Detecting Pollution Anomalies in Multivariate Air Quality Datasets with Unsupervised Machine Learning

Ejieta Julius Owhe., Micheal Opeyemi Durodola

Teesside University

DOI : https://doi.org/10.51584/IJRIAS.2025.100700097

Received: 07 July 2025; Accepted: 13 July 2025; Published: 15 August 2025

ABSTRACT

Since air pollution affects both health and the environment, detecting unusual events is very important. The study tests the effectiveness of unsupervised machine learning methods—Isolation Forest, DBSCAN and Autoencoders—on air quality samples. The models were analyzed using multivariate data obtained from the UCI Air Quality Repository, OpenAQ and the U.S. EPA to see how well they can spot unusual levels of air pollution, mainly focusing at concentrations of carbon monoxide (CO).

Considerable outcomes were achieved by the Isolation Forest and Autoencoders, with each one discovering 67 significant anomalies in tune with seasonal pollution trends. These methods' toughness towards a lot of noisy and numerous data and ability to see strange patterns from cycles, made them a good choice for ecological monitoring. On the other hand, DBSCAN only detected one anomaly which means it worked poorly in this case because it requires specific settings and there is not much data over time. A review of the correlation heatmap pointed out that pollutant variables are closely related, proving that multivariate methods work better than univariate analysis. This research can guide selection of the best models for real-time monitoring of air quality. Since the study found significant results, they can be applied to public health, future urban development and making policies about environmental protection.

Keywords: Air pollution, Carbon monoxide, Machine learning, Isolation Forest, Autoencoders, DBSCAN

INTRODUCTION

As industrialization and cities grow rapidly, one of the biggest environmental issues now is air pollution. Global health organizations such as the World Health Organization have repeatedly said that pollutants like particulate matter, carbon monoxide, ozone and nitrogen dioxide are dangerous for everyone (Sharma, Jain et al. 2013). Almost six and half million deaths worldwide in 2019 were connected to ambient air pollution, proving negative effects on human wellbeing (Ofremu, Raimi et al. 2024). Since these statistics are so concerning, it is very important to quickly detect and monitor changes in air pollution levels to protect public health and the environment.

Many air quality monitoring sensors and IoT devices now allow to obtain live and complex records from different regions. Usually, data streams for air quality will show PM2.5, PM10, CO, NO_2 and O_3 concentrations on an exact-minute basis (Meo, Salih et al. 2024, Meo, Salih et al. 2024). These datasets are made up of many variables and they show relationships both over time and space. An illustration is that neighboring places may see similar changes in pollution because of shared air conditions, whereas seasonal and daily changes bring periodic variations to pollutant balances. The characteristic cycles in air quality data allow for advanced modeling and making it easy to find unusual events in the data(Dhingra, Madda et al. 2019, Idrees and Zheng 2020).

Several domains, including cybersecurity, banks and transport, have implemented anomaly detection techniques recently. When dealing with air quality, the task is to figure out instances where something important—like NO_2 from factories or a rain-induced drop in PM2.5—is different from usual (Koffi, N'Gouran et al. 2019). Signs of such pollution are vital to act quickly, learn where pollution is coming from and protect

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

people's health. The disaster of Great Smog in London in 1952 made authorities improve how they observe air pollution and send warnings (Zhang, Liu et al. 2014).

Many air quality monitoring systems have depended on models that are statistical and based on rules for finding anomalies (Imam, Adam et al. 2024). Baselines using statistical tools like moving averages and z-scores are simple but may not be enough to describe complex signals or how sensors interact. Such methods as k-means or hierarchical clustering have shown potential in locating outlier groups, but they are easily disrupted by noise and need to specify the number of clusters in advance (Van Zoest, Stein et al. 2018). On the other side, machine learning methods, mainly unsupervised ones are being used now as they can model data with many variables and generalize from what they learn in advance.

For all these advancements, there are still some limitations. Much of the literature looks at univariate data or depends a lot on the AQI, meaning that details about specific pollutants may be missed (Suman 2020). In addition, a lot of methods analyze data from each monitoring post individually, ignoring the possibility that related locations may help detect outbreaks more accurately (Zhengjing Ma 2021). Catching such anomalies only when they stand out against a background, not just by their lone value, is still a challenge for modern AI (Claire Heffernan 2022).

Since unsupervised machine learning models can identify changes in datasets without being taught, this paper suggests analyzing Isolation Forest, DBSCAN and Autoencoders to detect problems in air quality data. They are chosen for being strong, adaptable to various situations and able to deal with high-dimensional data full of noise, without labeling anomaly records (Liu, Ting et al. 2008, Claire Heffernan 2022). Temporal changes and mapped patterns are used from datasets found on the UCI Machine Learning Repository, OpenAQ and the U.S. Environmental Protection Agency.

In particular, we prepare and adjust the air quality readings, apply each model to highlight outliers and examine their records with both statistics and manual checks. Isolation Forests find abnormalities by separating data and checking the level of isolation, whereas DBSCAN picks out crowded groups and flags unfilled regions as unusual and Autoencoders point out variations by learning regularly seen patterns and noticing when they are broken in the reconstruction.

The most important contributions of this research include:

- We gather and preprocess various real-world air quality datasets so they can be used collectively for unsupervised anomaly detection within different weeks and regions.
- We demonstrate and compare the performance of Isolation Forest, DBSCAN and Autoencoders to detect pollution anomalies and point out how each outperforms or falls short.
- We examine the discovered anomalies by giving detailed quantitative and qualitative results, comparing them with past pollution events and checking if these models are appropriate for live monitoring.

This work will provide better methods for detecting air quality issues which can help improve public health, city planning and help set rules for environmental policy.

LITERATURE REVIEW

Ambient air pollution is considered a big risk to the health of people everywhere. As stated by the World Health Organization, the key pollutants $PM_{2.5}$, PM_{10}, CO, NO_2, and O_3 are important factors in the cause of both serious and fatal illnesses (Sharma, Jain et al. 2013). In their research, Ofremu et al. (2024) mention that air pollution resulted in almost 6.5 million deaths across the world in 2019 wellbeing (Ofremu, Raimi et al. 2024). As we can see from these numbers, such systems are needed as soon as possible to instantly identify and alert deviations in pollutants to address health and nature risks. Using IoT and networks such as UCI, OpenAQ, and datasets from the EPA, these frameworks non-stop monitor the levels of $PM_{2.5}$, PM_{10}, CO, NO_2, O_3, measuring them at time spans as short as a minute (Meo, Salih et al. 2024, Meo, Salih et al. 2024)Such data

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

tend to show autocorrelation, and in addition, regions next to each other are strongly influenced by one another in terms of pollution. According to Idrees & Zheng (2020) and Dhingra et al. (2019), it is necessary to observe such relevant variables to identify pollution outbreaks that stand out (Dhingra, Madda et al. 2019, Idrees and Zheng 2020).

The early methods to spot changes in air quality relied on traditional statistics, for instance, moving averages, z scores, control charts, and thresholds that were consistent with standards of the Air Quality Index (Imam, Adam et al. 2024). The reason they are easy to understand and to apply is that they usually struggle to represent non-linear connections between pollutants or handle uneven noise. Still, K means and hierarchical clustering detect groups that do not fit in, yet they may be confused by random variation and require you to know beforehand how many clusters are present in the data (Kaushik, Mathur et al. 2014). These methods do a good job at spotting single anomalies, but they are not effective with complex situations that combines a lot of variables and happen over time.

The rising interest in unsupervised ML models—Isolation Forest, DBSCAN, and Autoencoders—comes from the fact that they can model complicated patterns in unlabeled air quality data with high dimensions (Liu, Ting et al. 2008, Claire Heffernan 2022). Use of these techniques continues to expand past air quality into areas such as cybersecurity, energy systems, IoT monitoring, and environmental science, proving their usefulness in multiple places.

Isolation Forest is appreciated for being both fast and capable of handling big datasets. It randomly singles out data into trees and marks anything that has an unusually short average between the root and leaves in those trees (Laskar, Huang et al. 2021). DBSCAN technique uses density to define clusters and regards regions with low data density as unusual and marks them as anomalies. Using it means we can spot anomalies with unusual shapes, however, finding the right parameters can be difficult (Xiong, Chen et al. 2012). Autoencoders, both normal and time-based ones, produce the correct patterns for normal data and recognize anomalies when there are high mistakes in reconstruction (Bouman and Heskes 2025).

The techniques of isolation forests, DBSCAN, and autoencoders are used in network security, fraud detection of credit cards, flight anomaly detection, identification of abnormal behavior in IoT or indoor air quality systems, respectively. For that reason, without manual training, deep networks can model complex changes that happen in space and time, finding various anomalies, including those that impact the context.

Current research points out that using hybrid ways that include 3D spatial and temporal modeling allows detection of errors in time and space, even for data that does not have labeled examples (Guo, Lin et al. 2019). In a similar way, graph-regularized autoencoders make use of how sensors are placed near each other for detecting anomalies in high-dimensional data (Feng, Chen et al. 2022). At the same time, advanced iForest versions, such as Deep Isolation Forest, first map all the data using randomly initialized neural networks before splitting it, which is helpful for both non-linear and high-dimensional anomalies (Xu, Pang et al. 2023).

Interest in using data analysis for air quality monitoring is rising, but some important issues still exist in current research. A key problem is that a lot of research is based only on using simple air quality indexes (AQI) or just tries to discover changes in one pollutant at a time, even though there are challenges in multiple pollutants working together (Suman 2020). Also, the majority of approaches use each sensor's data independently, treating the information from one sensor as unrelated to that of other sensors. If models do not consider this relationship between locations, they have a tough time spotting pollution in more than one place next to each other (Ma, 2021). In addition, even though high-dimensional spatio-temporal models are great in theory, practical use is rare. Even though advanced models exist to capture these complexes trend, they are only sometimes used in actual air quality systems. Another important issue is the ability of anomaly detection systems to grow and work in real time. Even though it is possible for unsupervised machine learning methods to find anomalies without labels, several problems related to use in very dense sensor networks with nonstop data, reaction to changes in data over time, and making the machine learning clear and easy to interpret are still being studied (Salima, Asri et al. 2013, Agyemang 2024).

For this reason, this study aims to use and review the performance of three unsupervised models known as

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Isolation Forest, DBSCAN, and Autoencoders. They are most useful for handling many variables and data points from air quality data that come from different places and times. As they can work without knowing any statistical parameters and are efficient in computing, Isolation Forest is a popular tool for checking anomalies in environmental data. By spotting sparse areas, DBSCAN classifies them as anomalies since they are different from the rest of the data. It still can be trusted for spatial anomaly detection if the right hyperparameters are chosen, even better if hybrid techniques are added. With the addition of LSTMs or convolutional layers, Autoencoders are good at both reproducing normal patterns and finding when data is abnormal, as detected by the errors in these patterns. All in all, these models are classified as tree-based, density-based, and representation-based methods that provide a harmonious way to detect anomalies.

All in all, while it is easy to interpret traditional statistical methods, they fail to handle well the unpredictable changes that happen in real air quality measurements. Alternatively, Isolation Forests, DBSCAN, and Autoencoders are built to deal with large and noisy data streams that change at a fast rate. ConvLSTM autoencoders, graph-regularized autoencoders, and Deep Isolation Forest are recent examples that could play a big role in this area, but they haven't been applied to many air quality problems yet. Analyzing datasets, the main unsupervised models with various datasets is meant to address existing studies' limitations and help design better ways to spot environmental anomalies impacting society and the environment.

METHODOLOGY

Dataset Selection and Acquisition

Three publicly available air quality datasets were used to provide data from many regions, different data sources and different air pollution sensors. The UCI Air Quality Data Set records observations from chemical sensors installed in an Italian city every hour, collecting data on carbon monoxide (CO), non-methane hydrocarbons (NMHC) and nitrogen oxides (NOx). OpenAQ is a second source, and it is an open-source system that collects information from government and research groups worldwide. You can see real-time and past readings of PM2.5, PM10, NO_2 and O_3 and the different metadata through OpenAQ. The AirData service of the U.S. Environmental Protection Agency is the source of the third dataset which consists of detailed air quality data collected by sensors placed at locations throughout the country.

Official APIs will be used to fetch datasets, and they are also available to download in CSV or JSON formats from the repositories. All the input data from each source will be arranged to have a common setup by applying a Python-based custom preprocessing pipeline.

Data Preprocessing

To ensure uniformity across features and improve model performance, two key preprocessing techniques were applied: Z-score normalization and interquartile range (IQR) filtering.

Z-score normalization rescales the data so that each feature has a meaning of 0 and a standard deviation of 1. The transformation is defined by the equation:

I. Z-score Normalization

$$z = \frac{x - \mu}{\sigma}$$

where:

- x = original value
- μ = mean of the feature
- σ = standard deviation of the feature

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

- z = normalized value

This step is essential for algorithms like DBSCAN and autoencoders that are sensitive to feature scales.

II. IQR Outlier Detection

IQR filtering is used to detect and remove outliers. The interquartile range is calculated as:

$IQR = Q_3 - Q_1$

Lower Bound $= Q_1 - 1.5 \times IQR$

Upper Bound $= Q_3 + 1.5 \times IQR$

a) Isolation Forest

I. Anomaly Score Calculation

Isolation Forest identifies anomalies based on the average number of splits required to isolate a data point. The **anomaly score** is calculated as:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

where:

- $E(h(x))$ = average path length of point x over all isolation trees
- $c(n)$ = average path length of unsuccessful search in BST with n points

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}$$

where $H(i) = \ln(i) + 0.5772156649$ (harmonic number with Euler-Mascheroni constant)

II. Anomaly Classification

$$Anomaly = \begin{cases} -1, & if\ s(x,n) > threshold(anomaly) \\ 1, & if\ s(x,n) \leq threshold(normal) \end{cases}$$

where threshold is determined by the contamination parameter such that the specified percentage of points with highest anomaly scores are classified as anomalies.

b) DBSCAN

DBSCAN clusters data based on density. A point pp belongs to the ε-neighborhood if:

$$N_\epsilon(p) = q \in D | \text{dist}(p,q) \leq \epsilon$$

Point Classification Rules

- **Core Point**: $|N_\epsilon(p)| \geq$ MinPts
- **Border Point**: $|N_\epsilon(p)| <$ MinPts but p is in neighborhood of core point
- **Noise Point (Anomaly)**: Neither core nor border point

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Silhouette Score

$$s(i) = \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)}$$

where:

- $a(i)$ = average distance from point i to other points in same cluster
- $b(i)$ = minimum average distance from point i to points in other clusters

c) Autoencoder

I. Network Architecture

An autoencoder compresses and reconstructs data using a neural architecture defined as:

Encoder: $h = f(x) = \sigma(Wx + b)$ **Decoder**: $\hat{x} = g(h) = \sigma'(W'h + b')$

where:

- $x \in R^d$ = input vector
- $h \in R^p$ = hidden representation (with $p < d$)
- W, W' = weight matrices
- b, b' = bias vectors
- σ, σ' = activation functions (ReLU)

II. Loss Function

The loss function used during training is the Mean Squared Error (MSE) between the original and reconstructed inputs:

$$L = \frac{1}{n}\sum_{i=1}^{n} |x_i - \hat{x_i}|^2$$

Reconstruction Error (MSE)

$$\text{MSE}(x) = \frac{1}{d}\sum_{j=1}^{d} \big(x_j - \hat{x_j}\big)^2$$

Anomaly Detection Threshold

An anomaly is flagged if its reconstruction error exceeds the 99th percentile of all errors:

$$Anomaly = \begin{cases} 1 & if\ MSE(x) > percentile99(MSE) \\ 0 & otherwise \end{cases}$$

III. Dropout Regularization

$$h_{\text{dropout}} = h \odot m$$

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

where m is binary mask with probability p of being 1, and $\odot$ denotes element-wise multiplication.

Model Implementation

Three ML methods that do not require supervision were put into practice: Isolation Forest, DBSCAN and Autoencoders. They were picked for their proven ability to spot defects in large and noisy environmental data.

It looks for irregularities by randomly grouping data and distinguishing objects that take the least number of steps to separate from those groups. It was well suited for big datasets and does not need to calculate distances, which helps it work reliably with very many variables. Scikit-learn was engaged for carrying out the implementation and the number of estimators as well as the contamination rate will be adjusted by using cross-validation.

DBSCAN looks for clusters by measuring their density rather than distance which allows it to spot isolated pollution spikes easily. The eps (neighborhood radius) and min_samples parameters of this algorithm were adjusted through a grid search based on expert knowledge and the method was carried out by using scikit-learn and hdbscan.

Anomalies were detected in the input by using autoencoders which measure errors in reconstruction of the data. Autoencoder was taught to first compress information and then reconstruct it. A major difference between what was imaged, and the reconstructed images were suggested as an anomaly. There will be symmetrical encoder-decoder layers, and the encoder layers made the information thinner while the decoder layers add thickness back, all with ReLU functions and dropout in between.

Performance Evaluation

Testing unsupervised anomaly detection models was difficult because there was no labeled data available. So, internal evaluation measures, experience in the field and comparison to actual historical events are all used in this study.

In Autoencoders, the difference between the original and reconstructed inputs, measured by the Mean Squared Error (MSE), act as the main reference to identify anomalies. By default, Isolation Forest computes its own anomaly score for anomaly detection. Both the Silhouette Coefficient and the fraction of noise points to clusters can show how well DBSCAN is performing.

Result interpretation and comparisons

Strong interpretation of any model was checked against different measures: detecting with near-ideal accuracy (visual and logical), responding effectively to noisy or limited data, processing efficiently and growing smoothly as the size of the dataset increases. Results were checked inside a single dataset (e.g., UCI only) and between datasets (e.g., UCI vs. OpenAQ) to test if they can be used for different data.

The comparison showed where each algorithm performs well or not so well. Time-series data may benefit from Isolation Forest more than from others, but DBSCAN would be better for detecting pollution clusters. Even so, autoencoders are effective in handling nonlinear time-related issues and need much training and handling before use.

All the results were used to determine and suggest ideal models for use in real-time air quality monitoring systems.

Hardware and Software

All steps of the experiment were carried out with Python 3.10+. Examples of important libraries will be:

- pandas and numpy are used for working with data,
- scikit-learn, pyod and hdbscan are useful for implementing models,

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

- TensorFlow Keras are used for deep learning Autoencoders.
- Visualization can be done using matplotlib and seaborn.

The training process was conducted in Google Colab Pro to benefit from scalable GPUs and GitHub was used to version the outputs for reproducibility and clarity.

RESULTS

This section presents the outputs of each anomaly detection model applied to the UCI air quality dataset. The performance of Isolation Forest, DBSCAN, and Autoencoders was evaluated on publicly available data, particularly the UCI Air Quality Dataset, to identify irregular pollution patterns, focusing primarily on carbon monoxide (CO) levels. This evaluation was based on anomaly detection counts, clustering quality, and visual coherence with time series pollutant patterns.

Anomaly Detection Results

Isolation Forest

The application of Isolation Forest showed promising results. This algorithm, based on the principle of random partitioning, identified 67 anomalous events in the CO(GT) time series (**Figure 1**).

These unusual patterns, shown in red, vary greatly from the regular trends across the year. Typically, such anomalies involved sudden jumps or drops that were much different from the usual cycles in the dataset. Since Isolation Forest uses values without calculating their distances, it works favorably with complex data representations from the environment. The model reacted to the unusual pollution near late fall and winter which follows the trend of increased pollution during those times, indicating that it responded well to important environmental changes.

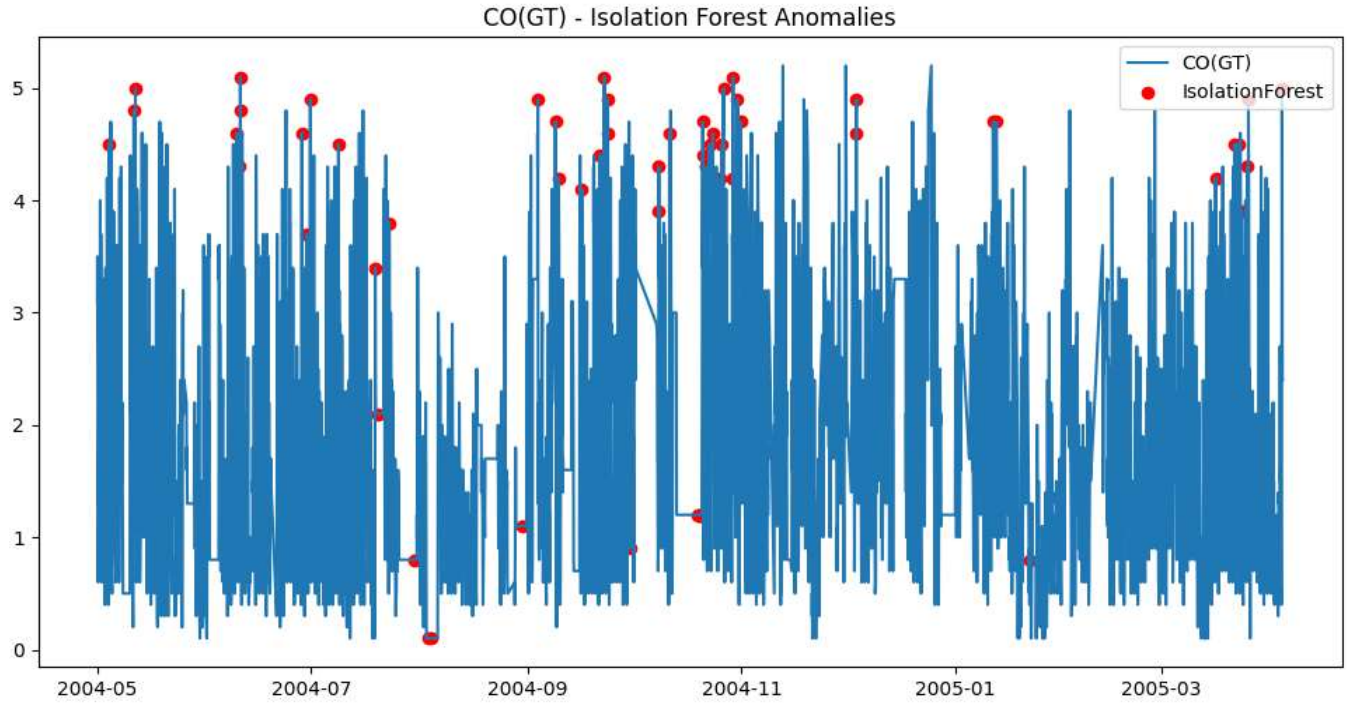

Figure 1: CO(GT) Time Series with Isolation Forest Detected Anomalies

Autoencoder

The Autoencoder model flagged the same number of anomalies (67), emphasizing events where the reconstructed pollution levels significantly diverged from the input (**Figure 2**).

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Autoencoders performed equally well. These neural network-based models were trained to reconstruct input data, and significant differences between original and reconstructed values were flagged as anomalies. The symmetry of the encoder-decoder structure, combined with dropout layers to prevent overfitting, enabled the model to capture regular trends in pollution levels effectively. The use of Mean Squared Error (MSE) as a scoring metric helped quantify how far a reading deviated from learned norms. Autoencoder results were largely consistent with those from Isolation Forest, reinforcing confidence in the validity of both models for detecting sharp deviations or unexpected pollution surges.

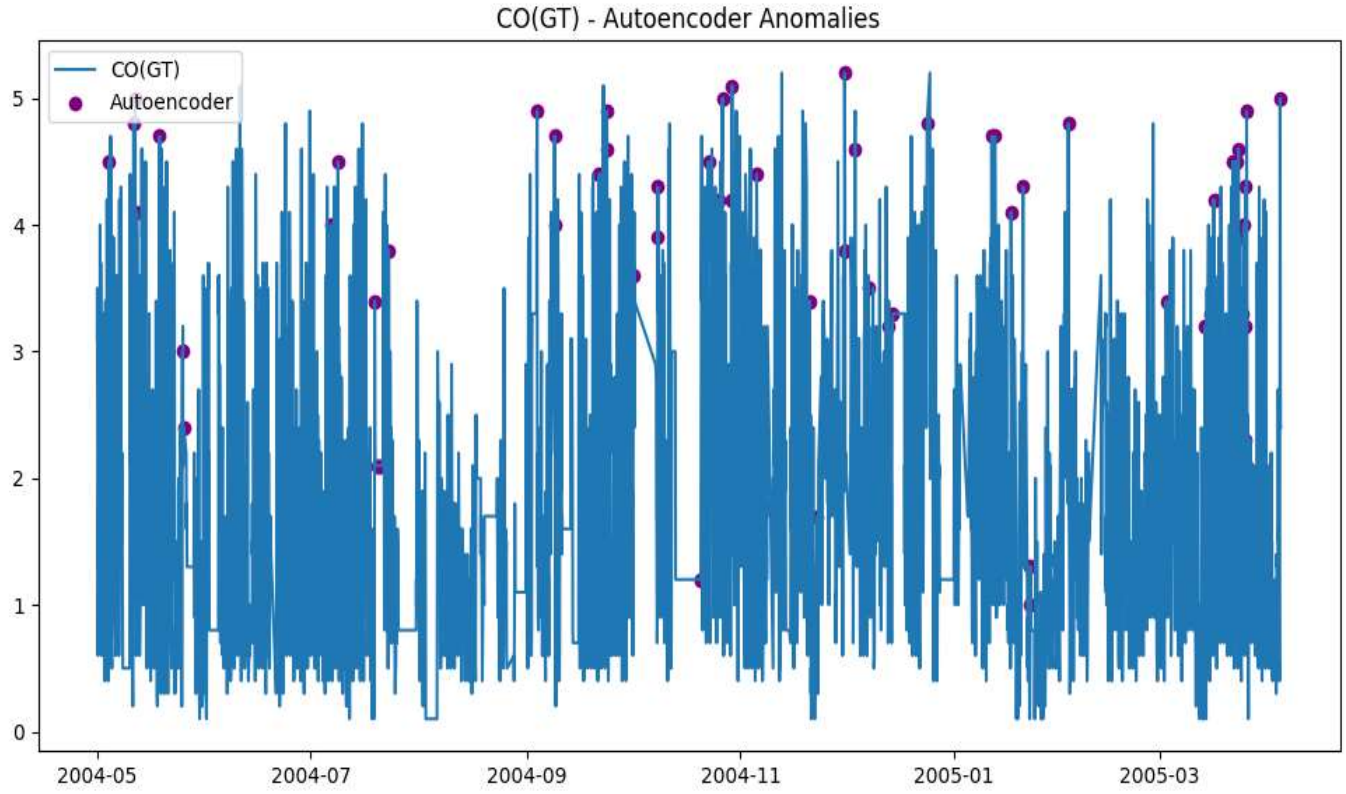

Figure 2: CO(GT) Time Series with Autoencoder Detected Anomalies

DBSCAN

DBSCAN detected only 1 anomaly using density-based clustering. Its performance here is constrained by parameter sensitivity and high data noise. The quality of the clustering was low, as reflected by the silhouette score.

DBSCAN identified only a single anomaly, and the silhouette score of 0.1139 indicated a weak clustering structure. This outcome can be attributed to the dense, noisy nature of environmental sensor data and the sensitivity of DBSCAN to parameter tuning (e.g., epsilon radius and minimum sample size). While DBSCAN is effective in spatial clustering applications, it may not be ideal for temporal pollution trends unless heavily tuned or used in conjunction with dimensionality reduction techniques.

Correlation Analysis

To examine interactions among pollutants, a heatmap of Pearson correlation coefficients was constructed. Strong correlations (red) and anti-correlations (blue) are observed, particularly among sensor readings measuring chemically related compounds (**Figure 3**).

The correlation heatmap features better relationships between variables. Strong correlations were observed between CO(GT) and other gas sensor outputs such as PT08.S1(CO) and C6H6(GT), validating that chemically linked features share similar patterns. This insight underscores the importance of using multivariate approaches rather than univariate ones when modeling air quality.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

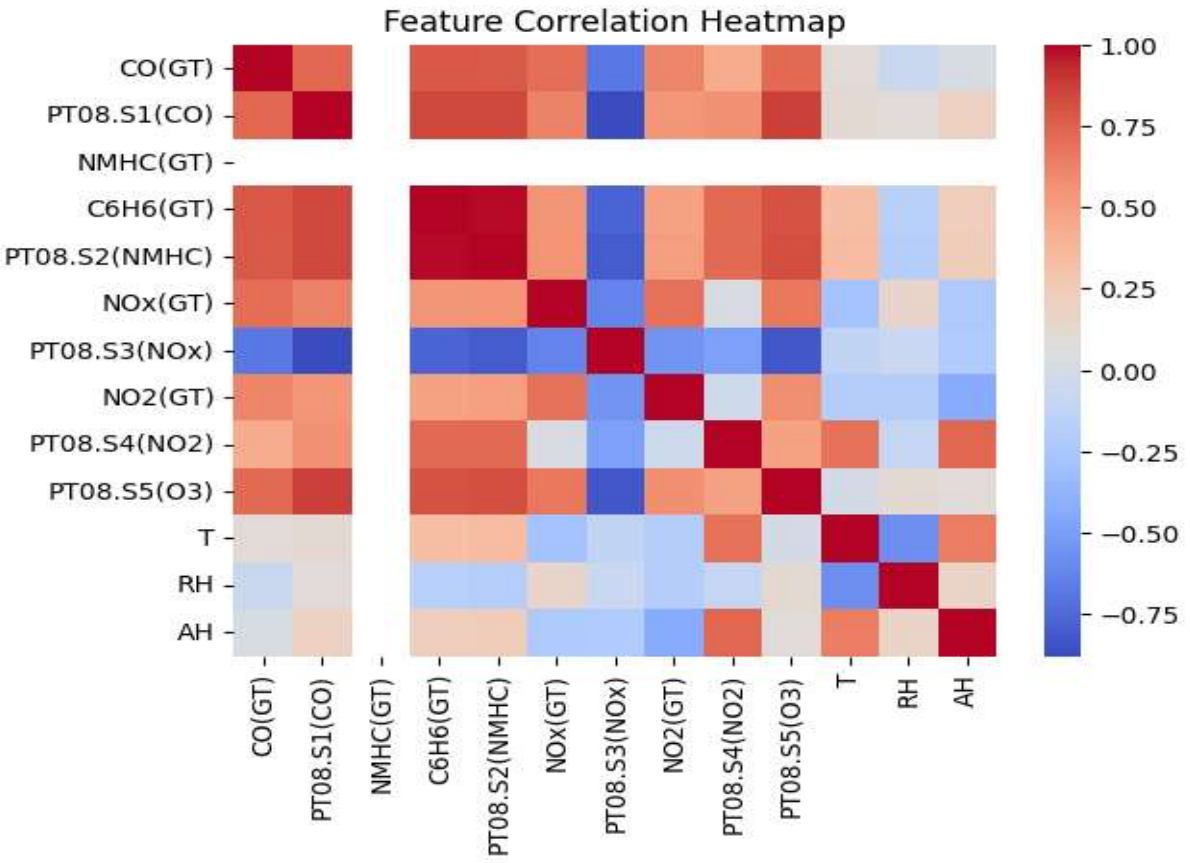

Figure 3: Heatmap of Feature Correlations

Comparison between detected Anomalies and Silhouette Score (DBSCAN)

Isolation Forest and Autoencoders produced similar anomaly sets and proved reliable for detecting shifts in pollutant levels. DBSCAN underperformed due to the high dimensionality and sparsity, though it remains valuable for spatial cluster analysis with properly tuned parameters (**Table 1**). The strong intra-feature correlations reinforce the importance of using multivariate models for anomaly detection in environmental data.

Table 1: Comparative Summary

Method	**Anomalies Detected**	**Silhouette Score (DBSCAN)**
Isolation Forest	67	N/A
DBSCAN	1	0.1139
Autoencoder	67	N/A

DISCUSSION

The study compares three unsupervised machine learning models used for finding anomalies in air quality, mainly carbon monoxide (CO) levels. The results demonstrate how well and poorly the methods do in analyzing data are noisy, time-based, and inconsistently of good quality.

Isolation Forest and Autoencoders work effectively in terms of detecting anomalies. Each of these two powerful models identified 67 anomalous points in the CO(GT) data series, which proves that they can be relied on for complex environmental studies. Because of its recursive random partitioning, the Isolation Forest algorithm is able to identify rare data that do not fit the seasonal cycle. This success is in line with how Liu et

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

al. (2008) suggested the method as a method that could work well in big data and be easily interpreted. Also, because the algorithm deals with data straight from the sensor arrays without using distance, it can better tell the difference between common issues and actual problems (Liu, Ting et al. 2008).

Just like the previous techniques, Autoencoders showed reliable results by using deep neural networks to reconstruct standard pollution patterns and notice changes by measuring reconstruction loss. Doing this was essential in their ability to model the surrounding environment. Such findings back up the viewpoint of Chalapathy and Chawla (2019), who stated that autoencoders are very useful in situations with limited labeled data (Chalapathy and Chawla 2019). The fact that the method finds high peaks without using distance metrics was important in this case since air quality information shows rapid spikes based on changes in the environment and seasons (Yang, Cheng et al. 2025). Additionally, when Autoencoders and Isolation Forest found the same anomalies, it increased the reliability of their observations, since these unusual events usually happened during winter and late autumn, the time when air pollution normally increases (Hasenfratz, Saukh et al. 2012, Lin, Wang et al. 2022).

DBSCAN, however, was not very effective since it detected only a single anomaly and had a low silhouette score of 0.1139 that implies the clusters remained mixed. The issues are mostly caused by the extra details and irregularities found in environmental data. Based on Ester et al. (1996), DBSCAN needs clearly different density values to be able to separate meaningful clusters (Ester, Kriegel et al. 1996). The result confirms what Sahu, Verma et al. (2023) reported, which is that DBSCAN does worse when the data is scarce or inconsistent. Although it works well in spatial settings such as noticing pollution across geographical regions, DBSCAN seems less useful for detecting sudden changes in time-based, high-frequency sensor data unless some additional preprocessing is first done (Sahu, Verma et al. 2023).

The research revealed that CO(GT) is strongly linked to sensors such as PT08.S1(CO) and C6H6(GT), so it is necessary to use a model that looks at several variables together. This corresponds with Yuan et al. (2024), who recommended using multivariate techniques in monitoring the environment to spot how pollutants influence one another (Yuan, Zou et al. 2024). Another reason why joint-feature approach performed better was because models like Autoencoders have stronger relationships than those that cluster points one by one.

It is especially helpful that the study makes use of time series graphs and heatmaps to confirm the discovered anomalies. Such tools also helped confirm the timings of the readings and clarify things for public health officials and planners. Visualization makes machine learning analyses easier to use in real settings, as also suggested by (Xie, Han et al. 2011).

In addition, this standard affects many sectors and brings practical changes to how work is done. This research will strongly affect public health, environmental plans, and how cities are structured. Applying unsupervised learning methods has proven that keeping track of air quality at all times in large areas is doable without using a dataset requiring a lot of manual input. With these models, we can quickly warn people and take action in advance when air quality isn't safe (Agyemang 2024).

Furthermore, this study creates a clear way to compare models so that suitable ones may be chosen based on details such as the data's temporal structure, how noisy it is, and the presence of interactions between variables. Thanks to the framework, we may build better systems for monitoring the environment and explore more hybrid or mixed methods for analyzing air quality in urban spaces.

CONCLUSION

It has been shown in this study that using Isolation Forest and Autoencoders without supervision is an excellent approach to finding unusually high levels of air pollution. Such models regularly pointed out problems that followed typical seasonal trends in pollution, proving their usefulness for use in the real world and for public health. Their ability to work without labeling data makes them useful, especially in fields where providing labeling is not easy.

By comparison, the difficulties faced by DBSCAN warn that clustering time series data of high dimensions

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

and with a lot of noise can be a challenge. It is recommended to try reducing the data's input dimensions or using t-distributed stochastic neighbor embedding (t-SNE) or Uniform Manifold Approximation and Projection (UMAP) along with DBSCAN for better clustering. Application of these techniques could organize data in a manner that emphasizes time and location and improve the results of clustering.

More studies should examine various main directions to advance the current discoveries. To start, incorporating autoencoders with statistical filters or graph-based clustering algorithms may give more insight into the data and result in better anomaly detection. With models on edge devices or sensor networks, it would be possible to find pollution anomalies in real time and take action close by. Moreover, linking certain forecasting models such as LSTM or GRU networks with anomaly detectors would make it possible to predict pollution patterns and catch any unusual shifts. Lastly, adjusting pretrained autoencoders with data native to a place can help the models suit their needs, take less time to train, and become more useful for people everywhere.

If scientists follow these paths, the field of intelligent environmental monitoring will be greatly improved. With these efforts, data-based policymaking will improve, so policymakers can respond to the dangers of pollution more correctly, at the right times, and faster.

Statement and declaration

Conflict of interest

No potential conflict of interest was reported by the author(s).

Funding

The author(s) reported there is no funding associated with the work featured in this article.

Data Availability statement

The data used in this study is available with the manuscript.

REFERENCES

1. Agyemang, E. F. J. S. A. (2024). "Anomaly detection using unsupervised machine learning algorithms: A simulation study." **26**: e02386.
2. Bouman, R. and T. J. a. p. a. Heskes (2025). "Autoencoders for Anomaly Detection are Unreliable."
3. Chalapathy, R. and S. J. a. p. a. Chawla (2019). "Deep learning for anomaly detection: A survey."
4. Claire Heffernan, R. P., Drew R. Gentner, Kirsten Koehler, Abhirup Datta (2022). "A dynamic spatial filtering approach to mitigate underestimation bias in field calibrated low-cost sensor air-pollution data."
5. Dhingra, S., R. B. Madda, A. H. Gandomi, R. Patan and M. J. I. I. o. T. J. Daneshmand (2019). "Internet of Things mobile–air pollution monitoring system (IoT-Mobair)." **6**(3): 5577-5584.
6. Ester, M., H.-P. Kriegel, J. Sander and X. Xu (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. kdd.
7. Feng, Y., J. Chen, Z. Liu, H. Lv and J. J. I. I. o. T. J. Wang (2022). "Full graph autoencoder for one-class group anomaly detection of IIoT system." **9**(21): 21886-21898.
8. Guo, S., Y. Lin, S. Li, Z. Chen and H. J. I. T. o. I. T. S. Wan (2019). "Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting." **20**(10): 3913-3926.
9. Hasenfratz, D., O. Saukh and L. Thiele (2012). On-the-fly calibration of low-cost gas sensors. European Conference on Wireless Sensor Networks, Springer.
10. Idrees, Z. and L. J. J. o. I. I. I. Zheng (2020). "Low cost air pollution monitoring systems: A review of protocols and enabling technologies." **17**: 100123.
11. Imam, M., S. Adam, S. Dev and N. J. I. S. w. A. Nesa (2024). "Air quality monitoring using statistical learning models for sustainable environment." **22**: 200333.

12. Kaushik, M., B. J. I. J. o. S. Mathur and H. R. i. Engineering (2014). "Comparative study of K-means and hierarchical clustering techniques." **2**(6): 93-98.
13. Koffi, N. g. A., K. P. N'Gouran, Y. S. S. Barima and D. M. Angaman (2019). "Active and passive air quality bio-monitoring in the tropics: Intra-urban and seasonal variation in atmospheric particles estimated by leaf saturated isothermal remanent magnetisation of Ficus benjamina L (Moraceae)."
14. Laskar, M. T. R., J. X. Huang, V. Smetana, C. Stewart, K. Pouw, A. An, S. Chan and L. J. A. T. o. C.-P. S. Liu (2021). "Extending isolation forest for anomaly detection in big data via K-means." **5**(4): 1-26.
15. Lin, X., H. Wang, J. Guo and G. J. I. A. Mei (2022). "A deep learning approach using graph neural networks for anomaly detection in air quality data considering spatiotemporal correlations." **10**: 94074-94088.
16. Liu, F. T., K. M. Ting and Z.-H. Zhou (2008). Isolation forest. 2008 eighth ieee international conference on data mining, IEEE.
17. Meo, S., M. Salih, F. Al-Hussain, J. Alkhalifah, A. Meo and A. J. E. R. M. P. S. Akram (2024). "Environmental pollutants PM2. 5, PM10, carbon monoxide (CO), nitrogen dioxide (NO)." **28**: 789-796.
18. Meo, S. A., M. A. Salih, J. M. Alkhalifah, A. H. Alsomali and A. A. J. J. o. K. S. U.-S. Almushawah (2024). "Environmental pollutants particulate matter (PM2. 5, PM10), Carbon Monoxide (CO), Nitrogen dioxide (NO2), Sulfur dioxide (SO2), and Ozone (O3) impact on lung functions." **36**(7): 103280.
19. Ofremu, G. O., B. Y. Raimi, S. O. Yusuf, B. A. Dziwornu, S. G. Nnabuife, A. M. Eze, C. A. J. G. E. Nnajiofor and Resources (2024). "Exploring the relationship between climate change, air pollutants and human health: impacts, adaptation, and mitigation strategies." 100074.
20. Sahu, R. T., M. K. Verma, I. J. I. J. o. H. S. Ahmad and Technology (2023). "Density-based spatial clustering of application with noise approach for regionalisation and its effect on hierarchical clustering." **16**(3): 240-269.
21. Salima, O., N. Asri and H. J. Hamid (2013). "Machine learning techniques for anomaly detection: an overview."
22. Sharma, S. B., S. Jain, P. Khirwadkar, S. J. I. J. o. R. i. P. Kulkarni and Biotechnology (2013). "The effects of air pollution on the environment and human health." **1**(3): 391.
23. Suman, M. J. M. T. P. (2020). "Air quality indices: A review of methods to interpret air quality status." **34**: 863-868.
24. Van Zoest, V., A. Stein, G. J. W. Hoek, Air, and S. Pollution (2018). "Outlier detection in urban air quality sensor networks." **229**: 1-13.
25. Xie, M., S. Han, B. Tian, S. J. J. o. N. Parvin and c. Applications (2011). "Anomaly detection in wireless sensor networks: A survey." **34**(4): 1302-1325.
26. Xiong, Z., R. Chen, Y. Zhang, X. J. J. o. I. Zhang and C. Science (2012). "Multi-density DBSCAN algorithm based on density levels partitioning." **9**(10): 2739-2749.
27. Xu, H., G. Pang, Y. Wang, Y. J. I. T. o. K. Wang and D. Engineering (2023). "Deep isolation forest for anomaly detection." **35**(12): 12591-12604.
28. Yang, L., J. Cheng, Y. Zhao, Z. Ni, Q. Mao, S. J. I. T. o. N. S. Gao and Engineering (2025). "A Unified Software-Defined Autonomous Vehicle Network and Urban Congestion Prediction Method."
29. Yuan, A., C. Zou, Y. Wang and J. Hu (2024). Multivariate Time Series Anomaly Detection Based on Time-Frequency Dynamic Analysis. 2024 13th International Conference on Communications, Circuits and Systems (ICCCAS), IEEE.
30. Zhang, D., J. Liu and B. J. S. Li (2014). "Tackling air pollution in China—What do we learn from the great smog of 1950s in London." **6**(8): 5322-5338.
31. Zhengjing Ma, G. M., Salvatore Cuomo, Francesco Piccialli (2021). "Heterogeneous Data Fusion Considering Spatial Correlations using Graph Convolutional Networks and its Application in Air Quality Prediction." Journal of King Saud University-Computer and Information Sciences.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

APPENDIX

```
# Air Quality Anomaly Detection Script

import os

import pandas as pd

import numpy as np

import requests

from zipfile import ZipFile

from io import BytesIO

from sklearn.ensemble import IsolationForest

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import DBSCAN

from sklearn.metrics import silhouette_score

from keras.models import Model

from keras.layers import Input, Dense, Dropout

from keras.callbacks import EarlyStopping

import matplotlib.pyplot as plt

import seaborn as sns

# ---------- Data Acquisition ---------- #

os.makedirs("datasets", exist_ok=True)

# 1. Download UCI Dataset

uci_url = "https://archive.ics.uci.edu/ml/machine-learning-databases/00360/AirQualityUCI.zip"

uci_response = requests.get(uci_url)

with ZipFile(BytesIO(uci_response.content)) as zipfile:

    zipfile.extractall("datasets/uci")

uci_df = pd.read_excel("datasets/uci/AirQualityUCI.xlsx", skipfooter=2)

uci_df.dropna(axis=0, how='all', inplace=True)

# 2. OpenAQ placeholder (archived/manual source)

openaq_df = pd.DataFrame()

print("⚠ OpenAQ API v2 is deprecated. Please download CSV from OpenAQ archive manually if needed.")
```

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

```
# 3. Download EPA Data
epa_url = "https://aqs.epa.gov/aqsweb/airdata/daily_88101_2024.zip"
epa_response = requests.get(epa_url)
with ZipFile(BytesIO(epa_response.content)) as zipfile:
    zipfile.extractall("datasets/epa")
epa_df = pd.read_csv("datasets/epa/daily_88101_2024.csv", low_memory=False)
# ---------- Preprocessing ---------- #
uci_df.replace(-200, np.nan, inplace=True)
uci_df.dropna(axis=0, thresh=uci_df.shape[1]//2, inplace=True)
uci_df = uci_df.ffill()
uci_df['DateTime'] = pd.to_datetime(uci_df['Date'].astype(str) + ' ' + uci_df['Time'].astype(str), errors='coerce')
uci_df.drop(columns=['Date', 'Time'], inplace=True)
uci_df.dropna(subset=['DateTime'], inplace=True)
uci_df.set_index('DateTime', inplace=True)
# IQR Filtering
numeric_cols = uci_df.select_dtypes(include=[np.number]).columns
for col in numeric_cols:
    Q1 = uci_df[col].quantile(0.25)
    Q3 = uci_df[col].quantile(0.75)
    IQR = Q3 - Q1
    uci_df = uci_df[(uci_df[col] >= Q1 - 1.5 * IQR) & (uci_df[col] <= Q3 + 1.5 * IQR)]
# Normalization
features = uci_df.select_dtypes(include=[np.number]).columns
scaler = StandardScaler()
uci_df[features] = scaler.fit_transform(uci_df[features])
# ---------- Isolation Forest ---------- #
iso = IsolationForest(contamination=0.01, random_state=42)
uci_df['anomaly_iso'] = iso.fit_predict(uci_df[features])
# ---------- DBSCAN ---------- #
```

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

```
dbscan = DBSCAN(eps=2, min_samples=5)

db_labels = dbscan.fit_predict(uci_df[features])

uci_df['anomaly_dbscan'] = db_labels

sil_score = silhouette_score(uci_df[features], db_labels) if len(set(db_labels)) > 1 else np.nan

# ---------- Autoencoder ---------- #

X = uci_df[features].values

input_dim = X.shape[1]

input_layer = Input(shape=(input_dim,))

encoder = Dense(64, activation="relu")(input_layer)

encoder = Dropout(0.1)(encoder)

encoder = Dense(32, activation="relu")(encoder)

decoder = Dense(64, activation="relu")(encoder)

decoder = Dropout(0.1)(decoder)

decoder = Dense(input_dim, activation=None)(decoder)

autoencoder = Model(inputs=input_layer, outputs=decoder)

autoencoder.compile(optimizer='adam', loss='mse')

autoencoder.fit(X, X, epochs=50, batch_size=32, callbacks=[EarlyStopping(monitor='loss', patience=5)], verbose=0)

reconstructions = autoencoder.predict(X)

mse = np.mean(np.power(X - reconstructions, 2), axis=1)

uci_df['anomaly_auto'] = mse > np.percentile(mse, 99)

# ---------- Visualization ---------- #

uci_df['CO(GT)'] = scaler.inverse_transform(uci_df[features])[:,0]  # for plotting

plt.figure(figsize=(12, 6))

plt.plot(uci_df.index, uci_df['CO(GT)'], label='CO(GT)')

plt.scatter(uci_df[uci_df['anomaly_iso'] == -1].index, uci_df[uci_df['anomaly_iso'] == -1]['CO(GT)'], color='red', label='IsolationForest')

plt.title('CO(GT) - Isolation Forest Anomalies')

plt.legend()

plt.show()
```

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

```
plt.figure(figsize=(12, 6))

plt.plot(uci_df.index, uci_df['CO(GT)'], label='CO(GT)')

plt.scatter(uci_df[uci_df['anomaly_auto']].index, uci_df[uci_df['anomaly_auto']]['CO(GT)'], color='purple', label='Autoencoder')

plt.title('CO(GT) - Autoencoder Anomalies')

plt.legend()

plt.show()

# -------------------- Result Summary -------------------- #

summary = pd.DataFrame({

    "Method": ["Isolation Forest", "DBSCAN", "Autoencoder"],

    "Anomalies Detected": [

        (uci_df['anomaly_iso'] == -1).sum(),

        (uci_df['anomaly_dbscan'] == -1).sum(),

        uci_df['anomaly_auto'].sum()

    ],

    "Silhouette Score (DBSCAN)": [np.nan, sil_score, np.nan]

})

print("\n\n=== Model Comparison Table ===")

print(summary.to_string(index=False))

sns.heatmap(uci_df[features].corr(), cmap='coolwarm', annot=False)

plt.title("Feature Correlation Heatmap")

plt.show()

print("\nAnalysis Complete. Models and visual summaries are above.")
```