Assignment Report: Generative Models and Phishing Web-Pages Detection a Real-World Challenge

Muhammad Abdullah 21L-6225 BSDS-8A

February 19, 2025

1. Introduction

This assignment explores two distinct machine learning tasks:

- **Generative Models**: I used Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) to generate realistic images of handwritten digits (MNIST) and fashion items (FashionMNIST). Additionally, i analyze their latent spaces using dimensionality reduction techniques like PCA.
- **Phishing Detection**: I also addressed a real-world problem by training a Variational Autoencoder (VAE) to detect phishing URLs in the phishing dataset. The VAE learns to reconstruct legitimate URLs and identifies anomalies (phishing URLs) based on reconstruction errors with accuracy of 70%.

The goal is to demonstrate how generative models can be applied to both synthetic data generation and anomaly detection in real-world scenarios.

2. Methodology

2.1 Generative Models for MNIST and FashionMNIST

2.1.1 Generative Adversarial Network (GAN)

The GAN architecture consists of two neural networks:

- **Generator**: Learns to map random noise vectors (latent space) to realistic images.
- **Discriminator**: Learns to distinguish between real images from the dataset and fake images generated by the generator.

2.1.2 Variational Autoencoder (VAE)

The VAE architecture consists of an encoder-decoder structure:

- **Encoder**: Maps input images to a probabilistic latent space (mean `mu` and log variance `logvar`).
- **Decoder**: Reconstructs the input image from a sample drawn from the latent space.

2.2 Phishing Detection Using VAE

The phishing detection task involves identifying phishing URLs in the phishing dataset. The approach is as follows:

1. **Data Preprocessing**: The dataset contains features extracted from URLs, with labels indicating whether each URL is legitimate or phishing.
2. **Feature Scaling**: Features are normalized using StandardScaler to ensure consistent input for the VAE.
3. **Model Architecture**: A VAE is trained to reconstruct legitimate URLs. The reconstruction error is used to identify anomalies (phishing URLs).
4. **Thresholding**: A threshold is determined based on the reconstruction errors of legitimate URLs. URLs with reconstruction errors exceeding the threshold are classified as phishing.

