Learning and Nonlinear Models - Journal of the Brazilian Society on Artificial Intelligence (SBIA), Vol. 24, Iss. 1, pp. 36–45, 2026
https://doi.org/10.21528/lnlm-vol24-no1-art3 © Brazilian Society on Artificial Intelligence

USING TEMPORAL CONVOLUTIONAL AUTOENCODERS FOR SOUND-BASED FAULT DETECTION IN INDUSTRIAL MACHINERY

Igor C. Assunção, Rodrigo P. Monteiro
Programa de Pós-Graduação em Engenharia de Sistemas
Universidade de Pernambuco
E-mails: {igor.cassuncao, rodrigo.paula}@upe.br

Andrea M. N. C. Ribeiro
Departamento de Eletrônica e Sistemas
Universidade Federal de Pernambuco
E-mails: andrea.marianogueira@ufpe.br

Diego M. P. F. Silva
Escola de Tecnologia e Comunicação
Universidade Católica de Pernambuco
E-mails: diego.silva@unicap.br

Abstract – We propose using two deep autoencoder architectures, *i.e.* the Temporal Convolutional Network Autoencoder (TCN-AE) and Wasserstein Autoencoder with Generative Adversarial Network (WAE-GAN), to perform anomaly detection in the industrial context. In this work, we detect anomalies by introducing a feature engineering pipeline that structures univariate acoustic signals into sequential multivariate representations via Mel Frequency Cepstral Coefficients (MFCCs). The trained models were evaluated using 10-fold cross-validation, and their performances were assessed through an empirical analysis of hyperparameter sensitivity, mapping prominent structural configurations for each neural architecture. The optimized TCN-AE showed superior performance, achieving an average AUC-ROC of 0.8716 for pump devices and 0.7834 for fan devices, thereby surpassing state-of-the-art literature benchmarks, while also demonstrating robust performance for the slider device with an average AUC-ROC of 0.9646. Both architectures presented limitations with the valve device due to its highly transient behavior. The results achieved confirm the effectiveness of sequential reconstruction-based models for continuous machine operations while highlighting specific open challenges in acoustic industrial monitoring.

Key Words – Predictive Maintenance, Industry 4.0, Anomaly Detection, Deep Autoencoders, Temporal Convolutional Networks.

1 Introduction

Rotating machines play an essential role in industry since they generate the motion and power required by industrial processes. Consequently, the maintenance policies for those equipment directly impact the product quality and industrial plant safety, accounting for up to 70% of total production costs [1]. On the other hand, selecting the most appropriate maintenance strategies in industrial environments is a challenging task.

In this context, predictive maintenance is a promising solution. This maintenance strategy uses data from equipment operation to predict potential equipment failures, allowing interventions to be scheduled efficiently and in advance. In some cases, early anomaly detection can prevent more serious problems [2], emphasizing the importance of proactive approaches. This philosophy is closely related to Condition-Based Maintenance (CBM) [3], which relies on continuous monitoring of the machine's operational state to perform tasks such as fault detection and diagnosis [4]. This strategy associates the fault occurrences with the anomalous operating patterns identified from information collected during the monitoring activity [5].

The advancement of Industry 4.0 technologies, driven by the adoption of Internet of Things (IoT) and increased connectivity, promoted the use of intelligent approaches to perform fault detection and diagnosis more effectively. In this new paradigm, which emphasizes predictive maintenance [6], advanced computational models and digital signal processing techniques are widely applied to fault diagnosis in industrial machinery. Particularly, Anomalous Sound Detection (ASD), *i.e.* the task of identifying whether the sound emitted by an object belongs to a normal or abnormal operating behavior, is emerging as a relevant field of research. Mnasri *et al.* [7] described real-world scenarios in which ASD techniques can be applied, such as industrial machinery monitoring for predictive maintenance. Literature examples include using Wavelet Packet Transform for feature extraction from temporal signals and fault classification in gearboxes [8], employing context-aware deep autoencoders for fault detection from log-Mel spectrograms [9], and combining Convolutional Neural Networks (CNNs) with support vector machines for fault diagnosis [10]. It is worth mentioning that identifying anomalous patterns in data collected from industrial machines has been a recurring subject of study, even before the popularization of deep learning [11].

https://doi.org/10.21528/lnlm-vol24-no1-art3 © Brazilian Society on Artificial Intelligence

Although traditional techniques, such as Isolation Forest e Local Outlier Factor, are still employed in the context of predictive maintenance, their application is commonly restricted to low-dimensional problems. On the other hand, the fault detection in industrial devices using sound data presents significant challenges. These include the multidimensional and temporal nature of sound signals, the common absence of labels indicating faults, the difficulty in understanding complex interrelationships in data, and the lack of knowledge about all possible types of faults [12]. Additionally, the industrial environment introduces noise and variability in operating conditions, which hinders accurate anomaly detection. To address the complexity and dimensionality of sound data, feature extraction is a critical step. Among the features most commonly found in the literature, we have the Mel-frequency Cepstral Coefficients (MFCCs) [7]. MFCCs are often used in speech and audio processing and represent the short-term power spectrum of a sound signal on a non-linear Mel frequency scale [13]. Incorporating MFCC feature extraction at the beginning of the pipelines allowed the traditional techniques to handle acoustic data more effectively [6].

