An Intelligent Hierarchical Framework for Efficient Fault Detection and Diagnosis in Nuclear Power Plants

Jean Carlos Tonday Rodriguez
Florida International University
Miami, Florida, United States
jtond004@fiu.edu

David Perry
Florida International University
Miami, Florida, United States
dperr040@fiu.edu

Mohammed Ashiqur Rahman
Florida International University
Miami, Florida, United States
marahman@fiu.edu

Syed Bahauddin Alam
University of Illinois Urbana-Champaign
Champaign, Illinois, United States
alams@illinois.edu

Abstract

The increasing demand for usable power and the pressure to reduce carbon dioxide emissions have increased interest in fossil fuel alternatives. Specifically, nuclear power has received more attention from energy agencies globally, resulting in positive growth for the sector. The need for improved safety systems has increased with the expansion of nuclear power plants (NPP). Traditional fault detection and diagnosis (FDD) methods require high upfront and operational costs. Integrating Machine learning (ML) strategies can present a robust and equally effective alternative while minimizing the necessary time and money. This paper presents a novel framework for fault detection in NPPs. Unlike existing FDD methods that usually rely on single-model designs, we propose a hierarchical framework using a combination of multi and single-class classifiers. For data-driven FDD, one primary consideration is handling noisy scenarios in NPP. We design an algorithm that integrates deep learning multi- and single-class classifiers to improve fault diagnosis robustness, especially under noisy sensor readings. We evaluate our framework across various models and explore the need for a hierarchical approach under noisy and clean data. Our deep learning solution produces comparable results when no noise is present and significantly improves performance as noise is added to the system.

CCS Concepts

• **Hardware → Error detection and error correction**; *Failure prediction*; • **Computing methodologies → Classification and regression trees**; **Neural networks**.

Keywords

Nuclear Power Plant; Fault Detection and Diagnosis; Machine Learning; Unsupervised Learning; Supervised Learning; Hierarchical Framework

CPSIoTSec'24, October 14–18, 2024, Salt Lake City, UT, USA

ACM ISBN 979-8-4007-1244-9/24/10
https://doi.org/10.1145/3690134.3694814

ACM Reference Format:
Jean Carlos Tonday Rodriguez, David Perry, Mohammed Ashiqur Rahman, and Syed Bahauddin Alam. 2024. An Intelligent Hierarchical Framework for Efficient Fault Detection and Diagnosis in Nuclear Power Plants. In *Proceedings of the Sixth Workshop on CPS&IoT Security and Privacy (CPSIoTSec'24), October 14–18, 2024, Salt Lake City, UT, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3690134.3694814

1 Introduction

Nuclear Power Plants (NPPs) have seen rapid growth in recent years due to the increasing requirement for power generation and rising efforts to reduce carbon dioxide emissions [1]. According to the International Energy Agency, in 2023, NPPs accounted for about ten percent of the total power generation in the world [2]. Additionally, the NPP sector is expected to expand by 25 percent in the upcoming years [3]. NPP's access to abundant and clean energy makes them helpful for limiting the rise of global warming in future power generation [4]. As a result, there is a need to secure the plants against human and natural faults to maintain safe operation. Safety incidents such as the Fukushima Dai-ichi reactor meltdown [5] result in extensive loss for the NPP. Therefore, fault detection and diagnosis (FDD) has become a critical topic in recent research regarding NPPs [6, 7] to mitigate the risk and losses associated with faults.

To improve the efficiency of FDD, NPPs have begun to implement modern computational techniques, improving performance through the movement of Industry 4.0 [8, 9]. Industry 4.0 introduces smart manufacturing through machine learning (ML), the Internet of Things, and cloud computing [10]. FDD, in particular, is a field that has benefited from recent innovations during the rise of artificial intelligence (AI) for fault detection. ML, especially using deep neural networks (DNN), can function better in highly complex systems such as NPP. Some existing FDD models using ML employ unsupervised learning [11] to identify a fault occurrence and supervised learning [12] for classification and fault diagnosis. Most models follow this structure with different approaches [13] but cannot develop a scheme to address model misclassification [14].