3. Hand-written Diagrams for GANs and VAEs Architectures

3.1 Training Architecture of GAN

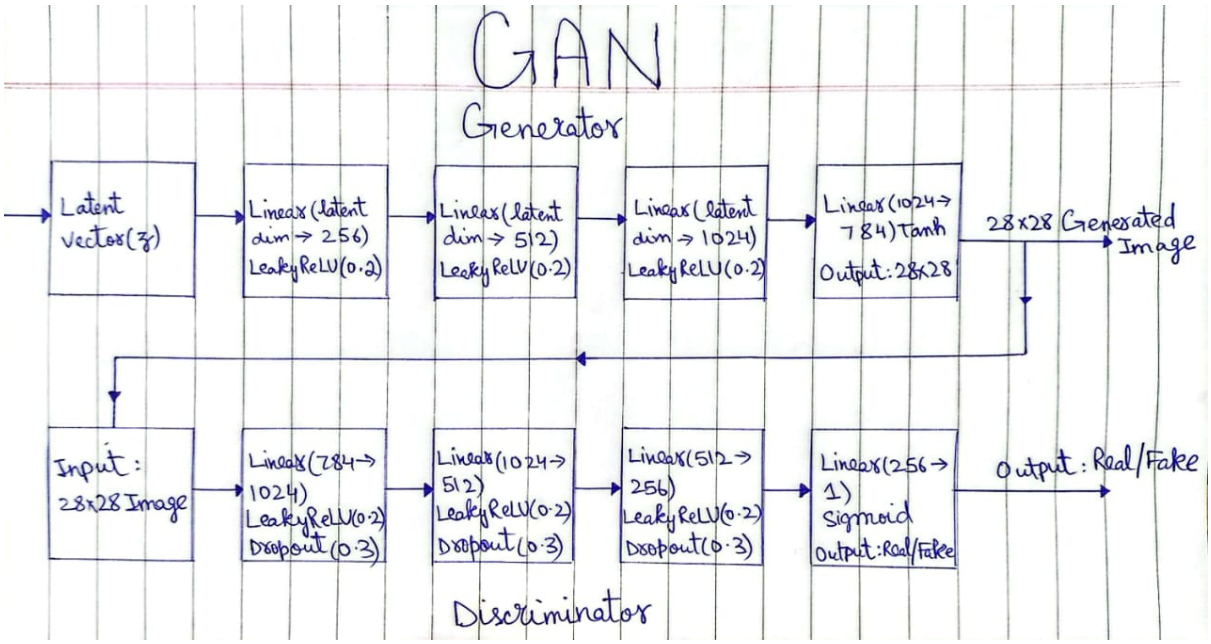

Figure 1: Training Architecture of GAN

3.2 Training Architecture of VAE

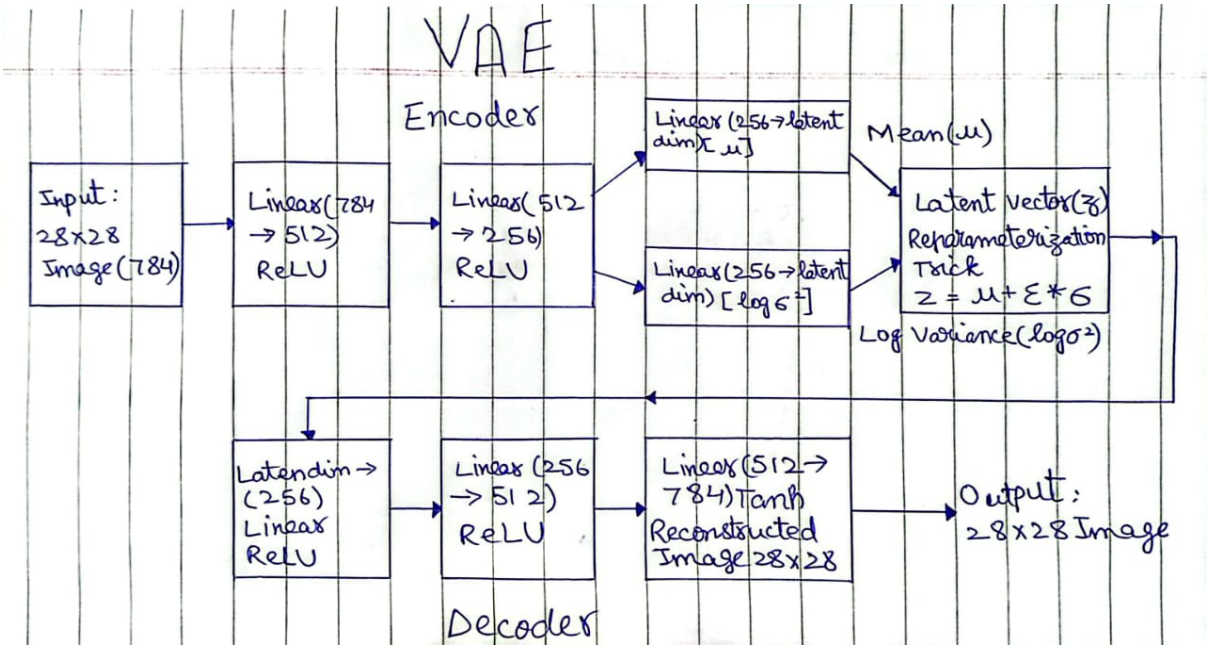

Figure 2: Training Architecture for VAE

4. Results

4.1 Image Generation Comparison: GAN vs. VAE

4.1.1 Generated Digits (MNIST)

Below is a comparison of digits generated by the GAN and VAE:

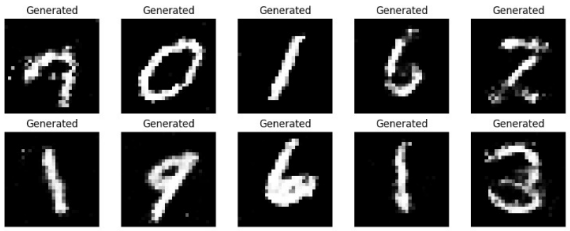

Figure 3: *
GAN-generated MNIST digits.

