Intrusion Detection System Based on Quantum Generative Adversarial Network

Franco Cirillo[a] and Christian Esposito[b]
University of Salerno, Via Giovanni Paolo II 132, Fisciano, Italy
{fracirillo, esposito}@unisa.it

Keywords: Quantum Machine Learning (QML), Quantum Generative Adversarial Network (QGAN), Intrusion Detection System (IDS), Anomaly Detection.

Abstract: Intrusion Detection Systems (IDS) are crucial for ensuring network security in increasingly complex digital environments. Among IDS techniques, anomaly detection is effective in identifying unknown threats. However, classical machine learning methods face significant limitations, such as struggles with high-dimensional data and performance constraints in handling imbalanced datasets. Generative Adversarial Networks (GANs) offer a promising alternative by enhancing data generation and feature extraction, but their classical implementations are computationally intensive and limited in exploring complex data distributions. Quantum GANs (QGANs) overcome these challenges by leveraging quantum computing's advantages. By utilizing a hybrid QGAN architecture with a quantum generator and a classical discriminator, the model effectively learns the distribution of real data, enabling it to generate samples that closely resemble genuine data patterns. This capability enhances its performance in anomaly detection. The proposed QGAN use a variational quantum circuit (VQC) for the generator and a neural network for the discriminator. Evaluated on NSL-KDD dataset, the QGAN attains an accuracy of 0.937 and an F1-score of 0.9384, providing a robust, scalable solution for next-generation IDS.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

Intrusion Detection Systems (IDS) play a critical role in ensuring the security and integrity of digital infrastructures by identifying malicious activities, unauthorized access, or policy violations (Abdulganiyu et al., 2023). In networked environments, the Network Intrusion Detection System (NIDS) is particularly vital. Among the various detection techniques, anomaly detection has proven to be an effective approach for intrusion detection (Rafique et al., 2024). Unlike signature-based systems that rely on predefined attack patterns, anomaly detection methods establish a baseline of normal system or network behavior. Any deviation from this baseline is flagged as a potential intrusion.

In recent years, machine learning (ML) has emerged as a powerful tool for both intrusion and anomaly detection. ML techniques excel at learning complex patterns, enabling systems to automatically classify network traffic as normal or malicious. However, ML's reliance on quality datasets and extensive computational resources can limit its scalability and applicability in dynamic environments (Muneer et al., 2024).

One of the most promising advancements in machine learning for intrusion detection is the application of Generative Adversarial Networks (GANs) (Gui et al., 2021). GANs are a type of neural network architecture consisting of two components: a generator, which creates synthetic data, and a discriminator, which distinguishes between real and generated data. The adversarial training between these two components allows GANs to stand out in generating realistic data samples. In the context of intrusion detection, GANs can generate synthetic network traffic to augment training datasets, address class imbalance issues, and improve the detection of subtle or rare anomalies. Additionally, GANs have been used directly as anomaly detection mechanisms by leveraging their ability to model the underlying distribution of normal data. Despite these benefits, ML techniques, including GANs, have inherent limitations. Classical ML models are constrained by the limitations of classical computing, which may struggle with

[a] https://orcid.org/0009-0006-9599-5996
[b] https://orcid.org/0000-0002-0085-0748

Cirillo, F. and Esposito, C.
Intrusion Detection System Based on Quantum Generative Adversarial Network.
DOI: 10.5220/0013397800003890
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 1*, pages 830-838
ISBN: 978-989-758-737-5; ISSN: 2184-433X
Proceedings Copyright © 2025 by SCITEPRESS – Science and Technology Publications, Lda.

the increasing complexity of data in high-dimensional spaces (Yamasaki et al., 2023). To address these challenges, researchers have turned to Quantum Machine Learning (QML) as a promising alternative (Cerezo et al., 2022). By harnessing the unique capabilities of quantum computers, QML has the potential to overcome the scalability and performance limitations of classical ML techniques. Within the domain of QML, Quantum Generative Adversarial Networks (QGANs) have emerged as a particularly innovative approach (Ngo et al., 2023). QGANs extend the principles of classical GANs into the quantum domain, incorporating quantum generators and/or discriminators. These networks have shown promise in generating high-quality synthetic data and in improving the efficiency of anomaly detection systems, as they can achieve comparable or superior performance to classical GANs with fewer trainable parameters (Herr et al., 2021). Furthermore, their ability to operate effectively with limited data makes them highly suitable for cybersecurity applications, where datasets are often sparse or imbalanced.

This work makes the following key contributions:

- Introduction of a novel QGAN-based model specifically designed for intrusion detection, utilizing a quantum generator and a classical discriminator.
- Development of a mapping function to transform the generator's quantum outputs into meaningful samples that align with the features of an intrusion detection dataset.
- Extensive experimentation across various configurations, archiving an accuracy of 0.937 and F1 score of 0.9384, offering a significant advancement in the state-of-the-art for QGAN applications in intrusion detection using the NSL-KDD dataset.