Literature suggests that using reconstructed-based models, such as Autoencoders (AEs) [14], is an efficient approach for the ASD in industrial context. As highlighted by Nunes [2], AEs are unsupervised neural networks that learn efficient data encodings and have been widely used in ASD. Their architectures consist of an encoder that compresses the input into a low-dimensional latent representation and a decoder that reconstructs the original data from that code. The model is trained exclusively on normal operation data, minimizing the reconstruction error (*e.g.* Mean Squared Error) for this kind of information. The primary strategy for anomaly detection is to make the AE highly proficient at reconstructing normal patterns. Consequently, when anomalous data is provided, the AE has difficulty in reconstructing it, resulting in a higher error that can be used to detect faults.

In this context, we propose using AE-based architectures on ASD in the industrial context. We assessed the Temporal Convolutional Network Autoencoder (TCN-AE) and Wasserstein Autoencoder with Generative Adversarial Network (WAE-GAN), inspired by the work of Silva *et al.* [6], which demonstrated their superior performance in anomaly detection using multivariate data from the Porto municipal subway system in Portugal. In this manuscript, we detail the practical application, training, and evaluation of the TCN-AE and WAE-GAN models using the Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection (MIMII) dataset. We included a key methodological step: transforming the original univariate sound data into a multivariate time-series representation using series of MFCCs. This feature extraction process is essential, as it structures the data into a format compatible with the sequential, multivariate architectures of the TCN-AE and WAE models. Their performances are evaluated using metrics such as ROC-AUC, F1-Score, Accuracy, Precision, and Recall, and the results were benchmarked against established literature approaches.

We can summarize the main contributions of this work as follows:

1. Propose and evaluate the use of TCN-AE and WAE-GAN for sound anomaly detection;
2. Provide an in-depth empirical analysis of hyperparameter sensitivity and network behavior to identify structural properties governing robust model configurations for ASD;
3. Using MFCC to adapt reconstruction-based anomaly detection models that operate on multivariate time series for univariate time series.

This document is organized as follows: Section 2 presents related work that supports the chosen approach and comparison methods. Section 3 details the methodology employed, from data processing to model training and evaluation. Section 4 presents and discusses the experimental results. Finally, Section 5 presents the conclusions, the work's contributions, its limitations, and directions for future research.

2 Related Work

2.1 ASD in Industrial Predictive Maintenance

Predictive maintenance emerged as a prominent strategy in Industry 4.0, aiming to anticipate equipment failures to optimize production and safety. As highlighted by Silva *et al.* [6], the transition to Industry 4.0 transformed industrial manufacturing through significant advances in automation and intelligent systems. This paradigm shift emphasized the detection of future failures while machines were still considered to be in normal state of operation. Within this context, ASD emerged as a promising approach, defined as the task of identifying whether the sound emitted by an object was normal or anomalous [2].

A relevant benchmark for this task is the work of Silva *et al.* [15], which conducted a comparative analysis of approaches like Hitachi, RANSynCoders, and GANF. In this study, the authors highlighted that incorporating MFCCs allowed the models to handle the acoustic data more effectively.

The relevance of ASD is accentuated by its potential for early intervention, which can prevent a variety of subsequent problems [2]. Indeed, the economic impact of proactive fault detection is substantial, since some production issues can lead to dramatic financial losses, such as the 400 million Euros per week in lost sales reported by Volkswagen due to downtime [16]. Thus, ASD techniques are applied in various practical scenarios, including the monitoring of industrial machinery for predictive maintenance [7]. This approach aligns with the principles of Condition-Based Maintenance (CBM), which Alaswad and Y. Xiang describe as fundamental for early fault detection and diagnosis [4]. The core strategy of CBM is to base maintenance actions on the actual condition of the asset, which is determined by processing information collected through continuous monitoring [17].

Learning and Nonlinear Models - Journal of the Brazilian Society on Artificial Intelligence (SBIA), Vol. 24, Iss. 1, pp. 36–45, 2026
https://doi.org/10.21528/lnlm-vol24-no1-art3 © Brazilian Society on Artificial Intelligence

2.2 Extracting Features from Sound Signals with MFCC

Processing sound data effectively for ASD requires a robust feature-extraction step that transforms the raw audio signal into a compact, informative representation [7]. MFCCs are a widely adopted standard for this task, representing the short-term power spectrum of a sound on a non-linear Mel frequency scale [13]. Systematic reviews confirm that MFCCs are among the most cited feature extraction methods in ASD, often used alongside Autoencoders [2]. The ability of MFCCs to reduce dimensionality and focus on relevant spectral features, even in the presence of background noise, makes them particularly suitable for industrial environments, with studies demonstrating that their incorporation improves deep learning models capacity to handle complex acoustic data [6].

3 Methodology

This section outlines the methodological procedures used to conceive and evaluate the sound anomaly detection models. The overall workflow, illustrated in Figure 1, follows a structured pipeline encompassing data processing, model definition, robust training and validation, and performance evaluation. The steps of the methodology pipeline are thoroughly described in the following subsections.

