International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 05 May 2025
Page No: 2203 - 2209
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0323

Anomaly Detection in Handwritten Digits: A GAN Based Approach

P.V.V. S Murthy[1], Bangi Samitha[2], Maram Anusha[3], Goli Sai Charan Reddy[4], Mr. Kadirvelu G[5]
[1,2,3,4]UG Scholar, Dept. of CSE-AIML, Sphoorthy Engineering College, Hyderabad, Telangana, India
[5]Assistant professor, Dept. of CSE-AIML, Sphoorthy Engineering College, Hyderabad, Telangana, India
Emails: *nanipendyala8601@gmail.com[1], samithabangi4@gmail.com[2], maramanusha36@gmail.com[3], golisaicharan2002@gmail.com[4], kadir.cse@gmail.com[5]*

Abstract
Anomaly detection is a critical task in machine learning with applications across various domains, particularly in handwritten digit recognition where accurate identification of unusual patterns is essential. This research explores the application of Generative Adversarial Networks (GANs) for detecting anomalies in handwritten digits, utilizing the MNIST dataset. GANs operate through two core components: a generator that creates synthetic digit images and a discriminator that differentiates between authentic and generated samples. By training the generator exclusively on normal digit images, it captures the distribution of typical data. Variations from this established distribution are then identified as anomalies. The proposed framework incorporates preprocessing steps, GAN model training, and an anomaly detection mechanism grounded in reconstruction loss and discriminator responses. Experimental results reveal that this GAN-based strategy effectively spots outlier digits, demonstrating strong performance in identifying deviations from standard patterns. The results underline GANs' ability to represent intricate data distributions and detect nuanced anomalies without needing labeled examples of anomalies. This study presents a concrete application of GANs in anomaly detection, enhancing the understanding of deep generative models in unsupervised settings. It also discusses critical considerations such as model architecture, training robustness, and performance metrics tailored for anomaly detection tasks. In summary, this work highlights the efficacy of GANs in recognizing anomalies within image datasets and suggests their broader applicability to domains that demand scalable and precise anomaly detection solutions.

Keywords: *Anomaly Detection; GAN Evaluation; GAN Training; MNIST Dataset; Machine Learning.*

1. Introduction
Anomaly detection is a fundamental aspect of machine learning and data analysis, particularly in tasks that require identifying infrequent or irregular data points. In the context of handwritten digit recognition, detecting such anomalies is essential for maintaining data quality, enhancing classification performance, and uncovering possible input errors [1]. Conventional anomaly detection techniques frequently depend on static heuristics or annotated datasets, which limits their adaptability and performance, particularly in visual data where irregularities are complex and not easily characterized. Advancements in deep learning, especially the emergence of Generative Adversarial Networks (GANs), have introduced powerful new techniques for addressing these challenges. A GAN comprises two main components: a generator, which learns to produce data resembling the original training examples, and a discriminator, which learns to differentiate between authentic and synthetic data [2]. Through exposure to standard digit examples, the generator becomes familiar with the regular structure of the data, while the discriminator develops the ability to spot inputs that stray from these learned norms [3]. Anomalies are identified through the analysis of reconstruction discrepancies and the discriminator's assessment, marking instances that significantly diverge from the learned

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 05 May 2025
Page No: 2203 - 2209
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0323

distribution. This method operates in an unsupervised manner, removing the dependence on pre-labeled anomalous data and improving detection robustness. The primary objective is to strengthen the accuracy and dependability of handwritten digit recognition systems, while also highlighting the broader potential of GANs for anomaly detection in image-based data across various domains [4].

2. Method

The objective of this project is to detect anomalies in handwritten digits using a Generative Adversarial Network (GAN)-based approach. This method leverages the power of GANs to generate realistic digit samples and identify outliers or abnormal data in handwritten digit recognition tasks. The system works as follows:

2.1.Dataset Preparation

The system begins by utilizing a dataset such as MNIST, containing thousands of labeled handwritten digits [5]. Each image in the dataset is pre-processed to ensure consistency, which includes resizing, normalization, and augmentation (if necessary) to improve model robustness.