Furthermore, NPPs are complex and contain significant noise in the collected sensor data. The sensor noise originates from the system's physical components, causing changes in the NPP subsystems, such as the steam generators [15]. Additionally, Noise in NPP directly affects the existing instrumentation and control solutions,

one being FDD. Traditional ML techniques, such as random forest models, are prone to over-fitting, resulting in lower accuracy when used for inference on noisy data [16]. Furthermore, in NPP systems, noise can be generated from many factors, such as mechanical operation or electromagnetic radiation in the NPP electronics [17]. As a result, to provide accurate and secure operation, all controllers and safety systems in the NPP must be capable of handling deviations. More complex models are also required to improve the noise-handling capabilities of AI-based FDD.

DNNs, which have increased usage in recent years, look to improve the robustness of the FDD process through larger models. DNN provides strengths in effectively analyzing, predicting, and classifying complex systems and equations that benefit NPP FDD. However, DNN models require a large amount of data or risk overfitting and losing generalizability [18]. Overfitting occurs when the model's performance on untrained data significantly differs from the data it was trained on. Overfitting becomes more prevalent when noise is present due to changes in data distribution, causing a further loss in FDD performance [19] for many cyber-physical systems. To leverage DNN's robustness and accuracy in complex systems, we look to implement a DNN-based model in a hierarchical framework to minimize the trend of overfitting and improve model robustness.

In this paper, we introduce a hierarchical framework for FDD in NPP to overcome the drawbacks of traditional FDD methods. Our framework utilizes supervised and unsupervised ML models to identify the operational condition of the NPP. The state of the NPP is described as one of the possible faults or the normal operation. We propose using a supervised multi-class classifier for initial state identification. The classification model learns the overall structure of all possible operational conditions of the NPP and selects the best label, or state, given sensor readings. The supervised classifier is then augmented using unsupervised models to obtain relationships of data belonging to the same state. The unsupervised approach compares the performance of models trained on different labels on the same sample to select the most probable condition by leveraging the class rewards. Unsupervised ML is used as it is more difficult to overfit and performs better under noisy scenarios. Furthermore, we explore using deep learning techniques to improve the robustness and accuracy of the FDD process. Our paper uses the proposed hierarchical approach to make the following contributions:

- We propose a hierarchical framework that uses supervised and unsupervised ML techniques for detecting and diagnosing faults in NPP operations under clean and noisy data.
- We propose a fault diagnosis algorithm composed of multi- and single-class classification models using Long Short Term Memory techniques to improve the performance under noisy data.
- We conduct a detailed evaluation of the proposed hierarchical framework by exploring the accuracy of different models under clean and noisy data.

The rest of the paper is organized as follows. We explore existing literature and related works in Section 2. We then provide an overview of a pressurized water reactor NPP, including the subsystems, possible faults, and the presence of noise, as well as the implementation of ML and DNN for FDD in Section 3. We represent our proposed hierarchical solution and the model architectures in Section 4. Section 5 provides our detailed evaluation of the performance and strength of the proposed solution in optimal and noisy environments. Finally, Section 6 concludes our work and provides future direction.

2 Related Works

Utilizing classical ML and DNN technologies for FDD in mechanical systems has gained traction due to their ability to function in complex scenarios such as NPPs [20–22]. However, traditional FDD methods are still widely used in such systems. Traditional fault diagnosis in the nuclear energy sector includes hardware redundancy, model, and signal-processing-based procedures [23]. Hardware redundancy involves multiple sensors in a system gathering the same data, where any discrepancy between the redundant readings signifies potential faults. Alternatively, a model-based approach generates a mathematical representation of the system, which compares the predicted steady-state output with the real sensor values, such as the one by Gross et al., [24]. Signal-processing-based procedures extract information related to instrumentation channels during plant operation and allow monitoring subsystems from a statistical approach. However, when new data is presented, traditional FDD is not transferable between NPPs and struggles with generalizability.