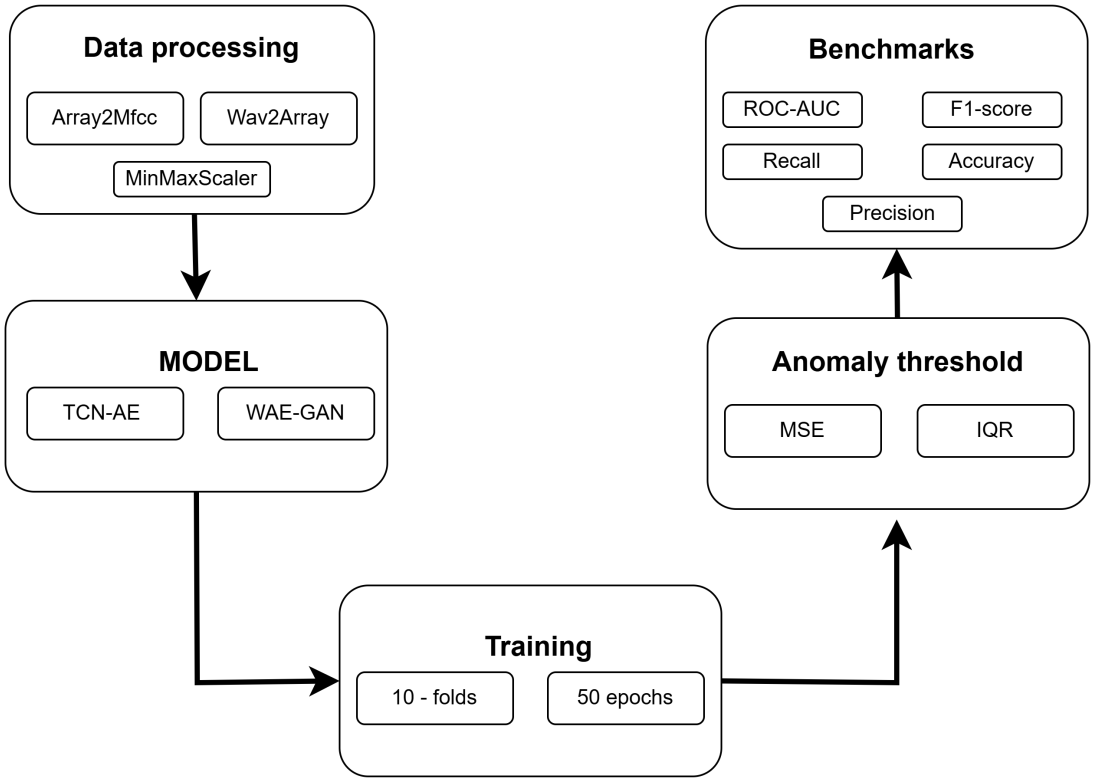


Figure 1: Methodology pipeline diagram

3.1 Data Processing

3.1.1 MIMII Dataset

The MIMII dataset [18] presents data collected from different groups of devices: valve, pump, fan, and slide. Their information is shown in Table 1. Each group contains hundres of recordings of normal and anomalous sounds. These anomalous sounds include faults such as contamination, leakage, rotational imbalance, and rail damage. The devices in a group are identified by the IDs 00, 02, 04, and 06. Recordings of 10 seconds were made at a sampling rate of 16 kHz and 16 bits per sample. The data is available in .wav format, representing continuous sound recordings. This dataset allows the development and testing of sound-based fault detection systems.

3.1.2 Train and Test Split

An initial data split was performed to create independent sets for cross-validation and final evaluation. The split followed a one-class anomaly detection protocol, in which only normal samples were used for training and validation. Specifically, 80% of the normal samples formed the Primary Training Set, while the Primary Test Set was composed of the remaining 20% of the normal samples and 100% of the anomalous samples.

https://doi.org/10.21528/lnlm-vol24-no1-art3 © Brazilian Society on Artificial Intelligence

Table 1: Information about the industrial devices in the MIMII Dataset [18].

Machine Type	Operations	Examples of Anomalous Conditions
Valve	Repeatedly opening/closing with different timings	More than two types of contamination
Pump	Suction from/discharge into a water pool	Leakage, contamination, clogging, etc.
Fan	Normal operation	Imbalance, voltage change, clogging, etc.
Slide	Repeatedly sliding at different speed	Rail damage, loose belt, lack of lubrication, etc.

Data partitioning protocol:

- 80% of the normal samples → 10-fold cross-validation.
- 20% of the normal samples → final evaluation only.
- 100% of the anomalous samples → final evaluation only.

Subsequently, a K-fold cross-validation strategy was applied exclusively to this Primary Training Set. For this split, the MTSA [19] was used. In this study, we restricted our analysis to data originating from the ID_00 identifiers of each of the four machine types. While the MIMII dataset contains multiple device IDs representing different physical instances of the same machinery class, the ID_00 subset was selected as a standard baseline configuration. This constraint allows for a direct and reproducible comparison against benchmarks in recent literature, which establish their baseline findings on these exact data partitions. Each has a set of sound data under normal and anomalous conditions, as shown in Table 2.

Table 2: Distribution of MIMII Dataset sound segments by machine type, model, and condition.

Machine Type	**Model ID**	Segments in Normal Condition	Segments in Anomalous Condition
Valve	00	991	119
Pump	00	1006	143
Fan	00	1011	407
Slider	00	1064	356

3.1.3 Feature Extraction

The raw sound signals in .wav format were first converted to arrays, and then converted to series of MFCCs. In this step, the sound signals were processed using a sampling rate of 16000 Hz, consistent with the MIMII Dataset standard. The MFCCs were then extracted by using the MTSA module with the default parameters: a window size of 2048 samples, a hop length of 512 samples, and 128 Mel filter banks. This process resulted in a feature matrix with dimensions of 20 coefficients x 313 time windows for each 10-second audio signal. Crucially, the extraction of 20 coefficients follows established industrial baselines [15, 18], as lower-order cepstral coefficients successfully capture the macro spectral envelope (mechanical friction, unbalance, and resonance). Higher-order coefficients typically capture high-frequency background noise, which triggers overfitting. Restricting this to 20 coefficients regularizes the models and optimizes edge computing efficiency.