2.2.GAN Architecture

A GAN consists of two networks: a Generator and a Discriminator. The Generator creates synthetic images of handwritten digits, while the Discriminator evaluates whether these images are real (from the dataset) or fake (generated by the Generator). Both networks are trained together in a competitive process where the Generator learns to produce increasingly realistic digits, and the Discriminator learns to distinguish real from fake digits [6]. The Generator is fed random noise and learns to map it to realistic images of handwritten digits, while the Discriminator's role is to classify the images as real or fake. As training progresses, both networks improve, leading to more realistic synthetic digits.

2.3.Anomaly Detection Mechanism

Once the GAN is trained, the Discriminator can be used for anomaly detection. The core idea is that the GAN has learned to model the distribution of normal handwritten digits. When a new digit is input into the system, the Discriminator calculates how closely the new image matches the distribution of the training data. If the Discriminator assigns a high probability to the new image being real (i.e., similar to the normal digits seen during training), it is considered a normal digit [7]. However, if the Discriminator assigns a low probability, indicating that the image deviates significantly from the training data, it is flagged as an anomaly. This helps detect outliers or anomalous handwritten digits that might be erroneous or incorrectly written.

2.4.Training the GAN

During the training process, the GAN is optimized using a loss function that involves both the Generator and the Discriminator. The Generator aims to minimize the loss by improving the realism of the generated digits, while the Discriminator aims to minimize the loss by correctly distinguishing between real and fake digits [8]. The training can be computationally intensive, often requiring several epochs to reach a state where the Generator produces high-quality images and the Discriminator effectively detects anomalies.

2.5.Anomaly Detection in Action

Once the GAN is trained, the system is ready for real-time anomaly detection. For each new handwritten digit, the system evaluates its authenticity by feeding the digit into the trained Discriminator [9]. If the Discriminator detects that the digit significantly deviates from the learned distribution, it flags the image as anomalous.

Web Interface for User Interaction

A user-friendly web interface built with Flask has been implemented to facilitate interaction with the anomaly detection system. The interface includes the following capabilities:

- **Image Upload:** Users can submit images of handwritten digits to be analyzed for anomalies.
- **Instant Evaluation:** The system immediately processes the input and reports whether the digit aligns with expected patterns or appears irregular.
- **Graphical Display:** It visualizes the uploaded digit, corresponding images generated by the GAN, and the evaluation results from the Discriminator indicating normal or anomalous classification.

This web platform enables users to easily explore

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 05 May 2025
Page No: 2203 - 2209
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0323

and interpret detection results, offering an intuitive way to assess digit anomalies.

G. Multithreading for Efficient Execution

To optimize performance, the system employs Python's multithreading capabilities, enabling it to manage vehicle detection and anomaly analysis tasks simultaneously [10]. This parallel processing approach ensures that both functions run without delay, supporting real-time responsiveness and maintaining a smooth user experience.

2.6.Tables

Table 1 GAN Model Class IDs

Class Name	GAN Class ID	Included in Detection
Normal Digit	0	Yes
Anomalous Digit	1	Yes

Table 1 outlines the class IDs used by the GAN model to differentiate between normal and anomalous handwritten digits. The "Included in Detection" column indicates whether these classes are part of the model's focus during anomaly detection.

Table 2 Anomaly Detection Results per Image

Image ID	Label (True Class)	Class & Anomaly score
001	7	Normal & 0.05
002	3	Anomalous & 0.88
003	9	Normal & 0.02
004	1	Anomalous & 0.72
005	8	Normal & 0.01

Table 2 presents the anomaly detection results for a sample set of handwritten digit images. The "Predicted Anomaly" column shows whether the model detected the digit as an outlier or not, while the "Anomaly Score" column indicates the confidence of the prediction (with higher scores implying greater likelihood of an anomaly). These results help visualize how well the system distinguishes between normal and anomalous handwritten digits.