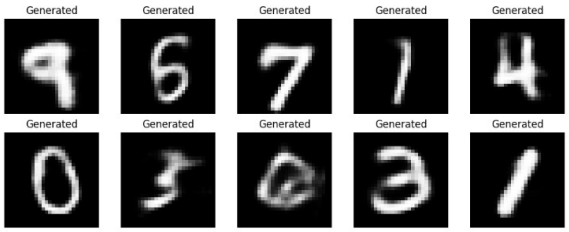

Figure 4: *
VAE-generated MNIST digits.

Figure 5: Comparison of GAN and VAE-generated MNIST digits.

4.1.2 Latent Space Visualization (MNIST Digits)

Below is a comparison of latent space visualizations for MNIST digits:

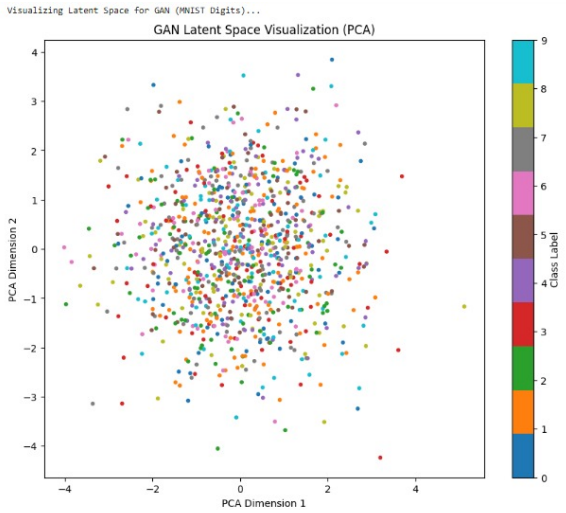

Figure 6: *
GAN latent space for MNIST digits.

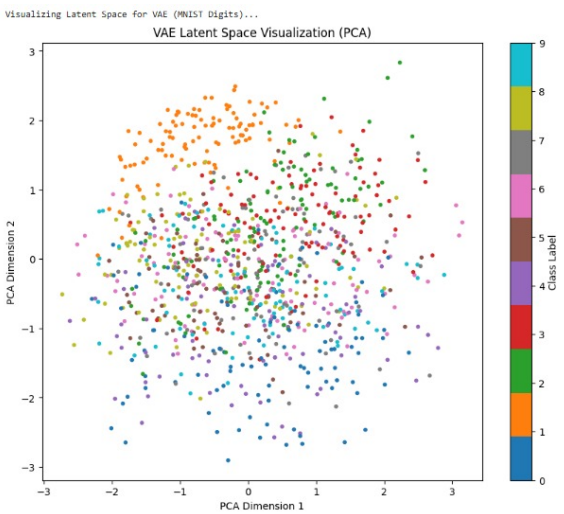

Figure 7: *
VAE latent space for MNIST digits.

Figure 8: Comparison of GAN and VAE latent space visualizations for MNIST digits.

4.1.3 Loss Curves (MNIST Digits)

Below is a comparison of training loss curves for MNIST digits:

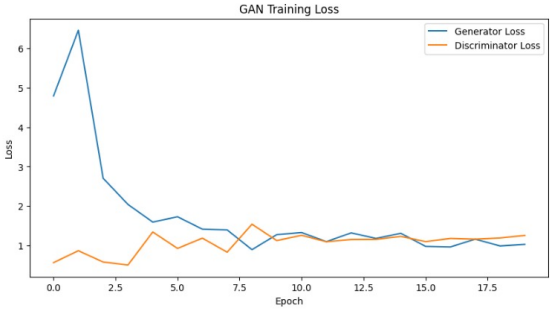

Figure 9: *
GAN training loss for MNIST digits.

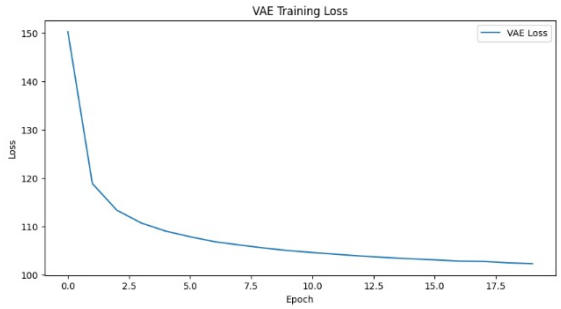

Figure 10: *
VAE training loss for MNIST digits.

Figure 11: Comparison of GAN and VAE training loss curves for MNIST digits.

