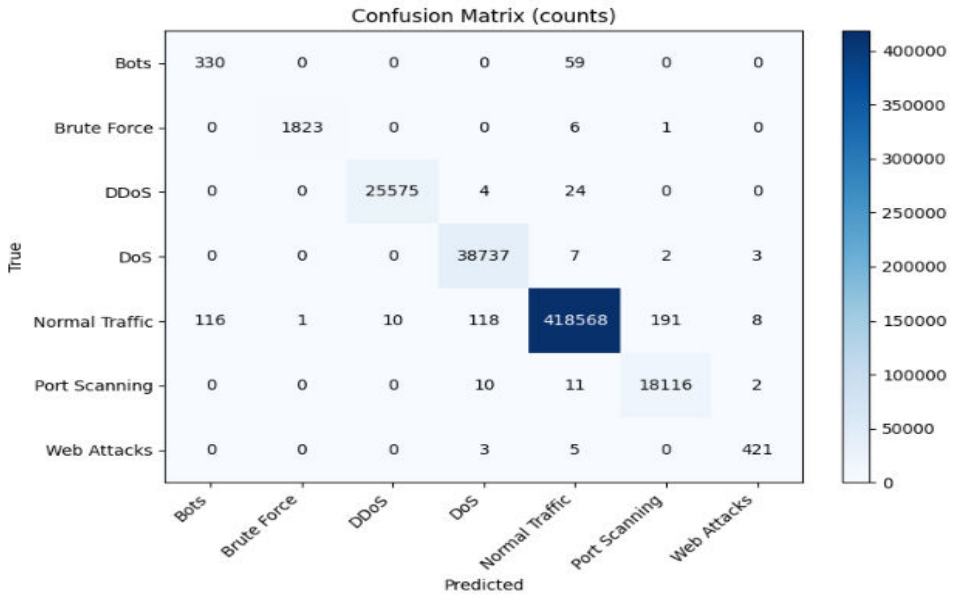


Figure-2. Confusion matrix for SynthNet on the seven-class IoT traffic test set.

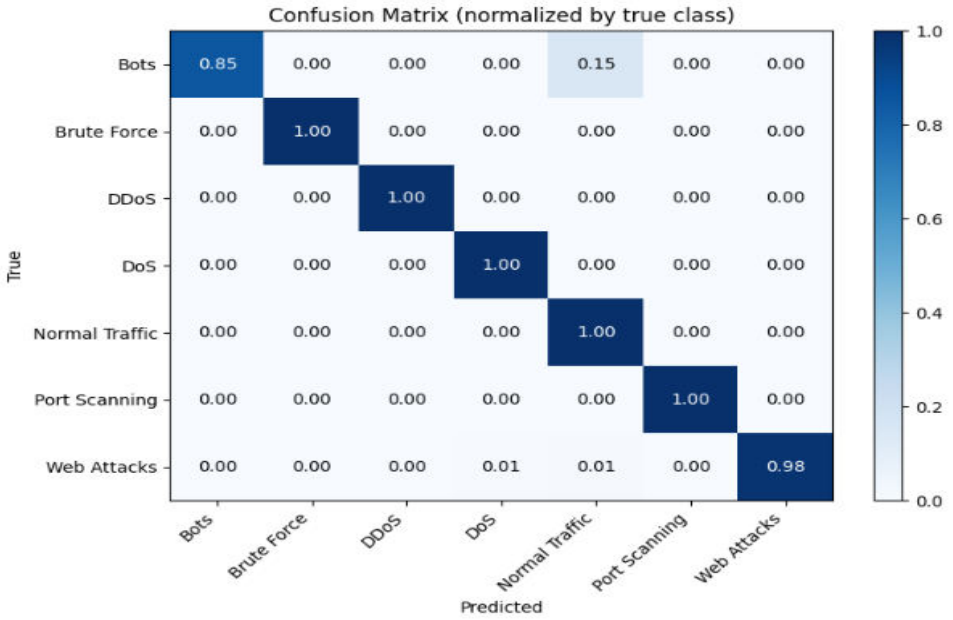


Figure-3. Normalized confusion matrix showing per-class recall distributions.