3.1.4 Data Normalization

The MFCC features were normalized to the [0,1] range using scikit-learn's MinMaxScaler [20]. In each cross-validation fold, the scaler was fitted exclusively on the training partition of that fold. The fitted scaler was then used to transform the corresponding validation partition. After model selection, the scaler was fitted again on the entire training set and used to transform the held-out test set. This procedure prevents data leakage from test data into the training process.

3.1.5 Data Dimensionality

Each normalized MFCC matrix, with dimensions (20, 313), was treated as a single input sequence for the models. In this representation, the 20 MFCC coefficients are treated as the attributes (or channels), and the 313 steps represent the sequence length. The final dataset for each machine ID_00 is therefore a collection of n matrices, where n is the total number of .wav files (normal and anomalous) for that machine. This collection is structured as a 3D tensor with the final shape (n, 20, 313), which is the format used by the models.

https://doi.org/10.21528/lnlm-vol24-no1-art3 © Brazilian Society on Artificial Intelligence

3.2 Model

3.2.1 Architecture

Deep learning models, especially autoencoders, have been widely used in anomaly detection [2]. This study implements and compares two advanced autoencoder architectures designed for sequential and mutivariate data: TCN-AE and WAE-GAN [6].

TCN-AE: This is a variation of the standard Autoencoder that replaces traditional recurrent layers with Temporal Convolutional Network (TCN) layers in both encoder and decoder. This architectural choice is motivated by the unique properties of TCNs, which are a variant of CNNs specifically adapted for sequential data. TCNs employ two key techniques: **a.** causal convolutions, which ensure that outputs at time t are only dependent on inputs from time t and earlier, and **b.** dilated convolutions, which exponentially increase the receptive field. In the architecture detailed by [6] and adopted by this work, these TCNs are constructed from residual blocks. Residual shortcuts are also employed, which are essential for training the deep networks required. As an autoencoder, the layers project the sequence into a compressed latent space and subsequently reconstruct it. This combination allows the network to capture long-range temporal dependencies, and the entire model is optimized by minimizing the Mean Squared Error (MSE) [6].

WAE-GAN: According to Silva *et al.* [6], the WAE-GAN's strength lies in the adversarial training, which enables the detection of more subtle variations in time series. While this architecture of Autoencoder also uses reconstruction error as an anomaly metric, this capability is enhanced by the regularization of the latent space imposed by an adversarial component. The WAE-GAN model used in this study was inspired by the work of Silva *et al.* [6], also using TCNs for the encoder and decoder functions and an LSTM network for the discriminator.

Both models leverage TCN layers to capture long-range temporal dependencies. The key architectural differences between the two models as tested in this study are summarized in Table 3.

Table 3: Architectural comparison of the evaluated models.

Component	TCN-AE	WAE-GAN
Encoder	TCN Layers	TCN Layers
Decoder	TCN Layers	TCN Layers
Latent Space	Resulting from Encoder	Fixed Embedding
Extra Component	—	LSTM Discriminator
Loss Function	Reconstruction Loss (MSE)	Reconstruction Loss (MSE) + Adversarial Loss

3.3 Training

To ensure a robust and fair comparison, the models were trained and evaluated using a multi-step process that included hyperparameter optimization via grid search, 10-fold cross-validation, and a clear protocol for anomaly classification, as detailed in the following subsections.

3.3.1 Hyperparameter Optimization

To identify the best configuration of each model architecture, we conducted a grid search over the predefined hyperparameter space. This step is important to ensure that the final performance comparison is based on each model's best potential, rather than an arbitrary or default setup. We detail the search space in Table 4.

Table 4: Hyperparameter search space used for model optimization.

Hyperparameter	Model	Tested Values
Learning Rate (lr)	Both	{0.001, 0.0005, 0.0001}
Batch Size	Both	{16, 32, 64, 128}
Encoder/Decoder Layers	Both	{3, 4, 5}
Embedding Dimension	WAE-GAN	{16, 32, 64}
Discriminator Ratio	WAE-GAN	{0.5, 1.0}

For each combination of the parameters presented in Table 4, a full 10-fold cross-validation was performed, and the configuration yielding the highest average AUC-ROC was selected for the final analysis.

https://doi.org/10.21528/lnlm-vol24-no1-art3 © Brazilian Society on Artificial Intelligence

3.3.2 K-Fold Cross-Validation Strategy

A 10-fold cross-validation strategy was applied strictly to the Primary Training Set, which contains only normal samples. In each fold, this set was divided into a local training partition (90% of the Primary Training Set) and a local validation partition (the remaining 10%). The deep autoencoders were fitted exclusively on the local training partition, while the local validation partition was used only for internal validation and hyperparameter analysis.

The Primary Test Set was not included in the cross-validation loop. After the cross-validation procedure, the final evaluation was performed exclusively on the Primary Test Set, composed of the held-out 20% normal samples and 100% of the anomalous samples. Therefore, the local validation samples were not combined with the final test samples when computing the reported metrics. The results are reported as the mean and standard deviation across the 10 fold-trained models, each evaluated on the same independent Primary Test Set.

3.4 Anomaly threshold

The anomaly score for each test sample was defined using the MSE reconstruction error. A classification threshold was dynamically established for each fold based on the distribution of training errors. Specifically, we adopted the same thresholding strategy used in our reference work [6], which employs the standard outlier detection rule from Tukey's method. Any sample with a reconstruction error exceeding the threshold defined in Equation (1) was classified as anomalous.

$$\text{Threshold} = Q3 + 1.5 \times \text{IQR} \tag{1}$$

In which:

- $Q3$ (third quartile) is the value that delimits the lowest 75% of reconstruction errors from training;
- IQR (interquartile range) is the range between the third $Q3$ and first quartile $Q1$ of the training reconstruction errors ($IQR = Q3 - Q1$), representing the variation of the central 50% of errors.