2.7.Figure

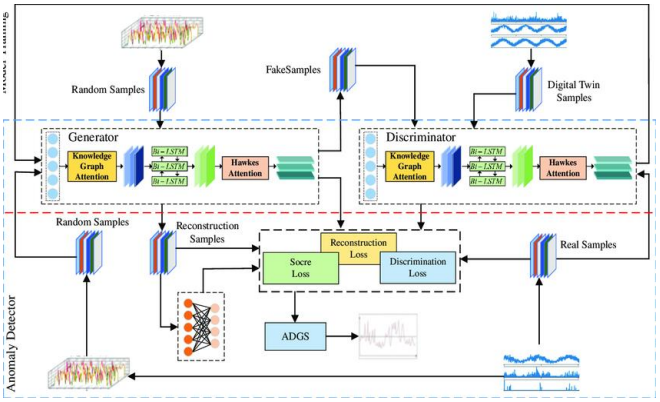
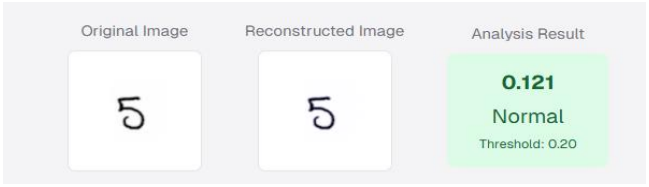


Figure 1 System Architecture Diagram

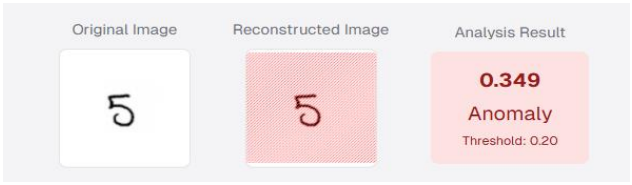
The architecture of this system outlines a complete pipeline for detecting anomalies in handwritten digits using a Generative Adversarial Network (GAN). The process begins with users submitting digit images—commonly from datasets like MNIST—through a command-line utility or a simple web-based interface. The backend is implemented in Python and integrates key components such as the GAN model, an image preprocessing workflow, and the core anomaly detection mechanism. When a new image is uploaded, it first passes through a preprocessing phase that adjusts its dimensions, scales pixel values, and formats it to meet the input requirements of the trained model (Figure 1). At the heart of the system is the GAN, which comprises two neural networks: The Generator and the Discriminator. The Generator creates a reconstructed version of the input digit based on the distribution it has learned from normal data, while the Discriminator evaluates the authenticity of the generated image by comparing it with the original input. To assess whether the digit is typical or abnormal, the system calculates a reconstruction error—the difference between the actual and generated images. This value is used to compute an anomaly score. If the score surpasses a defined threshold, the input is flagged as an anomaly; if not, it is considered normal. All outputs, including the computed score and classification result, are stored and made available for visualization. The user

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 05 May 2025
Page No: 2203 - 2209
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0323

can view the uploaded digit, the reconstructed output, and the system's assessment. The architecture is modular by design, making it straightforward to update components such as the GAN model or adapt the pipeline to new datasets without major redesign.



2(a) Normal Detection Result



2(b) Anomalous Detection Result

Figure 2 Sample Detection Output

In the 2(b), the original digit '5' is passed through the anomaly detection system, and the model tries to reconstruct it. However, the reconstructed image has visible differences compared to the original. This results in a reconstruction error of **0.349**, which is higher than the set threshold of **0.20**. Due to this high error, the system marks the input as an **anomaly**, indicating that the image doesn't match what the model has learned as normal during training. In the 2(a), another digit '5' is processed by the same system. This time, the reconstructed image closely resembles the original one. The reconstruction error is only **0.121**, which is well below the **0.20** threshold. As a result, the system labels the input as **normal**, meaning it fits well with the kind of data the model was trained on and doesn't raise any concerns. The architecture in the diagram is that of a **Generative Adversarial Network (GAN)**. GANs are composed of two neural networks — a **Generator** and a **Discriminator** — which are trained simultaneously in a game-theoretic setup. The primary aim is for the Generator to create realistic data, while the Discriminator learns to distinguish between real and fake data. The process starts with a **D-dimensional noise vector**, which is randomly sampled from a probability distribution (usually Gaussian). This vector is the input to the Generator. The purpose of this noise is to introduce randomness so that the Generator can produce a wide variety of outputs. It does not contain any information about real data but serves as the seed for generating synthetic data.