The paper is structured as follows: Section 2 introduces QGANs and the dataset used. Section 3 reviews the relevant related works. Section 4 details the data preprocessing steps and the proposed QGAN model. Section 5 analyzes the evaluation results. Lastly, Section 6 summarizes the contributions and suggests future research directions.

2 BACKGROUND

2.1 Quantum Generative Adversarial Networks (QGANs)

QGANs are an extension of classical GANs, adapted to leverage the principles of quantum computing (Zoufal et al., 2019). They can be implemented in two primary configurations: full quantum (Kalfon et al., 2024) and hybrid. In a full quantum QGAN, both the generator and discriminator are quantum systems, leveraging parameterized quantum circuits for data generation and evaluation. In contrast, hybrid QGANs combine quantum and classical elements (Ngo et al., 2023). A typical hybrid QGAN consists of two components:

- Quantum Generator (G): A parameterized quantum circuit (PQC) that generates quantum states representing data samples.
- Classical Discriminator (D): A classical neural network that evaluates the similarity between generated samples and real data.

The discriminator's objective is to maximize the correct classification of real samples as real and generated samples as false. Using binary cross-entropy (BCE), the loss function for the discriminator, $\mathcal{L}_D$, is defined as:

$$\begin{aligned}\mathcal{L}_D = -\mathbb{E}_{x\sim p_{\text{real}}(x)}[\log D(x)]+\\ -\mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]\end{aligned} \tag{1}$$

Where:

- $D(x)$: Probability output by the discriminator that x is real.
- $D(G(z))$: Probability output by the discriminator that the generated sample $G(z)$ is real.

This loss comprises two terms: the BCE loss for real data: $-\log D(x)$, where the discriminator is penalized when it fails to classify real data as real, and the BCE loss for generated data $-\log(1-D(G(z)))$, where the discriminator is penalized when it classifies generated data as real.

The generator's objective is to "fool" the discriminator, such that it cannot distinguish between real and generated samples. This is achieved by maximizing $D(G(z))$, which is equivalent to minimizing $-\log(D(G(z)))$. The generator loss, $\mathcal{L}_G$, is therefore defined as:

$$\mathcal{L}_G = -\mathbb{E}_{z\sim p_z(z)}[\log D(G(z))] \tag{2}$$

In a QGAN, the generator G is implemented using a PQC. This involves encoding the latent variables z, which are typically sampled from a simple prior distribution, such as a uniform or Gaussian distribution, into quantum states $|\psi(z)\rangle$, applying a series of quantum gates parameterized by θ, and measuring the quantum circuit to generate output samples.These latent variables act as a compressed representation or "seed" for generating data, and they capture the stochastic nature of the data generation process. The discriminator remains classical, mapping the input data to a probability score using standard neural network layers.

2.2 Dataset

NSL-KDD is the intrusion detection dataset used in this work, which consists of a carefully selected subset of records from the original KDD dataset. NSL-KDD resolves many of the limitations present in KDD-99, such as the presence of duplicate records, while still maintaining a comprehensive representation of various attack types. NSL-KDD was introduced by Tavallaee et al. (Tavallaee et al., 2009) as a means to overcome the shortcomings of KDD-99, particularly to enhance the reliability of network intrusion detection system evaluations. The dataset also features a more balanced record selection, where the number of records from each difficulty level is inversely proportional to their representation in the original KDD dataset. Additionally, NSL-KDD is smaller in size compared to KDD-99, consisting of fewer but unique records, which reduces computational costs and enhances efficiency for training machine learning models. The NSL-KDD dataset is split into two primary subsets: Training and Testing datasets. The training set contains 125,972 records, while the test set contains 22,542 records. Each dataset is categorized by different attack types like DoS (Denial of Service), Probe, R2L (Remote to Local), U2R (User to Root).

3 RELATED WORK

Recent advancements in machine learning, particularly GANs, have opened new avenues for enhancing NIDS. GANs are widely used in unsupervised learning tasks and are capable of generating synthetic data that closely resembles the original dataset. Their integration into intrusion detection has shown promise in improving feature extraction, augmenting datasets, and addressing limitations in both signature-based and anomaly-based detection systems.

One of the primary applications of GANs in NIDS is the generation of synthetic data to enhance training processes. The authors of (Shahriar et al., 2020) introduced an ANN-based GAN to generate synthetic samples for training an IDS on the NSL-KDD dataset. Their results demonstrated that an IDS incorporating GAN-generated data significantly outperformed standalone systems in attack detection. Similarly, in the work (Lee and Park, 2021), GANs are employed to mitigate the class imbalance problem in NIDS datasets, finding GANs to be more effective than traditional oversampling techniques.