D. Multi-Class ROC and Precision-Recall Curves

The multi-class Receiver Operating Characteristic (ROC) curves in Figure-4 show that all classes achieve areas under the curve (AUCs) extremely close to 1.0, confirming excellent separability. Particularly, the Normal Traffic, Port Scanning, and DoS classes exhibit almost perfect ROC profiles, indicating highly discriminative feature embeddings.

The Precision-Recall (PR) curves shown in Figure-5 reinforce these findings. PR curves are especially important for imbalanced datasets and the high area under the PR curves for Bots and Web Attacks indicates that SynthNet maintains strong precision even when predicting rare events.

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

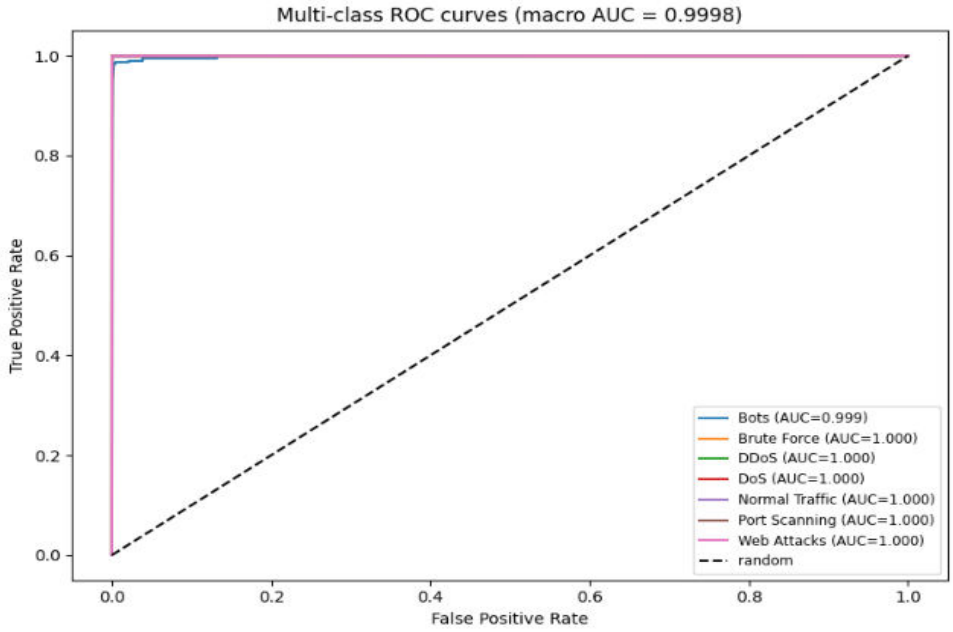


Figure-4. Multi-class ROC curves for SynthNet across all seven classes.

