GENERATIVE ADVERSARIAL NETWORKS FOR UNSUPERVISED ANOMALY DETECTION IN ENERGY TIME SERIES DATA

Praveen P. Handigol
Robert Bosch Centre for Cyber-Physical Systems
Indian Institute of Science
Bengaluru, India
praveenph@iisc.ac.in

Pandarasamy Arjunan
Robert Bosch Centre for Cyber-Physical Systems
Indian Institute of Science
Bengaluru, India
samy@iisc.ac.in

ABSTRACT

The growing energy consumption in buildings, which accounts for a significant portion of global energy use, highlights the need for efficient energy management solutions. Smart metering systems have facilitated the collection of vast amounts of energy time-series data, offering valuable insights into consumption patterns. However, anomalies in this data, resulting from issues such as faulty hardware or improper operational practices, can lead to significant energy wastage. This study investigates the use of deep learning models, specifically Generative Adversarial Networks (GANs), for unsupervised anomaly detection in energy time-series data. Specifically, we employ a 1D Deep Convolutional GAN (DCGAN) to model normal consumption patterns and detect anomalies by leveraging the reconstruction error. Additionally, a 1D CNN Autoencoder and a Convolutional Variational Autoencoder are also trained and the results are compared with Isolation Forest and Local Outlier Factor which serve as baseline models. These methods are evaluated on the LEAD 1.0 dataset, consisting of hourly electricity readings from 200 commercial buildings. The results demonstrate that deep learning methods perform better compared to traditional unsupervised machine learning methods in identifying sequential anomalies. DCGAN trained with conventional loss function achieved the highest F1 score of 0.697 when compared to the rest of the methods.

1 INTRODUCTION

Buildings consume significant energy, accounting for approximately 40 % of total energy usage worldwide [Cao et al. (2016), Shaikh et al. (2014)]. This demand is steadily increasing and is expected to rise by 28 % by 2040 [Conti et al. (2016)]. As a result, minimizing energy consumption and reducing related emissions in buildings is crucial to achieve sustainability goals. The adoption of smart meters has become widespread in recent years in order to meet the growing demand for effective energy management.

The widespread adoption of smart metering systems has resulted in vast amounts of energy time-series data, offering valuable insights into building energy consumption patterns [Pan & Zhang (2020)]. However, anomalies in this data, such as faulty hardware or improper operational practices, can significantly impact energy management. Studies show that in commercial buildings, these issues can contribute to 15-30 % of energy wastage [Katipamula & Brambley (2005), Schein et al. (2006)]. Efficient energy management requires recognizing normal consumption patterns and identifying deviations to minimize operational costs. Failure to detect anomalies can lead to inaccurate forecasts[Wang et al. (2019)] and unreliable predictive models, making anomaly detection essential for effective energy management[Himeur et al. (2021)].

Traditional methods for anomaly detection, such as statistical techniques, clustering algorithms, and linear models, often struggle with the complexity of time series data . Statistical methods typically rely on assumptions about the distribution of data, which may not hold in real-world scenarios, especially when data exhibits complex, non-linear patterns [Chandola et al. (2009)]. Traditional machine learning techniques, including decision trees or support vector machines, are often limited by their inability to capture long range dependencies and temporal correlations inherent in time series data [Blázquez-García et al. (2020)]. Clustering algorithms may fail to detect subtle anomalies, as

they focus on grouping similar data points, overlooking minor deviations that may indicate critical anomalies [Braei & Wagner (2020)].

Deep learning-based methods have gained popularity for anomaly detection in time series due to their ability to capture complex, non-linear patterns in data[Blázquez-García et al. (2020), Braei & Wagner (2020)]. Generative models, particularly Generative Adversarial Networks (GANs) [Goodfellow et al. (2014)], have gained attention for anomaly detection in time series due to their ability to learn complex data distributions. GANs are trained on normal subsequences, with the generator mapping random noise to realistic data points. During reconstruction, the generator's latent space is adjusted using gradient descent to minimize the error between the input and its reconstruction. The difference is used to calculate an anomaly score, with higher scores indicating anomalies. This approach effectively identifies deviations from normal patterns, enhancing anomaly detection in time series data.

In this paper, we present a comparative analysis of deep learning architectures for unsupervised anomaly detection in building energy time-series data. Leveraging hourly electricity consumption patterns from 200 commercial buildings in the LEAD 1.0 dataset, we implement 1D Deep Convolutional GANs (DCGAN), 1D CNN Autoencoder, and Convolutional Variational Autoencoder to identify operational inefficiencies. Our framework employs inverse mapping through latent space optimization to compute anomaly scores, evaluated against traditional methods like Isolation Forest and Local Outlier Factor with a 24-hour tolerance window. Experimental results demonstrate DCGAN's superior performance (F1 score: 0.697), highlighting its effectiveness in detecting sequential anomalies compared to baseline approaches.

2 METHODOLOGY

2.1 LEAD DATASET

This study utilizes the LEAD 1.0 dataset[Gulati & Arjunan (2022)], which provides hourly electricity meter readings from commercial buildings over a period of up to one year. The dataset, open sourced and available on GitHub, contains data from 200 buildings, with each building contributing approximately 8,747 data points. Anomaly annotations are included, identifying anomalous timestamps within each building's time series. Around 2% of the overall data is classified as anomalous. The percentage of anomalies per building varies, with the mean anomaly rate per building being 2.13 % ± 1.15%. Refer Table 2 in the appendix for more details about the dataset.

2.2 DATA-PREPROCESSING AND MODEL INPUT

For each annual meter reading time series in the dataset, missing readings are first removed. The series is then divided into 25 contiguous, non-overlapping time segments, each with point-wise anomaly annotations. Segments without anomalies are used for training, while those containing anomalies are reserved for testing. Each segment is normalized to the range [-1,1], which is essential as the generator model employs a tanh activation function at its output, producing values within this range. A segment of time series data is processed using a rolling window of fixed length to generate model inputs. Let $S = \{x_1, x_2, \ldots, x_{n+w-1}\}$ represent a segment of the time series. Using a window size w, a collection of n overlapping time subsequences, X is generated as follows:

$$X = \{(x_1, \ldots, x_w), (x_2, \ldots, x_{w+1}), \ldots, (x_n, \ldots, x_{n+w-1})\} \tag{1}$$

These subsequences are then further processed to form the model input, resulting in a tensor with the shape $(\text{Batch size}, \text{features } (1), w)$.

2.3 1D-DCGAN ARCHITECTURE

The 1D Deep Convolutional GAN (DCGAN) architecture is inspired from the work Radford et al. (2016). The network architecture used in this study consists of a generator and discriminator, each with three symmetrical hidden layers, as illustrated in Figure 1. The generator employs a series of transposed convolutional layers to upsample the input noise vector into a 1D time series output. It begins with a 1D transposed convolutional layer, which has 100 input channels (representing the noise vector) and 256 output channels, followed by ReLU activation. This is followed by three more transposed convolutional layers with progressively smaller output channels (128, 64, and 1), and ends with a Tanh activation function.

The discriminator comprises four 1D convolutional layers, each followed by Leaky ReLU activation. While batch normalization is commonly used in intermediate layers of a DCGAN to stabilize training and enhance convergence, it is typically omitted from the generator's output layer and the discriminator's input and output layers. The latent space is composed of 100 dimensions, with the prior distribution p_z assumed to be an independent standard multivariate Gaussian over this latent space.

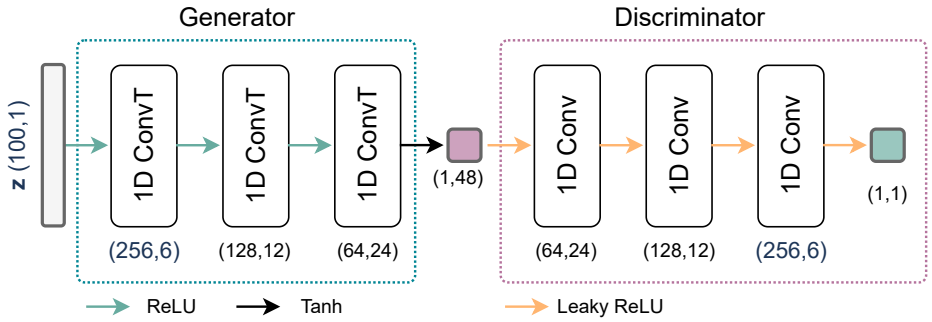

Figure 1: 1D DCGAN Architecture for Anomaly Detection

2.4 Inverse Mapping

Inverse mapping aims to estimate the input that generated a specific output. In deep learning, this involves reversing the forward mapping learned by a neural network. In this work, we focus on applying gradient descent in the latent (input) space. To invert a given query data X, we begin with random noise in the latent space, and iteratively update it using gradient descent based on a differentiable reconstruction loss $L(X, G(Z))$. After several iterations, Z is considered an approximation of $G^{-1}(X)$.

2.5 Anomaly Score

The reconstruction loss is computed using MSE, which also serves as a direct method for calculating the anomaly score. The goal extends beyond determining whether a suitable input exists to generate a sample resembling X. We aim to ensure that the input can be generated by sampling from the latent space distribution p_z, which is Gaussian-centered at the origin. The overall anomaly score is given by:

$$\text{Anomaly Score}(X) = \alpha \cdot \ L(X, G(Z)) + \beta \cdot \|Z\|^2 \tag{2}$$

where α and β regulate the influence of each subpart. The input subsequence is marked as anomalous if the anomaly score exceeds a certain threshold.

2.6 Anomaly Detection

Comparing the ground truth with the predictions can be a difficult task. To tackle this issue, a method similar to the one proposed by [Gu & Jazizadeh (2022)] is employed. The model's detected anomalous time windows (subsequences) are first converted into anomalous timestamps for comparison with the ground truth. This process involves several steps. First, a test segment is selected for evaluation. Next, overlapping subsequences are created using a specified window size w. Anomaly scores are then computed for each subsequence. If the anomaly score exceeds a predefined threshold, the corresponding middle timestamp of the subsequence is marked as critical points. Following this, Kernel Density Estimation (KDE) is applied to generate a distribution over the test segment for the critical points, and the density is scaled to range between 0 and 1. Finally, any points in the scaled KDE that exceed a certain threshold are identified as predicted anomalies, and their corresponding timestamps are marked accordingly.

A limitation of this approach is the reduced KDE values at the beginning and end of the segment. This reduction occurs because the midpoint of an anomalous sub-sequence is marked as a critical point, leading to fewer critical points at the segment's boundaries. To address this, the time series is augmented by adding half the window size from the previous and subsequent segments, ensuring more accurate anomaly detection at the edges.

2.7 Model Evaluation

While anomaly labels in time series data are typically assigned as single points, sequential anomalous events often extend across intervals, making it challenging to pinpoint exact onset times. To accommodate this temporal ambiguity, relaxation tolerance parameter r_t is incorporated when assessing detection accuracy. Therefore, as long as the model's prediction is near the labelled anomaly, it should not be penalized. Precision, recall, and F1 score are used to evaluate the model, with adjustments to the calculations of True Positives (TP), False Negatives (FN), and False Positives (FP) based on the tolerance.

Table 1: Average metrics over 200 buildings for different methods with tolerance r_t of 24 hours

Model	Precision	Recall	F1 Score
DCGAN (Conventional)	0.769	0.696	0.697
DCGAN (Wasserstein Loss)	0.764	0.660	0.662
Convolutional Variational Autoencoder	0.787	0.649	0.665
1D CNN Autoencoder	0.785	0.650	0.655
Local Outlier Factor	0.525	0.849	0.620
Isolation Forest	0.466	0.589	0.472

$$\text{TP: } |d - p_{\text{closest}}| \leq r_t \tag{3}$$

$$\text{FN: } |d - p_{\text{closest}}| > r_t \tag{4}$$

$$\text{FP: } |p - d| > r_t \tag{5}$$

where d represents the location of a labelled ground truth anomaly, and p_{closest} denotes the nearest predicted anomaly. Ground truth anomalies with a predicted anomaly within the tolerance range are considered TP. Anomalies without a nearby prediction are classified as FN. An FP is an anomaly prediction that does not correspond to any ground truth anomaly within its vicinity, where p is the predicted anomaly location and d_{closest} is the closest actual annotated anomaly location.

3 Experimental Setup and Results

All model implementation is carried out using PyTorch, a widely used deep learning library in Python. In the experimental setup, the Adam optimizer is employed for training all the models, with a beta value of 0.5 and a learning rate of 0.0002. The WGAN's "ncritic" parameter is set to 5, and the clipping value is 0.01. A batch size of 128 is used throughout the training. The analysis focuses on sub-sequences with a window size of 48 and a latent space of 100 dimensions. The models are trained for 200 epochs. Evaluation hyperparameters for anomaly scores are fine-tuned based on the model's performance on the test set. Bayesian optimization is employed, where the evaluation hyperparameters serve as the inputs for the black-box optimization function. The F1 score, resulting from the set hyperparameters, is used as the objective value to optimize. The reproducible code and data are released in our GitHub repository[1].

Experiments are conducted by first training models for each building in the dataset of 200 buildings. Reconstruction is performed using Euclidean distance (MSE) over 1000 gradient descent iterations, and reconstruction errors are calculated. Anomalous timestamps are identified using the method described previously. The evaluation is carried out with a tolerance of $r_t = 24$.

Table 1 summarizes the results, with deep learning methods performing better when compared to traditional unsupervised machine learning methods. DCGAN variants and autoencoder architectures achieved superior balance between precision and recall, with F1 scores ranging from 0.655 to 0.697. The Wasserstein loss DCGAN configuration showed marginally lower performance (F1=0.662) compared to conventional DCGAN (F1=0.697), though prior research suggests Wasserstein variants may offer improved training stability. The results suggest convolutional architectures effectively capture sequential patterns in the input data distribution.

4 Conclusion and Future Work

This study demonstrates that deep learning approaches, particularly DCGAN and autoencoders, effectively detect anomalies in building energy consumption data, outperforming traditional methods like Isolation Forest. The evaluation framework, incorporating a temporal tolerance parameter ($r_t = 24$), accounts for annotation ambiguities in time-series anomalies. Deep Learning models excel at capturing complex patterns in the LEAD 1.0 dataset, enabling accurate identification of operational inefficiencies contributing to energy waste.

Future research could explore implementing new architectures and hybrid architectures combining GANs and autoencoders. Investigating the correlation between detected anomalies and actual energy savings would enhance practical applicability. Testing the model on a larger set of buildings and incorporating real-time sequential anomaly detection would enhance scalability and practical

[1] https://github.com/AI-IoT-Lab/gan-for-energy-anomaly-detection-ccml25

applicability. Extending the framework to multivariate time series and addressing class imbalance in anomaly distributions are additional promising directions.

ACKNOWLEDGMENTS

We would like to thank the providers of the LEAD 1.0 dataset for making the data available for research. We also acknowledge the contributions made by Hardik Prabhu towards this project. Special thanks to our colleagues at the Robert Bosch Centre for Cyber-Physical Systems for their guidance and feedback throughout this work.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. URL https://arxiv.org/abs/1701.07875.

Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A. Lozano. A review on outlier/anomaly detection in time series data, 2020. URL https://arxiv.org/abs/2002.04236.

Mohammad Braei and Sebastian Wagner. Anomaly detection in univariate time-series: A survey on the state-of-the-art, 2020. URL https://arxiv.org/abs/2004.00433.

Xiaodong Cao, Xilei Dai, and Junjie Liu. Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade. *Energy and Buildings*, 128:198–213, 2016. ISSN 0378-7788. doi: https://doi.org/10.1016/j.enbuild.2016.06.089. URL https://www.sciencedirect.com/science/article/pii/S0378778816305783.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL https://doi.org/10.1145/1541880.1541882.

John Conti, Paul Holtberg, Jim Diefenderfer, Angelina LaRose, James T. Turnure, and Lynn Westfall. International energy outlook 2016 with projections to 2040. Technical report, USDOE Energy Information Administration (EIA), Washington, DC (United States). Office of Energy Analysis, 05 2016. URL https://www.osti.gov/biblio/1296780.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Yueyan Gu and Farrokh Jazizadeh. Degan: Time series anomaly detection using generative adversarial network discriminators and density estimation, 2022. URL https://arxiv.org/abs/2210.02449.

Manoj Gulati and Pandarasamy Arjunan. Lead1.0: a large-scale annotated dataset for energy anomaly detection in commercial buildings. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, e-Energy '22, pp. 485–488, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393973. doi: 10.1145/3538637.3539761. URL https://doi.org/10.1145/3538637.3539761.

Yassine Himeur, khaled Ghanem, Abdullah Alsalemi, Faycal Bensaali, and Abbes Amira. Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. *Applied Energy*, 287:116601, 04 2021. doi: 10.1016/j.apenergy.2021.116601.

Srinivas Katipamula and Michael R Brambley. Methods for fault detection, diagnostics and prognostics for building systems - a review part i. *HVAC & R Research, 11(1):3-25*, 11(1), 01 2005. doi: 10.1080/10789669.2005.10391123. URL https://www.osti.gov/biblio/15011268.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Yue Pan and Limao Zhang. Data-driven estimation of building energy consumption with multi-source heterogeneous data. *Applied Energy*, 268:114965, 2020. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2020.114965. URL https://www.sciencedirect.com/science/article/pii/S0306261920304773.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016. URL https://arxiv.org/abs/1511.06434.

Jeffrey Schein, Steven T. Bushby, Natascha S. Castro, and John M. House. A rule-based fault detection method for air handling units. *Energy and Buildings*, 38(12):1485–1492, 2006. ISSN 0378-7788. doi: https://doi.org/10.1016/j.enbuild.2006.04.014. URL https://www.sciencedirect.com/science/article/pii/S0378778806001034.

Pervez Hameed Shaikh, Nursyarizal Bin Mohd Nor, Perumal Nallagownden, Irraivan Elamvazuthi, and Taib Ibrahim. A review on optimized control systems for building energy and comfort management of smart sustainable buildings. *Renewable and Sustainable Energy Reviews*, 34:409–429, 2014. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2014.03.027. URL https://www.sciencedirect.com/science/article/pii/S1364032114001889.

Yi Wang, Qixin Chen, Tao Hong, and Chongqing Kang. Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid*, 10:3125–3148, 05 2019. doi: 10.1109/TSG.2018.2818167.

APPENDICES

A DESCRIPTION OF DATASET

This study utilizes the LEAD 1.0 dataset[Gulati & Arjunan (2022)], which provides hourly electricity meter readings from commercial buildings over a period of up to one year. The dataset, open sourced and available on GitHub, contains data from 200 buildings, with each building contributing approximately 8,747 data points. data points. Anomaly annotations are included, identifying anomalous timestamps within each building's time series. The dataset encompasses buildings with 12 different primary usage types and varies significantly in size, from 898 to 484,376 square feet. As a result, energy consumption also varies widely, with average meter readings ranging from 0.786 to 4657.14, and an overall standard deviation of 396.

Description	**Value**
Meter readings frequency	Hourly for 1 year
Total meter readings (datapoints)	1749494
Percentage of missing meter readings	6%
Average meter readings (datapoints) per building	8747 ± 153
Average meter readings value across the entire dataset (in kW h)	179.9 ± 395.91
Building with the lowest average meter reading (in kW h)	0.786
Building with the highest average meter reading (in kW h)	4657.14
Percentage of anomalies in the entire dataset	2%
Mean percentage of anomalies per building	2.13 ± 1.15 %
Highest percentage of anomalies present in a building	8.82%
Lowest percentage of anomalies present in a building	0.5%
Number of buildings	200
Types of buildings based on primary usage	12
Building with the largest size (area in sq. feet)	484376
Building with the smallest size (area in sq. feet)	898

Table 2: Summary of the dataset.

B SIMPLE GAN TRAINING

A Generative Adversarial Network (GAN)[Goodfellow et al. (2014)] consists of two components: a generator G and a discriminator D, which operate in a competitive framework. The goal of a GAN is for the generator to learn to create data that closely resembles real samples from a target distribution, without directly modeling that distribution. The generator receives random noise as input and generates synthetic data aimed at mimicking real data. Meanwhile, the discriminator is trained to differentiate between real data from the actual dataset and fake data produced by the generator. The following shows the value function $V(G, D)$ for the adversarial game played between these two networks.

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [1 - \log D(G(z))] \tag{6}$$

C GAN TRAINING WITH WASSERSTEIN LOSS

Wasserstein GAN (WGAN) [Arjovsky et al. (2017)] introduces an alternative training approach to improve the stability and reliability of GAN training. It utilizes Wasserstein distance to quantify the difference between two probability distributions. This method addresses common challenges such as vanishing gradients and mode collapse, making it a preferred choice for improving the training efficiency and performance of GAN models. The generator and discriminator are required to optimize the following min-max function:

$$V(G, D) = \min_G \max_{D \in \mathcal{L}_{\text{Lip}}(K)} \mathbb{E}_{x \sim p_{\text{data}}(x)}[D(x)] - \mathbb{E}_{y \sim p_g(y)}[D(y)] \tag{7}$$

It is crucial to understand that in WGAN, the discriminator is no longer restricted to being a classifier; it can be any function, provided it satisfies the Lipschitz constraint. To enforce this constraint during training, gradient clipping is employed. Furthermore, the discriminator's loss function can be used as a measure of convergence. In WGAN, the absolute value of the discriminator's loss approximates the Wasserstein distance between p_{data} and p_g.

D 1D CNN AUTOENCODER

The 1D Convolutional Autoencoder architecture consists of an encoder and a decoder, designed to learn compact representations of sequential data. The decoder has the same architecture of the 1D DCGAN Generator where latent vector z is passed as input to the decoder and the output obtained in time series data of window size w. The encoder is the same mirror image of the decoder which takes in time series data of window size w as input and compresses it to a latent respresentation of 100 dimensions.

E CONVOLUTIONAL VARIATIONAL AUTOENCODER

The Variational Autoencoder (VAE) [Kingma & Welling (2022)] architecture consists of an encoder, decoder, and a reparameterization trick for latent space sampling. The encoder uses four 1D convolutional layers to reduce the input size, with ReLU activations and batch normalization applied after each layer. The final convolution outputs the latent representation, which is then used to compute the mean and log-variance via two fully connected layers. The decoder uses four transposed convolutional layers to reconstruct the input from the latent space, with ReLU activations, batch normalization, and a Tanh activation at the output layer.

The reparameterization trick generates a latent vector z from the mean and log-variance, enabling backpropagation through the stochastic sampling process. The VAE is trained using a loss function that combines reconstruction loss (mean squared error) and the KL divergence between the learned latent distribution and a standard Gaussian prior.

$$L(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) = -D_{\text{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{z})) + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})}[\log p_{\boldsymbol{\phi}}(\boldsymbol{x}^{(i)}|\boldsymbol{z})] \tag{8}$$