GANs have also been directly applied as detection mechanisms. The authors of (Patil et al., 2022) proposed a bidirectional GAN-based framework for anomaly detection, evaluated using the KDDCUP-99 dataset (Tavallaee et al., 2009), and demonstrated its superior performance compared to other deep learning models. Truong et al. (Truong-Huu et al., 2020) advanced this approach by utilizing two GAN models with custom neural network architectures for generators and discriminators. Their experiments on CIC-IDS 2017 and UNSW-NB15 datasets (Moustafa and Slay, 2015) showcased the efficacy of GAN-based systems against traditional unsupervised methods.

Building upon GAN advancements, Quantum GANs (Lloyd and Weedbrook, 2018; Dallaire-Demers and Killoran, 2018) have emerged as a promising extension, particularly for image generation. Recent works illustrate their potential to outperform classical counterparts even with fewer trainable parameters. The authors of (De Falco et al., 2024) applied QGANs to latent diffusion models, using classical diffusion processes for noise generation and quantum denoising with three VQC. Their results showed that the number of qubits, set to 10 in their study, significantly influenced performance. Similarly, in (Chang et al., 2024) it is introduced a QGAN architecture where the generator is quantum, and the discriminator operates classically in latent space, achieving superior image quality with fewer training epochs or smaller datasets. Finally, the work (Vieloszynski et al., 2024) proposed a hybrid model integrating an autoencoder to map images into a lower-dimensional latent space, allowing the quantum generator to efficiently process MNIST data with a lightweight architecture of seven layers and 140 parameters. Despite the promising advances of QGANs in image generation, their application to anomaly detection remains relatively underexplored, with only a few studies addressing this intersection.

In (Bermot et al., 2023) it is demonstrated the use of QGANs for detecting specific physical particles. They tested their QGAN on up to eight features, achieving 0.88 accuracy on noiseless simulators for seven features, outperforming classical GANs. They also validated its feasibility on actual quantum hardware for three features. However, the results were dataset-specific, limiting applicability across domains. The authors of (Kalfon et al., 2024) introduced a novel high-dimensional encoding approach called Successive Data Injection (SuDaI), which expands encoded data size without increasing qubits. They utilized a state-fidelity QGAN with both generator and discriminator implemented as quantum circuits. The design incorporated the SWAP test to compare generated and real data representations. Using the Numenta Anomaly Benchmark (NAB) database, they analyzed temporal anomaly detection but did not report specific performance metrics, leaving its practical effectiveness unclear. The work (Rahman et al., 2023) proposed a QGAN-based IDS but highlighted challenges due to the quantum generator producing discrete output values. For each feature represented by n qubits, the results were limited to 2^n discrete values, potentially reducing fidelity in real-world applications. Their work lacked specific implementation details, reproducibility, extensive testing, and performance metrics such as accuracy. In this work (Herr et al., 2021) the authors presented a hybrid quantum–classical anomaly detection model using Variational Quantum-Classical Wasserstein GANs (WGANs) with gradient penalty. Their method extended the AnoGAN framework (Schlegl et al., 2019) for anomaly detection, integrating recent GAN advancements with hybrid quantum–classical neural networks trained via variational algorithms. Testing on a credit card fraud dataset, they evaluated performance with an anomaly score combining generator and discriminator losses, achieving an F1 score of 0.85. While their results were promising, the dataset's simplicity limited the validity of the results in more complex and diverse real-world scenarios.

This work is motivated by the need to thoroughly evaluate QGANs in the intrusion detection domain, focusing on their scalability, robustness, and performance. Existing studies have been limited to simplified datasets and narrow feature sets, leaving their applicability to real-world intrusion detection systems underexplored. Additionally, challenges such as improving the fidelity of quantum generator outputs and achieving high accuracy and reliability in this domain remain unresolved. This study aims to optimize QGANs for intrusion detection by assessing their performance on high-dimensional datasets and providing a comprehensive analysis of their effectiveness through key metrics such as accuracy and F1-score.

4 METHODOLOGY

4.1 Data Preprocessing

The NSL-KDD dataset, a widely used benchmark in intrusion detection, is loaded as a structured dataset. The data comprises a mix of numerical and categorical features, as well as target labels indicating whether a given instance corresponds to a normal or anomalous network activity.

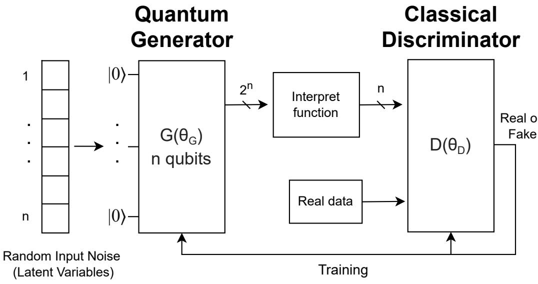

Figure 1: Proposed QGAN training process.