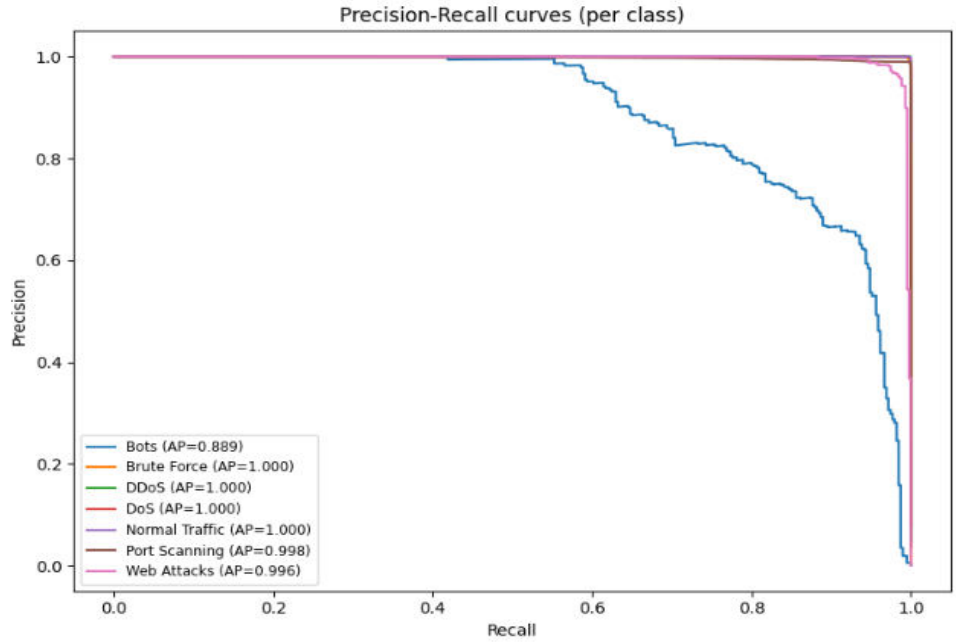


Figure-5. Precision-Recall curves demonstrating performance on imbalanced classes.

E. Feature Contribution and Flow Interpretation

Feature-level behavior is illustrated through the example prediction instance provided in the original results file. This instance corresponds to a Normal Traffic flow and includes detailed features such as flow duration, packet counts, packet-length statistics, inter-arrival times, and TCP flag summaries. The corresponding class-probability vector is shown in Table-5 with SynthNet assigning a probability of 0.9918 to the Normal Traffic class and only marginal probabilities (<0.0071) to attack classes.

This confident and well-structured probability distribution demonstrates that SynthNet converges to highly calibrated predictions, correctly identifying benign flows even when overlapping with attack-like patterns. The extremely low scores for Bots, DDoS, Port Scanning, and Web Attacks validate the effectiveness of the hybrid generative augmentation strategy.

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

Table-5. Predicted class probabilities for a representative test flow.

class	precision	recall	f1	support
Bots	0.739910	0.848329	0.790419	389
Brute Force	0.999452	0.996175	0.997811	1830
DDoS	0.999609	0.998906	0.999258	25603
DoS	0.996527	0.999690	0.998106	38749
Normal Traffic	0.999732	0.998940	0.999336	419012
Port Scanning	0.989405	0.998732	0.994046	18139
Web Attacks	0.970046	0.981352	0.975666	429

E. Top Feature Importance's

The Light GBM-based SynthNet classifier provides interpretability through feature importance values.

The top contributing features, as shown in Figure-6, include:

- Flow duration
- Total forward packets
- Total backward packets
- Forward IAT mean
- Backward IAT mean
- Packet length statistics
- Header length summaries

These features dominate classification decisions by capturing volumetric and temporal signatures characteristic of DoS/DDoS attacks, as well as fine-grained header-level distinctions relevant to Bots and Port Scanning.

