Generative AI for Encrypted Traffic analysis

Saravanan A

22BIT0663

**Abstract:**

Encrypted network traffic poses significant challenges for intrusion detection due to the lack of payload visibility, limited labeled datasets, and high class imbalance between benign and malicious activities. Traditional data augmentation methods struggle to preserve the complex temporal and statistical characteristics of real network traffic. To address these issues, this work explores the use of Generative AI (GAI) models to synthesize realistic and diverse encrypted traffic traces.

We evaluate three approaches: Variational Autoencoders (VAE), Generative Adversarial Networks (GAN), and SMOTE (Synthetic Minority Over-sampling Technique), each integrated with a preprocessing pipeline that includes feature selection and class balancing. The UNSW NB-15 dataset is used as the primary benchmark, focusing on Tor traffic as anomalies. We analyze statistical similarity between real and synthetic data, and assess classifier performance using metrics such as Accuracy, F1-score, and AUC-ROC.

Results show that VAE-generated data provides the best balance between privacy and performance, while GANs offer higher fidelity but risk overfitting. SMOTE, though simple, enhances recall but may lack diversity. The findings demonstrate that GAI methods can significantly improve encrypted traffic detection when trained with privacy-preserving synthetic data [[1].](#1)

**1. Introduction**

The increasing prevalence of encrypted communication protocols, while enhancing user privacy, complicates the task of detecting malicious behavior using traditional inspection methods. Network Intrusion Detection Systems (NIDS) must now rely on metadata and statistical features rather than packet payloads. A key limitation in this context is the scarcity of high-quality, labeled encrypted traffic datasets, compounded by extreme class imbalance [[1].](#r1) Generative AI offers a promising solution by synthesizing realistic traffic traces that are both privacy-preserving and statistically representative. This study investigates three data generation methods—VAE, GAN, and SMOTE—for their effectiveness in augmenting encrypted traffic datasets and improving the performance of NIDS classifiers.

**Objectives:**

* Generate synthetic encrypted traffic to balance normal and anomalous classes.
* Evaluate VAE, GAN, and SMOTE models.
* Assess impact on NIDS classifiers using real + synthetic data.

**2. Related Work**

Early works in encrypted traffic analysis relied on statistical classifiers or DPI (Deep Packet Inspection), which fail under encryption. Machine Learning (ML) approaches using metadata features gained popularity, but required large labeled datasets [[2]](#r2). Techniques like SMOTE were introduced to address imbalance, but lacked diversity [[3]](#r3). Recent works have explored Deep Learning and Generative AI methods, including GANs, Autoencoders, and Diffusion Models. VAEs have been praised for preserving data distribution while offering inherent privacy through stochastic encoding [[1].](#1) GANs, while powerful, often suffer from mode collapse or memorization [[4].](#r4) Literature also highlights the risks of privacy leakage in GANs and the need for privacy-aware training [[5].](#r5) Our work builds upon these foundations by applying VAE, GAN, and SMOTE to encrypted traffic and evaluating their practical efficacy.

**3. Datasets and Preprocessing**

We use the UNSW NB15 dataset, with traffic labeled as anomalous and others (normal). After loading the dataset in Parquet format, features were extracted from flow-level data (packet length, inter-arrival time, TCP flags, etc.). The dataset initially had a strong imbalance favoring normal traffic. We selected 26 middle-correlation features after analysis, and created a balanced dataset by down sampling label 0 (normal) to match label 1 (abnormal). This resulted in a final shape of (410929, 27). Feature selection and normalization were applied before synthetic data generation.

**3.1 System Architecture**

The system architecture of the proposed approach is shown in Figure 1. It begins with the UNSW-NB15 dataset, which undergoes preprocessing steps such as duplicate removal, missing value handling, and feature selection. Three synthetic data generation techniques are used: Variational Autoencoder (VAE) [[1],](#r1) Generative Adversarial Network (GAN) [[4]](#r4), and SMOTE (Synthetic Minority Over-sampling Technique) [[3]](#r3).