3.5 Metrics

To assess the final classification performance, we employed a standard set of metrics, as detailed in machine learning literature [21]: **1.** Accuracy, **2.** Precision, **3.** Recall, **4.** F1-Score, and **5.** AUC-ROC. For each fold-trained model, the classification metrics were computed on the independent Primary Test Set, composed of the held-out 20% normal samples and 100% of the anomalous samples. Accuracy, Precision, Recall, and F1-Score were calculated from the binary anomaly predictions obtained after applying the reconstruction-error threshold. AUC-ROC was computed from the continuous reconstruction-error scores. The results were presented in terms of the mean and standard deviation across the 10 fold-trained models. In addition, a 95% confidence interval of AUC-ROC was calculated for the comparison with the benchmark models.

4 Results and Discussion

Here, we present the results of the performance evaluation of the TCN-AE and WAE-GAN models for anomalous sound detection across different types of industrial machines in the MIMII dataset, using ID_00.

4.1 Model Performance

The anomaly detection performance of each model, regarding the four machine types (valve, pump, fan, and slide) is detailed in Table 5. This table presents the mean and standard deviation for each metric, calculated for the 10 models obtained via the 10-fold cross-validation. The results in Table 5, calculated on the Primary Test Set, suggest that the TCN-AE model consistently outperforms the WAE-GAN model in most metrics and machine types. Regarding the slider device, TCN-AE achieved an average F1-Score of 0.9711 and an average AUC-ROC of 0.9646. Meanwhile, the valve device posed the most significant challenge, with both models achieving AUC-ROC values below 0.5. Notably, for this device, both models performed poorly, with Precision, Recall, and F1-Score values close to zero. Those results suggest an inability of both models to identify anomalies in the valve device under the tested configuration.

Regarding the pump and fan devices, TCN-AE also demonstrated promising performance. For the pump, TCN-AE achieved average F1-Score of 0.8487 and AUC-ROC of 0.8716. For the fan, the average values were 0.7869 for F1-Score and 0.7834 for AUC-ROC. The WAE-GAN model, although showing lower results than TCN-AE for pump and fan, still demonstrated some detection capability for the fan, with an AUC-ROC of 0.7476, and for the pump, with an AUC-ROC of 0.6811, despite the lower F1-Score.

Figure 2 shows extreme scenarios in the performance analysis. In both images, the dashed red line represents the anomaly threshold, calculated using Equation (1) with the training reconstruction error data, *i.e.* the loss value. All samples with reconstruction error superior to the anomaly threshold are classified as anomalies. For the Slide Rail device (B), a clear separation is observed between the test reconstruction errors of normal (samples $\leq$ 390) and anomalous samples (samples > 390), consistent with the high AUC-ROC (0.96) and F1-Score (0.97) performance of the TCN-AE model. In contrast, for the Valve device (A),

https://doi.org/10.21528/lnlm-vol24-no1-art3 © Brazilian Society on Artificial Intelligence

Table 5: Performance (Mean ± Standard Deviation of 10 models) of the TCN-AE and WAE-GAN models for IDs_00 of the MIMII dataset machines. Best results for each metric per machine are in bold.

Model	Machine (ID 00)	Accuracy	Precision	Recall	F1-Score	AUC-ROC
TCN-AE	Valve	**0.6435 ± 0.0119**	**0.1961 ± 0.02523**	**0.0613 ± 0.0146**	**0.0928 ± 0.0182**	**0.4769 ± 0.0070**
	Pump	**0.9095 ± 0.0075**	0.9678 ± 0.0057	**0.7559 ± 0.0224**	**0.8487 ± 0.0145**	**0.8716 ± 0.0112**
	Fan	**0.7718 ± 0.0222**	0.8720 ± 0.0124	**0.7179 ± 0.0416**	**0.7869 ± 0.0266**	**0.7834 ± 0.0189**
	Slider	**0.9677 ± 0.0067**	0.9439 ± 0.0109	**1.0000 ± 0.0000**	**0.9711 ± 0.0058**	**0.9646 ± 0.0074**
WAE-GAN	Valve	0.4457 ± 0.0015	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.4607 ± 0.0022
	Pump	0.5878 ± 0.0028	**1.0000 ± 0.0000**	0.2979 ± 0.0034	0.4590 ± 0.0041	0.6811 ± 0.0027
	Fan	0.3574 ± 0.0247	**0.9874 ± 0.0080**	0.2005 ± 0.0320	0.3320 ± 0.0448	0.7476 ± 0.0086
	Slider	0.3764 ± 0.0364	**1.0000 ± 0.0000**	0.1893 ± 0.0471	0.3156 ± 0.0690	0.8322 ± 0.0016

the distributions of test reconstruction errors from both normal and anomalous are close, making it impossible to define a suitable threshold, as reflected in the low metric performance.