To standardize numerical features, scaling techniques are applied to ensure uniformity and to mitigate the influence of outliers. Robust scaling, which is resilient to the effects of extreme values, is used to transform numerical columns. This scaling process centers and scales the data based on interquartile ranges, preserving the distribution's core properties. Categorical features, such as protocol types, services, and flags, are converted into numerical representations using one-hot encoding. Therefore, the QGAN framework requires input data that are normalized and compact. To achieve this, feature values are scaled to a [0, 1] range using Min-Max scaling.

To further optimize the dataset for QGAN processing, PCA is applied. PCA reduces the feature space to a lower-dimensional latent representation, retaining only the components that capture the majority of the data's variance. Different numbers of principal components have been tested to assess their impact on model performance, and the results of these evaluations will be presented in the next section. The explained variance ratio is analyzed to confirm the adequacy of the selected components.

4.2 QGAN Configuration

The proposed QGAN model is an adaptation for the intrusion detection problem, specifically a dedicated interpret function and a selection for quantum circuits has been done. The structure is depicted in Figure 1 and consists of two primary components: a quantum generator implemented as a Variational Quantum Circuit and a classical discriminator implemented as a neural network. This hybrid design leverages quantum mechanics' inherent probabilistic nature for data generation and the computational power of the classical neural network for discrimination tasks.

The process begins with a random noise vector that serves as input to the generator. This noise introduces variability to generate diverse samples. The generator is a quantum circuit acting on n-qubits. Starting from the initial state $|0\rangle$, the circuit applies parameterized quantum gates defined by θ_G weights.

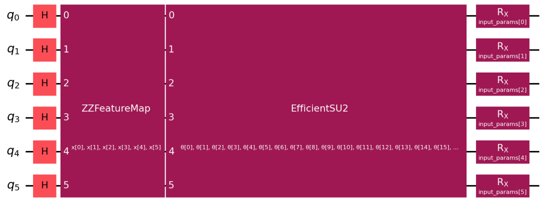

Figure 2: Quantum circuit for the generator.

This generates a quantum state:

$$G(\theta_G)|0\rangle^{\otimes n},$$

which represents the generated false data, encoded in a Hilbert space of dimension 2^n.

The quantum state produced by the generator cannot be directly evaluated classically. Therefore, an interpret function is applied to map the quantum state to classical data.

The discriminator is a classical neural network that receives input data, either from the generator or real data. It evaluates the data and outputs a probability score indicating whether the input is benign or an attack.

The generator and discriminator are trained adversarially. While the discriminator $D(\theta_D)$ learns to distinguish between real data (from the training set) and false data (from the generator), the generator $G(\theta_G)$ learns to fool the discriminator by improving the quality of its generated data, minimizing the difference between real and false samples. This iterative process enables the generator to produce increasingly realistic data. The QGAN training process is described in Algorithm 1.

The generator is constructed as a variational quantum circuit designed to produce synthetic data from a noise vector. The Figure 2 depicts an instance of the several tested configurations. Each qubit in the circuit represents a feature of the dataset. The number of qubits, therefore, equals the number of features. The circuit begins by applying Hadamard gates to all qubits, bringing them into a superposition state. This ensures that the circuit starts with a uniform distribution over the possible states. A feature map is applied to enhance the expressiveness of the circuit by encoding latent variables into the quantum state. In this circuit the ZZFeatureMap utilizes entanglement across qubits, introducing interdependence among qubits. The ansatz defines the trainable portion of the quantum circuit. In the scheme of this specific configuration, EfficientSU2 architecture is used, characterized by a combination of single-qubit rotation gates and two-qubit entangling gates. By repeating these layers multiple times, the circuit gains expressive power, allowing it to approximate intricate probability distributions. The number of repetitions (hyperparameter) controls the trade-off between circuit depth and the capability to represent complex distributions.

The generator incorporates trainable parameters associated with the rotation gates in the ansatz. Additionally, noise is injected into the circuit via rotation gates, parameterized by a vector representing the noise input. The combination of noise input and trainable weights enables the generator to sample from a wide range of data distributions.

The circuit is converted into a Quantum Neural Network (QNN) using a quantum sampler to evaluate the circuit and compute gradients, which are then updated leveraging Adam optimizer. This QNN framework enables integration with classical optimization techniques for training. The sampler evaluates the circuit multiple times (shots), ensuring robust estimates of expectation values.