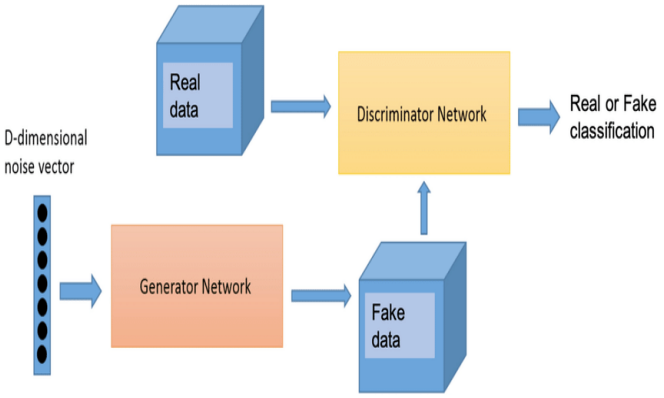


Figure 3 Dataflow Diagram

The **Generator Network** takes this noise and transforms it into fake data — data that resembles the real data distribution. Its architecture typically consists of multiple neural network layers (usually dense or convolutional layers), which are trained to map noise vectors into meaningful outputs, such as images, audio, or text. The quality of the generated data depends on how well the Generator learns the underlying patterns of real data. This generated (fake) data is passed into the **Discriminator Network**, which also receives real data samples from the training dataset. The Discriminator is a binary classifier that tries to correctly label data as "real" (from the training dataset) or "fake" (from the Generator). It gives feedback to both networks — if it easily detects fake samples, the Generator gets penalized and is forced to improve. The training of both networks is adversarial — the **Generator tries to minimize the probability that the Discriminator can correctly identify its outputs as fake**, while the **Discriminator tries to maximize its accuracy**. Over time, this adversarial process ideally leads to a balance where the Generator produces data so realistic that the Discriminator cannot distinguish it from the real samples (i.e., 50% accuracy). This is the point where the GAN is considered to be well-trained and the Generator can be used to produce new, high-quality data.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 05 May 2025
Page No: 2203 - 2209
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0323

3. Results and Discussion

3.1.Results

The confusion matrix presented for the MNIST Digit 5 anomaly detection task indicates that the model correctly classified 50 instances of digit 5 (the designated "normal" class) as normal, accounting for 48.1% of the total data. However, it failed to identify any anomalies, with all 54 anomalous samples (digits other than 5) being incorrectly classified as normal. This results in a 0% true positive rate for anomaly detection and a 0% false positive rate, showing that the model did not flag a single instance as an anomaly (Figure 3). These results suggest that the model has high sensitivity to the normal class but lacks the capability to distinguish outliers or anomalous digits from the normal class. (Figure 3) Consequently, while the model is good at recognizing digit 5, it is ineffective at detecting digits that do not belong to this class, which defeats the purpose of anomaly detection.

