Enhancing Predictive Maintenance in Water Treatment Plants through Sparse Autoencoder Based Anomaly Detection

[1]Hussein Z. Almngoshi, [2]Balaji V, [3]Ramesh R, [4]Arokia Jesu Prabhu L, [5]Venubabu Rachapudi and [6]Eswaramoorthy V
[1]Department of Computer Science, College of Computer Science and Information Technology, University of Kerbala, Karbala, Iraq.
[2]Department of Computer Science and Engineering, GITAM University, Bengaluru, Karnataka, India.
[3]Department of Science and Humanities (Mathematics), Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India.
[4]Department of Computer Science and Engineering Artificial Intelligence & Machine Learning, CMR Institute of Technology, Hyderabad, Telangana, India.
[5]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.
[6]Department of Information Technology, Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India.
[1]hussein.z@uokerbala.edu.iq, [2]balajipucs@gmail.com, [3]rameshrajappa1982@gmail.com, [4]arokiajeruprabhu@gmail.com, [5]venubabu.r@gmail.com, [6]eswarinfotech@gmail.com

Correspondence should be addressed to Balaji V : balajipucs@gmail.com.

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202404027
Received 25 July 2023; Revised from 24 September 2023; Accepted 18 February 2024.
Available online 05 April 2024.

Abstract – The deployment of Machine Learning (ML) for improving Water Treatment Plants (WTPs) predictive maintenance is investigated in the present article. Proactively detecting and fixing functional difficulties which might cause catastrophic effects has historically been an endeavour for reactive or schedule-based maintenance methods. Anomaly Detection (AD) in WTP predictive maintenance frameworks is the primary goal of this investigation, which recommends a novel approach based on autoencoder (AE)-based ML models. For the objective of examining high-dimensional time-series sensor data collected from a WTP over a long time, Sparse Autoencoders (SAEs) are implemented. The data collected involves an array of operational measurements that, evaluated together, describe the plant's overall performance. With the support of the AE, this work aims to develop a practical framework for WTP operation predictive maintenance. Anomalies are all system findings from testing that might result in flaws or malfunctions. The research article analyses January and July 2023 WTP data from Jiangsu Province China. The AE paradigm had been evaluated using F1-scores, recall, accuracy, and precision. SAE has substantially improved AD functionality.
Keywords – Water Treatment Plants, Machine Learning, Anomaly Detection, Sparse Autoencoders, Precision, Recall, F1-score.

I. INTRODUCTION

The formation of pure water to drink whose contaminants are free of toxins hinges on Water Treatment Plants (WTPs), which are additionally tasked with ensuring health care to the people, protecting environmental sustainability and promoting business development [1]. With regard to the important role that these frameworks serve for human society, it is vital that their maintenance be kept in good repair on a more regular schedule. Nevertheless, predefined methods, which are a crucial part of standard maintenance methods, regularly fail to accurately represent the real-time performance of the system. This may result in to irrelevant maintenance or faults that hadn't been planned [2]. WTP routine maintenance has to cope with dynamic operational factors, multimodal sensor data, and the requirement to rapidly identify small anomalies that could suggest future issues [3]. Conventional techniques can be difficult to forecast this kind of failure due to the intricate layout in data collected during operation [4]. Anomaly Detection (AD) is capable of recognising exceptional anomalies that signify major problems [5]. This paper investigates the application of AE-based systems, primarily the Sparse Autoencoder (SAE), to set up AD in the WTP employing operational measurement metrics. Such structures attempt to reclaim regular operations and detect any out-of-the-ordinary anomalies that might suggest an interruption. The adaptability of AE concepts and the multidimensional and temporal features of sensor data are employed as proof of a Machine Learning (ML) technique for increasing WTP predictive maintenance speed and precision.

The proposed work has been used to explore the implementation of ML models for predictive maintenance in Water Treatment Plants (WTPs), which proactively detect anomalies and predict potential system failures. Three ML models, Autoencoders (AE), Variational Autoencoders (VAE), and Sparse Autoencoders (SAE), are all being trained on time-series sensor data to learn standard operational patterns and try to identify the deviations that are key in indicating possible malfunctions. The study used historical data from a WTP in Jiangsu, China, spanning January to July 2023. Performance metrics such as accuracy, precision, recall, and F1-score have been used to evaluate the models; out of all the models from the results, the SAE model has shown the best results, which is attributed to its sparsity constraint enforcing focus on crucial features.

The paper is organized as follows: **Section II** presents the literature review, **Section III** presents the background of the study, **Section IV** presents the methodology, **Section V** presents the experiment analysis, and **Section VI** concludes the work.

II. LITERATURE REVIEW

In [6-8] have investigated the use of AEs in the domain of three DoF delta robots for employing in convolutional layers for the purpose of Feature Extraction (FE) and used a sigmoid function for the work of AD that has highlighted the utility of AEs in the process of maintenance in prediction and fault localization without reliance on R2F data. Their approach also includes the method for calculating the remaining useful life (RUL), possibly using Gaussian processes based on the Health Indicator (HI) values to demonstrate the multifaceted applications of AEs in maintenance strategies.

The [9-10] have developed a Dynamic Predictive Maintenance Scheduling (DPMS) method that uses deep auto-encoders and deep forest for failure prognosis, which showcases the method's effectiveness in maintenance and decision-making based on the system degradation FE from raw sensor data. This method is effectively validated using NASA's aircraft engine datasets and has possibly outperformed several other *state-of-the-art* methods, which underscored the potential of Deep Learning (DL) in predictive maintenance for reducing costs and facilitating precise maintenance decisions.

In [11-12] have presented a parallel-stacked autoencoder model (PSAM) focusing on generating low-dimensional features, particularly from high-dimensional vehicle data targeted for fault prediction, particularly in powertrain components. The incorporation of embeddings over the categorical variables has further enhanced all performance of their Artificial Neural Network (ANN) models, which has led to signifying advancements in powertrain failure prediction and data size reduction.

The [13-14] have proposed an SAE-based predictive maintenance framework for the Air Production Unit (APU) system used in the Metro do Porto train. Their research was done to differentiate the predictive capabilities of analog and digital sensors in identifying failures equipped with digital sensors that show superior performance in detecting air leakage and other failures, which are indicating about the importance of sensor selection in the predictive maintenance models.

In [15-16] have explored a combined process of Recurrent Neural Network (RNN) together with the autoencoder approach for AD in power plant equipment with the objective of using the Mahalanobis Distance (MD) for anomaly condition determination. Their work has emphasized the effectiveness of GRU models over the LSTM in modelling normal behaviour, thereby offering a comprehensive framework for diagnostic and prognostic equipment management.

III. BACKGROUND

Unsupervised Learning based Failure Detection (ULFD)

ULFS models have all focused-on AD, malfunctions, and/or failures within systems without relying on labelled datasets. Unsupervised techniques aim to learn the standard operational patterns of a WTP system and use them to flag deviations as potential failures, which leverages the inherent data structure and distribution. The core principle of unsupervised failure detection is that the assumption over the standard operational data has significantly outnumbered anomalies or failures [17]. Modelling the expected behaviour of such a system could show significant deviation from this model that can be considered as an anomaly or a precursor to failure. Techniques such as clustering, density estimation, and neural networks are all employed to capture the underlying structure of the data and identify outliers.

Time-series Data

This data type comprises all sequences of values collected over a consistent period of time intervals and used for capturing the operational dynamics of systems, such as flow rates, pressure, and chemical concentrations. The essence of such time-series data often lies in its ability over the aspect to reflect temporal variations, trends, and patterns that are crucial for establishing normal operational baselines. The application of time-series data, particularly in unsupervised learning, has always involved the AD by analyzing deviations from these baselines. Unlike the supervised learning models [18], the unsupervised models, including the AE, usually process time-series data without labelled examples indicating normal and abnormal states. They learn to identify patterns and anomalies through the data's inherent properties, focusing on temporal dependencies and fluctuating parameters. The challenge with time-series data is its complexity, which also includes high dimensionality and the presence of noise. Practical analysis has always been a requirement using sophisticated preprocessing, FE, and dimensionality reduction techniques. Unsupervised models are particularly adept at handling these complexities, learning from the data's structure to distinguish between normal variability and indicators of potential failures.

Auto Encoder

The AE is considered a type of ANN that is used for unsupervised learning and efficient data coding. The primary goal of an AE is to learn about a representation (encoding) for a set of data, which is typically applied to dimensionality reduction or feature learning. AE are particularly effective in AD tasks, which include predictive maintenance in WTP, all by learning to reconstruct the standard operational data and identifying deviations as anomalies.

For an input time series, data represented as $X_i = \{x_1, x_2, \dots, x_T\}$, where T denotes the time steps and x_t represents the data at time step t, an AE aims to learn a compressed representation of X_i. The architecture of an AE comprises two primary components: the encoder and the decoder.

- Encoder: This component transforms the input time series data X_i into a hidden, lower-dimensional representation $H_i = \{h_1, h_2, \dots, h_n\}$, where $n < T$ and h denotes the features in the latent space. The encoder's function, which is denoted as f, is to transform the input time series data X_i into a hidden representation H_i. This transformation can be mathematically represented as Equ (1).

$$H_i = f\left(X_i; \Theta_f\right) \tag{1}$$

where Θ_f represents the parameters (weights and biases) of the encoder network. The function f typically involves a series of transformations, including linear mappings and nonlinear activations designed to compress the input data into a lower-dimensional latent space.

- Decoder: Following the above-discussed encoding process, the following decoder function is built to reconstruct the original input X_i from the compressed representation H_i. The reconstructed time series data are denoted as $R_i = \{r_1, r_2, \dots, r_T\}$, is obtained by applying a function g. The decoder function, which is denoted as g, is used in this process to reconstruct the original input time series data from the latent representation H_i. This reconstruction process is expressed as Equ (2).

$$R_i = g\left(H_i; \Theta_g\right) \tag{2}$$

where Θ_g represents the parameters of the decoder network. Similar to f function, the above function g too involves a series of transformations that map the compressed representation H_i back to the original data space that is aiming to produce a reconstruction R_i that closely mirrors the input X_i.

Both the functions such as f and g are being learned during the training process, and the inside objective in the model is to minimize a loss function that quantifies the difference between the original time series data X_i and its reconstruction R_i. By adjusting the parameters Θ_f and Θ_g, the AE model learns to encode and decode the input data and identifies the essential patterns in the latent space, ensuring the minimization of reconstruction errors. The training of an autoencoder revolves around minimizing the difference between the original time series data X_i and its reconstruction R_i. This discrepancy is quantified using a loss function, such as the Mean Squared Error (MSE), formulated as Equ (3).

$$L(X_i, R_i) = \frac{1}{T}\sum_{t=1}^{T} (x_t - r_t)^2 \tag{3}$$

where x_t is the actual data point at time t and r_t is the corresponding reconstructed data point.

Sparse Autoencoder (SAE)

A SAE, as shown in **Fig 1**, is also a variant of the basic AE model that mainly introduces sparsity constraints over the activations of the hidden layers. This constraint has helped to encourage the model to learn more meaningful and representative data features by limiting the number of active neurons in the hidden layer at any given time. Sparsity in an AE is mainly achieved by adding a regularization term to the loss function; this step penalizes the model if the activations of the hidden units deviate from a predefined sparsity level, ρ. The sparsity level is a small value close to '0', indicating the proportion of neurons that are expected to be active (*i.e.*, have non-zero activations) on average.

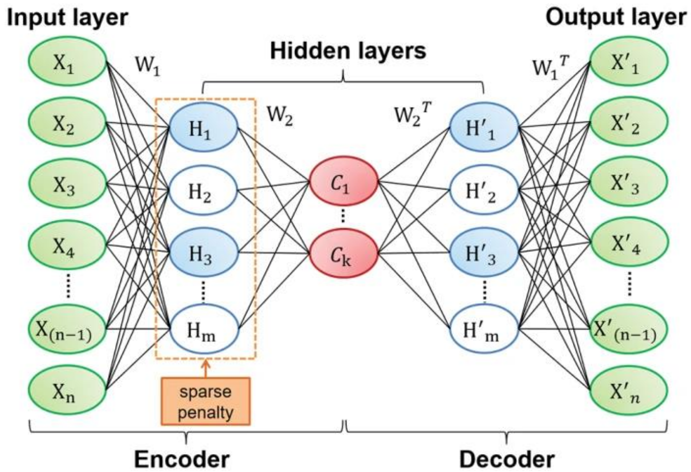

Fig 1. AE model

(i) Encoder: The encoder part of a SAE maps the input vector $X = \{x_1, x_2, \ldots, x_n\}$ to a hidden representation H. This mapping can be expressed by the function Equ (4)

$$H = \sigma(W_1 \cdot X + b_1) \tag{4}$$

Here, W_1 represents the weight matrix connecting the input layer to the hidden layer, b_1 is the bias vector of the encoder, and 'σ' denotes a nonlinear activation function, such as the sigmoid function. The encoder's purpose is to compress the high-dimensional input data X into a lower-dimensional latent space representation H.

(ii) Sparsity Constraint: The sparsity constraint is imposed on the activations of the neurons in the hidden layers. For each neuron i in the hidden layer, the average activation over a batch of inputs is considered and denoted as $\hat{\rho}_i$. The sparsity penalty enforces that $\hat{\rho}_i$ remains close to a pre-set sparsity target $'\rho'$, which is typically a small value close to '0'. The Kullback-Leibler (KL) divergence is employed to measure the difference between the actual activation level $\hat{\rho}_i$ and the desired level $'\rho'$, contributing the following term to the loss function.

$$\text{Sparsity\_Penalty } = \beta \sum_{i=1}^{m} \left[\rho \log\left(\frac{\rho}{\hat{\rho}_i}\right) + (1-\rho)\log\left(\frac{1-\rho}{1-\hat{\rho}_i}\right)\right] \tag{5}$$

In this Equ (5), β is a coefficient that balances the sparsity penalty against the reconstruction loss, and m denotes the total number of neurons in the hidden layer.

(iii) Decoder: The decoder segment aims to reconstruct the input data from the compressed representation H, and this reconstruction can be represented by the function Equ (6)

$$X' = \sigma(W_2^T \cdot H + b_2) \tag{6}$$

Here, X' is the reconstructed input, W_2^T denotes the transposition of the weight matrix connecting the hidden layer to the output layer, b_2 is the bias vector for the decoder, and $'\sigma'$ represents the same nonlinear activation function used in the encoder. The decoder's objective is to generate a reconstruction X' that closely approximates the original input vector X.

(iv) Loss Function: The overall loss function of a SAE is the sum of the reconstruction loss and the sparsity penalty. The reconstruction loss for each input vector X compared to its reconstruction X' is typically calculated using the Mean Squared Error (MSE), given by Equ (7).

$$\text{Reconstruction Loss} = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_i')^2 \tag{7}$$

Combining the reconstruction loss with the sparsity penalty, the loss function the SAE seeks to minimize is Equ (8).

$$\text{Loss} = \text{Reconstruction\_Loss} + \text{Sparsity\_Penalty} \tag{8}$$

This loss function ensures that the SAE not only learns to reconstruct the input data accurately but also adheres to the sparsity constraint, promoting the activation of a minimal number of neurons in the hidden layer. This approach encourages the AE to learn more robust and meaningful features in the data, facilitating applications such as feature extraction, dimensionality reduction, and anomaly detection.

IV. METHODOLOGY

Water Treatment Plant (WTP) Overview

The WTP chosen for this study is located in Jiangsu, China, and incorporates a series of processes employed to treat industrial and municipal wastewater. As depicted in Fig. 2, the treatment process has to start with the collection of industrial wastewater and city wastewater and then transitioning it through mechanical and biological treatment stages, culminating in the disinfection and final filtration before the treated water is discharged. Within this WTP, there are several critical areas where failures could occur, impacting the efficiency and safety of the water treatment process:

- Pumping Stations: Vital for initiating the wastewater treatment process by moving water to subsequent stages. A mechanical failure here could cause significant process delays or stoppages.
- Primary and Secondary Clarifiers are instrumental in separating sediments and facilitating biological treatment. Failures in clarifiers can lead to a domino effect, overloading other processes.
- Aeration Tanks: A core component for biological treatment, where a breakdown can result in the inadequate breakdown of organic matter.
- Chlorination and Disinfection Chambers: Essential for ensuring that the water is free of harmful microorganisms before discharge. Inaccurate dosing or system failures could pose public health risks.
- Sludge Digesters and Methane Storage: These areas manage the by-products of the water treatment process. Malfunctions can disrupt waste stabilization and methane harvesting, which is critical for both treatment efficacy and energy efficiency.
- Sludge Dewatering and Disposal: Proper functioning is crucial for environmental compliance. Issues in this stage can lead to improperly handling sludge, causing environmental hazards.
- Grease Traps and Sand/Grit Chambers: They prevent the accumulation of solids that can clog the system. Blockages or mechanical wear could reduce overall plant efficiency.
-

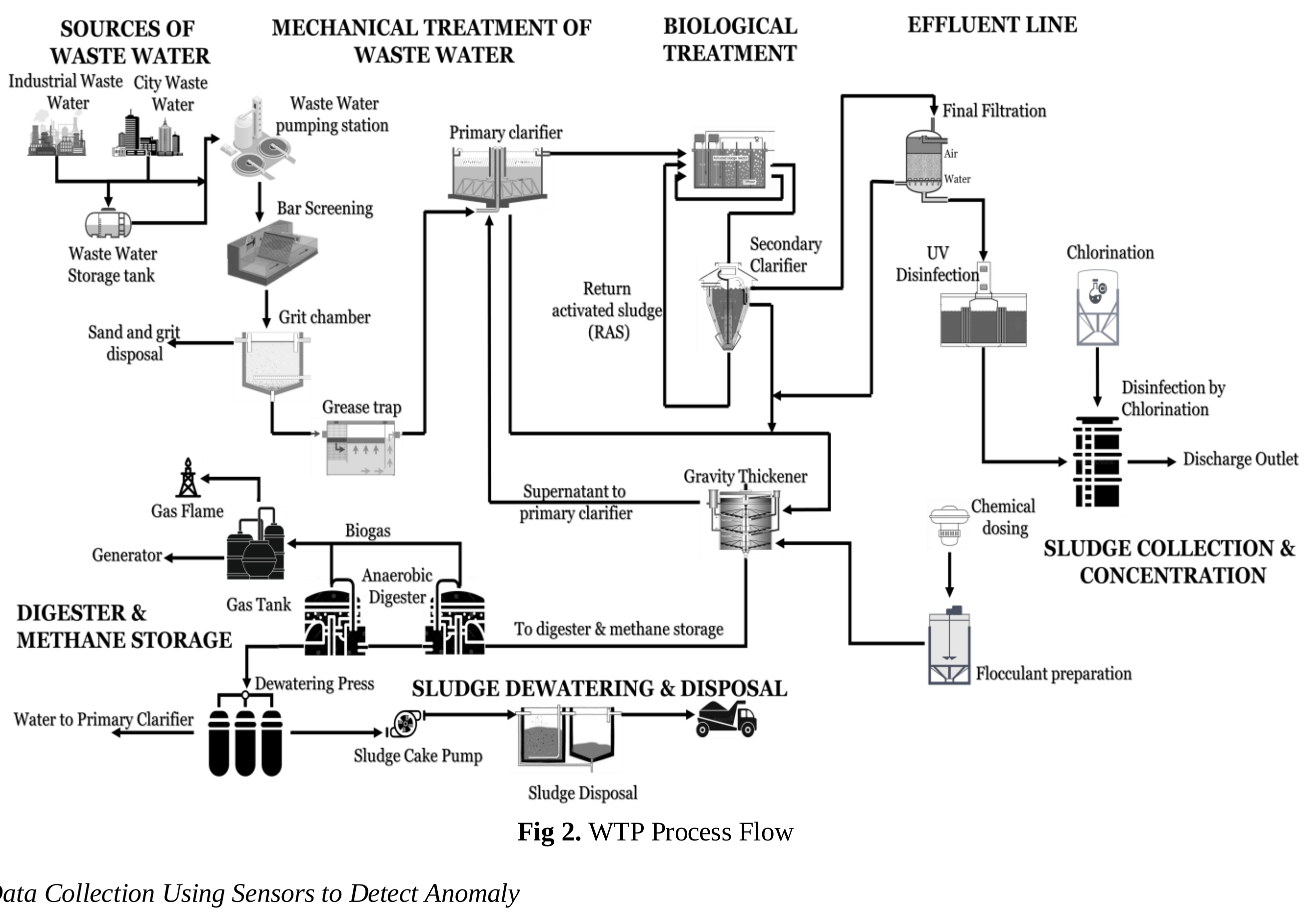

Fig 2. WTP Process Flow

Data Collection Using Sensors to Detect Anomaly

The data were collected for the chosen WTP during the period from January 2023 to July 2023, and the time series data were sourced from an array of sensors installed in the WTP. The following list in Table 1 describes the sensors deployed and the data collected through those sensors.

Table 1. Sensor Type Description

Sensor Type	Description	Data Measured
Flow Meters (Digital)	Digital sensors for water flow rate with data logging.	Volume (m^3/h)
Pressure Sensors (Analog)	Detect the pressure of water in different parts of the system.	Pressure (hPa, psi)
pH Sensors (Digital)	Digital readout of water's pH level, integrated into control systems.	pH level
Turbidity Sensors (Digital)	Digital sensors with real-time data on water clarity to a central system.	NTU (Nephelometric Turbidity Units)
Temperature Sensors (Digital)	Digital sensors with network connections for system integration.	Temperature (°C, °F)
Level Sensors (Digital - Ultrasonic/Radar Type)	Digital sensors use ultrasonic or radar signals to measure water levels.	Level (meters, cm)

The sensors have logged signals at a 1 *Hz* frequency that transmit data to the server every five minutes via TCP/IP protocol. A total of 2,92,073 data samples were recorded during the period between January 2023 to July 2023.

Feature Engineering

The raw data that have undergone preprocessing to ensure quality and consistency have included cleaning, removing outliers, and correcting errors. Normalization was used to scale the data to fit it within a specific range using Min-Max scaling. Given the enormous volume of data, it was segmented based on distinct operational phases of the WTP, such as filtration cycles, chemical dosing periods, and sedimentation times.

- Operational Cycle Identification: The WTP's important functional cycles have been mapped out, the most significant high-activity times were captured, and emergency or reduced-activity times were monitored.
- Segmentation and Binning: Additional data was segmented into "bins" of equal size within these functional phases. For the purpose of identifying correlations that could suggest malfunctions or equipment decline, this method allows an in-depth evaluation of the system's activity during specific operative stages.
- Feature Extraction: At every phase, the WTP's functioning condition has been defined by identifying and analysing empirical features, such as the mean and variance of sensor data within each bin. These features are advantageous due to their ability to disclose data regarding the treatment method's performance and good health, which can be employed to determine when service is required.
- Vector Formation: Input variables for the SAE framework have been created using data from all the integrated features. The framework has employed these patterns, which are often comprehensive reviews of the plant's data for operation, to demonstrate to the SAE how to function correctly and identify when AD requires service.
- The method resulted in the selection of 16 features for the SAE design. The features that have been selected are outlined in **Table 2:**

Table 2: Features Selected.

Feature No.	Feature Description	Data Source	Relevance to WTP Operation
1	Mean Flow Rate	Flow Meter	Indicates the average water flow through the plant.
2	Variance in Flow Rate	Flow Meter	Highlights fluctuations in water flow.
3	Mean Pressure Level	Pressure Sensor	Reflects average system pressure.
4	Pressure Variance	Pressure Sensor	Identifies pressure instability.
5	Average pH Level	pH Sensor	Measures the mean acidity or alkalinity of water.
6	pH Level Variability	pH Sensor	Detects fluctuations in pH levels.
7	Mean Turbidity	Turbidity Sensor	Indicates the clarity of water.
8	Turbidity Variance	Turbidity Sensor	Highlights changes in water clarity.
9	Average Temperature	Temperature Sensor	Reflects the mean water temperature.
10	Temperature Fluctuations	Temperature Sensor	Identifies temperature instability.
11	Mean Water Level	Level Sensor	Indicates average water level in tanks/reservoirs.
12	Water Level Variability	Level Sensor	Detects changes in water levels.
13	Filtration Cycle Duration	Flow Meter, Pressure Sensor	Captures the time taken for a complete filtration cycle.
14	Backwash Cycle Duration	Flow Meter, Level Sensor	Measures the duration of the backwash process infiltration.
15	Chemical Dosing Duration	pH Sensor, Turbidity Sensor	Indicates the length of chemical treatment phases.
16	Sedimentation Process Duration	Turbidity Sensor, Level Sensor	Reflects the time sedimentation takes to clear particulates from water.

SAE for Predictive Maintenance of WTP

The predictive maintenance framework for WTP, as shown in the algorithm, employed an SAE to process time series sensor data for the early detection of potential equipment or process failures. The process begins with collecting sensor data, which are represented as a time series T mainly recorded from various operational points within the WTP. This data then undergoes preprocessing, which includes normalization using min-max scaling to ensure uniformity. Additionally, feature engineering is applied to the normalized data T_{norm}, which in turn creates a set of enriched features F that have encapsulated domain-specific insights and operational nuances. These features, combined with the normalized data, form a comprehensive dataset P that is ready for model training. The SAE model is configured with a defined number of hidden layers and neurons alongside a sparsity regularization parameter λ. The model is trained over the source dataset P using the Adam optimizer, fine-tuning the network to minimize reconstruction error while adhering to the sparsity requirement.

Once trained, the SAE model is then deployed for AD upon a new time series sensor data T_{new}. The model attempts to reconstruct this received data, and by calculating the reconstruction error for each data point against its original, it AD based on a predefined threshold. Anomalies flag potential issues within the WTP's operational processes, triggering an alert. The following algorithm presents the functioning of the SAE algorithm for WTP maintenance.

Algorithm for SAE for predictive maintenance of WTP

Input: Time series sensor data T from WTP

Output: Anomaly alerts and recommendations for maintenance

Data Preprocessing

$NormalizeData(T)$:

- For each time series data point $t_i \in T$, normalize t_i using min-max scaling.
- Return the normalized time series data T_{norm}.

$FeatureEngineering\ (T_{norm})$:

- Create additional features F based on domain knowledge from T_{norm}.
- Combine T_{norm} and F into preprocessed data P.
- Return P.

Sparse Autoencoder Model

$Define\ Model\ Parameters$:

- L : Number of hidden layers.
- $neurons$ [ ]: Array defining the number of neurons per layer.
- λ : Sparsity regularization parameter.

$TrainSAE\ (P)$:

- Initialize an empty SAE model.
- Configure model with L hidden layers and λ for sparsity.
- Train model on P with neurons defined in $neurons$ [ ] using the Adam optimizer.
- Ensure sparsity by incorporating λ in the loss function during training.
- Return model.

Anomaly Detection

$AD\ (model, T_{new})$:

- Normalize T_{new} using the same method as in Normalize_Data.
- Reconstruct T_{new} using model.
- Calculate the reconstruction error for each point in T_{new} against its reconstruction.
- If error > threshold:
    - Flag the AD to the corresponding time series segment(s).
- Return anomaly flags and identified segments.

Predictive Maintenance

$AnalyzeAlerts(anomaly\_flags)$:

- For each flagged anomaly, analyze the corresponding time series segment.
- Identify potential equipment or process issues based on anomalies.

$ScheduleMaintenance(issue\_type, severity)$:

- Based on the analysis, schedule maintenance activities prioritizing by issue_type and severity.

Main Function

$Main()$:

- $T =$ Collect time series sensor data from WTP.

- $P = FeatureEngineering\ (NormalizeData\ (T))$.
- $model\ = TrainSAE(P)$.
- Continuously:
  - T_{new} = Collect new time series sensor data.
  - $anomaly\_flags\ = AD\ (model, T_{new})$.
  - If anomalies are detected:
    - $issues\ = AnalyzeAlerts\ (anomaly\_flags)$.
    - For each issue in issues:
      - Schedule maintenance based on issue.
      -

V. Experiment Analysis

The experiments were conducted using a computer that has the following specifications: an Intel Core i7 processor with a base clock speed of 3.6 GHz, 16 GB of DDR4 RAM, and a 512 GB SSD for storage—an NVIDIA GeForce GTX 1080 Ti. The algorithm was implemented in Python, an object-oriented, high-level programming language, Pandas for data manipulation, and TensorFlow for building and training neural networks. The data available from January to May 2023 is used to train the SAE network, which was then tested to predict the failures in June and July 2023. The SAE model was trained using the following hyperparameters, as shown in **Table 3**:

Table 3. Hyperparameters

Hyperparameter	**Value**
Number of Layers	3
Neurons per Layer	[64, 32, 64]
Activation Function	ReLU
Regularization Parameter (λ)	0.01
Learning Rate	0.001
Batch Size	128
Epochs	100

The model's effectiveness was assessed using the following metrics:

- Accuracy: Measures the overall correctness of the model, defined as Equ (9)

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{9}$$

- Precision: The ratio of TP to all positive predictions, calculated as Equ (10)

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{10}$$

- Recall (Sensitivity): The proportion of TP identified correctly, defined as Equ (11)

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{11}$$

- F1-score: The harmonic mean of precision and recall, expressed as Equ (12)

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

- Mean Squared Error (MSE) for Reconstruction: Measures the reconstruction error, crucial for AD, defined as EQU (13)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{x}_i)^2 \tag{13}$$

  where n is the number of samples, x_i is the actual value and $\hat{x}_i$ is the reconstructed value by the model.

Fig 3 illustrates the accuracy of three different ML models—AE, VAE, and SAE—over 100 training epochs. The graph proved that the SAE model outperforms the other two models, which maintain. The AE and VAE models also exhibited similar performance at the beginning, but it is noted that as training progresses, the AE model falls behind, with VAE overtaking it slightly and maintaining a narrow lead. The SAEs have shown superior performance, which could be attributed to their ability to enforce sparsity in the hidden layer, which could have helped the model avoid irrelevant features and concentrate on the most essential features. The VAE model has shown slightly higher accuracy than AE, possibly due to its probabilistic approach to learning the data representation.

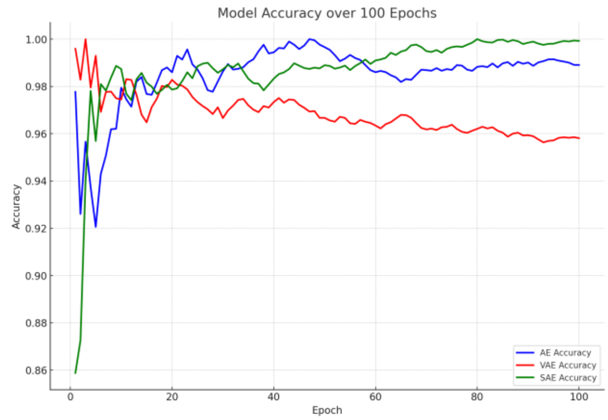

Fig 3. Accuracy Against Epochs

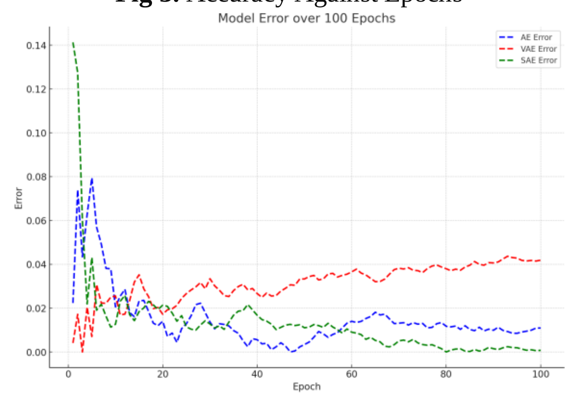

Fig 4. Error Against Epoch Analysis

As shown in **Fig 4,** the error rates have been compared for all the same three models, such as AE, VAE, and SAE, across 100 epochs. The figure shows that the SAE model has the lowest error rate, which suggests it is the most effective model for accurately reconstructing the input data. The error rates for the AE and VAE are higher, with VAE showing a slight advantage over AE.

Considering that lower error rates typically correlate with better model performance, the SAE's consistently lower error rate indicates that it will likely provide the most reliable predictions in an operational setting. Its error rate also appears to stabilize quickly and remains low throughout the training process, which is indicative of robust learning and generalization capabilities. On the other hand, while the AE and VAE improve over time, their error rates suggest that they may not capture the complexities of the data as effectively as the SAE. This analysis would support the choice of SAE for deployment in a water treatment plant's predictive maintenance system, potentially leading to improved operational efficiency and reduced downtime.

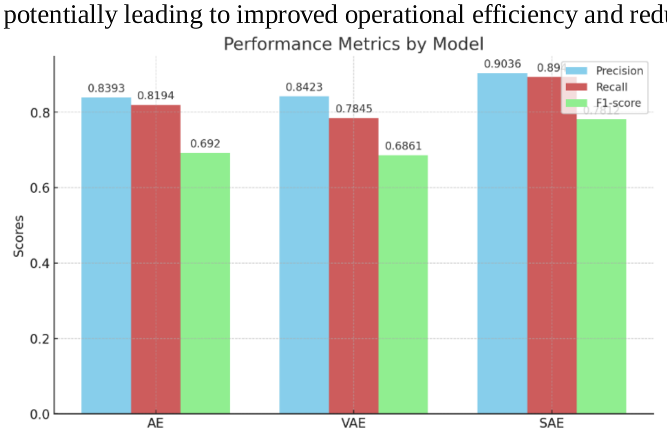

Fig 5. Performance Comparison Against Various Metrics

The chart, as shown in **Fig 5**, visualizes the performance metrics of Precision, Recall, and F1-score for three different models: AE, VAE, and SAE.

Analyzing the results:

- Precision: SAE has the highest precision at 0.9036, indicating that it has the highest proportion of TP results relative to the number of TP results it claims. AE and VAE have slightly lower precision scores of 0.8393 and 0.8423, respectively.
- Recall: Again, SAE has the highest score, with 0.894, which shows it is the most capable of correctly identifying all relevant instances. AE has a slightly lower recall than SAE at 0.8194, while VAE has the lowest recall at 0.7845, which shows that it misses more relevant cases than the other two models.
- F1-score: The F1-score is a balance between precision and recall. SAE's F1-score is the highest at 0.7812, indicating it has the best balance between precision and recall. AE and VAE have F1-scores of 0.692 and 0.6861, respectively; this means they do not balance the two as effectively as SAE.

VI. CONCLUSION AND FUTURE WORK

Optimising the success rate of predictive maintenance in Water Treatment Plants (WTP) is the main goal of the present study, which addressed the implementation of Sparse Autoencoders (SAE) within the context of Machine Learning (ML). Study results demonstrated that SAEs are suitable for Anomaly Detection (AD), that analyses time-series sensor data for faults or anomalies. According to the F1-score, recall, accuracy, and precision, they tested autoencoder (AE) designs. The most reliable AD detection model is SAEs. Findings indicate SAE use in WTP maintenance procedures may enable driven by data, preventative repairs. The process lowers delay, breakdowns in equipment, and expenses related to operation while increasing integrity and productivity. This research has encouraged concerns about using ML for industrial servicing to develop smart, adaptable systems which satisfy current development of infrastructure requirements. The analysis demonstrates that ML can set up WTP solutions, enhancing driven-by-technology service techniques.

In future research, investigators plan to develop real-time ML models for evaluating these approaches across several water treatment applications.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding agency is associated with this research.

Competing Interests

There are no competing interests.

References:

[1]. Wysowska, E., Wiewiórska, I., & Kicińska, A. (2021). The impact of different stages of water treatment process on the number of selected bacteria. Water Resources and Industry, 26, 100167.

[2]. Daily, J., & Peterson, J. (2017). Predictive maintenance: How big data analysis can improve maintenance. Supply Chain Integration Challenges in Commercial Aerospace: A Comprehensive Perspective on the Aviation Value Chain, 267-278.

[3]. Nunes, P., Santos, J., & Rocha, E. (2023). Challenges in predictive maintenance–A review. CIRP Journal of Manufacturing Science and Technology, 40, 53-67.

[4]. Arafat, M. Y., Hossain, M. J., & Alam, M. M. (2024). Machine learning scopes on microgrid predictive maintenance: Potential frameworks, challenges, and prospects. Renewable and Sustainable Energy Reviews, 190, 114088.

[5]. Erhan, L., Ndubuaku, M., Di Mauro, M., Song, W., Chen, M., Fortino, G., ... & Liotta, A. (2021). Smart anomaly detection in sensor systems: A multi-perspective review. Information Fusion, 67, 64-79.

[6]. Fathi, K., van de Venn, H. W., & Honegger, M. (2021). Predictive maintenance: an autoencoder anomaly-based approach for a 3 DoF delta robot. Sensors, 21(21), 6979.

[7]. Yu, H., Chen, C., Lu, N., & Wang, C. (2021). Deep auto-encoder and deep forest-assisted failure prognosis for dynamic predictive maintenance scheduling. Sensors, 21(24), 8373.

[8]. Revanur, V., Ayibiowu, A., Rahat, M., & Khoshkangini, R. (2020). Embeddings based parallel stacked autoencoder approach for dimensionality reduction and predictive maintenance of vehicles. In IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning: Second International Workshop, IoT Streams 2020, and First International Workshop, ITEM 2020, Co-located with ECML/PKDD 2020, Ghent, Belgium, September 14-18, 2020, Revised Selected Papers 2 (pp. 127-141). Springer International Publishing.

[9]. Davari, N., Veloso, B., Ribeiro, R. P., Pereira, P. M., & Gama, J. (2021, October). Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry. In 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA) (pp. 1-10). IEEE.

[10]. Santoso, B., Anggraeni, W., Pariaman, H., & Purnomo, M. H. (2022). RNN-Autoencoder Approach for Anomaly Detection in Power Plant Predictive Maintenance Systems. International Journal of Intelligent Engineering & Systems, 15(4).

[11]. S. Chang, B. Du and L. Zhang, "A Sparse Autoencoder Based Hyperspectral Anomaly Detection Algorithm Using Residual of Reconstruction Error," IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019, pp. 5488-5491, doi: 10.1109/IGARSS.2019.8898697.
[12]. A. R. Rezvanian, M. Imani and H. Ghassemian, "Patch-based Sparse and Convolutional Autoencoders for Anomaly Detection in Hyperspectral Images," 2020 28th Iranian Conference on Electrical Engineering (ICEE), Tabriz, Iran, 2020, pp. 1-5, doi: 10.1109/ICEE50131.2020.9261057.
[13]. M. Ganesh, A. Kumar and V. Pattabiraman, "Autoencoder Based Network Anomaly Detection," 2020 IEEE International Conference on Technology, Engineering, Management for Societal Impact using Marketing, Entrepreneurship and Talent (TEMSMET), Bengaluru, India, 2020, pp. 1-6, doi: 10.1109/TEMSMET51618.2020.9557464.
[14]. K. Wu, M. Cao, P. Wang and Z. Wang, "Convolutional One-Dimensional Variational Autoencoder Based Intrusion detection," 2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), AB, Canada, 2021, pp. 488-493, doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech52372.2021.00087.
[15]. Varatharaj Myilsamy *et al.*, State-of-Health Prediction for Li-ion Batteries for Efficient Battery Management System Using Hybrid Machine Learning Model. J. Electr. Eng. Technol. 19, 585–600 (2024). https://doi.org/10.1007/s42835-023-01564-2
[16]. S. Chouhan and B. Rudra, "Autoencoder-Based Anomaly Detection in ECG Image Time Series Data: A Comparative Evaluation of Three Different Architectures," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-6, doi: 10.1109/ICCCNT56998.2023.10306672.
[17]. K. Alrawashdeh and C. Purdy, "Fast Activation Function Approach for Deep Learning Based Online Anomaly Intrusion Detection," 2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), Omaha, NE, USA, 2018, pp. 5-13, doi: 10.1109/BDS/HPSC/IDS18.2018.00016.
[18]. H. Park, J. Jeong, K. -W. Oh and H. Kim, "Autoencoder-Based Recommender System Exploiting Natural Noise Removal," IEEE Access, vol. 11, pp. 30609-30618, 2023, doi: 10.1109/ACCESS.2023.3262026.