ML techniques offer a more accurate and robust alternative to FDD than conventional methods. ML models, such as the one described by Elshenway et al., require minimal prior knowledge about the specific NPP and can be employed to predict the plant's behavior [11]. The shift towards ML signifies an advancement in fault detection, presenting a more streamlined and robust approach to ensure the safety and reliability of NPPs. Prior investigation into ML for fault detection in 2021 by L. Elshenawy [11] provided insight into FDD using unsupervised ML techniques. The research focused on a limited number of common faults, principal component analysis (PCA) performance for detection, and multivariate contribution plots for diagnosis. Furthermore, ML techniques for single-class classification, such as outlier detection, have been used previously for NPP FDD [25, 26]. For example, Lv et al. propose a fast-density peak clustering method to overcome the faults of standard outlier detection algorithms in power data research [27]. Huang et al. apply outlier detection to secure process control in industrial cyber-physical systems from inconsistencies like noisy sensor readings [28]. Similarly, multi-class classification algorithms have also seen growth. For example, the work by Li et al. studied five diagnosis models for their wide use and performance in classification applications for NPP FDD [29]. Unlike previous research, the work explored in this paper investigates the combination of different ML techniques for FDD through a hierarchical framework to improve performance.

Deep learning techniques have also been explored for FDD, such as Qian et al., who deployed a deep gated recurrent unit network for fault diagnosis in NPPs [30]. Other research has focused on employing long short-term memory (LSTM) architecture to diagnose NPP faults. However, DNN has been shown to require significant data overhead to reduce overfitting. A hierarchical approach can be taken to increase the robustness of the DNN through a more precise fault verification step without the need for more training

data. Liu et al. propose using LSTM to develop FDD systems based on deep learning [31], which shows high accuracy when the sensor data is clean. However, this paper does not consider the presence of noise in the system, making their proposed solution less robust. Similarly, Rajkumar et al. propose an LSTM classification model mainly focused on detecting faults in small nuclear reactors [32]. This paper only considers one NPP component, removing some complexity in identifying overall faults that affect multiple NPP subsystems. Unlike the existing work, our paper aims to generate a more robust architecture that efficiently identifies complex faults in NPPs, even under noisy scenarios.

In addition to Deep learning and AI techniques, methods for noise handling have become widely used in NPP FDD. The prevalence of noise in NPP systems makes handling it a required step for the correctness of FDD. One method of implementing noise reduction is training FDD models on artificially added noisy samples. For example, Ghosh et al. propose using noise analytics and artificial neural networks to predict individual mechanical components' detection precision in NPP [33]. On the other hand, Li et al. consider the robustness of five different data-driven methods for FDD in noisy NPPs to measure the performance of each model [29]. However, both papers include noise in the training dataset to add noise to the model's training process. This means that the model transferability in different noise levels is not guaranteed. The lack of generalizability is further shown by the distinct variations of noise in NPP, which may change depending on the physical characteristics of the components and electrical systems. In contrast, our work utilizing the dataset in [34] trains the models on five noise levels, overcoming this lack of generalizability.

A secondary technique for improving robustness is utilizing denoising modules. For example, Shaheryar et al. consider using denoising-based regularization parameters to reduce the impact of noise on sensor validation model [35]. Another research in this area includes Zhong et al., who propose using an additional NPP FDD system module to detect and filter noise features in the data [36]. While effective at reducing noise, this approach can also remove critical information from the data, reducing accuracy. Our approach looks to build a more robust solution using a hierarchical method to improve its generalizability under different noises without denoising strategies or including noise during training.

3 Background

This section briefly overviews the NPP's operational structure and how ML can be implemented for FDD. Figure. 1 showcases a three-loop pressurized water reactor (PWR) NPP with one brief overview of the NPP's operational structure team generator.

3.1 PWR Operational Overview

The primary function of an NPP is to produce power, which requires careful management of energy and heat. NPPs rely on various subsystems to oversee heat transfer to ensure a steady and secure power supply while maintaining temperature and pressure within the safe range. This paper focuses on the PWR NPP, examining each subsystem to showcase where faults can occur.

PWR plants transfer energy from the reactor vessel to the steam generator using a high-pressure liquid coolant. The pressurizer is

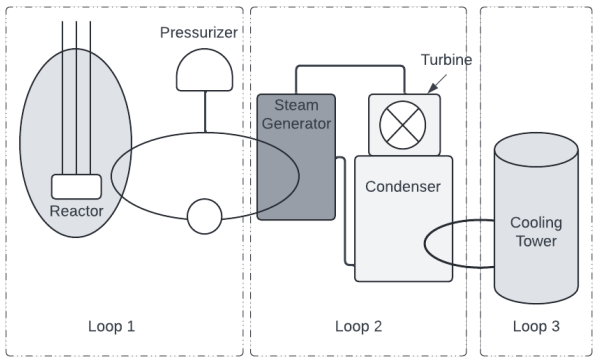

Figure 1: General design of PWR NPP including reactor cooling system and power generation loop.