The interpret function transforms the generator's output, which corresponds to measurements of all possible combinations of n qubits, into a lower-dimensional representation of cardinality n. Each of the 2^n values produced by the generator represents the probability of observing a specific combination of qubit states. For each qubit i, the interpret function extracts its marginal probability by summing over all measurement outcomes where qubit i is in state 1. Let $P(x)$ represent the probability of observing a particular n-qubit state $x = (x_1, x_2, \ldots, x_n)$, where $x_i \in \{0, 1\}$. Then, the marginal probability p_i for qubit i is computed as:

$$p_i = \sum_{x:x_i=1} P(x),$$

where the summation runs over all 2^n possible states x such that the i-th qubit is 1. The resulting vector $(p_1, p_2, \ldots, p_n)$, of dimension n, represents the marginal probabilities of each qubit being in state 1. Each component p_i lies in the range $[0, 1]$, making it directly comparable to the features in the real dataset.

This dimensionality reduction captures relevant probabilistic information about each qubit while discarding higher-order correlations between multiple qubits. The transformation simplifies the discriminator's input without sacrificing the ability to distinguish between real and generated data. The transformed n-dimensional data is then fed into the discriminator.

The discriminator is a classical feedforward neural network designed for binary classification, distinguishing between real and generated data. It takes input vectors matching the dataset's feature size and processes them through dense hidden layers with LeakyReLU activations, which introduce nonlinearity while mitigating vanishing gradients. The output layer uses a sigmoid activation to produce a probability score indicating whether the input is real.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Trained with Binary Cross-Entropy loss, the discriminator iteratively improves through adversarial feedback with the generator. Evaluation metrics such as loss values, accuracy, and the F1 score monitor performance, while the quality of generated data is validated by comparing its statistical properties with real data.

In the following section it will be discussed all the tested configurations and the relative results. The experiments are replicable, and the source code is available on (Franco Cirillo, 2024).

```
Input  : Number of qubits $n$, batch size $b$,
         learning rates $l_r^G, l_r^D$, epochs $N$, real
         data $X$.
Output: Trained generator $G$ and
         discriminator $D$.
Initialize: Create generator $G$ using quantum
  circuit and noise parameters $z$;
Define discriminator $D$, a classical
  feedforward neural network;
Initialize Adam optimizers Adam$_G$ and
  Adam$_D$ with learning rates $l_r^G, l_r^D$;
Define Binary Cross-Entropy Loss $\mathcal{L}_{BCE}$;
for epoch $e \leftarrow 1$ to $N$ do
    Step 1: Train Discriminator with Real
      Data;
    Sample a batch $X_{\text{real}} \subseteq X$ of size $b$;
    Compute real loss:
      $\mathcal{L}_D^{\text{real}} \leftarrow \mathcal{L}_{BCE}(D(X_{\text{real}}), 1)$;
    Update discriminator: Adam$_D \leftarrow \nabla \mathcal{L}_D^{\text{real}}$;
    Step 2: Train Discriminator with
      Generated Data;
    Generate noise $z \sim \mathcal{N}(0, 1)$;
    Generate data: $X_{\text{gen}} \leftarrow G(z)$;
    Apply interpret function:
      $X_{\text{gen}} \leftarrow \text{InterpretFunction}(X_{\text{gen}})$;
    Compute gen loss:
      $\mathcal{L}_D^{\text{gen}} \leftarrow \mathcal{L}_{BCE}(D(X_{\text{gen}}), 0)$;
    Update discriminator: Adam$_D \leftarrow \nabla \mathcal{L}_D^{\text{gen}}$;
    Step 3: Train Generator;
    Generate new data: $X_{\text{gen}} \leftarrow G(z)$;
    Apply interpret function:
      $X_{\text{gen}} \leftarrow \text{InterpretFunction}(X_{\text{gen}})$;
    Compute generator loss:
      $\mathcal{L}_G \leftarrow \mathcal{L}_{BCE}(D(X_{\text{gen}}), 1)$;
    Update generator: Adam$_G \leftarrow \nabla \mathcal{L}_G$;
    Step 4: Monitor Training Progress;
    Generate samples $X_{\text{gen}} \leftarrow G(z)$ and apply
      interpret function;
    Compare statistical properties and
      distributions of $X_{\text{gen}}$ with $X$;
    Record losses: $\mathcal{L}_D^{\text{real}}, \mathcal{L}_D^{\text{gen}}, \mathcal{L}_G$;
    Evaluate metrics (accuracy, F1-score);
end
```

Algorithm 1: QGAN Training Algorithm with Quantum Generator and Classical Discriminator.

5 RESULTS AND DISCUSSION

To evaluate the proposed QGAN model, several hyperparameters have been taken in consideration. Table 1 presents a comprehensive analysis of various configurations implemented, highlighting the performance of each setup in terms of accuracy and F1 score. These configurations are among the most significant experiments conducted in the study, providing insights into the impact of different architectural and hyperparameter choices on the QGAN's performance. For all the experiments, the training process was executed over 80 epochs to ensure sufficient convergence and reliable results, and a ZZFeatureMap has been applied for encoding noise allowing the quantum generator to effectively learn complex data distributions.

A clear trend can be observed where increasing the number of generator repetitions (reps) generally leads to slightly improved performance metrics. For example, moving from 6 to 9 generator repetitions resulted in one of the highest performance outcomes (accuracy = 0.937, F1 score = 0.9384). However, while the gains in performance are incremental, the computational time required to train the model increases substantially. This trade-off must be carefully considered when deciding on the optimal number of generator repetitions for real-world applications.

Results also show that configurations with at least 32 discriminator neurons tend to achieve more stable performance. Using fewer neurons, such as 8 or 16, often results in instability in the loss functions during training, potentially leading to less reliable convergence. For instance, configurations with 32 or 64 discriminator neurons consistently yield higher accuracy and F1 scores compared to those with 8 neurons. Additionally, configurations with 64 or more discriminator neurons slightly outperform those with 32, although the improvements are not always substantial.

The learning rates for both the generator and discriminator (denoted as lr_g and lr_d) play a crucial role in determining the stability and convergence of the QGAN. While most configurations use balanced learning rates (e.g., 0.01 for both), experiments with significantly smaller or unbalanced learning rates (e.g., 0.001/0.005 or 0.003/0.008) show mixed results. These settings sometimes lead to minor performance

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Table 1: Performance of QGAN with different configurations.

Features	**Ansatz**	**Generator Reps**	**Discriminator Layers**	**Learning Rates (Gen/Disc)**	**Performance (Accuracy / F1)**
6	EfficientSU2	3	16	0.01 / 0.01	0.914 / 0.9189
5	EfficientSU2	3	16	0.01 / 0.01	0.909 / 0.916
6	EfficientSU2	3	8	0.01 / 0.01	0.899 / 0.9001
6	EfficientSU2	3	32	0.01 / 0.01	0.911 / 0.9178
6	EfficientSU2	3	64	0.01 / 0.01	0.923 / 0.9241
6	RealAmplitudes	3	32	0.01 / 0.01	0.915 / 0.9202
6	EfficientSU2	6	32	0.01 / 0.01	0.9210 / 0.9187
6	RealAmplitudes	3	32	0.001 / 0.001	0.904 / 0.9087
6	EfficientSU2	6	64	0.01 / 0.01	0.9270 / 0.9215
6	EfficientSU2	6	64	0.001 / 0.005	0.914 / 0.9007
6	EfficientSU2	6	64	0.005 / 0.001	0.909 / 0.8991
6	EfficientSU2	10	8	0.01 / 0.01	0.913 / 0.9012
6	EfficientSU2	6	128	0.01 / 0.01	0.917 / 0.9015
5	EfficientSU2	8	16	0.01 / 0.001	0.793 / 0.7812
6	EfficientSU2	6	64	0.001 / 0.005	0.909 / 0.9161
6	EfficientSU2	10	8	0.01 / 0.01	0.922 / 0.9275
6	EfficientSU2	6	64	0.003 / 0.008	0.913 / 0.918
6	RealAmplitudes	3	32	0.007 / 0.007	0.914 / 0.9189
6	EfficientSU2	9	128	0.01 / 0.01	**0.937 / 0.9384**
6	EfficientSU2	6	64	0.006 / 0.01	0.917 / 0.9222
5	EfficientSU2	8	64	0.01 / 0.001	0.899 / 0.9083

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

drops, highlighting the importance of carefully tuning the learning rates.

Regarding the ansatz, EfficientSU2 and RealAmplitudes are compared in the table. Overall, EfficientSU2 demonstrates slightly better performance across most configurations, particularly when combined with 6 principal components. Reducing the number of features to 5 generally leads to a slight decrease in performance, indicating that retaining more features provides the model with richer information to generate better outputs, but it is not without limitations.

The highest-performing configuration combines 6 PCA features, the EfficientSU2 ansatz, 9 generator repetitions, and 128 discriminator neurons. This setup achieves an accuracy of 0.937 and an F1 score of 0.9384.

The plot shown in Figure 3 represents the loss of the generator and the discriminator (for both real and generated data) over the course of training, focusing on the configuration with the best performance. The generator's loss starts relatively high at the beginning of the training process, as it initially produces poorly generated samples. As the epochs progress, the generator's loss gradually decreases, indicating its improved ability to produce data that closely resembles the real dataset. By the end of the training process, the generator achieves a relatively stable loss, suggesting that it has effectively learned the underlying data distribution.

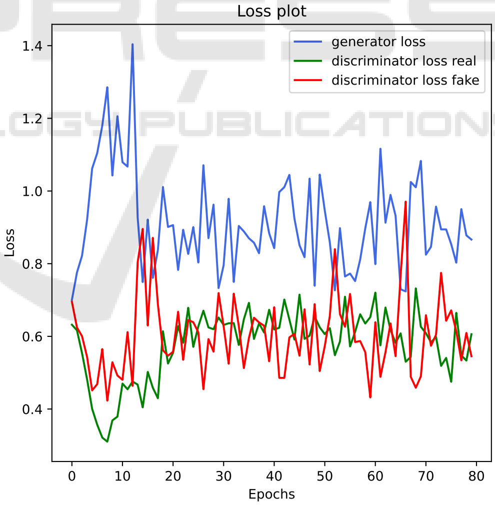

Figure 3: Generator and Discriminator loss during the training process.

The discriminator's loss for real data starts lower at the beginning, as the discriminator can easily distinguish between real and generated samples when the generator is weak. However, as the generator improves, the discriminator's task becomes more challenging, leading to an increase in its loss for real data. Toward the later epochs, the loss stabilizes, indicating

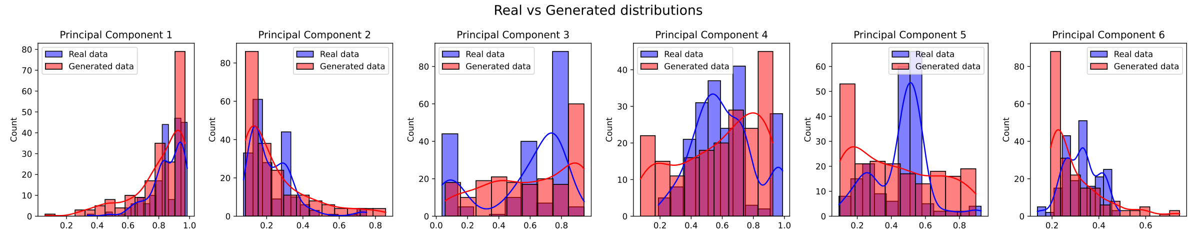

Figure 4: Distribution comparison of feature values with real and generated samples.

that the discriminator has adapted to the generator's improved outputs.

Instead, the loss for generated data follows a complementary trend to the generator's progress. Initially, the discriminator easily identifies fake data, resulting in a low loss. As the generator's outputs improve, the discriminator's loss for fake data increases, reflecting its reduced confidence in distinguishing fake data from real data. By the end of training, this loss also stabilizes, balancing the adversarial dynamics.

The overall stability in loss values across the epochs demonstrates that the adversarial training between the generator and discriminator has reached equilibrium.

The provided plots in Figure 4 compare the distributions of real data and generated data for each of the six principal components (PCA features) extracted from the dataset for this best parameter configuration. These visualizations evaluate how closely the generated data aligns with the real data in terms of feature distribution, providing an important assessment of the generator's ability to replicate the underlying patterns of the original dataset.

The plots demonstrate a strong alignment between the real and generated data distributions, indicating that the generator successfully learns the essential statistical features of the dataset. In most cases, the generated data closely follows the shape, range, and density of the real data, particularly in high-density regions where the majority of data points are concentrated. This suggests that the generator is effective at capturing the primary structure of the dataset.

Minor deviations are observed in certain areas, especially in lower-density regions or the extremes of the distributions, where the generator occasionally underestimates or overestimates specific ranges. These variations reflect potential areas for improvement, such as fine-tuning hyperparameters or increasing the training iterations to better capture the nuances of the dataset.

The results validate the effectiveness of the QGAN configuration in generating synthetic data that closely mirrors the real dataset. Loss trends confirm stable adversarial training, and distribution plots show strong alignment between real and generated data, with minor deviations in low-density regions. Overall, the model proves highly effective, balancing accuracy and realism, though computational trade-offs must be considered.

6 CONCLUSION AND FUTURE WORK

This work presents an adaptation of the hybrid QGAN framework for intrusion detection, showcasing its potential to address key challenges in anomaly detection systems. The quantum generator, implemented using a VQC, effectively learns the data distribution and generates realistic samples, while the classical discriminator evaluates these samples to refine the generator's performance. This structure not only enhances anomaly detection capabilities but also maintains compatibility with standard evaluation frameworks. The QGAN was tested on the NSL-KDD dataset, under various configurations, including changes to the number of qubits, circuit repetitions, ansatz structures, and feature maps, achieving an accuracy of 0.937 and an F1 score of 0.9384, demonstrating its effectiveness and scalability.

Future research will extend these experiments to explore a wider range of configurations and datasets, focusing on optimizing performance across diverse environments. Further, integrating advanced quantum hardware and incorporating domain-specific feature engineering could unlock even greater potential for QGAN-based intrusion detection systems.