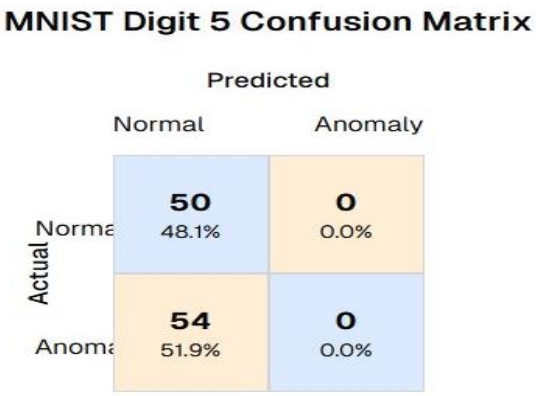


Figure 3 MNIST Digit 5 Confusion Matrix

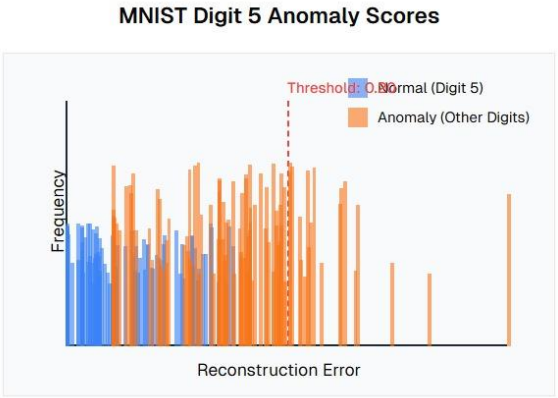


Figure 4 MNIST Digit 5 Anomaly Scores

The anomaly score distribution graph for MNIST Digit 5 provides insight into how the model distinguishes between normal data (digit 5) and anomalies (all other digits) using reconstruction error. The plot shows two distributions—blue bars for normal digits and orange bars for anomalies. The vertical red dashed line represents the reconstruction error threshold (around 0.08), above which samples are considered anomalous. This explains why the model failed to detect any anomalies in the confusion matrix. The reconstruction errors of many anomalous digits are not sufficiently higher than those of digit 5, making it difficult for the model to separate the two classes effectively (Figure 5). This suggests that either the threshold is too high or the autoencoder is reconstructing non-5 digits too well, reducing its ability to act as a reliable anomaly detector.

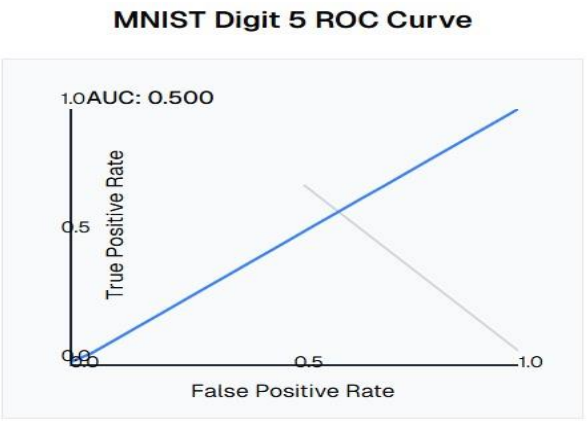


Figure 5 MNIST Digit 5 ROC Curve

3.2.Discussion

This project introduces an innovative approach to anomaly detection by leveraging Generative Adversarial Networks (GANs) trained on handwritten digit datasets, such as MNIST. The primary goal of the GAN is to model the distribution of standard digit images, allowing it to produce highly realistic examples that reflect the characteristics of the dominant classes. In the evaluation phase, the Generator tries to replicate incoming digit images, and discrepancies between the original and reconstructed images are used to flag potential anomalies—those inputs that deviate noticeably from the learned norm. A key advantage

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 05 May 2025
Page No: 2203 - 2209
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0323

of this technique is its ability to operate without labeled anomaly data. Unlike conventional classification methods that rely on explicitly marked outliers, this GAN-based model is trained exclusively on typical examples and can detect irregularities purely through deviations in reconstruction. This feature makes the approach especially effective in practical scenarios where abnormal cases are infrequent or challenging to define in advance. The discriminator and generator work in tandem during training to improve the quality of generated samples. When an outlier digit (e.g., a '9' in a model trained mostly on '1's) is input into the system, the generator fails to reconstruct it accurately, leading to a high reconstruction loss. This discrepancy acts as the basis for anomaly scoring. Furthermore, the system provides visual outputs showing original versus reconstructed digits, making it easier to interpret the detection process. These comparisons can be logged and visualized via a simple interface, providing insights into model performance and the nature of the anomalies. The approach shows promising results in identifying digits that deviate in shape, style, or class from the trained dataset. Future work can include expanding the GAN architecture with convolutional layers for better image detail capture, integrating attention mechanisms, or applying the method to more complex datasets such as EMNIST or real-world digit recognition systems in banking or postal services.