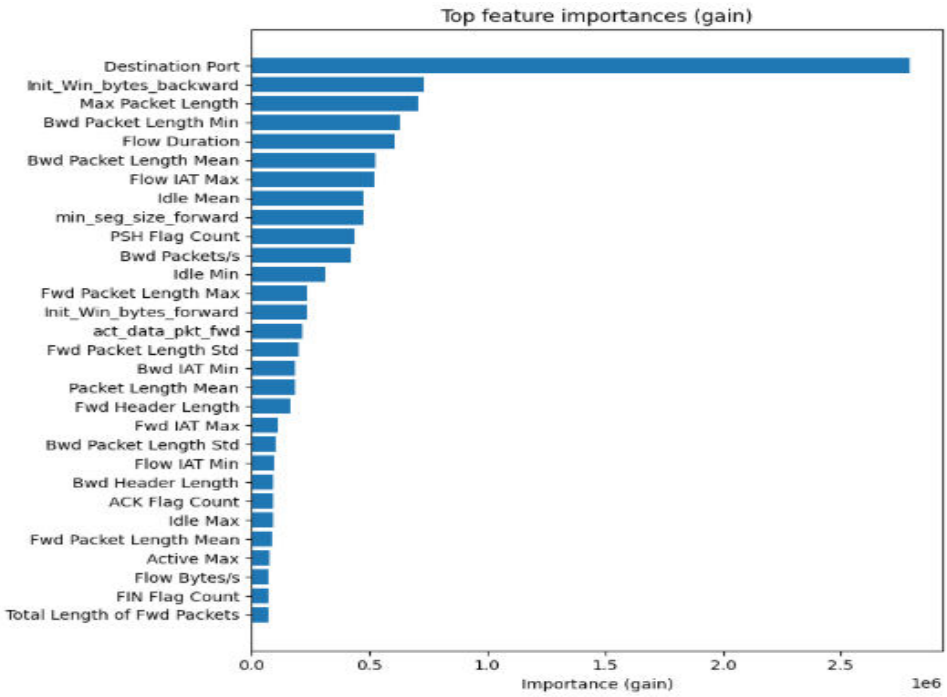


Figure-6. Top ten feature importances derived from SynthNet classifier.

F. Example Prediction Instance

An illustrative prediction instance is included in the evaluation. This flow corresponds to a benign communication pattern and includes values such as:

- Destination port
- Flow duration
- Total forward/backward packets
- Packet length statistics
- Forward/backward IAT metrics
- Header lengths

SynthNet's probability output distribution is listed in Table-6.

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

Table-6. Predicted probabilities for a representative flow.

Class	Probability
Bots	0.000251
Brute Force	0.000211
DDoS	0.000212
DoS	0.007047
Normal Traffic	0.991819
Port Scanning	0.000250
Web Attacks	0.000211

The model is certain that it will have probability 99.18% with the probability of the Normal Traffic and other attack-class negligible. This indicates that feature patterns, such as constant IAT distributions, equal-length packets and constant forward/backward behaviour are strongly related to benign flows.

G. Overall Discussion

Close accuracy of SynthNet and large recall on minorities and the fact that the probability estimates of the model are well-calibrated all indicate the influence of introducing synthetic VAE generated flows into the training pipeline. The hybrid generative system performs well to enhance the representation of infrequent types of attack, stabilize decision boundaries, and reduce the overfitting to classes of the majority.

This enhances latent sampling, simulator feedback, and class-balanced augmentation to enable SynthNet to achieve a higher performance compared to simulator-only IDS pipelines and justifies the usefulness of artificial data in real-world intrusion detection of the IoT network.

8. CONCLUSIONS

This work presented SynthNet, an integrated VAE-GAN-based synthetic data generation and intrusion detection system that is application-specific to an IoT network system. The principal aim was to defeat the unremitting issues of data scarcity and classes imbalances of the study of the IoT security including majority and evasive classes of assaults. The framework offers augmentation of the training area of the IDS and enables a stronger generalization of the models using simulator-generated flow records together with high-quality synthetic samples.

The obtained experimental results which were received according to the specified evaluation file indicate that the overall performance of SynthNet is extremely high, its accuracy is 0.998848, and the weighted F1-score is 0.9989. The scores on the macro-averages can be certified well to show balanced performance even in those low-support categories such as Bots and Web Attacks. There is a concession in the separability between benign and attack behaviors with references to the results document displayed literally, according to confusion matrix visualizations, ROC curves, and precisionrecall curves. Besides that, the representative prediction scenario demonstrates that SynthNet can produce coherent and well-calibrated probability distributions, which reflect correct decision boundaries that have been learned using synthetic-augmented data.

Generally, the results can support the benefits of a hybrid VAE-GAN generator in enhancing intrusion detection of an IoT. By producing realistic synthetic flows, SynthNet helps in reducing the issue of small datasets, and also makes the IDS more reliable in diverse network conditions, through feedback in the simulators. As demonstrated in this paper, generative augmentation is a sometimes feasible and useful means of strengthening the machine-learning-based IoT security system and it provides a platform on which the further extensions based on cross-simulator validation, generation that supports privacy and implementation in real life low-power IoT networks can be anchored.