ACKNOWLEDGEMENTS

This work was partially supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government (MOTIE) (RS-2023-00303559, Study on developing cyber-physical attack response system and security management system to maximize real-time distributed resource availability), and by project SER-

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

ICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team.

REFERENCES

Abdulganiyu, O. H., Ait Tchakoucht, T., and Saheed, Y. K. (2023). A systematic literature review for network intrusion detection system (ids). *International journal of information security*, 22(5):1125–1162.

Bermot, E., Zoufal, C., Grossi, M., Schuhmacher, J., Tacchino, F., Vallecorsa, S., and Tavernelli, I. (2023). Quantum Generative Adversarial Networks For Anomaly Detection In High Energy Physics. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 01, pages 331–341.

Cerezo, M., Verdon, G., Huang, H.-Y., Cincio, L., and Coles, P. J. (2022). Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576.

Chang, S. Y., Thanasilp, S., Saux, B. L., Vallecorsa, S., and Grossi, M. (2024). Latent style-based quantum gan for high-quality image generation. *arXiv preprint arXiv:2406.02668*.

Dallaire-Demers, P.-L. and Killoran, N. (2018). Quantum generative adversarial networks. *Physical Review A*, 98(1):012324.

De Falco, F., Ceschini, A., Sebastianelli, A., Le Saux, B., and Panella, M. (2024). Quantum latent diffusion models. *Quantum Machine Intelligence*, 6(2):85.

Franco Cirillo (2024). Qgan code. https://github.com/francocirill/qgan.

Gui, J., Sun, Z., Wen, Y., Tao, D., and Ye, J. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE transactions on knowledge and data engineering*, 35(4):3313–3332.

Herr, D., Obert, B., and Rosenkranz, M. (2021). Anomaly detection with variational quantum generative adversarial networks. *Quantum Science and Technology*, 6(4):045004.

Kalfon, B., Cherkaoui, S., Laprade, J., Ahmad, O., and Wang, S. (2024). Successive data injection in conditional quantum GAN applied to time series anomaly detection. *IET Quantum Communication*, 5(3):269–281.

Lee, J. and Park, K. (2021). Gan-based imbalanced data intrusion detection system. *Personal and Ubiquitous Computing*, 25(1):121–128.

Lloyd, S. and Weedbrook, C. (2018). Quantum generative adversarial learning. *Physical review letters*, 121(4):040502.

Moustafa, N. and Slay, J. (2015). Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6. IEEE.

Muneer, S., Farooq, U., Athar, A., Ahsan Raza, M., Ghazal, T. M., and Sakib, S. (2024). A critical review of artificial intelligence based approaches in intrusion detection: A comprehensive analysis. *Journal of Engineering*, 2024(1):3909173.

Ngo, T. A., Nguyen, T., and Thang, T. C. (2023). A survey of recent advances in quantum generative adversarial networks. *Electronics*, 12(4):856.

Patil, R., Biradar, R., Ravi, V., Biradar, P., and Ghosh, U. (2022). Network traffic anomaly detection using pca and bigan. *Internet Technology Letters*, 5(1):e235.

Rafique, S. H., Abdallah, A., Musa, N. S., and Murugan, T. (2024). Machine learning and deep learning techniques for internet of things network anomaly detection—current research trends. *Sensors*, 24(6):1968.

Rahman, M. A., Shahriar, H., Clincy, V., Hossain, M. F., and Rahman, M. (2023). A Quantum Generative Adversarial Network-based Intrusion Detection System. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1810–1815, Torino, Italy. IEEE.

Schlegl, T., Seeböck, P., Waldstein, S. M., Langs, G., and Schmidt-Erfurth, U. (2019). f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54:30–44.

Shahriar, M. H., Haque, N. I., Rahman, M. A., and Alonso, M. (2020). G-ids: Generative adversarial networks assisted intrusion detection system. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 376–385.

Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A. (2009). A detailed analysis of the kdd cup 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications*, pages 1–6. Ieee.

Truong-Huu, T., Dheenadhayalan, N., Pratim Kundu, P., Ramnath, V., Liao, J., Teo, S. G., and Praveen Kadiyala, S. (2020). An empirical study on unsupervised network anomaly detection using generative adversarial networks. In *Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence*, pages 20–29.

Vieloszynski, A., Cherkaoui, S., Laprade, J.-F., Nahman-Lévesque, O., Aaraba, A., and Wang, S. (2024). Latentqgan: A hybrid qgan with classical convolutional autoencoder. *arXiv preprint arXiv:2409.14622*.

Yamasaki, H., Isogai, N., and Murao, M. (2023). Advantage of quantum machine learning from general computational advantages.

Zoufal, C., Lucchi, A., and Woerner, S. (2019). Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information*, 5(1):103.