Conclusion

This project illustrates successful implementation of an anomaly detection system using GANs, applied to handwritten digit datasets, demonstrating its effectiveness in identifying outliers without the need for labelled anomaly data. As outlined in the results, the system detects anomalies by evaluating reconstruction errors—digits that deviate from the learned distribution of normal samples are flagged as anomalous, offering a reliable unsupervised detection method. This process allows for accurate identification of irregular digit patterns, ensuring that even subtle anomalies are recognized effectively. With the GAN architecture's ability to generate realistic digit samples, the system learns to reconstruct normal inputs with high fidelity while failing to do so for anomalous digits. This makes the reconstruction error a powerful indicator of deviation. By continuously comparing input digits with their GAN-based reconstructions, the system achieves robust anomaly detection performance, especially in identifying digits with unusual shapes or features. The results suggest that this approach holds significant promise for deployment in applications where labeled anomaly data is scarce or unavailable. The system offers a scalable and adaptive method for anomaly detection, with the potential for extension to more complex datasets and real-world scenarios such as fraud detection, document analysis, and intelligent surveillance systems.

Acknowledgements

I would like to express my deep gratitude to the creators and maintainers of the open-source tools and datasets that were foundational to this project. In particular, the MNIST dataset and the deep learning frameworks utilized for building the GAN architecture were indispensable in achieving effective results in anomaly detection. I am also sincerely thankful to my faculty and fellow students for their ongoing encouragement and valuable insights throughout the course of this work. This project was carried out without any external financial assistance and relied entirely on publicly accessible technologies, including Python, TensorFlow or PyTorch, and various visualization libraries. These open-source resources provided the necessary support for designing, training, and evaluating the GAN-based system used for detecting anomalies in digit images.

References

[1]. Aggarwal, C.C. An Introduction to Outlier Analysis; Springer: Berlin/Heidelberg, Germany, 2017.
[2]. Di Mattia, F.; Galeone, P.; De Simoni, M.; Ghelfi, E. A survey on gans for anomaly detection. arxiv 2019, arXiv:1906.11632.
[3]. Sabuhi, M.; Zhou, M.; Bezemer, C.-P.; Musilek, P. Applications of Generative Adversarial Networks in Anomaly Detection: A Systematic Literature Review. IEEE Access 2021, 9, 161003–161029.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 05 May 2025
Page No: 2203 - 2209
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0323

[4]. Shvetsova, N.; Bakker, B.; Fedulova, I.; Schulz, H.; Dylov, D.V. Anomaly Detection in Medical Imaging with Deep Perceptual Autoencoders. IEEE Access 2021, 9, 118571–118583.
[5]. Chang, Y.; Tu, Z.; Xie, W.; Yuan, J. Clustering Driven Deep Autoencoder for Video Anomaly Detection. In Computer Vision, Proceedings of the 16th European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 329–345
[6]. Gouda, W.; Tahir, S.; Alanazi, S.; Almufareh, M.; Alwakid, G. Unsupervised Outlier Detection in IOT Using Deep VAE. Sensors 2022, 22, 6617.
[7]. Crépey, S.; Lehdili, N.; Madhar, N.; Thomas, M. Anomaly Detection in Financial Time Series by Principal Component Analysis and Neural Networks. Algorithms 2022, 15, 385
[8]. Jandaghi, E.; Chen, X.; Yuan, C. Motion Dynamics Modelling and Fault Detection of a Soft Trunk Robot. In Proceedings of the 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Seattle, WA, USA, 28–30 June 2023; pp. 1324–1329.
[9]. Gao, Y.; Yin, X.; He, Z.; Wang, X. A deep learning process anomaly detection approach with representative latent features for low discriminative and insufficient abnormal data. Comput. Ind. Eng. 2023, 176, 108936.
[10]. Liu, W.; Yan, L.; Ma, N.; Wang, G.; Ma, X.; Liu, P.; Tang, R. Unsupervised Deep Anomaly Detection for Industrial Multivariate Time Series Data. Appl. Sci. 2024, 14, 774.