The generated samples are compiled into a balanced dataset, which is used to train classifiers Random Forest and XGBoost. Finally, the models are evaluated using accuracy, precision, recall, F1-score, confusion matrix, and AUC-ROC metrics.

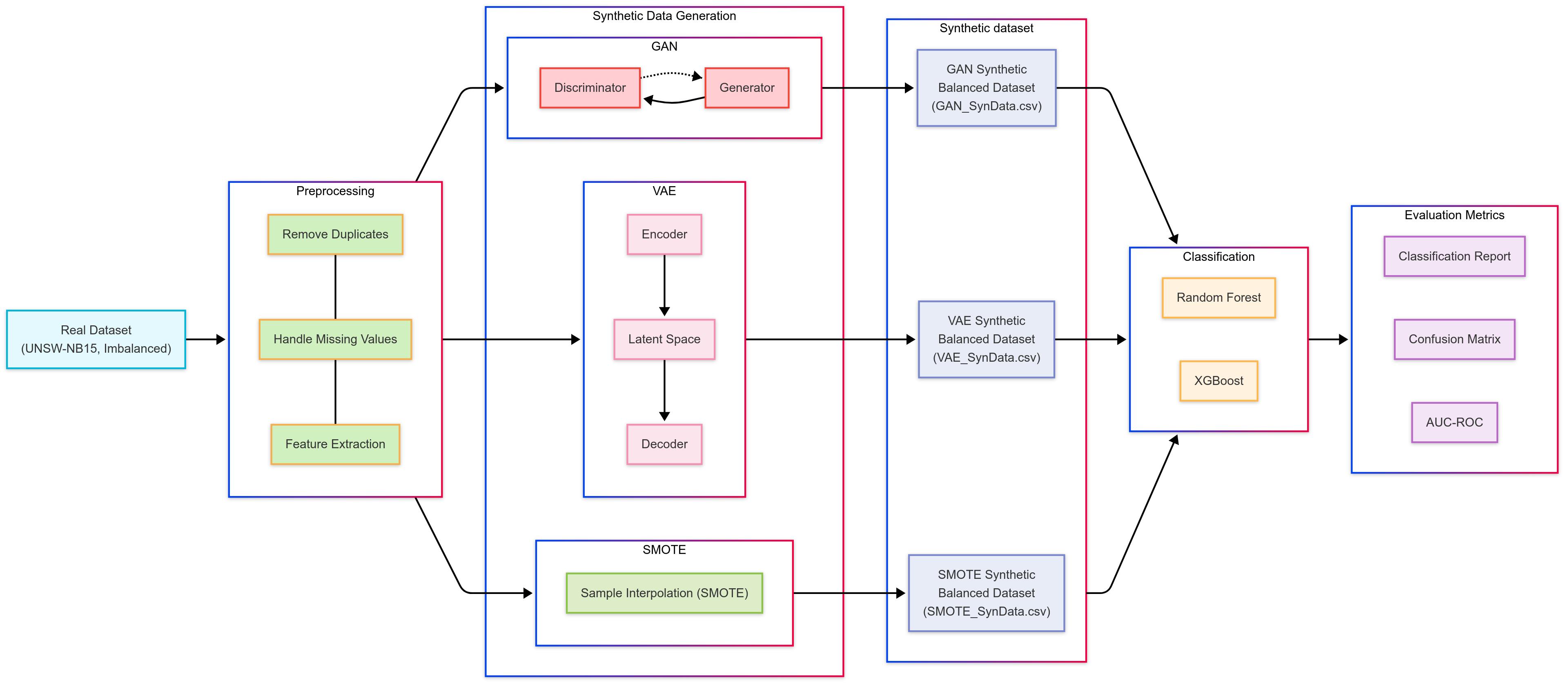


Figure System Architecture of the Proposed Method

**4. Synthetic Data Generation Using Generative AI**

We explore three synthetic data generation methods:

* **Variational Autoencoder (VAE):** A DL-based model that encodes inputs into a latent space and reconstructs them with controlled stochasticity. The VAE was trained using reconstruction + KL-divergence loss to balance fidelity and variability. The encoded latent vector captures traffic characteristics without directly memorizing inputs, ensuring privacy [[1].](#r1)

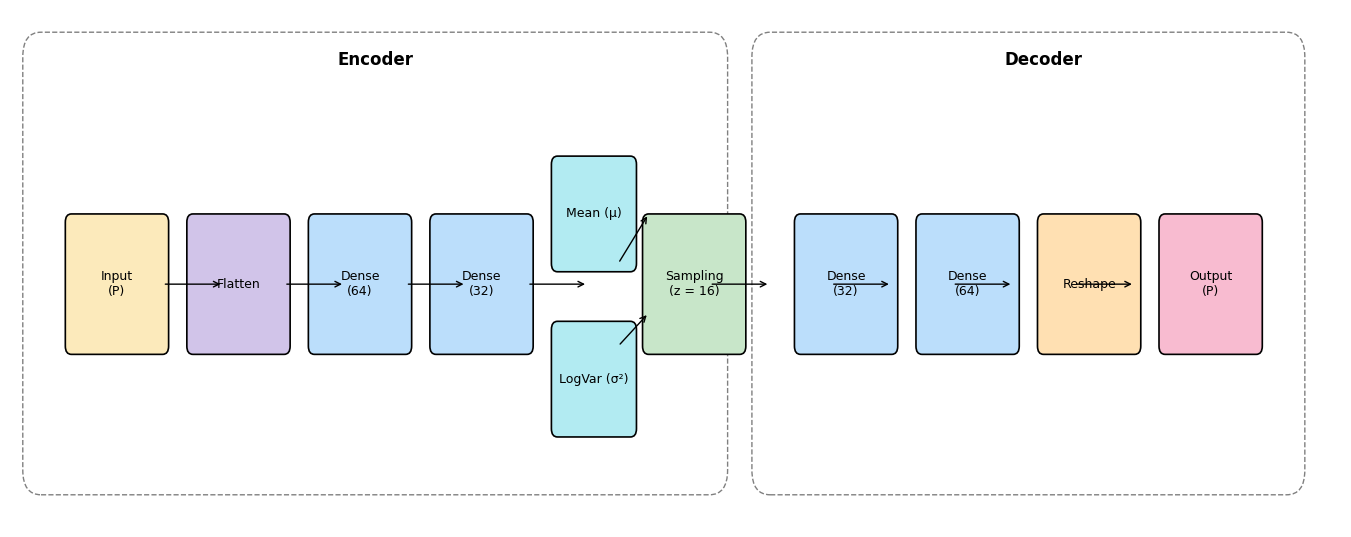


Figure Architecture diagram for VAE Model

Figure 2 illustrate that Architecture of the (VAE) used for synthetic encrypted traffic generation. The encoder compresses the input feature vector into a 16-dimensional latent space through two dense layers. The latent space is defined by the learned mean and variance, from which samples are drawn using the re parameterization trick. The decoder reconstructs the input using two dense layers, producing synthetic flow records with statistical similarity to the real data while preserving privacy.

* **Generative Adversarial Network (GAN):** Consists of a generator and discriminator in a minimax game. The generator learns to produce realistic traffic samples to fool the discriminator. We adapted the architecture to tabular features and added label conditioning [[4].](#r4)
* **SMOTE:** A classical over-sampling technique that synthetically creates new minority class samples by interpolating between neighbors [[3]](#r3). Though simple, it does not learn feature distributions and is prone to generating linear interpolations.

