Enhancing Pharmaceutical Batch Processes Monitoring with Predictive LSTM-Based Framework

Daniele Antonucci[1,2], Davide Bonanni[3], Domenico Palumberi[3], Luca Consolini[4] and Gianluigi Ferrari[4]

[1]*Department of Information Engineering and Architecture, University of Parma, Via delle Scienze 181/a, Italy*

[2]*Department of Electrical and Information Engineering , Politecnico of Bari, Via Re David 200, 70125 Bari, Italy*

[3]*GlaxoSmithKline s.p.a., Strada Provinciale Asolana, 90, San Polo Torrile, Parma, Italy*

[4]*Department of Information Engineering and Architecture, Università di Parma, Via delle Scienze 181/a, Parma, Italy*

Keywords: Anomaly Detection, Predictive Maintenance, Machine Learning, Artificial Intelligence, AutoEncoder, Time-Series Forecasting, Hyperparameter Optimization, Fault Detection.

Abstract: Monitoring industrial processes and understanding deviations is critical in ensuring product quality, process efficiency, and early detection of anomalies. Traditional methods for dimensionality reduction and anomaly detection, such as Principal Component Analysis (PCA) or Partial Least Squares (PLS), often struggle to capture the complex and dynamic nature of batch data. In this study, we propose a novel approach that combines an AutoEncoder (AE), based on Long Short-Term Memory (LSTM) layers, with a rolling threshold for anomaly evaluation. Unlike conventional threshold methods that rely on global statistical parameters, the applied threshold leverages rolling median and rolling Median Absolute Deviation (MAD) to adaptively detect deviations, making it more resilient to outliers and distribution shifts. The LSTM-AE demonstrates superior performance in anomaly detection with respect to PCA and more recent model approaches, specifically for the reference dataset, obtained from a GlaxoSmithKline (GSK) production plant. Additionally, an LSTM regression model is employed to forecast future data points, which are then fed into the LSTM-AE to enable a predictive approach. This framework leverages the temporal dependencies captured by LSTM layers and reconstruction efficiency of the AE, facilitating a predictive anomaly detection in real-world applications.

1 INTRODUCTION

An anomaly is an unexpected deviation from normal system behavior, representing data points or events that stray from the operational baseline. These anomalies can indicate critical issues—such as faults, errors, or fraudulent activities—that may lead to degraded performance, failures, safety risks, or financial losses. Pharmaceutical processes, in particular, are complex and strictly regulated. Anomalies in these settings can result from equipment malfunctions, environmental changes, human errors, or variations in raw material properties, making swift detection essential for maintaining process integrity. However, detecting anomalies is particularly challenging in batch operations, which involve dynamic, stage-specific behaviors and batch-to-batch variability (see (Majozi, 2009)). An anomaly in one stage might be typical in another, further complicating detection. Additionally, batch-to-batch variability—driven by factors like raw material quality, environmental conditions, or equipment wear—further obscures the identification of subtle anomalies (Mockus et al., 2015).

Over the past decade, various data-driven methods have been employed for anomaly prediction. Traditional statistical techniques, such as PCA (Greenacre et al., 2022) and PLS (Pirouz, 2006), have been widely used, even though their linear nature limits their ability to capture non-linear interactions in batch data. Extensions like kPCA (Schölkopf et al., 1997) and non-linear PLS have been developed but often come with high computational costs. Recent advances in Machine Learning (ML) and Deep Learning (DL)—particularly AutoEncoders (AEs) and Recurrent Neural Networks (RNNs) (Aghaee et al., 2024)—have shown promise in capturing both non-linear patterns and temporal dependencies in dynamic systems.

Antonucci, D., Bonanni, D., Palumberi, D., Consolini, L. and Ferrari, G.
Enhancing Pharmaceutical Batch Processes Monitoring with Predictive LSTM-Based Framework.
DOI: 10.5220/0013665500003982

In *Proceedings of the 22nd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2025) - Volume 1*, pages 15-24
ISBN: 978-989-758-770-2; ISSN: 2184-2809

This paper introduces a robust framework that combines LSTM and AE methodologies (detailed in Section 3) for batch process monitoring. It demonstrates improved performance in managing the complex dynamics of batch process monitoring in comparison to traditional approaches (discussed in Section 4.1). Furthermore, a customized threshold mechanism, based on rolling median and Median Absolute Deviation (MAD) is implemented to enhance anomaly detection accuracy and reduce false positives. Notably, the framework also employs an LSTM regression model to predict future process variables, which are subsequently fed into the AE to enable predictive anomaly detection.

Paper Outline

In Section 2, we discuss existing approaches for anomaly detection in batch processes, including statistical and ML-based methods, and we outline the motivation behind our proposed framework. Section 3 introduces the architecture, implementation, and integration of our framework composed by an LSTM-AE model alongside the LSTM regression model, designed for both real-time monitoring and prediction. The training and testing of these models utilize batch process data sourced from the pharmaceutical company GSK (as detailed in Section 3.1). Section 4 describes the experimental setup, evaluation metrics, and provides a comparative analysis of our approach against alternative methods. Finally, Section 5 concludes the paper by summarizing key findings, discussing their implications, and suggesting future research.

2 LITERATURE REVIEW

Despite the inherent non-linearity of industrial processes, PCA remains popular for process modeling due to its simplicity and ease of use (Russell et al., 2000). An alternative for batch processes, as shown in (Jeffy et al., 2018), employs a multi-way PCA technique by aligning and concatenating batches into an unfolded 2D matrix, suited for following transformations. The advantages of PCA in industrial applications include robust irregularity detection even with sparse data, scalability for process efficiency, and ease of interpretation for real-time monitoring and control.

Transitioning from linear to non-linear methods introduces challenges in deploying optimized, interpretable models for fault detection in highly non-linear systems. Non-linear techniques offer enhanced modeling capabilities but often demand greater computational resources and larger training datasets. For instance, Kernel PCA (kPCA)—presented in (Choi et al., 2005)—addresses linear PCA's limitations by modeling complex non-linear relationships. However, kPCA requires significant computational resources due to the need to compute and store a kernel matrix that scales quadratically with the number of samples, and its high-dimensional feature space can complicate interpretation.

Building on these non-linear techniques, methods such as one-class Support Vector Machine (SVM) (Li et al., 2003) and Support Vector Data Description (SVDD) (Zhao et al., 2013) have been adopted to enhance anomaly detection. Both these methods use kernel functions to map data into a high-dimensional space based solely on fault-free samples, with a key distinction in the boundary: one-class SVM constructs a hyperplane while SVDD defines a hypersphere. In (Inoue et al., 2017), one-class SVM is compared with Deep Neural Networks (DNNs) for detecting anomalies in the context of a water treatment plant. The study demonstrates that while both methods can be effective, each has its own trade-offs in terms of false positives and sensitivity to different fault scenarios. In (Kilickaya et al., 2024), a deep variant of SVDD is developed to detect anomalies in industrial machinery based on audio signals. By mapping log-Mel spectrograms into a feature space and learning a compact hypersphere that encloses normal behavior, the method achieves excellent detection performance under various noise conditions. These studies not only demonstrate how one-class SVM and SVDD can effectively model normal operational states and identify deviations as anomalies across various types of industrial data, but they also pave the way for the adoption of kernel-based and DL approaches in anomaly detection and dimensionality reduction tasks.

In the past decade, AEs have emerged as one of the most effective methods for anomaly detection in non-linear systems due to their ability to learn compact and expressive representations of complex data (Sakurada and Yairi, 2014). By reconstructing the underlying data distribution through a decoder, AE-based models can inherently detect deviations from learned patterns—a feature particularly valuable in batch operations. Studies such as (Said Elsayed et al., 2020) and (Nguyen et al., 2021) have demonstrated the potential of AE-based approaches to enhance fault detection and process monitoring.

Statement of Contribution

AEs serve two main functions: (I) reconstruction-based detection and (II) prediction (Liu et al., 2023). In this work, we introduce a framework that combines an LSTM-AE for reconstruction-based detection with a separate LSTM regression model for variable prediction. The models are trained on a batched dataset with variable batch lengths, reorganized into a two-dimensional matrix that preserves the temporal structure without using interpolation or padding. This approach tackles the challenges posed by nonlinear dynamics, high dimensionality, and temporal dependencies in batch data. Although metrics such as Hotelling T^2 and Squared Prediction Error (SPE) are used for outlier detection in linear models (Zeng et al., 2019), Hotelling T^2 is rarely applied with AEs because its assumptions—linear relationships, Gaussian latent distributions, and a well-defined covariance matrix—do not hold in DNNs. Therefore, we utilize a non-parametric threshold computed using a rolling median and rolling MAD for more reliable anomaly detection.

Overall, our framework is capable of: (I) achieving robust reconstruction by emphasizing key features while filtering out noise; (II) implementing an anomaly detection mechanism that improve the detection of anomalies and deterioration over time and reduces false positives; and (III) capturing temporal dependencies and long-term patterns by predicting process variables.

3 PROPOSED MODEL

This section presents our framework for anomaly detection and prediction in batch processes, detailing the data preprocessing steps and overall data flow (see Figure 1). The framework starts with raw data, which is initially standardized using a scaling method. The data then proceeds through two training phases: one for the LSTM regression model and another for the LSTM-AE. For the regression model, data is formatted with past and future steps to accurately forecast sequences of points. For the AE, the reconstructed output is subsequently evaluated against a threshold to detect anomalies.

3.1 Data Architecture

The dataset for this study comes from a GSK manufacturing process that produces proteins for a biologic drug. In this process, protein cells are cultivated in large bioreactors under precisely controlled

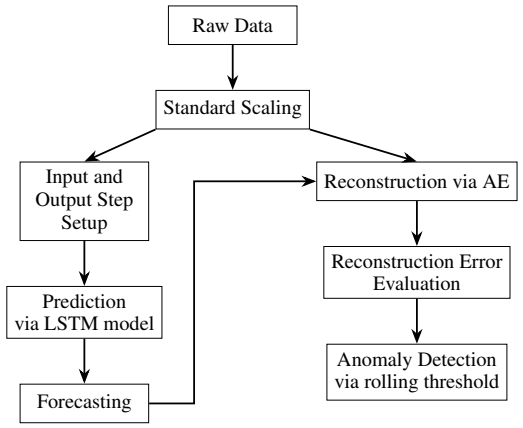

Figure 1: Data flow of both prediction and reconstruction phases.

conditions—such as pressure, pH, and oxygen levels—to ensure optimal cell growth. Once the cells produce sufficient proteins, they are extracted, purified through a series of filtration and chromatography steps to eliminate impurities, and then rigorously tested for quality and safety. Each variable must operate within its designated Normal Operating Condition (NOC) to maintain the process's integrity. A preliminary variable selection, conducted by the company, has identified critical process variables—referred to as dynamic variables—for monitoring. These include measurements from pH sensor and controller, Dissolved Oxygen sensor and controller, Vessel Weight, Vessel Pressure, and levels of Air, CO_2, O_2, and N_2, making a total of 10 key variables. Given the multivariate analysis and prediction, the pH signal (shown in Figure 2) was chosen as the reference signal since it exhibits the highest variance among all signals.

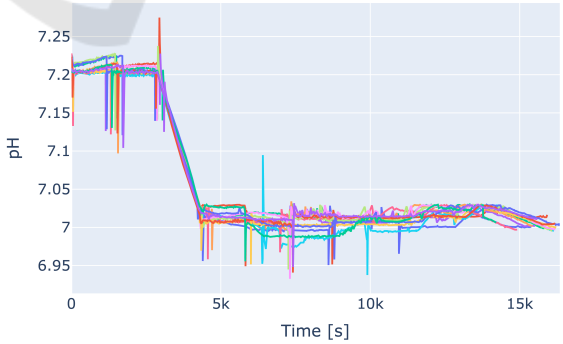

Figure 2: Representation of pH sensor readings across different batches, with varying signal durations.

The data is structured in batches, with each batch representing a distinct production cycle and capturing time series data from multiple sensors, with sampling period equal to 5 minutes. Although each

batch follows a similar pattern, subtle variations exist, requiring the model to detect anomalies by discerning these differences. The dataset comprises 14 training batches and 6 test batches, all adhering to the process's NOCs. As noted earlier, each batch terminates upon reaching the required protein quantity, resulting in variable batch lengths. Aligning these batches is challenging, often requiring the introduction of artificial sequences of points to standardize the lengths. However, this process can distort the original data distribution, introducing more noise or misleading patterns that may compromise the model's ability to learn meaningful representations and, ultimately, reduce the reliability of anomaly detection. To address this issue, we can employ an unfolding technique commonly used in PCA applications (Lee et al., 2004). This approach transforms a 3D data matrix (batches I × time steps K × variables J) into a 2D matrix ([batches I × time steps K] × variables J), preserving the temporal dependencies within each batch while enabling cross-batch analysis to detect trends or anomalies. By using this method, the model can effectively process both temporal and batch-level patterns, thus enhancing pattern recognition and anomaly detection. As illustrated in Figure 3, batch-wise unfolding—where variable differences between consecutive time steps are analyzed—can only be applied to batches of equal length. In contrast, variable-wise unfolding highlights inter-batch patterns, helping to preserve and reveal deviations across batches.

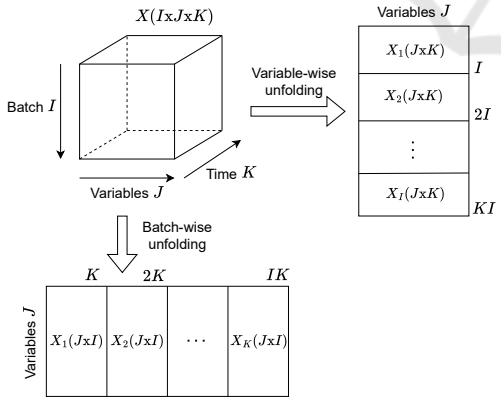

Figure 3: Unfolding approaches for batch processes.

3.2 Data Preprocessing

To avoid an unbalanced training phase, appropriate data preprocessing is required. In this work, the only required preprocessing step is feature scaling using `StandardScaler`, as the dataset already consists of pre-selected critical process sensors, ensuring that only the most relevant features are included. `StandardScaler` standardizes the data by subtracting the mean and scaling to unit variance, placing all features on a comparable scale. The transformation is defined as:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

where x represents the original feature signal, μ is the mean, and σ is the variance. This standardization prevents features with different scales or units from disproportionately influencing the model, thereby improving training convergence and enhancing result interpretability (Ahsan et al., 2021).

3.3 AutoEncoder (AE)

The AE is an unsupervised, feed-forward neural network that consists of two main components: an encoder and a decoder. The *encoder* compresses high-dimensional input data into a lower-dimensional latent space, effectively extracting the most relevant information. The *decoder* then reconstructs the original data from this compact representation, yielding a similar version of the input data. The model is trained to minimize reconstruction error, which serves as a measure of data fidelity. Additionally, the size of the latent space plays a crucial role in the performance of AEs, as more aggressive compression can lead to a greater loss of information.

During the anomaly detection phase, the AE evaluates both actual and predicted data points. It continuously monitors reconstruction errors: deviations that exceed an established threshold are flagged as potential anomalies. Figure 4 illustrates the proposed model implementation, with the hidden layers and activation functions defined following the optimization phase (detailed in Section 3.5). Figure 5 demonstrates that the model accurately reconstructs the original signals with overall minimal reconstruction error. However, the error increases in regions where the model struggles to capture rapid variations—such as infrequent peaks or spikes— or sequence variations that were not common in the training batches.

3.4 Long Short-Term Memory (LSTM) Regression Model

LSTM networks are a specialized type of RNN well-suited for modeling sequence data, including time-series. They are designed to overcome the vanishing and exploding gradient problems in traditional RNNs (Noh, 2021), enabling the learning of long-range temporal patterns. As a result, LSTMs have become one

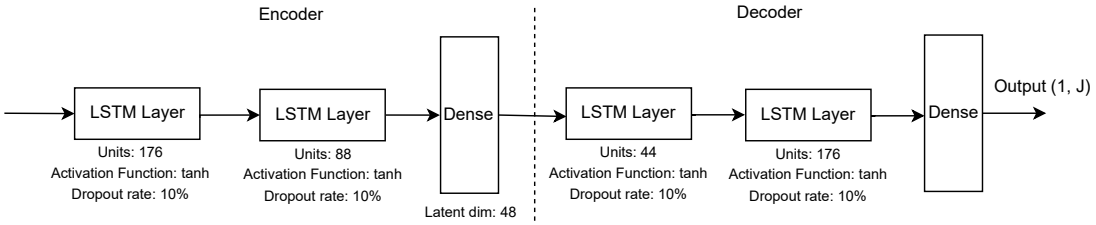

Figure 4: Autoencoder structure.

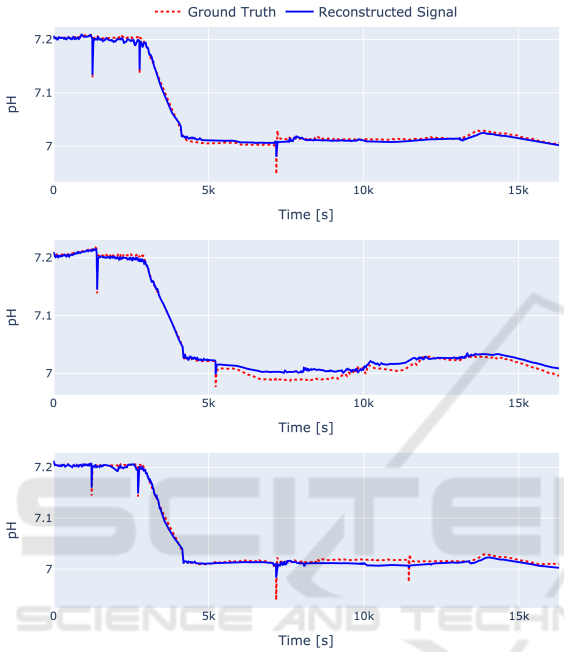

Figure 5: Comparison of AE's reconstruction ability of different pH signals.

of the most popular and effective RNN architecture for time-series forecasting (Torres et al., 2021). Numerous studies have adopted LSTM-based models to predict time-dependent data, demonstrating robust results across diverse application domains (Kong et al., 2024).

In our framework, the dataset is also fed into the LSTM model. With its inherent capability to capture long-term dependencies, the model learns a mapping from sequences of past observations (inputs) to a continuous target variable (output). Traditional ML models often do not support multi-output prediction, since they are optimized and designed for one-target case. Input and output lags can be particularly beneficial for batch processes, where understanding both recent changes and long-term trends is critical for monitoring and control.

3.5 Parameter Optimization

The performance of Neural Network (NN) models hinges on their architecture and hyperparameters. To find the best optimal set of parameters, HyperParameter Optimization (HPO) is essential. Bayesan Optimization (BO) is a Sequential Model-Based Optimization (SMBO) technique well-suited for tuning expensive black-box functions, such as those encountered in DL. In BO, a surrogate model—often a Gaussian Process (GP)—is used to approximate the true objective function. Since evaluating the objective function (for instance, training an ML model) is resource-intensive, the surrogate model significantly reduces computational costs by providing an efficient estimation. BO excels in exploring complex, high-dimensional parameter spaces by effectively balancing exploration and exploitation (Wu et al., 2019). These benefits are particularly evident when compared to traditional methods like grid search and random search. Grid search becomes computationally prohibitive as the number of hyperparameters increases due to exponential growth in evaluations, while random search can inefficiently allocate resources by sampling suboptimal regions without focusing on promising configurations. The reconstruction capability of the AE and the prediction accuracy of the LSTM model are optimized using BO, using GP as the surrogate model. The objective is to minimize the Mean Squared Error (MSE) between the input data and the target output—the AE strives to reproduce the input data accurately and the LSTM aims at forecasting future data points effectively. Formally, the optimization problem is defined as follows:

$$\theta^{\star} = \arg\min_{\theta} f(\theta) \tag{2}$$

$$f(\theta) = \mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N} \|x_i - \hat{x}_i\|^2 \tag{3}$$

where: $f(\theta)$ is the objective function to be minimized; θ is the hyperparameter configuration vector; x_i and $\hat{x}_i$ represent the ground truth and reconstructed values for each sample i; and N is the total number of samples (with each sample corresponding to a set of

sensor readings). The optimal configuration $\theta^\star$ is the one that yields the lowest observed objective value. To approximate the objective function, the following GP surrogate model is employed:

$$\hat{f}_n(\theta) \sim \mathcal{GP}\big(m(\theta),\, k(\theta,\theta')\big) \tag{4}$$

where: n is the current optimization; $m(\theta) = 0$ (normalized data) is the mean function; and the covariance function is defined as:

$$k(\theta,\theta') = \sigma^2 \left(1 + \frac{\sqrt{5}\,r}{\ell} + \frac{5r^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}\,r}{\ell}\right) \tag{5}$$

where: $r = \|\theta - \theta'\|$; $v > 0$ is the smoothness parameter (normally $v = 2.5$); ℓ is the length-scale parameter, which scales the Euclidean distance r; and σ is the variance. Once the surrogate model provides an estimate, a new set of hyperparameters is chosen to further minimize the objective function. This selection is guided by an acquisition function, typically the Expected Improvement (EI) function, defined as:

$$\mathrm{EI}(\theta) = \mathbb{E}\left[\max\left(0, f_{\text{best}} - \hat{f}_n(\theta)\right)\right] \tag{6}$$

where $f_{\text{best}} \triangleq \min_{i=1,\ldots,n} f(\theta_i)$ is the best observed value of the objective function up to the current iteration. In this study, the following hyperparameters were optimized, with the MSE as the loss function.

- **Model Depth:** determines the number of layers of the model.
- **Units per Layer:** specifies the number of neurons in each layer.
- **Activation Function:** defines the transformation applied at each layer.
- **Learning Rate:** sets the step size for the optimization algorithm.
- **Latent Dimension:** specifies the number of neurons in the encoder's final layer, defining the size of the latent space into which the input is mapped.

The optimal hyperparameter configuration for the training dataset is presented in Figure 4, with learning rate equal to $8.6e^{-4}$. This configuration enables the AE to achieve efficient performance as described in Section 4.2.

3.6 Threshold Computation

In unsupervised anomaly detection models (e.g., autoencoders), selecting an appropriate threshold for the reconstruction error is critical to distinguish normal variations from true anomalies. One straightforward strategy is to define the anomaly cutoff at a high percentile of the reconstruction error distribution obtained from training data. This approach ensures that only a small fraction (e.g., 1%) of normal data would be mistakenly classified as anomalies by design. This should prioritize catching extreme outliers while limiting false alarms (Sabzehi and Rollins, 2024). For multivariate anomaly scoring (or when considering the vector of reconstruction errors across multiple features), thresholding can be based on the Mahalanobis Distance (MD) (Ghorbani, 2019). The MD measures how far a point is from the center of a distribution while accounting for the covariance structure of the data. The Mahalanobis thresholding approach has the advantage of capturing correlations among variables (or error components), making it more sensitive to unusual combinations of feature values that univariate methods might miss. However, it assumes a reasonably well-estimated covariance matrix; in high dimensions or with limited data, robust covariance estimation or dimensionality reduction may be necessary to apply this method effectively.

In this study, we employ an adaptive threshold technique based on rolling median and MAD of the reconstruction error. The rolling median serves as a local baseline of "normal" behavior, while the rolling MAD provides a scale of typical variability in that period. We then flag a data point as anomalous if its reconstruction error deviates from the current median by more than a chosen factor times the MAD. In other words, the threshold at time t is defined as:

$$\begin{aligned} M_t &= \text{median}\{x_j \mid j \in W(t)\} \\ \text{MAD}_t &= \text{median}\{|x_j - M_t| \mid j \in W(t)\} \\ \tau_t &= M_t + k \times \text{MAD}_t, \end{aligned} \tag{7}$$

where: $W(t)$ denotes the set of indices within the rolling window at time t; and k is a scaling factor that adjusts the threshold to suit the dataset's specific requirements. This rolling median/MAD approach yields a time-varying threshold that can adapt to gradual shifts or trends in the data while still being resistant to short-term spikes. It provides a simple, non-parametric way to detect deviations that are extremes with respect to local normal variations by carefully setting the scaling factor and the window size based on data distribution. For our dataset: k is set to 10 to catch anomalies that are not mistaken for new normal observations, caused by frequent spikes in the training data, which may lead to increased reconstruction error; and the length of W equal to 30 to account for local variations.

3.7 Evaluation Metrics

To evaluate the performance of the proposed model, we utilize the following two metrics.

- **Mean Squared Error (MSE)**: mainly used in the optimization and training phase, it measures the average squared difference between the predicted values ($\hat{y}_i$) and the true values (y_i). It can be expressed as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{8}$$

  where N is the number of samples.

- **Area Under the Curve Receiver Operating Characteristic (AUCROC)**: it represents the model's ability to distinguish between positive and negative classes. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various thresholds. The AUC is calculated as:

$$\text{AUC} = \int_0^1 \text{TPR}\, d(\text{FPR}) \tag{9}$$

  where:

$$\begin{aligned} \text{TPR} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \quad \text{(True Positive Rate)} \\ \text{FPR} &= \frac{\text{FP}}{\text{FP}+\text{TN}} \quad \text{(False Positive Rate)} \end{aligned} \tag{10}$$

  The AUCROC is a single scalar value ranging from 0 to 1, where 1 indicates a perfect distinction between normal and anomaly points.

These metrics, together, provide a comprehensive evaluation of the model's accuracy and its ability to explain the variance in the data, as shown in Section 4.1.

4 RESULTS

In order to provide an overall performance comparison between our proposed AE and both traditional and recent approaches, we leverage ADBench (Han et al., 2022), a comprehensive public benchmark for anomaly detection. ADBench evaluates the performance of 30 anomaly detection algorithms, of which 14 are unsupervised, across 57 datasets, encompassing a wide variety of real-world and synthetic scenarios. In this benchmark, anomalies are simulated via four distinct mechanisms: (I) local anomalies, which deviate from the patterns of their immediate neighborhoods; (II) global anomalies, generated by sampling from a uniform distribution; (III) dependency anomalies, where the natural correlations among input features are deliberately disrupted; and (IV) clustered anomalies, in which anomalous points occur in concentrated groups. This setup allows for a thorough evaluation of model performance under diverse conditions, offering valuable insights of anomaly detection capabilities.

4.1 Anomaly Detection with Generic Dataset

As illustrated in Table 1, the average AUCROC scores for various unsupervised models—evaluated on datasets characterized by distinct anomaly types—demonstrate that our approach consistently delivers robust results even without specialized tuning. Moreover, further performance improvements are expected following HPO. For instance, while models like KNN excel at detecting independent anomalies and PCA proves effective for clustered anomalies, both may struggle when confronted with complex, interrelated anomaly patterns.

4.2 Anomaly Detection with GSK Dataset

To assess ADBench's models using our dataset, synthetic anomalies must be incorporated into the test set since the original data contains no outliers and cannot be directly imported into the benchmark without anomalies. Specifically, for each synthetic anomaly, we randomly select one feature and one time step, and inject anomalies corresponding to 1% of the test data size. At the chosen time step, a spike is introduced—its magnitude is determined by the data's standard deviation and scaled by a predefined deterioration factor. Table 2 shows AUCROC score of each model with the custom dataset.

4.3 Batch Prediction & Reconstruction

Based on the data flow depicted in Figure 1, data points are analyzed in two distinct approaches: (I) prediction followed by detection, and (II) real-time detection. For prediction, the window corresponding to each input lag is fed into the LSTM model, which then produces the subsequent outputs, defined by the output lag. To analyze various combinations of input and output values, a grid search is performed over a range of potential input-output lag pairs, depending on the sampling time of 5 minutes. For each pair, a BO optimization is carried out to facilitate a comparative evaluation of the results. By looking at Table 3,

Table 1: Average AUCROC evaluated on 57 datasets across five anomaly types: Default, Local, Global, Dependency, and Cluster. For each type of anomaly, the best value is highlighted.

Model	Type of Anomaly				
	Original	Local	Global	Dependency	Cluster
IForest	**0.7349**	0.8859	0.9973	0.7820	0.9680
OCSVM	0.6922	0.8618	0.9871	0.6238	0.9569
CBLOF	0.7396	0.8918	0.9970	0.8357	0.8769
COF	0.6437	0.9065	0.9493	0.8877	0.5211
COPOD	0.7177	0.8557	0.9907	0.6065	0.9681
ECOD	0.7187	0.8785	0.9908	0.5950	0.9447
HBOS	0.7122	0.8527	0.9931	0.5982	0.9668
KNN	0.7058	0.9117	**0.9991**	0.8959	0.8339
LOF	0.6384	**0.9359**	0.9189	0.8814	0.4446
PCA	0.7194	0.8662	0.9933	0.6288	**0.9782**
SOD	0.6880	0.8703	0.9889	0.8940	0.7526
DeepSVDD	0.5612	0.6142	0.7507	0.6133	0.5652
DAGMM	0.6277	0.7927	0.9164	0.6526	0.9354
Proposed AE	0.7242	0.8968	0.9867	**0.8984**	0.8586

Table 2: AUCROC score with GSK batch dataset.

Method	AUCROC
IForest	0.7555
OCSVM	0.6957
CBLOF	0.9107
COF	0.9581
COPOD	0.6721
ECOD	0.6985
HBOS	0.6643
KNN	0.9766
LOF	0.9772
PCA	0.6702
SOD	0.8932
DeepSVDD	0.6297
DAGMM	0.6570
Proposed AE	**0.9844**

it is evident that the error increases as the output lag grows with a fixed input lag, caused by the increasing number of predictions and inherently difficulty of multi-point prediction. Although extending the input history does not linearly enhance the model's learning capacity, our findings indicate that a 30-minute input history (six data points) is sufficient to forecast 15 minutes ahead (three data points). In contrast, predicting 1 hour (twelve data points) accurately requires an input history of 1 hour and 30 minutes (eighteen data points). Figure 6 demonstrates that accuracy drops as additional outputs are predicted, with a notable drop in performance after 30 minutes, after which the predictions tend to become stationary. Moreover, Figure 7 shows the reconstructed pH signal from a 5-minute prediction step along with its threshold evaluation. Notably, the average AUCROC across the test set with synthetic anomalies is 0.8040, 0.5541, and 0.5507 for the rolling threshold, percentile, and MD methods, respectively.

Table 3: Batch prediction MSE on test data based on input and output lags.

		Output Predictions			
		1	3	6	12
Input Lag	6	0.0036	**0.0061**	0.0099	0.0209
	12	**0.0033**	0.0067	0.0098	0.0211
	18	**0.0033**	0.0063	0.0094	**0.0191**
	24	**0.0033**	0.0075	**0.0089**	0.0203

5 CONCLUSIONS

In this study, we introduced an anomaly detection approach that leverages an LSTM-AE for real-time monitoring in batch processes. The proposed framework addresses the challenges of non-linear dynamics, high dimensionality, and temporal dependencies by using reconstruction error-based detection with a rolling threshold. This method robustly preserves essential information while reducing false positives and detecting gradual deterioration. Validated in real-world scenarios, the LSTM-AE shows promise as an alternative to traditional and ML approaches for identifying subtle, complex anomalies in industrial batch processes. Furthermore, a NN model, based on LSTM layers, is integrated to forecast future data points by analyzing both historical trends and anticipated future steps, providing an accurate prediction of at least 15 minutes ahead. The predictions are then fed into the LSTM-AE, further enhancing its abil-

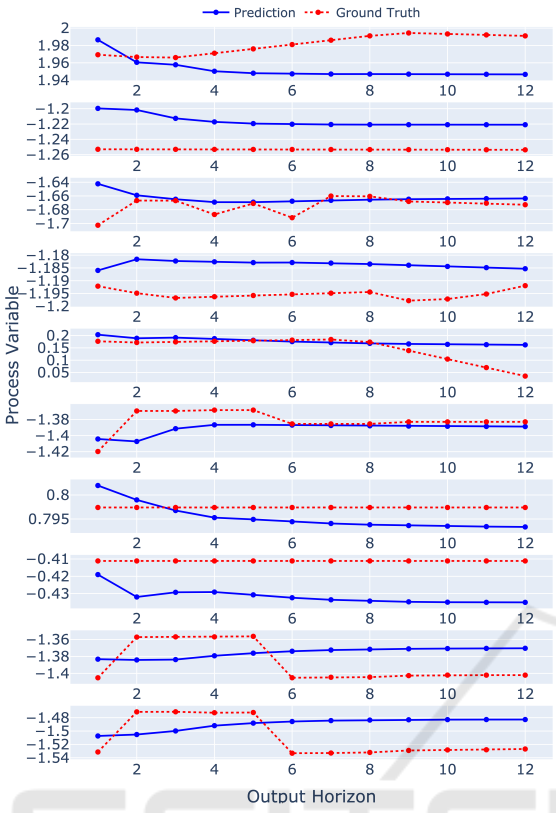

Figure 6: Predicted variables across 12 output steps (1 hour) with 1 hour and 30 minutes of input history.

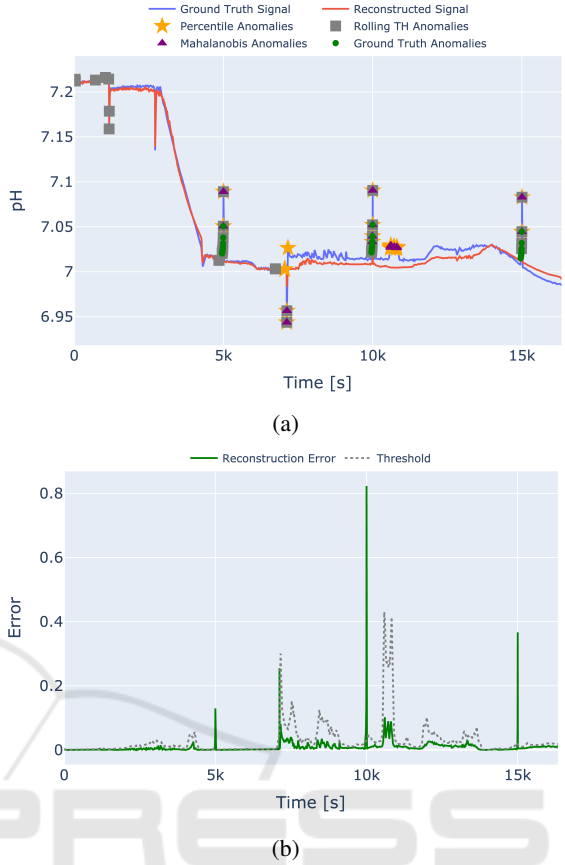

Figure 7: (a) Comparison of selected threshold mechanism between predicted pH signal and its reconstruction with synthetic anomalies; (b) reconstruction error with the rolling threshold.

ity to anticipate deviations. This work underscores the potential of DL techniques to revolutionize process monitoring and anomaly detection with advanced predictive capabilities. Future research could explore latent space analysis for improved anomaly prediction, incremental learning with real-time data, and broader deployment across various processes by analyzing critical process variables. Finally, this framework may offer valuable insights in real-time industrial environments into its operational efficiency and scalability, especially when real faults or anomalies are provided to increase model's knowledge, tweaking the threshold accordingly.

ACKNOWLEDGEMENTS

This work is based on a project funded by: (I) the European Union – Next Generation Eu - under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 3.3 - Decree No. 352 (09th April 2022) of Italian Ministry of University and Research - Concession Decree No. 2153 (28th December 2022) of the Italian Ministry of University and Research, Project code D93D22001390001, within the Italian National Program PhD Programme in Autonomous Systems (DAuSy); and (II) GlaxoSmithKline Manufacturing S.p.a.

CONFLICT OF INTEREST

Two of the authors are employees of the GSK group of companies. This was undertaken at the request of and sponsored by GlaxoSmithKline Biologicals SA. The remaining authors declare that they have no competing interests.

REFERENCES

Aghaee, M., Mishra, A., Krau, S., Tamer, I. M., and Budman, H. (2024). Artificial intelligence applications for fault detection and diagnosis in pharmaceutical bioprocesses: a review. *Current Opinion in Chemical Engineering*, 44:101025.

Ahsan, M. M., Mahmud, M. P., Saha, P. K., Gupta, K. D., and Siddique, Z. (2021). Effect of data scaling methods on machine learning algorithms and model performance. *Technologies*, 9(3):52.

Choi, S. W., Lee, C., Lee, J.-M., Park, J. H., and Lee, I.-B. (2005). Fault detection and identification of nonlinear processes based on kernel pca. *Chemometrics and intelligent laboratory systems*, 75(1):55–67.

Ghorbani, H. (2019). Mahalanobis distance and its application for detecting multivariate outliers. *Facta Universitatis, Series: Mathematics and Informatics*, pages 583–595.

Greenacre, M., Groenen, P. J., Hastie, T., d'Enza, A. I., Markos, A., and Tuzhilina, E. (2022). Principal component analysis. *Nature Reviews Methods Primers*, 2(1):100.

Han, S., Hu, X., Huang, H., Jiang, M., and Zhao, Y. (2022). Adbench: Anomaly detection benchmark. *Advances in neural information processing systems*, 35:32142–32159.

Inoue, J., Yamagata, Y., Chen, Y., Poskitt, C. M., and Sun, J. (2017). Anomaly detection for a water treatment system using unsupervised machine learning. In *2017 IEEE international conference on data mining workshops (ICDMW)*, pages 1058–1065. IEEE.

Jeffy, F., Gugaliya, J. K., and Kariwala, V. (2018). Application of multi-way principal component analysis on batch data. In *2018 UKACC 12th International Conference on Control (CONTROL)*, pages 414–419. IEEE.

Kilickaya, S., Ahishali, M., Celebioglu, C., Sohrab, F., Eren, L., Ince, T., Askar, M., and Gabbouj, M. (2024). Audio-based anomaly detection in industrial machines using deep one-class support vector data description. *arXiv preprint arXiv:2412.10792*.

Kong, Y., Wang, Z., Nie, Y., Zhou, T., Zohren, S., Liang, Y., Sun, P., and Wen, Q. (2024). Unlocking the power of lstm for long term time series forecasting. *arXiv preprint arXiv:2408.10006*.

Lee, J.-M., Yoo, C., and Lee, I.-B. (2004). Fault detection of batch processes using multiway kernel principal component analysis. *Computers & chemical engineering*, 28(9):1837–1847.

Li, K.-L., Huang, H.-K., Tian, S.-F., and Xu, W. (2003). Improving one-class svm for anomaly detection. In *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693)*, volume 5, pages 3077–3081. IEEE.

Liu, M., Zhu, T., Ye, J., Meng, Q., Sun, L., and Du, B. (2023). Spatio-temporal autoencoder for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):5516–5526.

Majozi, T. (2009). Introduction to batch chemical processes. *Batch Chemical Process Integration: Analysis, Synthesis and Optimization*, page 1–11.

Mockus, L., Peterson, J. J., Lainez, J. M., and Reklaitis, G. V. (2015). Batch-to-batch variation: a key component for modeling chemical manufacturing processes. *Organic Process Research & Development*, 19(8):908–914.

Nguyen, H. D., Tran, K. P., Thomassey, S., and Hamad, M. (2021). Forecasting and anomaly detection approaches using lstm and lstm autoencoder techniques with the applications in supply chain management. *International Journal of Information Management*, 57:102282.

Noh, S.-H. (2021). Analysis of gradient vanishing of rnns and performance comparison. *Information*, 12(11):442.

Pirouz, D. M. (2006). An overview of partial least squares. *Available at SSRN 1631359*.

Russell, E. L., Chiang, L. H., and Braatz, R. D. (2000). Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 51(1):81–93.

Sabzehi, M. and Rollins, P. (2024). Enhancing rover mobility monitoring: Autoencoder-driven anomaly detection for curiosity. In *2024 IEEE Aerospace Conference*, pages 1–7. IEEE.

Said Elsayed, M., Le-Khac, N.-A., Dev, S., and Jurcut, A. D. (2020). Network anomaly detection using lstm based autoencoder. In *Proceedings of the 16th ACM symposium on QoS and security for wireless and mobile networks*, pages 37–45.

Sakurada, M. and Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, pages 4–11.

Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer.

Torres, J. F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., and Troncoso, A. (2021). Deep learning for time series forecasting: a survey. *Big data*, 9(1):3–21.

Wu, J. et al. (2019). Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology*, 17(1):26–40.

Zeng, L., Long, W., and Li, Y. (2019). A novel method for gas turbine condition monitoring based on kpca and analysis of statistics t2 and spe. *Processes*, 7(3):124.

Zhao, Y., Wang, S., and Xiao, F. (2013). Pattern recognition-based chillers fault detection method using support vector data description (svdd). *Applied Energy*, 112:1041–1048.