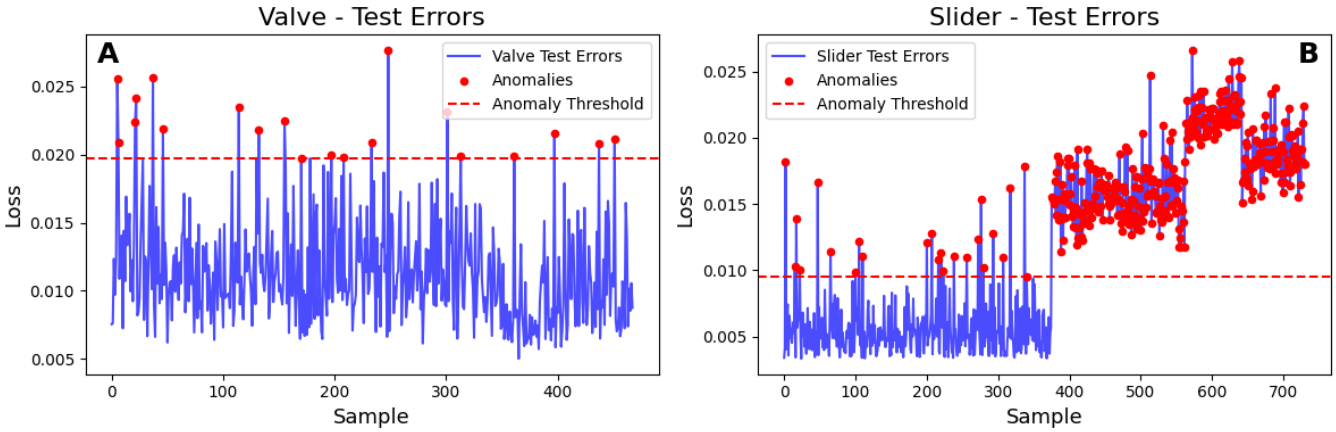


Figure 2: Comparing the distribution of reconstruction errors and the anomaly threshold based on Equation (1) of training errors for the worst-performing device, Valve ID 00 (A) and the best-performing device, Slide Rail ID 00 (B).

4.2 Comparative Analysis

The results of TCN-AE and WAE-GAN models were compared with those achieved by some reference models evaluated by [15] on the same subset of the MIMII dataset ID_00. Table 6 presents the results using the AUC-ROC as the reference metric.

The comparison reveals that the TCN-AE model used in this work is not only competitive but also outperforms the reference models (Hitachi, RANSynCoders-MFCC, GANF-MFCC) for the pump devices, scoring an AUC-ROC of 0.87 compared to the best result achieved by the reference models (0.79), and for the fan, scoring an AUC of 0.78 compared to the best reference model (0.64). For the slide device, the TCN-AE (0.96) performed similarly to GANF-MFCC (0.94) and slightly below Hitachi (0.99), which was the standout in [15]'s study for this machine.

As observed in Subsection 4.1, the valve device proved to be challenging for all models. The TCN-AE (0.476) and WAE-GAN (0.461) models in this study performed worse than the models from the reference work (which obtained AUC-ROC between 0.53 and 0.58), reinforcing the inherent complexity of anomalous sound detection in this specific device.

Table 6: Comparing the performance (average AUC-ROC and 95% Confidence Interval) of the models evaluated in this work with literature approaches for ID_00 of the MIMII dataset machines. Literature results were provided by Silva et al. (2024) [15]. The best result for each machine type is highlighted in bold.

Machine Type (ID 00)	**TCN-AE** (This Study)	**WAE-GAN** (This Study)	**Hitachi** [15]	**RANSynCoders-MFCC** [15]	**GANF-MFCC** [15]
Valve	0.48 (0.47, 0.48)	0.46 (0.46, 0.46)	0.56 (0.45, 0.68)	0.53 (0.50, 0.55)	**0.58 (0.48, 0.67)**
Pump	**0.87 (0.86, 0.88)**	0.68 (0.68, 0.68)	0.77 (0.75, 0.79)	0.79 (0.75, 0.82)	0.73 (0.68, 0.77)
Fan	**0.78 (0.77, 0.80)**	0.75 (0.74, 0.75)	0.60 (0.59, 0.61)	0.64 (0.59, 0.69)	0.54 (0.52, 0.57)
Slider	0.97 (0.96, 0.97)	0.83 (0.83, 0.83)	**0.99 (0.99, 0.99)**	0.70 (0.67, 0.73)	0.94 (0.92, 0.97)

https://doi.org/10.21528/lnlm-vol24-no1-art3 © Brazilian Society on Artificial Intelligence

4.3 Discussion

The experimental results revealed three key findings: (1) the TCN-AE architecture demonstrated superior performance compared to the WAE-GAN across most devices; (2) the optimized TCN-AE is not only effective but surpasses state-of-the-art models for the 'pump' and 'fan' devices; and (3) the 'valve' device represents a significant challenge for all tested models, including those from the literature. The superior performance of the TCN-AE in detecting anomalies on the slider device (AUC-ROC of 0.96) can be partly explained by insights from hyperparameter optimization. Our analysis of the search space revealed that the performance of TCN-AE is susceptible to its architectural complexity. As illustrated by the correlation analysis in Figure 3, we observed a strong negative correlation between the number of encoder layers and the model's final performance, except for precision. This suggests that simpler, less deep TCN architectures were more effective, likely by avoiding overfitting and creating a more generalized representation of normal operating sounds. The hyperparameter sensitivity analysis shown in Figure 3 and Figure 4 was computed automatically during the grid search execution. Although the evaluations included the passive test set to monitor architectural generalization across configurations, the final model selection was strictly determined by the automated cross-validation average, thereby eliminating human-in-the-loop bias and preventing optimistic parameter tuning.