4.1.4 Specific Digit Generation (Digit 5 vs. Digit 2)

Below is a comparison of specific digit generation (GAN for digit 5 vs. VAE for digit 2):

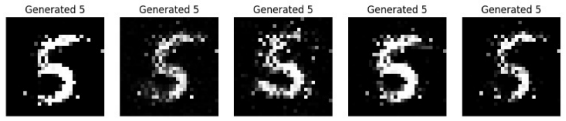

Figure 12: *
GAN-generated digit 5.

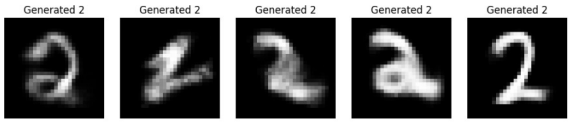

Figure 13: *
VAE-generated digit 2.

Figure 14: Comparison of GAN and VAE for specific digit generation.

4.1.5 FashionMNIST Shoes

Below is a comparison of FashionMNIST shoe generation:

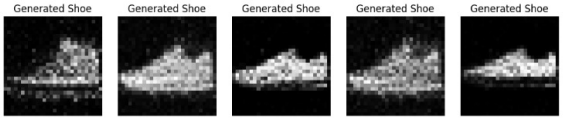

Figure 15: *
GAN-generated FashionMNIST shoes.

Figure 16: *
VAE-generated FashionMNIST shoes.

Figure 17: Comparison of GAN and VAE-generated FashionMNIST shoes.

4.1.6 Latent Space Visualization (FashionMNIST)

Below is a comparison of latent space visualizations for FashionMNIST:

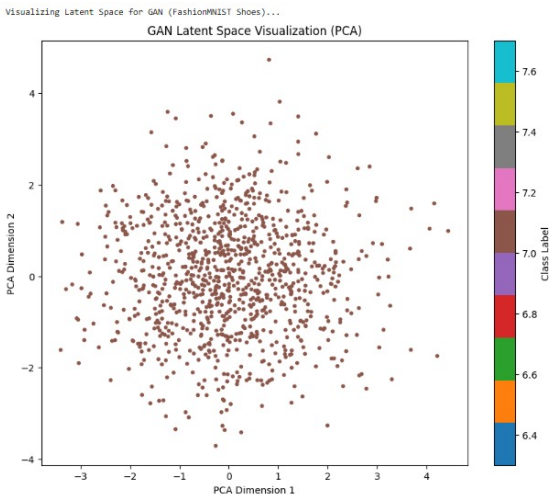

Figure 18: *
GAN latent space for FashionMNIST.

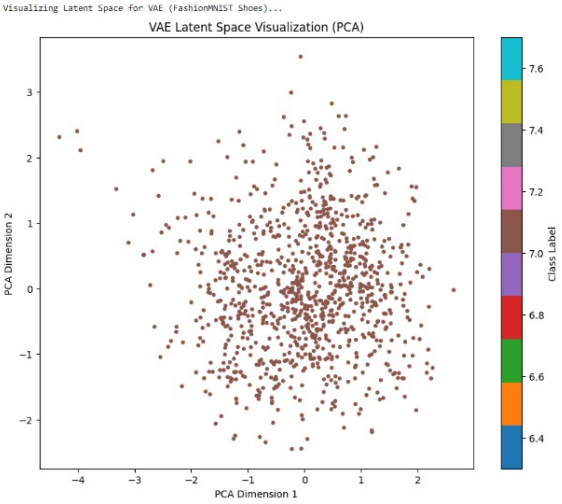

Figure 19: *
VAE latent space for FashionMNIST.

Figure 20: Comparison of GAN and VAE latent space visualizations for FashionMNIST.

4.1.7 Loss Curves (FashionMNIST)

Below is a comparison of training loss curves for FashionMNIST:

Figure 21: *
GAN training loss for FashionMNIST.

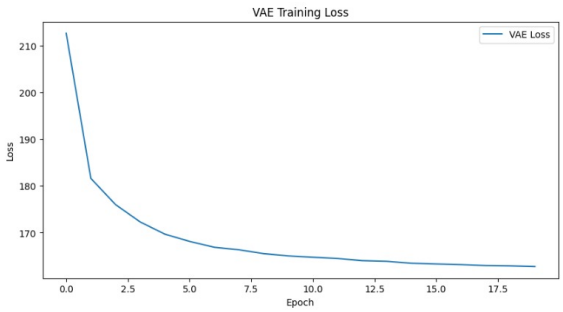

Figure 22: *
VAE training loss for FashionMNIST.