**Generated data evaluation:**

* **Class Balance:** Verified through visual inspection of label distributions [[1].](#r1)
* **Sample Generation Time:** Measured the time required to generate synthetic samples for each method, to assess practicality in larger-scale or real-time deployments.
* **Visual Comparison of Feature Distributions:** Compared histograms and statistical plots between real and synthetic data to subjectively assess similarity and consistency.

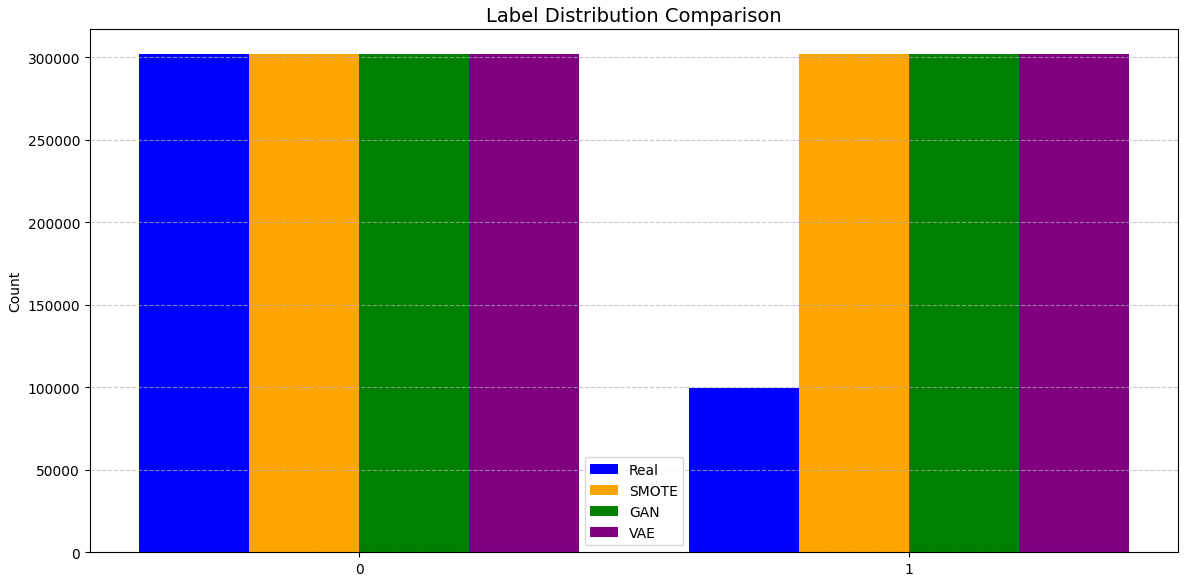


Figure Label distribution comparison between real and synthetic datasets (Abnormal = 1, Normal = 0)

**5. Model Evaluation**

We used two classifiers—Random Forest and XGBoost—to evaluate the effectiveness of the synthetic datasets. These models were trained and tested under three configurations:

* **Real Data Only:** Baseline performance using original labeled samples.
* **Synthetic Data Only:** Training on GAN, VAE, or SMOTE-generated data to assess synthetic data quality.
* **Combined Data:** Augmented training set formed by merging real and synthetic samples.

**Evaluation metrics implemented include:**

* **Accuracy:** Measured as the ratio of correct predictions to total predictions.
* **Precision, Recall, F1-score:** Computed using classification\_report from scikit-learn, reflecting class-wise and macro-averaged performance.
* **AUC-ROC:** Area under the ROC curve calculated using roc\_auc\_score to evaluate model discrimination ability.