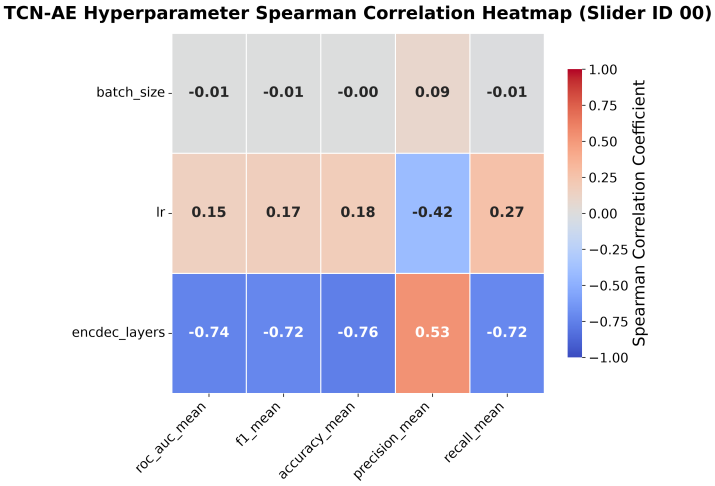


Figure 3: Spearman correlation heatmap for TCN-AE hyperparameters on the 'slider' device. There is a strong negative correlation of encoder layers with performance metrics.

On the other hand, the WAE-GAN exhibited a different behavior. Its performance depended less on architectural depth but was highly sensitive to training dynamics, particularly the learning rate and batch size (Figure 4). This is consistent with the known challenges of training GAN-based models, where achieving stability in the adversarial process is often more critical than the specific network depth. This dependency on training parameters may explain its lower and less consistent performance compared to the more stable TCN-AE.

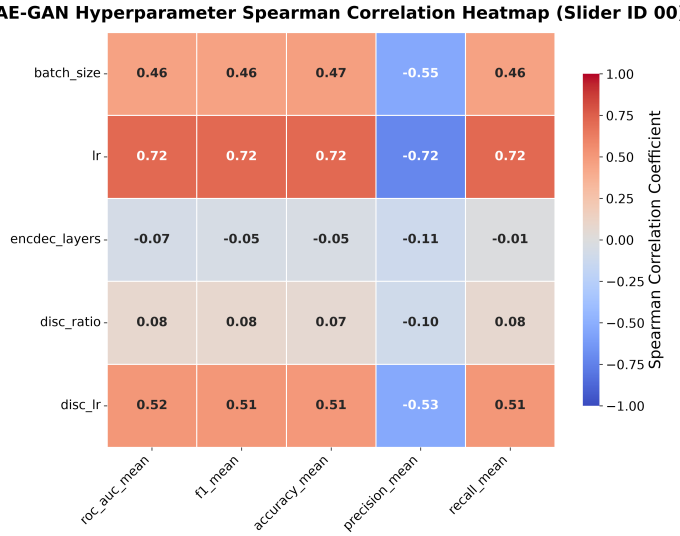


Figure 4: Spearman correlation heatmap for WAE-GAN hyperparameters on the 'slider' device. Performance is strongly correlated with training parameters like 'lr' and 'batch_size'.

https://doi.org/10.21528/lnlm-vol24-no1-art3 © Brazilian Society on Artificial Intelligence

4.4 Industrial Valve Performance Analysis

Finally, the consistent failure of all models on the 'valve' device is a critical finding that warrants a deeper signal processing and physical interpretation. As demonstrated in Table 5, both architectures exhibited an underperformance equivalent to random guessing, with an AUC-ROC of 0.4769 for TCN-AE and 0.4607 for WAE-GAN. From a mechanical perspective, industrial valves operate under highly non-stationary, impulsive, and transient events short in duration, characterized by the rapid mechanical impacts of opening and closing cycles. Under a signal processing perspective, the feature extraction stage based on MFCCs relies on short-time Fourier transforms (STFT) followed by non-linear Mel filter banks, which smooth the acoustic information over fixed temporal windows. Consequently, the fine-grained acoustic signatures and transient high-frequency peaks that may distinguish faulty valve operation can be attenuated and averaged within these windows. This temporal smoothing reduces the separability between normal and anomalous MFCC representations, making reconstruction-based autoencoders less effective for this device and contributing to the observed low metric performance. Future research focusing on valves must diverge from standard MFCC boundaries and explore multi-resolution feature extractions, such as Wavelet Transforms or high-resolution spectrograms capable of capturing short-duration acoustic transients.

4.5 Computational Complexity and Edge Computing Feasibility

To evaluate Industry 4.0 edge viability, we conducted a cost analysis. Table 7 provides the measured overhead parameters.

Table 7: Computational complexity and inference profile comparison for a 10s audio segment.

Metric Attribute	TCN-AE	WAE-GAN
Trainable Parameters	125,716	231,348
Average Training Time per Epoch (s)	0.77 ± 0.37	1.77 ± 1.11
Inference Latency for 10s Audio (ms)	5.94 ± 2.11	11.92 ± 4.82
Model Disk Space Size (MB)	0.49	0.92

The empirical measurements in Table 7 demonstrate an edge deployment advantage for TCN-AE. Requiring only 125,716 trainable parameters (0.49 MB on disk), the model is ideal for microcontrollers. Thanks to its feed-forward convolutional structure, TCN-AE achieves a latency of 5.94 ms, enabling real-time edge diagnostics. Conversely, WAE-GAN requires 231,348 parameters (0.92 MB) and a longer training cycle (1.77 s/epoch) due to the adversarial regularizations, proving that TCN-AE is the more resource-efficient option.

5 Conclusion

This paper presented a comparative analysis of a Temporal Convolutional Network Autoencoder and a Wasserstein Autoencoder with Generative Adversarial Network for sound anomaly detection in industrial machinery. By employing a robust methodology based on hyperparameter optimization and 10-fold cross-validation, we demonstrated that the TCN-AE was the more effective architecture for this task. Our optimized TCN-AE achieved higher AUC-ROC scores of 0.87 (pump) and 0.78 (fan), surpassing literature baselines, and 0.96 on the slider, which is competitive with prior results.