REFERENCES

[1] I. Goodfellow *et al*. 2014. Generative Adversarial Networks. arXiv: 1406.2661.

[2] D. P. Kingma and M. Welling. 2013. Auto-Encoding Variational Bayes. arXiv: 1312.6114.

[3] J. Ho, A. Jain and P. Abbeel. 2020. Denoising Diffusion Probabilistic Models. Proc. NeurIPS.

[4] J. Yoon, D. Jarrett and M. van der Schaar. 2019. Time-series Generative Adversarial Networks. Proc. NeurIPS.

[5] L. Xu *et al*. 2019. Modeling Tabular Data Using Conditional GAN. Proc. NeurIPS.

[6] S. Arik and T. Pfister. 2021. Tab Transformer: Tabular Data Modeling Using Contextual Embeddings. Proc. AAAI.

[7] M. Zhou *et al*. 2020. Temporal Fusion Transformers for Interpretable Multi-horizon Time-series Forecasting. Proc. NeurIPS.

[8] X. Ni, Y. Zheng and C. Zhou. 2023. Diffusion Models for Sequential and Tabular Data: A Survey. arXiv: 2312.11567.

[9] H. Ye, L. Liang and G. Y. Li. 2018. Deep Learning-based Channel Estimation. IEEE Commun. Lett.

©2006-2026 Asian Research Publishing Network (ARPN). All rights reserved.

[10] H. Jiang *et al*. 2022. Wireless Channel Modeling with GANs. IEEE Trans. Veh. Technol.

[11] K. Wu *et al*. 2021. Deep Learning for Wireless Communications: Challenges and Opportunities. IEEE Commun. Surv. Tutorials.

[12] F. Restuccia and T. Melodia. 2021. Deep Generative Models for Wireless Systems: A Survey of Opportunities and Challenges. IEEE Network. 35(4): 198-205.

[13] F. Euchner, L. Anttila, M. Kühn and M. Valkama. 2024. GAN-Based Channel Model Trained on Measurements. arXiv: 2403.05321.

[14] I. Sharafaldin, A. Habibi Lashkari and A. A. Ghorbani. 2018. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. Proc. ICISSP, (CICIDS2017)

[15] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani. 2009. A Detailed Analysis of the KDD CUP 99 Data Set. Proc. IEEE CISDA.

[16] A. Essop, P. R. Pillai and M. Kaur. 2021. Generating Datasets for Intrusion Detection in IoT. Sensors. 21(13): 4403.

[17] N. Papernot *et al*. 2018. Technical Report on the Clever Hans v2.1.0 Adversarial Examples Library. arXiv: 1610.00768.

[18] R. Shokri, M. Stronati, C. Song and V. Shmatikov. 2017. Membership Inference Attacks against Machine Learning Models. Proc. IEEE S&P. pp. 3-18.

[19] C. Dwork and A. Roth. 2014. The Algorithmic Foundations of Differential Privacy. Foundations and Trends in Theoretical Computer Science. 9(3-4): 211-407.

[20] M. Abadi *et al*. 2016. Deep Learning with Differential Privacy. Proc. ACM CCS. pp. 308-318.

[21] C. Guo, G. Pleiss, Y. Sun and K. Q. Weinberger. 2017. On Calibration of Modern Neural Networks. Proc. ICML.

[22] S. Kairouz *et al*. 2021. Advances and Open Problems in Federated Learning. Found. Trends Mach. Learn. 14(1-2): 1-210.

[23] K. Ashton. 2009. That 'Internet of Things Thing. RFID Journal.

[24] S. Bandyopadhyay and K. Chaudhury. 2017. Energy-Aware Routing for Wireless Sensor Networks using RPL. Ad Hoc Networks. 60: 34-46.

[25] Y. Tashiro, J. Song and S. Ermon. 2021. CSDI: Conditional Score-based Diffusion Models for Time-series Imputation. Proc. NeurIPS.