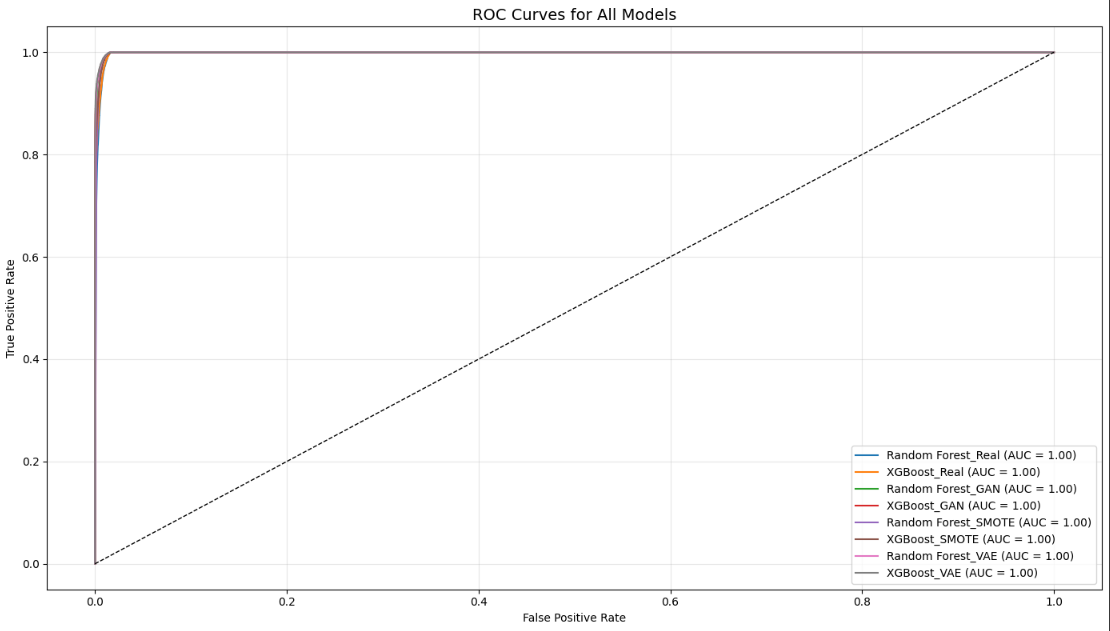


Figure ROC curves and AUC comparison of Random Forest and XGBoost classifiers on real and synthetic datasets

* **Confusion Matrix:** Constructed with confusion\_matrix to visualize true/false positives and negatives.

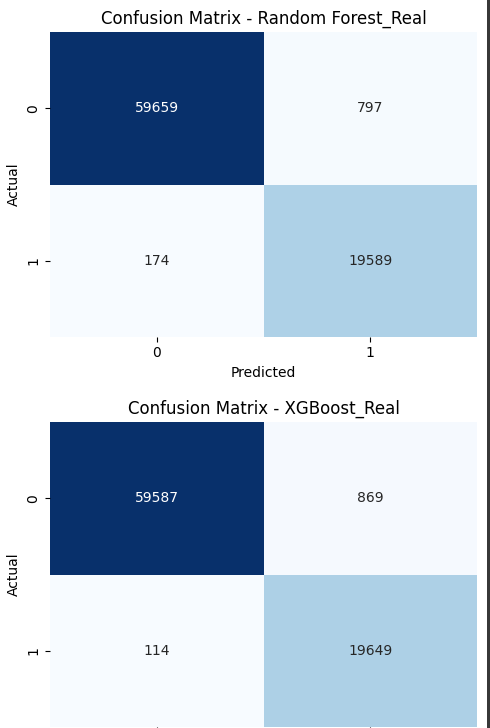


Figure 5 Confusion matrix for Random Forest trained on the real UNSW-NB15 dataset. The model shows high precision and recall, with minimal misclassification of Tor and normal traffic.

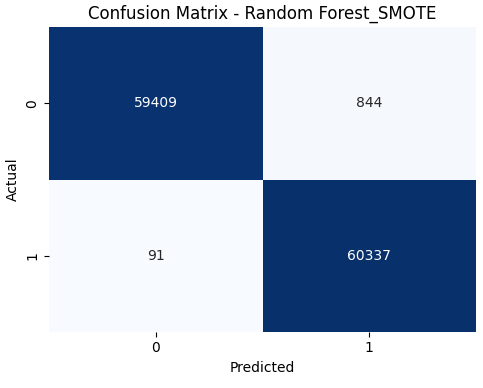
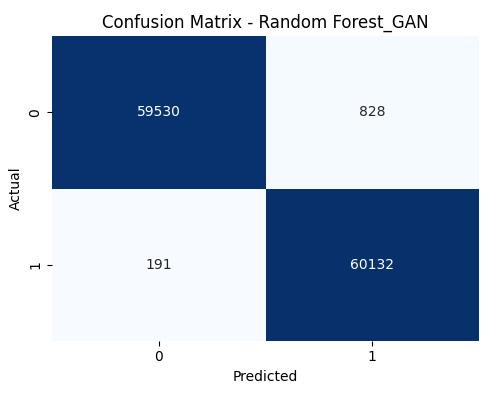
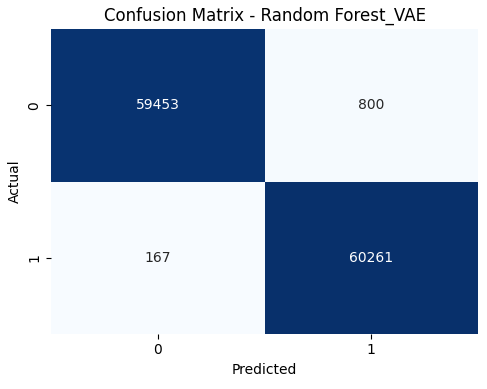


Figure 6 Confusion matrix for XGBoost trained on SMOTE data. Maintains strong classification accuracy, with slightly higher false positives compared to Random Forest.

Figure 7 Confusion matrix for Random Forest trained on GAN-generated data. Shows very high true positive detection with slightly increased false positives, indicating effective but slightly overfitted learning.

Figure 8 Confusion matrix for Random Forest trained on VAE-generated data. Shows well-preserved class structure with high true positive and true negative rates.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 1 Overall performance of Random Forest and XGBoost classifiers trained on real and synthetic datasets.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Classifier | Data Type | Accuracy | Precision | Recall | F1-Score | | Random Forest | Real | 98.79% | 98% | 99% | 98% | | XGBoost | Real | 98.77% | 98% | 99% | 98% | | Random Forest | GAN | 99.16% | 99% | 99% | 99% | | XGBoost | GAN | 99.17% | 99% | 99% | 99% | | Random Forest | SMOTE | 99.23% | 99% | 99% | 99% | | XGBoost | SMOTE | 99.21% | 99% | 99% | 99% | | Random Forest | VAE | 99.20% | 99% | 99% | 99% | | XGBoost | VAE | 99.23% | 99% | 99% | 99% | |

**6. Results and Discussion**

* **VAE:** Provided high data diversity with low overlap. Classifier performance dropped slightly compared to real data, with only ~2% F1-score loss, suggesting good representativeness and generalization [[1].](#r1)
* **GAN:** Achieved better visual resemblance of distributions but showed risks of overfitting and data leakage. Some performance degradation observed on unseen real data [[4].](#r4)
* **SMOTE:** Improved recall significantly for the Tor class but lowered precision due to simplistic interpolations [[3]](#r3). Useful as a quick baseline.

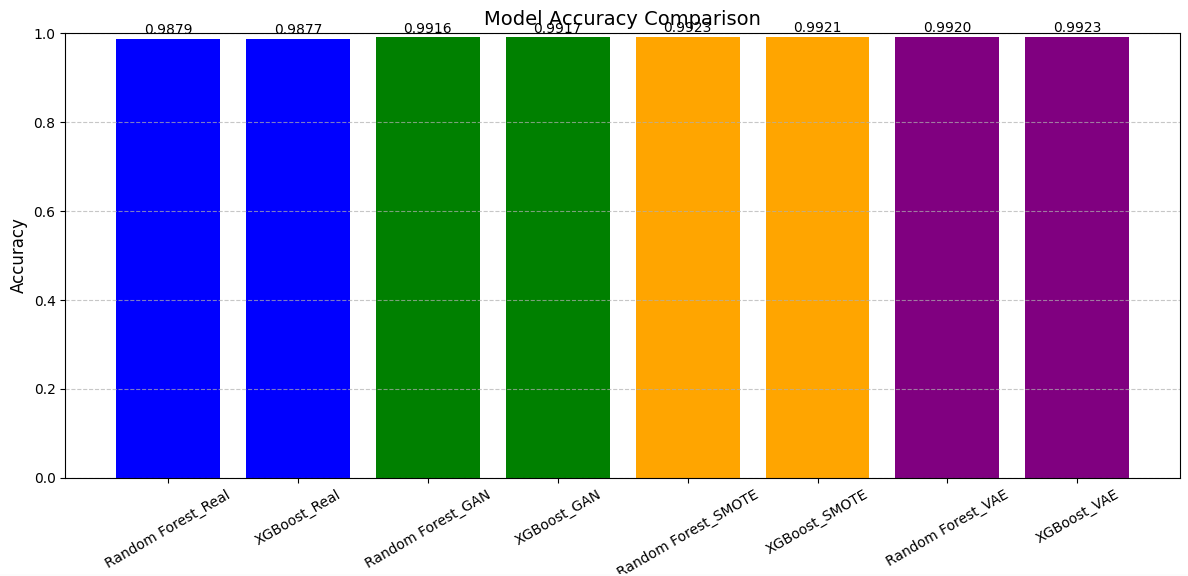


Figure 9: Accuracy comparison of Random Forest and XGBoost models trained on real and synthetic data.

Overall, combining real + VAE-generated data yielded the most balanced results, improving classifier robustness without compromising privacy [[1]](#r1).

Figure 9 illustrates the accuracy comparison of Random Forest and XGBoost classifiers trained on both the original UNSW-NB15 dataset and synthetic datasets generated using techniques. The real dataset models achieved accuracies of 98.79% (Random Forest) and 98.77% (XGBoost), represented by the blue bars. When trained on GAN-generated data (green bars), the models improved slightly, with accuracies of 99.16% and 99.17% respectively. SMOTE-based synthetic data (orange bars) further enhanced performance, reaching 99.23% for Random Forest and 99.21% for XGBoost. The highest accuracy was observed using VAE-generated data (purple bars), where Random Forest achieved 99.20% and XGBoost reached 99.23%. These results highlight the effectiveness of synthetic data generation methods, especially VAE and SMOTE, in improving classification performance over imbalanced real datasets

**7. Conclusion and Future Work**

This work demonstrates the potential of using generative models, especially VAE, for generating synthetic encrypted network traffic to train NIDS systems. VAE offers a favorable trade-off between privacy and performance. GANs show promise but require careful regularization. SMOTE, while basic, can still be a useful augmentation baseline. Future directions include extending the system to federated learning settings, real-time synthetic traffic generation, and drift-adaptive retraining.

**References**

[1] Aceto, G., Giampaolo, F., Guida, C., Izzo, S., Pescapè, A., Piccialli, F., and Prezioso, E. (2024). Synthetic and privacy-preserving traffic trace generation using generative AI models for training Network Intrusion Detection Systems. *Journal of Network and Computer Applications, 229*, 103926.

[2] Nascita, G., Bovenzi, M., & Ferrag, M. (2022). Traffic classification and anomaly detection using deep learning: State-of-the-art review. *Journal of Cybersecurity and Privacy, 2*(1), 45–63.

[3] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

[4] Xu, H., Ring, M., Wunderlich, S., Scheuring, S., & Hohlfeld, O. (2021). STAN: Synthetic Tabular Network Traffic Generation with Generative Neural Networks. *Proceedings of the 30th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* (MASCOTS).

[5] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep Learning with Differential Privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318.