Our primary contribution lies not only in identifying the models with best performance but also in using MFCCs to adapt models traditionally used on multivariate time series for the univariate context. In addition, this work offers insights into why the superior models may perform better. The hyperparameter analysis suggests that the TCN-AE's performance is related to its architectural simplicity, implying that simpler models may be less prone to overfitting on this type of data. In contrast, the WAE-GAN's performance seemed highly dependent on its training parameters. This insight underscores the importance of methodological rigor.

Furthermore, the consistent failure of all models on the 'valve' device strongly suggests that the limitation may not be in the model architecture but in the feature representation. The inability of MFCCs to capture the subtle acoustic signature of the 'valve' anomaly highlights a clear direction for future research. Future work should therefore focus on exploring more advanced feature extraction techniques for the 'valve' device. Additionally, investigating more efficient optimization methods, such as Bayesian Optimization, and applying explainability techniques (*e.g.*, SHAP or LIME) would further enhance the interpretability and trustworthiness of these models for real-world predictive maintenance.

REFERENCES

[1] M. Bevilacqua and M. Braglia. "The analytic hierarchy process applied to maintenance strategy selection". *Reliability Engineering & System Safety*, vol. 70, no. 1, pp. 71–83, 2000.

[2] E. C. Nunes. "Anomalous sound detection with machine learning: A systematic review". *arXiv preprint arXiv:2102.07820*, 2021.

[3] J. Koochaki, J. A. Bokhorst, H. Wortmann and W. Klingenberg. "Condition based maintenance in the context of opportunistic maintenance". *International Journal of Production Research*, vol. 50, no. 23, pp. 6918–6929, 2012.

[4] S. Alaswad and Y. Xiang. "A review on condition-based maintenance optimization models for stochastically deteriorating system". *Reliability engineering & system safety*, vol. 157, pp. 54–63, 2017.

[5] H.-E. Kim, A. C. Tan, J. Mathew, E. Y. Kim and B.-K. Choi. "Machine prognostics based on health state estimation using SVM". *Asset condition, information systems and decision models*, pp. 169–186, 2012.

[6] M. E. Silva, B. Veloso and J. Gama. "Predictive maintenance, adversarial autoencoders and explainability". In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 260–275. Springer, 2023.

[7] Z. Mnasri, S. Rovetta and F. Masulli. "Anomalous sound event detection: A survey of machine learning based methods and applications". *Multimedia Tools and Applications*, vol. 81, no. 4, pp. 5537–5586, 2022.

[8] F. Yu, Y. cai Liu and Q. Zhao. "Compound Fault Diagnosis of Gearbox Based on Wavelet Packet Transform and Sparse Representation Classification". *2020 Chinese Control And Decision Conference (CCDC)*, pp. 5339–5343, 2020.

[9] E. Rushe and B. Mac Namee. "Deep context-aware novelty detection". *arXiv preprint arXiv:2006.01168*, 2020.

[10] R. Monteiro, C. Bastos-Filho, M. Cerrada, D. Cabrera and R.-V. Sánchez. "Convolutional neural networks using fourier transform spectrogram to classify the severity of gear tooth breakage". In *2018 International conference on sensing, diagnostics, prognostics, and control (SDPC)*, pp. 490–496. IEEE, 2018.

[11] V. Chandola, A. Banerjee and V. Kumar. "Anomaly detection: A survey". *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[12] R. Chalapathy and S. Chawla. "Deep learning for anomaly detection: A survey". *arXiv preprint arXiv:1901.03407*, 2019.

[13] R. Zaheer, I. Ahmad, D. Habibi, K. Y. Islam and Q. V. Phung. "A survey on artificial intelligence-based acoustic source identification". *IEEE Access*, vol. 11, pp. 60078–60108, 2023.

[14] M. A. Yahya, A. R. Moya and S. Ventura. "Deep learning for multivariate time series anomaly detection: an evaluation of reconstruction-based methods". *Artificial Intelligence Review*, vol. 58, no. 12, pp. 400, 2025.

[15] C. Silva, M. Coelho, U. Barros, A. M. N. Ribeiro, R. de Paula Monteiro and D. Pinheiro. "A Unified Framework for Replicability of Anomaly Detection in Multiple Time Series: Enhancing GANF and RanSynCoders Pipelines with MFCC Feature Extraction for Acoustic Data". In *2024 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pp. 1–6. IEEE, 2024.

[16] L. Meitz, J. Senge, T. Wagenhals, T. Schöler, J. Hähner, J. Edinger and C. Krupitzer. "A framework and research challenges for predictive maintenance in industry 4.0". *Computers & Industrial Engineering*, p. 111193, 2025.

[17] F. Psarommatis, G. May and V. Azamfirei. "Envisioning maintenance 5.0: Insights from a systematic literature review of Industry 4.0 and a proposed framework". *Journal of Manufacturing Systems*, vol. 68, pp. 376–399, 2023.

[18] H. Purohit, R. Tanabe, K. Ichige, T. Endo, Y. Nikaido, K. Suefusa and Y. Kawaguchi. "MIMII Dataset: Sound dataset for malfunctioning industrial machine investigation and inspection". *arXiv preprint arXiv:1909.09347*, 2019.

[19] D. Pinheiro. "Multiple Time Series Analysis". `https://pypi.org/project/mtsa/`, 2025. [Online; Accessed March 13, 2025].

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] T. Fawcett. "An introduction to ROC analysis". *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.