Figure 23: Comparison of GAN and VAE training loss curves for FashionMNIST.

4.2 Phishing Detection Results

4.2.1 ROC Curve

The performance of the VAE-based phishing detector is evaluated using the ROC curve:

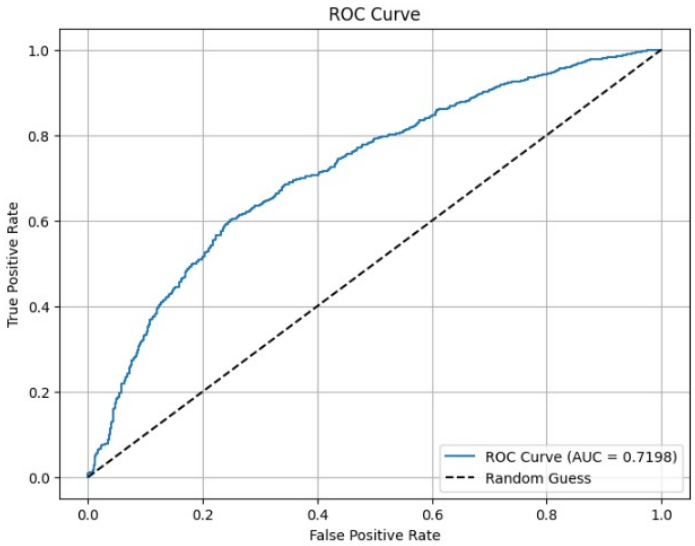

Figure 24: ROC curve for phishing detection.

4.2.2 Classification Report

The classification report summarizes the performance of the phishing detector:

```
Classification Report:
              precision    recall  f1-score   support

         0.0       0.64      0.77      0.70      1148
         1.0       0.71      0.487      0.63      1138

    accuracy                           0.67      2286
   macro avg       0.68      0.67      0.67      2286
weighted avg       0.68      0.67      0.67      2286
```

5. Discussion

5.1 Generative Models

- **Image Quality**: GANs produce sharper images compared to VAEs, which tend to generate slightly blurry images.
- **Latent Space**: VAEs provide a structured and interpretable latent space, while GANs do not explicitly enforce smoothness in the latent space.

5.2 Phishing Detection

- The VAE-based approach effectively distinguishes between legitimate and phishing URLs based on reconstruction errors.
- The choice of threshold significantly impacts performance. A higher threshold reduces false positives but increases false negatives.
- Challenges include handling imbalanced datasets and ensuring robust feature extraction from raw URLs.

5.3 Real-World Implications

The phishing detection model demonstrates the potential of generative models for anomaly detection in cybersecurity. Future work could involve:

- Incorporating additional features (e.g., domain age, SSL certificate information).
- Exploring other anomaly detection techniques, such as Isolation Forests or One-Class SVM.

6. Conclusion

In this assignment, I successfully implemented and compared GANs and VAEs for generating images of handwritten digits and fashion items. Additionally, I applied a VAE-based approach to detect phishing URLs in the phishing dataset. Both tasks highlight the versatility of generative models in addressing diverse machine learning challenges.

For future work we could involve exploring advanced architectures (e.g., Wasserstein GANs, β-VAEs) and integrating domain-specific knowledge into the phishing detection pipeline.

References

[1] J. Brownlee, "What Are Generative Adversarial Networks (GANs)?", Machine Learning Mastery. Available: `https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/`.

[2] S. G., "PyTorch GAN - Generating MNIST Digits", Kaggle. Available: `https://www.kaggle.com/code/sinclairg/pytorch-gan-generating-mnist-digits`.

[3] TechTarget, "What is a Variational Autoencoder (VAE)?". Available: https://www.techtarget.com/searchenterpriseai/definition/variational-autoencoder-VAE.

[4] Rkuo2000, "MNIST VAE", Kaggle. Available: https://www.kaggle.com/code/rkuo2000/mnist-vae.

[5] S. Work, "Web Page Phishing Detection Dataset", Kaggle. Available: https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset.