Proactive Anomaly Detection in Nuclear Power Plants using Deep Autoencoders: Enhancing Explainability with LLMs

Tapendra Baduwal
Westcliff University, USA
Kathmandu, Nepal

ABSTRACT

In real-world applications, such as nuclear power plants, failure data are often limited. Unlike supervised learning, which also requires failure examples, deep autoencoder unsupervised learning is therefore employed, involving training the model on a normal operational dataset by calculating the reconstruction error and setting a threshold to analyze new unseen data. Any dataset exceeding this threshold is classified as abnormal, and the top five contributing features are identified based on the highest reconstruction errors. The proposed deep autoencoder employs an architecture based on the activation functions of the Leaky Rectified Linear Unit (LeakyReLU) and Exponential Linear Unit (ELU) to mitigate the problem of 'dying neurons' and effectively capture complex, non-linear correlations between features. To enhance explainability, large language models (LLMs) are leveraged to analyze potential accident types and highlight likely areas of concern. Experiments were conducted on nuclear power plant accident data (NPPAD), generated using widely adopted PCTRAN simulation software. Comparative evaluations were conducted using Principal Component Analysis (PCA), Isolation Forest, ReLU-based autoencoders, and Deep autoencoders. Among these approaches, the proposed deep autoencoders achieved the best performance. These methods support a proactive anomaly detection method that empowers plant operators to detect potential accidents, identify their root causes, and make data-driven decisions, thereby improving safety, security, and timely maintenance.

Keywords

Predictive Maintenance, Anomaly detection, Unsupervised learning, Deep autoencoder, Reconstruction threshold, LLMs

1. INTRODUCTION

Predictive Maintenance is a proactive strategy that aims to forecast potential equipment failures and perform necessary repairs before problems arise [1]. This approach effectively minimizes downtime, reduces costs, and improves productivity, making it especially advantageous for the nuclear sector, where safety is of the utmost importance. Nuclear power plants (NPPs) are complex systems that require specialized expertise in various technical fields for their placement, design, construction, commissioning, and ongoing maintenance. Their complexity requires a coordinated approach that involves experts in nuclear engineering, environmental science, structural design, safety management, and regulatory compliance to ensure safe and efficient operation over the plant's entire lifespan. Fundamentally, these plants use nuclear fission to generate heat, which is then used to produce steam that drives turbines, converting nuclear energy into electricity [2]. To monitor the operational safety of NPPs, AI-driven alarm systems are essential for detecting abnormal conditions. This setup enables plant operators to make informed decisions by identifying potential problems and their root causes, thereby enhancing safety, security, and allowing timely maintenance.

As per the World Nuclear Association, nuclear energy currently provides about 9% of the world's electricity from about 440 power reactors, making it the second-largest source of low-carbon power and contributing to about one-quarter of low-carbon electricity as of 2024. In recent years, nuclear power has emerged as the second largest source of clean energy after hydropower. More than 50 nations operate about 220 research reactors that use nuclear energy [3]. Besides research purposes, these reactors are also employed in the production of medical and industrial isotopes, along with serving as training facilities.

In supervised learning, the model is trained on a labeled dataset, where input and output parameters are tagged with labels identifying characteristics, attributes, or classifications. In unsupervised learning, the model uses unlabeled datasets, containing only features without labels identifying characteristics, attributes, or classifications [4] [5].

Researchers mainly focus on models such as classifiers that distinguish normal and abnormal data, principal component analysis (PCA), isolation forest (iForest), and various deep learning methods, including autoencoders and long short-term memory (LSTM) models. PCA is a widely used classical method for computing reconstruction errors, but it is limited to linear relationships and may not fully capture the complexities of real-world data. In such cases, a popular alternative is to use autoencoders, which are neural network models that can compress and approximately restore data, making them more flexible in handling non-linear relationships [6]. One key challenge in the nuclear power plant domain is the lack of a publicly available dataset for evaluating the performance of various algorithms. To tackle these challenges, we utilize the Nuclear Power Plant Accident Data (NPPAD) generated using PCTRAN, a widely used simulation software for nuclear power plants. This dataset covers a comprehensive collection of data for normal operating conditions and a wide range of common accident scenarios that can occur in pressurized water reactor nuclear power plants [7] [8].

In the proposed methods, the deep autoencoder, a specialized artificial neural network model, is first trained using the normal operational dataset by calculating the reconstruction error and establishing a threshold for analyzing new, unseen data. After training, the model shows the ability to accurately reconstruct normal data with low reconstruction error. However, the model struggles with abnormal data, resulting in higher reconstruction errors that exceed the established threshold and are classified as anomalies. Effective training of the neural network requires a sufficient amount of data. To collect the normal operational dataset in a real-world scenario, the system must be operated under normal conditions for one to two refueling cycles. Reconstruction errors are used not only for detecting anomalies but also for explaining them. Features with larger reconstruction errors are considered more suspicious, as they contribute significantly to the total reconstruction error. By analyzing the contribution of each feature to the overall reconstruction error, it becomes possible to identify the key features responsible for the anomaly. This process provides an explainable framework for understanding anomalies, enabling domain experts to focus on specific features that require attention and take targeted corrective actions.

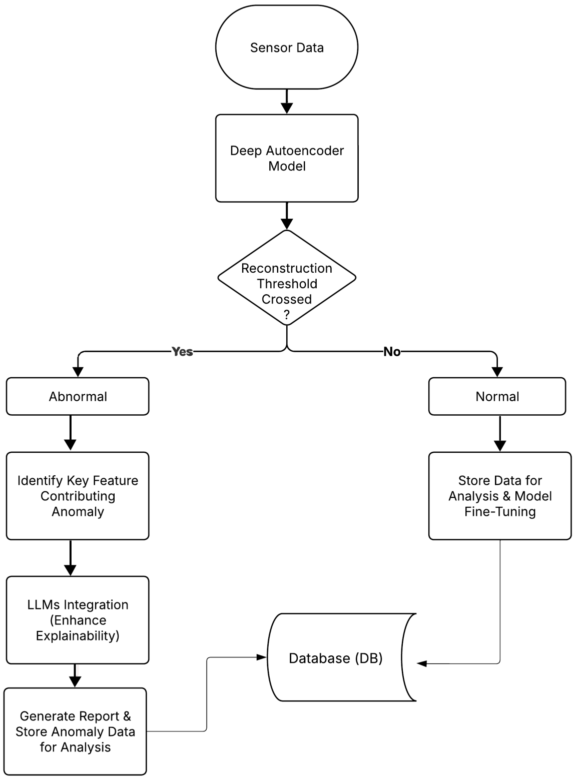
Sensor Data
Deep Autoencoder Model
Reconstruction Threshold Crossed ?
Yes
No
Abnormal
Normal
Identify Key Feature Contributing Anomaly
Store Data for Analysis & Model Fine-Tuning
LLMs Integration (Enhance Explainability)
Database (DB)
Generate Report & Store Anomaly Data for Analysis

Fig. 1. Comprehensive Workflow for Anomaly Detection and Analysis

Transformer-based Large Language Models (LLMs) have recently shown remarkable capabilities in natural language processing and in other areas as well. The transformer architecture is the basic framework for all LLMs and was first introduced in the 2017 paper “Attention is All You Need.” LLMs show impressive contextual understanding by interpreting top contributing features related to anomalies, accident types, and changes in feature values. This research utilizes large language models (LLMs) to analyze potential accident types and highlight critical areas of concern, supporting plant operators in making informed, data-driven decisions [9] [10] [11].

1.1 Problem Statement: Nuclear power plants operate in high-risk environments where even a single undetected anomaly can lead to serious safety risks. Due to security constraints, there is a continuing challenge in the nuclear power plant sector regarding the lack of an open dataset to evaluate various algorithms. In the real world, the absence of failure data poses a significant challenge for supervised learning models, which rely on labeled data for both anomalies and normal conditions.

1.2 Motivation: Advancements in technology have created new opportunities to address challenges in nuclear power plant safety and maintenance. This has motivated the exploration and integration of innovative approaches, such as deep autoencoder reconstruction threshold-based anomaly detection and LLMs to enhance explainability and report generation.

1.3 Contribution: The significant contributions of this study are summarized below:

a) Identify the key features contributing to the anomaly based on the highest reconstruction errors. Furthermore, LLMs are leveraged to analyze potential accident types and pinpoint likely areas of concern.
b) Deep autoencoder, combining Leaky Rectified Linear Unit (LeakyReLU) and Exponential Linear Unit (ELU) activation functions, mitigate the ’dying neurons’ problem and effectively captures complex and non-linear correlations between features.
c) For a comprehensive analysis, normal and anomalous behaviors were evaluated by testing the models on 17 distinct accident sets covered by NPPAD.

Overall, this research presents an efficient anomaly detection method for critical industries, leveraging the predictive capabilities of deep learning alongside the enhanced explainability offered by LLMs integration. The solution enables proactive anomaly detection, helping plant operators identify potential risks early, optimize maintenance schedules, and mitigate safety incidents in nuclear power plants.

2. LITERATURE REVIEW

PCA is a well-known classical method for dimensionality reduction and anomaly detection. It identifies anomalies based on reconstruction errors, where the encoder and decoder are linear models. Researcher Takeishi (2019) proposed utilizing Shapley values, a simple game-theoretic approach, to fairly distribute the total error among features by analyzing their marginal contributions. The study introduced an efficient value function based on the PCA probabilistic framework, which accounts for feature correlations and conditional distributions, demonstrating enhanced interpretability over raw reconstruction errors. This method effectively aids in identifying feature contributions, advancing fault detection, and interpreting anomaly analysis [6].

Isolation Forest is often employed as a state-of-the-art anomaly detector in real-world applications due to its simplicity and efficiency. However, its linear isolation method struggles with high-dimensional, non-linear, or complex data, leading to difficulty in detecting hard anomalies. In 2023, Xu et al. introduced Deep Isolation Forest (DIF), a method that employs deep neural networks to transform the data, enabling the capture of non-linear relationships and better handling of complex, high-dimensional datasets. The transformed data is then passed to the iForest model, which performs the anomaly detection in this new representation space. The weight matrix W_i is generated based on the base weight matrix W_0, and these weight matrices are used to transform the input data into new representations [12].

Researchers Qi et al. (2023) explored the critical role of fault diagnosis in ensuring the safety of nuclear power plants (NPPs). This paper reviews fault diagnosis techniques from the perspective of artificial intelligence (AI) and fault diagnosis techniques are classified into knowledge-driven and data-driven approaches. Knowledge-driven methods utilizing the experience of domain experts include early if-then rules-based principles. The data-driven section provides a detailed survey of fault diagnosis methods based on single and hybrid algorithms like artificial neural networks(ANNs), support vector machines(SVMs), decision tree, PCA, clustering, etc. Due to the complex relation-capturing nature of hybrid algorithms such as ANN+X, where X stands for other auxiliary algorithms, are emerging as a popular direction driven by advancements in deep learning technology by researchers, since single algorithms often fall short in meeting diagnostic needs [13].

A notable study by Li et al. (2022) developed an anomaly detection method for nuclear power plant (NPP) operations using an unsupervised deep generative model, specifically leveraging Variational Autoencoders (VAE) and Isolation Forest (iForest) techniques. VAE encode a continuous, probabilistic representation of that latent space or lower-dimensional space. In these methods, VAE trains on a normal condition dataset and calculates reconstruction error, if the input data is similar to the training normal operations data, the VAE will successfully reconstruct it with low error. However, if the data deviates from this normal pattern, indicating an abnormal nature with high error. After that, the iForest further analyzes the high reconstruction error from the VAE to detect anomalies definitively. Isolation Forest identifies anomalies by constructing isolation trees that randomly partition data, measuring how quickly data points can be isolated. Points that are isolated quickly are classified as anomalies [14].

Cancemi et al. (2023) explore deep learning techniques for unsupervised anomaly detection in nuclear power plant components using a digital twin of a pressurized water reactor (PWR) 2-loop simulator. The study aims to predict failures before safety systems activate by simulating various loss of cooling accident (LOCA) scenarios, adding Gaussian noise for realism. They employ an autoencoder architecture that uses ReLU activation functions, training it on normal operational conditions dataset to effectively reconstruct the original input data through its encoder and decoder components and then calculate reconstruction error and establish a threshold, after that any new data that exceeds this threshold is identified as an anomaly [15].

Chaudhary et al. (2024) explore anomaly detection in nuclear power plants, by highlighting the risks of cyberattacks in critical infrastructure. The study uses a Bi-LSTM model for early anomaly detection, based on reconstruction threshold and validated through simulations of cyberattacks on the Asherah Nuclear Power Plant simulator. For better explainable they apply shapley additive explanations AI methods to find key features contributing to anomaly detection. This research provides a robust framework for enhancing cybersecurity in critical industries through real-time monitoring and explainable models [16].

Recent studies by Liso et al. (2024) highlight the increasing significance of anomaly detection in sectors such as Industry 4.0, energy management, smart agriculture, cybersecurity, and bioinformatics. The analysis highlighted the growing adoption of autoencoders across various configurations and application scenarios, demonstrating their effectiveness and versatility in detecting anomalies. The study also identifies gaps in current knowledge and proposes future directions to consolidate research and develop a unified framework for anomaly detection [17]. In finance and banking, it can be applied to fraud detection, risk management, and stock market analysis. In healthcare, it enables real-time monitoring of abnormal vital signs, such as irregular heartbeats or oxygen levels. In the manufacturing sector, it supports predictive maintenance of machinery and the detection of production line defects. In cybersecurity, it aids in identifying unauthorized access, detecting malware, and analyzing user behavior for threats.

Emergency decision support techniques play an important role in complex and safety-critical systems such as nuclear power plants (NPPs). Xiao et al. (2024) review these techniques as a comprehensive framework involving operator training, risk assessment, fault detection, multi-criteria decision support, and accident consequence analysis. They discuss key systems like the Real-Time Online Decision Support System for Nuclear Emergencies (RODOS), the Accident Reporting and Guiding Operational System (ARGOS), and the Decision Support Tool for Severe Accidents (Severa). The authors highlight challenges such as integrating risk assessment, improving training methods, and utilizing LLMs. They propose a new decision support system that combines advanced probabilistic safety assessments (PSA) methods that are conducted by systematically analyzing potential accidents in nuclear power plants through fault and event tree analysis, data collection, and risk characterization. This process evaluates failure probabilities and consequences to inform safety decisions and emergency management strategies [18].

3. STATE-OF-THE-ART METHODS

3.1 PCA for Anomaly Detection: PCA transforms data into principal components, and the original data can be reconstructed from these components with minimal loss. Consider a dataset with two features, x and y, for which the covariance matrix is given as:

$$\text{Covariance Matrix } (C) = \begin{pmatrix} \text{cov}(x,x) & \text{cov}(x,y) \\ \text{cov}(y,x) & \text{cov}(y,y) \end{pmatrix}$$

$$\text{Where, } \text{cov}(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

Eigenvalues (λ_1, λ_2) and Eigenvectors (V_1, V_2) are derived from:

$$\det(C - \lambda I) = 0, \quad (C - \lambda I)V = 0$$

Where, I is the identity matrix of the same dimension as C

Eigenvectors $V_1 = \begin{pmatrix} V_{11} \\ V_{12} \end{pmatrix}$ and $V_2 = \begin{pmatrix} V_{21} \\ V_{22} \end{pmatrix}$ are obtained.

Feature vector is formed by sorting eigenvectors by descending eigenvalues:

$$\text{Feature Vector} = \begin{pmatrix} V_1 & V_2 \end{pmatrix}$$

Principal components are obtained by multiplying the transpose of the feature vector with the transpose of the standardized data.

$$\text{FinalData} = \text{Feature Vector}^T \times \text{Scaled Data}^T$$

$$\text{TransformedData} = \text{FinalData}^T$$

ReconstructedData = (TransformedData×Feature Vector^T) $\times\ Std$ + Original Mean

The optimal principal components k must be chosen. The number of principal components affects reconstruction accuracy: too many components minimize reconstruction error for all data, while too few result in poor reconstruction of the original data [6] [19] [20]. While PCA is a powerful tool, it has limitations, mainly because the feature vectors or principal components are restricted to being linear combinations of the existing features. If the data cannot be explained by linear combinations, PCA is less effective.

3.2 Isolation Forest: iForest is a popular anomaly detection method that identifies anomalies based on isolation difficulty in the data space. Anomaly scores are computed for each data point, with those exceeding a threshold classified as abnormal data.

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

Where:

n = Number of data points, x = Data point being evaluated

$E(h(x))$ = Average isolation path length of x in a tree

$c(n)$ = Average depth of data points in a tree or average value of h(x).

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

Where, $H(n)$ is the nth harmonic number, which can be approximated by $\ln(n) + \gamma$, with $\gamma \approx 0.577$ being the Euler-Mascheroni constant.

$$\text{If } E(h(x)) \ll c(n), \quad S(x,n) \approx 1, \quad \text{indicating an anomaly.}$$

$$\text{If } E(h(x)) \gg c(n), \quad S(x,n) \approx 0, \quad \text{indicating a normal point.}$$

$$\text{If } E(h(x)) = c(n), \quad S(x,n) = 0.5, \quad \text{indicating an uncertainty.}$$

An anomaly score ranges from 0 to 1, with scores near 1 indicating anomalies and scores near 0 indicating normal points. The tree used in iForest is known as an Extra(Extremely Randomized) Tree Regressor, in which splits are performed by selecting nodes randomly [12] [21] [22]. While iForest has significant advantages, it also has limitations, it may assign high anomaly scores to normal points near the edges of the data distribution due to their sparse surroundings, and it assumes feature independence when creating splits, which often does not align with the correlated nature of real-world datasets.

3.3 Autoencoders: It is a type of neural network architecture designed to compress or encode input data into essential features and reconstruct or decode the original input from the compressed representation. In the case of anomaly detection, reconstruction error measures the difference between the original input and its reconstructed output, with higher errors typically indicating anomalies. Encoder maps the input $x \in R^n$ to the latent space $z \in R^m$:

$$z = \sigma(Wx + b)$$

Decoder reconstructs the input $\hat{x}$ from the latent space z:

$$\hat{x} = \sigma'(W'z + b')$$

Where W = weight matrices, b = bias vectors, σ = activation functions, n = input dimension, m = latent space dimension.

Autoencoders use non-linear activation functions to capture complex, non-linear correlations between features, thereby increasing detection accuracy. Various types of autoencoders have been proposed by researchers, including variational autoencoders, denoising autoencoders, deep autoencoders, and sparse autoencoders [23] [24] [25].

The reviewed state-of-the-art methods provide significant advancements in anomaly detection, they each have inherent strengths and limitations. These approaches are specifically designed to address different tasks, and their effectiveness depends on the particular requirements of the application, such as data characteristics, complexity, and the nature of the anomalies being detected.

4. OVERVIEW OF THE DATASET

One of the key challenges in the nuclear power plant domain is the lack of a publicly available dataset for evaluating the performance of various algorithms. To tackle these challenges, we utilize the Nuclear Power Plant Accident Data (NPPAD), licensed under the MIT license and developed by Qi et al., using PCTRAN, a widely used and well-established simulation software for nuclear power plants [7] [8]. The dataset covers 18 types of operating conditions, including normal operation, various loss of coolant accidents (Hot Leg, Cold Leg), steam line breaks (inside and outside containment), hydrogen burn scenarios, loss of AC power, locked rotor events, and anticipated transients without scram. It also addresses turbine trips, steam generator tube ruptures (A and B), and other events like rod withdrawal and insertion, feedwater line breaks, moderator dilution, load rejection, and letdown line breaks in auxiliary buildings. There are 97 accident parameters variables in this dataset that pertain to accident conditions and represent essential features of a nuclear reactor accident, including but not limited to, reactor coolant temperatures, pressure levels in steam generators and the reactor building, flow rates for coolant, feedwater, and steam, reactor power metrics like thermal output and turbine load, critical reactivity factors, radiation levels, safety system flows, etc [7]. By studying them, a deeper understanding of the complex dynamics of reactor accidents can be gained, and areas for improvement in reactor design and safety protocols can be identified.

5. RESEARCH METHODOLOGY

5.1 Data Preprocessing Techniques: Since the Nuclear Power Plant Accident Data (NPPAD) [7] includes a normal condition dataset with only 302 rows, the neural network requires sufficient data to learn patterns effectively. Therefore, the amount of data needs to be increased using data augmentation techniques.

Add Gaussian Noise: There are mainly two benefits of adding gaussian noise in a dataset, one is to increase the size of the dataset by augmentation and introduces realism that might occur in real operational conditions. Gaussian noise is added to the numeric columns containing variable values, while excluding the time column and any columns with constant unique values. This approach simulates new data while ensuring the integrity of the normal condition data is preserved. It is calculated using two main parameters σ_{col} and μ_{col} of each feature (column) [26]:

$$\sigma_{\text{col}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{\text{col}})^2}$$

where:

x_i represents individual data points,
N is the total number of data points,
μ_{col} is the mean of the column, calculated as:

$$\mu_{\text{col}} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

After calculating the standard deviation for each column, it provides a measure of how much the values in that column typically vary from their mean. A noise multiplier of 0.001 is chosen to suit sensitive cases like nuclear power plant data, where small fluctuations are realistic. The noise values are generated randomly from a normal distribution with a mean of 0 value and a standard deviation determined by multiplying the noise multiplier with the column's standard deviation.

$$\mathcal{N}\left(0, \text{noise\_multiplier} \times \sigma_{\text{col}}\right).$$

$$\mathcal{N}(x_i \mid \mu_{\text{col}}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu_{\text{col}})^2}{2\sigma^2}}$$

The normal distribution determines the probability density of a random variable x within the curve. The function generates a bell-shaped curve, where the peak occurs at $x = \mu$. Values closer to the mean (μ) have higher probability densities, while values farther from the mean decrease in probability density exponentially [27]. Once the noise is generated, the new data value x'_i for each original value x_i is:

$$x'_i = x_i + \text{Noise}_i$$

Clipping: After the noise is added, the resulting value x'_i may fall outside the range of the original data. To ensure realistic values, noise-augmented data are constrained within the original column's minimum and maximum [26].

$$x'_i = \text{clip}(x'_i, x_{\text{min}}, x_{\text{max}})$$

where:

$$\text{clip}(x'_i, x_{\text{min}}, x_{\text{max}}) = \begin{cases} x_{\text{min}} & \text{if } x'_i < x_{\text{min}} \\ x'_i & \text{if } x_{\text{min}} \le x'_i \le x_{\text{max}} \\ x_{\text{max}} & \text{if } x'_i > x_{\text{max}} \end{cases}$$

Min-Max Scaling: Many machine learning models assume that features follow a normal distribution, but real-world datasets often have skewed distributions. Skewed features can be transformed using feature scaling techniques to improve model effectiveness. Min-Max Scaling, a form of normalization, resizes variables to a fixed range, typically [0, 1]. This is particularly useful when the data is not normally distributed [28].

$$x' = \frac{x_i - X_{\text{min}}}{(X_{\text{max}} - X_{\text{min}}) + \epsilon}$$

Where:

x_i is the original value,

X_{min} is the minimum value of the feature,

X_{max} is the maximum value of the feature,

ϵ is a small constant added to prevent division by zero

5.2 Model Architecture: The proposed method employs a deep autoencoder, an unsupervised learning model comprising six linear layers in both the encoding and decoding stages. The model utilizes a combination of LeakyReLU and ELU activation functions to introduce non-linearity. This involves training the model on a normal operational dataset by calculating the reconstruction error and setting the threshold for analyzing new unseen data. After training, the model shows the ability to accurately reconstruct normal data with low reconstruction error. However, it struggles with abnormal data, resulting in higher reconstruction errors that exceed the established threshold and are classified as anomalies [15] [23] [24] [25].

Algorithm 1 Deep Autoencoder

```
Encoder Architecture:
    Linear layer (input_dim $\to$ 128)
    LeakyReLU activation (negative_slope=0.1)
    Linear layer (128 $\to$ 96)
    LeakyReLU activation (negative_slope=0.1)
    Linear layer (96 $\to$ 80)
    LeakyReLU activation (negative_slope=0.1)
    Linear layer (80 $\to$ 64)
    ELU activation (alpha=1.0)
    Linear layer (64 $\to$ 48)
    ELU activation (alpha=1.0)
    Linear layer (48 $\to$ encoding_dim)
Decoder Architecture:
    Linear layer (encoding_dim $\to$ 48)
    ELU activation (alpha=1.0)
    Linear layer (48 $\to$ 64)
    ELU activation (alpha=1.0)
    Linear layer (64 $\to$ 80)
    LeakyReLU activation (negative_slope=0.1)
    Linear layer (80 $\to$ 96)
    LeakyReLU activation (negative_slope=0.1)
    Linear layer (96 $\to$ 128)
    LeakyReLU activation (negative_slope=0.1)
    Linear layer (128 $\to$ output_dim)
Hyperparameters and Configuration:
    Learning rate: 0.00001
    Batch size: 64
    Epochs: 400
    Optimizer: Adam
    L2 lambda: 0.001
    Encoding dim = 32
    Input dim = 97 = Output dim
```

Without activation functions, neural networks can only represent linear relationships, which prevents them from capturing complex, non-linear patterns in real-world data. A combined activation function is used for the following reasons:

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases}$$

$$\text{ELU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha(\exp(x) - 1) & \text{if } x < 0 \end{cases}$$

The derivatives of many activation functions (e.g tanh, sigmoid) are very close to 0, demonstrating the vanishing gradient problem. ReLU mitigates this issue on the positive side but still suffers from a similar problem for negative inputs, where gradients are 0. This is referred to as the dying ReLU problem instead [29] [30].
LeakyReLU primarily solves the dying neuron problem by allowing negative inputs to have a small, non-zero output, ensuring that all neurons remain active and contribute to the learning process. The output range is $(-\infty, \infty)$, and α is a small constant that controls the slope for negative inputs.
The ELU activation function addresses the dying neuron problem and vanishing gradient problem by maintaining non-zero dynamic gradients for negative inputs, where gradients represent the rate of change of a function with respect to its parameters or derivatives of activation functions and the output lies in the range of $[-1, \infty)$, and α determines the saturation level of the negative inputs [31].

5.3 Training and Validation loss: In this research, the model is first trained using data from normal operating conditions, as real-world failure data is limited. Mean squared error (MSE) is used as the loss function during training, where errors are squared and larger deviations are penalized more heavily than smaller ones, helping the model focus on minimizing the loss during optimization [32] [33].

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

Where:

y_i is the true value of the i-th data point,
$\hat{y}_i$ is the predicted value of the i-th data point.
n is the total number of data points.

The validation loss is calculated in a similar manner, using the validation dataset.

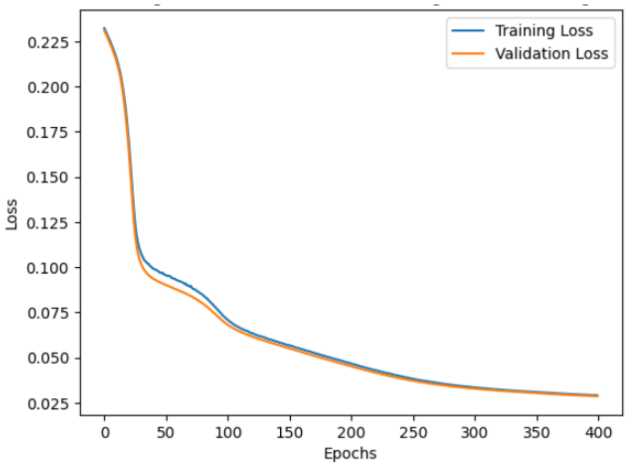

Fig. 2. Training and Validation loss Curve

After analyzing the training and validation loss curves, it was found that the curves remain very close throughout the training process. This suggests that the model generalizes well and is not overfitting to the training data. From epoch 100 onwards, the losses continue to decrease slowly and become almost stable in the range of epochs 350 to 400, with final values of 0.02878 for training loss and 0.02817 for validation loss.

5.4 Ridge Regularization(L2): Ridge regression is mostly used to reduce the overfitting in the model. In this technique, the cost function is altered by adding the penalty term to it. We can calculate it by multiplying with the lambda to the squared weight of each individual feature. The equation for the cost function in ridge regression will be:

$$\text{RidgeCostFunction} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{d}W_j^2$$

Here, $\lambda\sum_{j=1}^{d}W_j^2$ is the penalty, with W_j being the weights.
Increasing the value of λ (e.g., 0.0001, 0.01, 0.1, 1, 2, 3, etc.) increases the cost function. As the model minimizes the loss, it balances reducing residual error and penalizing large coefficients. This encourages less important features to have smaller coefficients, driving them closer to zero, while more relevant features maintain higher coefficients. A higher coefficient indicates a more important feature, while a smaller coefficient suggests less relevance to the model [24] [34].

5.5 Reconstruction Threshold: Reconstruction error measures the difference between the original input and its reconstructed output, with higher errors often indicating the presence of anomalies. Common metrics for calculating reconstruction error include mean squared error (MSE) and mean absolute error (MAE). MSE is highly sensitive to large errors due to its squaring effect. MAE, on the other hand, provides stability and robustness in evaluating how well each individual sample has been reconstructed. MAE was selected for this study due to its consistent performance. [32] [33].

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Where:

y_i is the true value of the i-th data point,
$\hat{y}_i$ is the predicted value of the i-th data point,
n is the total number of data points.

The reconstruction error threshold is computed as follows:

$$\mu = \frac{1}{n}\sum_{i=1}^{n}\text{reconstruction\_error}_i$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{reconstruction\_error}_i - \mu)^2}$$

$$\text{Threshold} = \mu + k \cdot \sigma$$

where k is a multiplier, commonly set to $k = 3$ for a 99.7% confidence level. The reconstruction threshold can vary with each training time due to random initialization of weights and random splits of training, testing, and validation data. However, the threshold remains within a specific range for each run. After training the model, the reconstruction threshold got fixed at 0.0838, and data exceeding this threshold were marked as anomalies.

5.6 Key Features Contributing to the Anomaly: After detecting abnormal data, the key features contributing to the anomaly are identified based on the highest reconstruction errors. For the accident type 'Loss of Coolant Accident (LOCA) hot leg', the top five features that contribute most to the anomaly, based on higher reconstruction errors, are shown below:

Features	**Abnormal Value**	**Reconstruction Error**	**Reconstruction Threshold**
WRCA	14895.537	4.106	0.0838
THB	386.813	3.473	
WRCB	30797.503	2.882	
THA	356.816	2.109	
PSGA	55.165	1.800	

Table 1: Top 5 Features Contributing to LOCA

Model proactively identifies potential accident types by analyzing changes in key feature values, enabling timely detection and mitigation. As shown in the table above, during a LOCA-Hot Leg, several critical features are affected, such as hot leg temperatures (THA and THB) increase due to insufficient coolant flow, coolant flow rates (WRCA and WRCB) decrease sharply from the loss of coolant volume, and steam generator pressure (PSGA) drops as the system pressure falls. These changes not only indicate the likelihood of a LOCA event but also help pinpoint the affected area, allowing for focused and efficient issue resolution.

5.7 Evaluation Metrics: For evaluation purpose, normal and accident-related abnormal test datasets were classified using the reconstruction threshold. The confusion matrix below compares the normal test data with the LOCA–Cold Leg abnormal dataset.

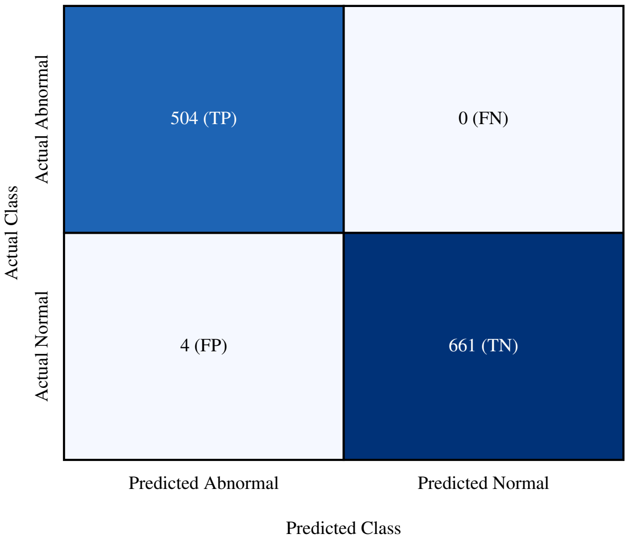

Fig. 3. Confusion Matrix

In the context of statistics, TP (the number of positive data points correctly identified or actual abnormal data predicted as abnormal)), TN (the number of negative data points correctly identified or actual normal data predicted as normal), FP (the number of negative data points wrongly marked as positive or actual normal data predicted as abnormal), and FN (the number of positive data points wrongly marked as negative or actual abnormal data predicted as normal) [35].

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

$$\text{Precision rate} = \frac{TP}{TP + FP}$$

$$\text{Recall rate} = \frac{TP}{TP + FN}$$

$$F_1 \text{ score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The tables below highlight the performance of different methods:

Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Isolation Forest	0.954	0.904	1.000	0.950
PCA	0.981	0.958	1.000	0.978
Autoencoder (ReLU)	0.990	0.978	1.000	0.989
Deep Autoencoder (LeakyReLU, ELU)	0.996	0.992	1.000	0.996

Table 2: Performance Metrics

The model was also tested across 17 different accident types from the NPPAD dataset, with zero false negatives, indicating that all anomalies were accurately identified. Furthermore, with sufficient normal condition data for training, the false positive rate has the potential to be reduced to zero, further enhancing the model reliability in practical applications.

6. ENHANCED EXPLAINABILITY WITH LLM

To enhance explainability, LLMs are leveraged to analyze possible accident types and pinpoint likely areas of concern using the key features contributing to the anomaly. LLMs like GPT, LLAMA, Phi, Mistral, Gemini, etc., excel in generating human-like text for diverse tasks such as programming, creative writing, report generation, customer support, and decision-making. LLMs are prediction engines that take a sequence of words and tries to predict the most likely sequence to come after that sequence. It does this by assigning probabilities to likely next sequences and sampling from them to choose one. The process repeats until a stopping condition is met [9] [10] [11] [36] [37].

6.1 Supervised fine-tuning (SFT): In SFT, a pre-trained LLM is further trained on a labeled dataset using supervised learning methods. This process involves three steps to optimize the model for specific tasks [38] [39].

Pre-training: The foundation model is trained on a large dataset to learn language patterns, grammar, and context by predicting the next word, building a broad understanding of language.

Data Labeling: A labeled dataset is created for supervised learning, guiding the model in adjusting its parameters.

Fine-tuning: The pre-trained model is further trained on a task-specific labeled dataset to improve performance in tasks like text classification, sentiment analysis, or question-answering.

Common supervised fine-tuning methods for LLMs include LoRA (Low-Rank Adaptation) and its memory-efficient variant, QLoRA (Quantized LoRA). Both are part of the Parameter-Efficient Fine-Tuning (PEFT) family, designed to improve fine-tuning efficiency [40] [41].

6.2 Low-Rank Adaptation(LORA): In natural language processing, a common approach is to pre-train models on large amounts of general data and then adapt them to specific tasks. However, as models get larger, fine-tuning all the parameters becomes harder and more expensive. For example, fine-tuning a model like GPT-3 with 175 billion parameters can be very costly. To solve this problem, researcher proposed Low-Rank Adaptation(LoRA) techniques, which freezes the pre-trained model weights and injects trainable lower-rank decomposition matrices into each layer of the Transformer architecture, greatly reducing the number of trainable parameters for downstream tasks. Compared to GPT-3 175B fine-tuned with Adam, LoRA can reduce the number of trainable parameters by 10,000 times and the GPU memory requirement by 3 times [42]. LoRA's approach to fine-tuning uses low-rank decomposition to represent weight updates with two smaller matrices, reducing the number of trainable parameters and making fine-tuning more efficient. QLoRA (Quantized LoRA)is a memory-efficient variant of LoRA that further reduces the memory requirements for fine-tuning large LLMs. Quantization is a technique to reduce computational and memory costs by using low-precision data types or fewer bits to represent data, such as 8-bit integers (int8) instead of 32-bit floats(float32) [40] [41] [43]. Reducing the number of bits means the resulting model requires less memory storage, consumes less energy, and operations like matrix multiplication can be performed much faster with integer arithmetic.

Once the deep autoencoder identifies the top contributing features, their increased or decreased values are provided to fine-tuned LLM model using impact-based system prompts to explain the causes of the anomaly, pinpoint critical areas with recommended maintenance actions, and produce a clear, human-readable report.

7. CONCLUSION AND FUTURE WORK

This research presents a proactive anomaly detection method for nuclear power plants using deep autoencoders with LeakyReLU and ELU activation functions, integrated with a reconstruction-threshold-based approach and large language models (LLMs) for enhanced explainability. The model effectively captures non-linear dependencies in normal operational data, identifies abnormal patterns through elevated reconstruction thresholds, and highlights key contributing features to support operator decision-making and timely maintenance actions. Experiments conducted on the PC-TRAN generated Nuclear Power Plant Accident Data (NPPAD) demonstrated strong performance across diverse accident scenarios and normal operations, enabling the establishment of a reliable threshold-based detection method. The integration of LLMs further strengthened the method by providing contextual analysis of potential accident types, improving decision support and operator awareness in high-risk environments. By combining unsupervised learning with explainable AI, this approach allows a shift from reactive to predictive maintenance strategies, enhancing operational safety, reducing downtime, and supporting resilient nuclear power plant management. Future work will focus on real-time implementation using streaming sensor data and edge computing devices to improve responsiveness and operational efficiency. Incorporating real-world operational data collected over one to two refueling cycles will enhance generalization and robustness, while accounting for environmental conditions, equipment lifecycle, and system disturbances will ensure adaptability under complex conditions. Additionally, the applicability of this method will be explored in other critical infrastructure sectors, and hybrid approaches that combine knowledge-driven and data-driven models will be investigated to further improve anomaly detection, interpretability, and predictive maintenance capabilities across diverse operational environments.

References

[1] Sepideh Pashami, Slawomir Nowaczyk, Yuantao Fan, Jakub Jakubowski, Nuno Paiva, Narjes Davari, Szymon Bobek, Samaneh Jamshidi, Hamid Sarmadi, Abdallah Alabdallah, et al. Explainable predictive maintenance. *arXiv preprint arXiv:2306.05120*, 2023.

[2] George Bereznai. Nuclear power plant systems and operation. *University of Ontario Institute of Technology, Oshawa*, 2005.

[3] World Nuclear Association. Nuclear power in the world today, September 2024. URL `https://world-nuclear.org/information-library/current-and-future-generation/nuclear-power-in-the-world-today`.

[4] Ramadass Sathya, Annamma Abraham, et al. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2):34–38, 2013.

[5] Ritu Sharma, Kavya Sharma, and Apurva Khanna. Study of supervised learning and unsupervised learning. *International Journal for Research in Applied Science and Engineering Technology*, 8(6):588–593, 2020.

[6] Naoya Takeishi. Shapley values of reconstruction errors of pca for explaining anomaly detection. In *2019 international conference on data mining workshops (icdmw)*, pages 793–798. IEEE, 2019.

[7] Ben Qi, Xingyu Xiao, Jingang Liang, Li-chi Cliff Po, Liguo Zhang, and Jiejuan Tong. An open time-series simulated dataset covering various accidents for nuclear power plants. *Scientific data*, 9(1):766, 2022.

[8] W Simulator. Pctran generic pressurized water reactor simulator exercise handbook. *Vienna: International Atomic Energy Agency*, 2019.

[9] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

[10] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

[11] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[12] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12): 12591–12604, 2023.

[13] Ben Qi, Jingang Liang, and Jiejuan Tong. Fault diagnosis techniques for nuclear power plants: a review from the artificial intelligence perspective. *Energies*, 16(4):1850, 2023.

[14] Xiangyu Li, Tao Huang, Kun Cheng, Zhifang Qiu, and Tan Sichao. Research on anomaly detection method of nuclear power plant operation state based on unsupervised deep generative model. *Annals of nuclear energy*, 167:108785, 2022.

[15] SA Cancemi, R Lo Frano, C Santus, and T Inoue. Unsupervised anomaly detection in pressurized water reactor digital twins using autoencoder neural networks. *Nuclear Engineering and Design*, 413:112502, 2023.

[16] Abhishek Chaudhary, Junseo Han, Seongah Kim, Aram Kim, and Sunoh Choi. Anomaly detection and analysis in nuclear power plants. *Electronics*, 13(22):4428, 2024.

[17] Adriano Liso, Angelo Cardellicchio, Cosimo Patruno, Massimiliano Nitti, Pierfrancesco Ardino, Ettore Stella, and Vito Renò. A review of deep learning-based anomaly detection strategies in industry 4.0 focused on application fields, sensing equipment, and algorithms. *IEEE Access*, PP:1–1, 01 2024. doi: 10.1109/ACCESS.2024.3424488.

[18] Xingyu Xiao, Jingang Liang, Jiejuan Tong, and Haitao Wang. Emergency decision support techniques for nuclear power plants: Current state, challenges, and future trends. *Energies*, 17(10):2439, 2024.

[19] Sidharth Prasad Mishra, Uttam Sarkar, Subhash Taraphder, Sanjay Datta, Devi Swain, Reshma Saikhom, Sasmita Panda, and Menalsh Laishram. Multivariate statistical data analysis-principal component analysis (pca). *International Journal of Livestock Research*, 7(5):60–78, 2017.

[20] Liton Chandra Paul, Abdulla Al Suman, and Nahid Sultan. Methodological analysis of principal component analysis (pca) method. *International Journal of Computational Engineering & Management*, 16(2):32–38, 2013.

[21] Mohammed Kareem and Lamia Muhammed. *Anomaly Detection in Streaming Data using Isolation Forest Tree Mohammed Shaker Kareem Supervised by*. PhD thesis, 07 2024.

[22] Fei Tony Liu, Kai Ting, and Zhi-Hua Zhou. Isolation forest. pages 413 – 422, 01 2009. doi: 10.1109/ICDM.2008.17.

[23] Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee, and Chiew Tong Lau. Autoencoder-based network anomaly detection. In *2018 Wireless telecommunications symposium (WTS)*, pages 1–5. IEEE, 2018.

[24] Umberto Michelucci. An introduction to autoencoders. *arXiv preprint arXiv:2201.03898*, 2022.

[25] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1–18, 2015.

[26] PyTorch Team. torchvision.transforms.v2.gaussiannoise, November 2024. URL `https://pytorch.org/vision/master/generated/torchvision.transforms.v2.GaussianNoise.html`.

[27] Eric W Weisstein. Normal distribution. *https://mathworld.wolfram.com/*, 2002.

[28] Lucas BV de Amorim, George DC Cavalcanti, and Rafael MO Cruz. The choice of scaling technique matters for classification performance. *Applied Soft Computing*, 133: 109924, 2023.

[29] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12): 310–316, 2017.

[30] Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*, 2019.

[31] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, 2022.

[32] Juan Terven, Diana M Cordova-Esparza, Alfonzo Ramirez-Pedraza, and Edgar A Chavez-Urbiola. Loss functions and metrics in deep learning. a review. *arXiv preprint arXiv:2307.02694*, 2023.

[33] Aryan Jadon, Avinash Patil, and Shruti Jadon. A comprehensive survey of regression-based loss functions for time series forecasting. In *International Conference on Data Management, Analytics & Innovation*, pages 117–147. Springer, 2024.

[34] Twan Van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

[35] Ž Vujović et al. Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, 12(6):599–606, 2021.

[36] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[37] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[38] Hugging Face. Sft trainer documentation, 2024. URL `https://huggingface.co/docs/trl/en/sft_trainer`.

[39] KLU AI. Supervised fine-tuning glossary, 2024. URL `https://klu.ai/glossary/supervised-fine-tuning`.

[40] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023.

[41] Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *arXiv preprint arXiv:2408.13296*, 2024.

[42] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[43] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.