primarily responsible for maintaining the coolant in a liquid state by keeping high pressure in the system. The pressurizer measures the water level in its tank as a guideline to control cooling sprays and heaters, which regulate the coolant's pressure. By maintaining the coolant liquid, the reactor will have access to consistent water to subside and transfer heat. A consistent coolant supply is needed as nuclear fission produces high energy; in an NPP, the reactor consumes fuel rods to generate power through nuclear fission. In a fission reactor, an atom is split through a neutron collision, using uranium to fuel the reaction. Nuclear fission can produce extensive heat to generate electrical power. If the heat is not controlled, the PWR can reach meltdown, resulting in catastrophic side effects.

In a PWR, the energy is transferred by the liquid coolant, which is carried through the reactor coolant system (RCS) with pumps located in the hot and cold legs of the system. The hot leg carries high-energy water from the reactor to the steam generator, while the cold leg carries the low-energy liquid back to the reactor. The steam generator uses U-bend tubes to transfer heat from the high-energy coolant to the fluid in the power generation loop. The steam generator obtains a low-pressure liquid from the feedwater pipe in the power generation loop that increases temperature until it evaporates into steam. The secondary loop does not necessitate constant high pressure, allowing feedwater to evaporate and flow into the turbine generator as a high-energy gas.

The turbine fan then spins using steam from the generator to provide electricity using a method similar to coal-fueled plants. After the steam leaves the turbine, it cools down and sinks into the condenser. The condenser uses a non-water coolant to turn the steam back into liquid feedwater, which the feedwater pump returns to the steam generator. While the secondary loop is less volatile to pressure changes than the RCS, it is directly related to the plant's power generation. It is mainly affected by changes to the electrical load of the system.

As shown, while PWR plants are segmented into major subsystems, each works in tandem with the others. Notably, the high-pressure region of the plant is stored within a containment structure and is prone to higher reactivity and temperature. The containment

region produces the heat through nuclear fission needed to generate steam and subsequent power and shares many safety systems amongst the components. On the other hand, the secondary loop transfers the steam to a turbine generator and directly interfaces with the electric load. Both the primary and secondary loops depend on each other for power generation. As a result of the interconnected nature of the plant, fault observations can be seen in multiple components regardless of where they initially occur.

3.2 Nuclear Power Plant Faults

NPPs are inherently complex and interconnected systems, where each component is vulnerable to faults. Additionally, fault effects can be seen in multiple areas, and differing faults can cause similar effects in multiple NPP components. As a result, it can be challenging to identify the NPP condition by analyzing each component individually, especially in the presence of noise. However, a more comprehensive model that examines all NPP systems cohesively can improve fault detection results in non-optimal conditions. We showcase the complexity of NPP faults by analyzing the dataset presented by Qi et al. [34], who simulates a PWR NPP in different runtime conditions, including 17 distinct faults; this dataset is used in this paper for our evaluation.

For example, the Loss of Coolant Accident (LOCA) is one of the faults from the dataset that showcases the difficulty in fault detection. Usually occurring due to mechanical failure, LOCA is generally categorized by a decrease in the quantity of coolant or a reduction in RCS pressure. The two major causes of LOCA are a coolant leak in the RCS transfer pipes or a pressurizer control system failure. LOCA causes a reduction in the RCS's temperature, as the coolant's volume is insufficient to transfer the heat properly. As the temperature decreases, a reduction in pressure will be seen, which may result in further evaporating coolant. In addition to the temperature and pressure decrease in the RCS, the reactor's reactivity will rise when the fuel rods are not sufficiently cooled to control the reaction. The steam generator will also decrease steam production as insufficient heat is transferred to the U-bend tubes. In extreme cases, a LOCA fault can result in a fuel rod meltdown, such as at the Chornobyl Power Plant [37]. The primary solution for minimizing meltdowns is a SCRAM, a rapid nuclear shutdown, at the first detection of a LOCA fault. If a reactor meltdown occurs, the plant's shutdown will not remove its effects, making early fault detection critical for the safe operation of NPPs.

Unlike LOCA, a rod withdrawal (RW) fault can occur because of an operator mistake or control system failure. In this case, the reactor's control rods, responsible for regulating the reactivity and temperature of the nuclear fission, are withdrawn from the system. RW will see a rapid rise of reactivity in the reactor core, increasing temperature and pressure on the RCS. If only the reactor were considered, LOCA and RW would see increased reactivity and temperature. However, when multiple components are analyzed, it can be seen that LOCA affects the RCS differently than RW. LOCA directly causes a decrease in RCS temperature and pressure, while RW does the opposite. Therefore, looking at only one component to identify faults is insufficient, as they can overlap in a singular subsystem and show differences holistically. An improved approach is to consider the state of all plant systems to differentiate faults.

3.3 Noise In Nuclear Power Plants

The difficulty of FDD in NPPs is due to the interconnectivity and overlap of different faults and functions by components, which are amplified by noise. In particular, NPP noise can cause deviation in the sensor readings from natural causes or sensor tuning, causing values to be altered as noise is added to the system. Noise can occur in an NPP due to physical elements such as mechanical and electrical systems. To reduce classification errors, the FDD must be capable of handling the noise in the components and sensors in NPPs. Component noise can be considered one of two types: mechanical due to manufacturing inconsistencies and electrical due to electromagnetic interference during sensing or transfer.

One cause of mechanical noise in the NPP system is flow-induced vibration in the physical components [38]. In contrast, electrical noise is mainly caused by the electromagnetic radiation introduced to the sensing equipment during instrumentation and control [39, 40]. Even in cases where compatibility of electromagnetic components is considered, the innate complexity of NPP and the vast amount of sensors and controllers makes it challenging to remove the noise completely. Noise can cause deviations in the correctness of sensing data in the NPP. Additionally, noise can decrease the accuracy of the FDD as differences between faults become harder to detect. As noise is generally part of the physical components in the NPP, denoising can result in a loss of granularity in the data. We propose robust DNN models to handle noise in FDD tasks without data-altering denoising techniques to maintain granularity.

In addition to noise generated by independent sensors, communication between controllers will also affect the NPP data. Communication error is the noise within the channel from the sensor device to the controllers in the NPP. This type of noise is more prevalent the longer the communication channel is and the more nodes it has to pass through. In addition, communication overhead and possible errors can result in difficulty identifying where the NPP noise originates, making it hard to implement denoising strategies using a centralized FDD module. The data obfuscation due to the presence of noise further showcases the requirement for robust and generalized FDD models.

3.4 Nuclear Power Plant Fault Detection and Classification

Faults can occur in any subsystem due to natural causes such as earthquakes or operator errors. To defend against prolonged NPP damages, the concept of FDD must be implemented. In traditional NPPs, expert diagnosis or physics-based approaches have previously handled fault detection. However, due to the rise of AI and the abundance of data, ML use for FDD has become more commonly explored [41].

The two main varieties of ML techniques are unsupervised and supervised learning. Unsupervised learning is traditionally used for anomaly detection within systems. For NPP fault detection, unsupervised learning is a powerful tool for identifying if a fault is occurring. Traditional unsupervised learning for anomaly detection involves clustering algorithms and single-class classification models. On the other hand, supervised learning uses labels for their predictive task. Supervised learning tends to be more commonly

used for classification and identification problems. Algorithms for supervised learning include decision trees and random or isolation forests. Unsupervised learning excels when the specific class of the data is not essential, while the alternative can be used to directly identify the class of the sample. At the same time, unsupervised learning can learn the critical features of the data better than supervised approaches.

Our work aims to establish a combination of both ML techniques to detect and identify a particular fault more accurately. A hierarchical approach can provide a greater understanding of the relationship between the features through unsupervised learning and the efficiency of the supervised approaches. This solution and the implementation of more robust models look to improve FDD in the presence of noise.

3.5 Deep Neural Network in Nuclear Power Plant Fault Diagnosis

DNN techniques are categorized by their large amount of data and parameters, which are used to find mappings within highly complex nonlinear systems effectively. DNN models can be described by a mathematical relation $f_w(\cdot) : X \rightarrow y$. The DNN model, f, contains a set of weights w, which maps the input X to a specific output y using a mathematical representation defined by the model architecture. DNN is typically used to model highly complex nonlinear relationships. As a result, most DNN models are considered black boxes, where the relationships are not inherently obvious. DNNs use a data-driven approach to update their parameters w and minimize the loss between the ground truth, y_{gt}, and $f(X)$, the prediction. DNNs have been used for various tasks, including regression, classification, and natural language processing.

In particular, DNN models can be leveraged on NPP data due to nonlinear relationships between the physical processes and control structure of the NPP. While faults in NPP, as previously stated, can be simplified when only considering a few possible ones, there are many errors in an NPP system, which are represented by diversion to the nonlinear dependencies within the control systems. As a result, DNN, given the complexity and non-linearity in the control loop of NPP, has been recently used to capture and represent all portions of the plant completely and thoroughly.

The most basic DNN is the Multi-Layer Perceptron (MLP). MLP uses multiple neurons at different layers or levels where each neuron performs weighted sum operations and applies the activation function. The more neurons and layers in an MLP model, the more required parameters. Larger models capture deeper dependencies from the training dataset. However, complex models can also overfit the training dataset with insufficient samples. Furthermore, given the nature of the time series of NPP controllers and sensors, recurrent networks can be used to capture temporal dependencies. The long short-term memory (LSTM) architecture is a recurrent neural network that uses memory cells, forget, and input gates to find dependencies between data in a time series format. Using the positional relationship within a series, LSTM shows better results for problems where data depends on past entries. Models such as LSTM are typically used for time series analysis of complex systems, given LSTM's strengths in analyzing long-term sequences. LSTM's use of memory cells allows it to analyze the sequence's temporal relationships between far-away samples. To use temporal models, a time window is generally built using historical samples to generate dependencies between current and past values. LSTMs are commonly used in supervised learning tasks for classification or regression. In the case of our proposed model, we use LSTM to learn the time series relationships of the control systems in an NPP

For unsupervised DNN architectures, one of the most commonly used methods is the Autoencoder (AE) neural network. AE uses DNN architectures to build a data-driven model reducing the input feature space $\mathbf{F}$ to a secondary reduced dimension $\mathbf{F_r}$ and then reconstruct it to the original space $\mathbf{F}$. AE is typically employed in dimensionality reduction, where the reduced space contains as much critical information as possible from the original data. AE can also be used for single-class classification, where the model, $f(\cdot) : X \rightarrow X$, learns to return X to itself by first reducing X's dimension through an encoder. If all training data X belongs to one class, the AE will only learn to reconstruct values for that class. As a result, if a high error is encountered while applying inference to the AE model, the tested sample could be considered an outlier. Furthermore, AE is designed to reduce the feature space through DNN methods that best represent the data. The models in AE modules, $f_w(\cdot) : X \rightarrow X_r$, can be defined depending on the data structure using any DNN architecture. In the case of sequential and temporal data analysis, the LSTM architecture previously described can be employed to learn the temporal and spatial relationship of the NPP sensor data. This paper explores using both LSTM and LSTM-AE models in a hierarchical nature to improve the robustness of FDD under noise.

4 Proposed Solution

This section provides an overview of our proposed hierarchical FDD solution. We showcase our solution in Figure. 2, which uses the current sensor readings of the NPP S^c and DNN models to classify the plant into a particular fault or normal condition.

4.1 Overview

Our hierarchical framework is divided into two components to leverage the benefits of unsupervised and supervised learning. Our framework uses supervised models to reduce inference runtime and unsupervised models to produce more accurate and robust predictions. We propose the architecture showcased in Figure. 2 combining both types of models to improve the accuracy of FDD in the presence of noise. In figure 2, S^c represents a test sample for inference of shape $(t, \mathbf{F})$, where t is the time window length of the sample, and $\mathbf{F}$ is the number of sensor readings at each timeslot. The sample S^c is scaled and used in the multi-class classifier, designed with the LSTM architecture described in section 3.5. LSTM is used for NPP FDD to find non-linear temporal relationships within long-time sequences, such as in NPP. We use the classifier to select the best n corresponding classes and F^n, the n most probable single-class classifiers. We then use all top selected models and choose the one with the best metric score. The goal of leveraging the robustness of unsupervised learning is to improve the accuracy of identification in the presence of noise. An LSTM-AE architecture is used to build the single class classifiers described in section 3.5 for the temporal analysis of the time series sensor data.

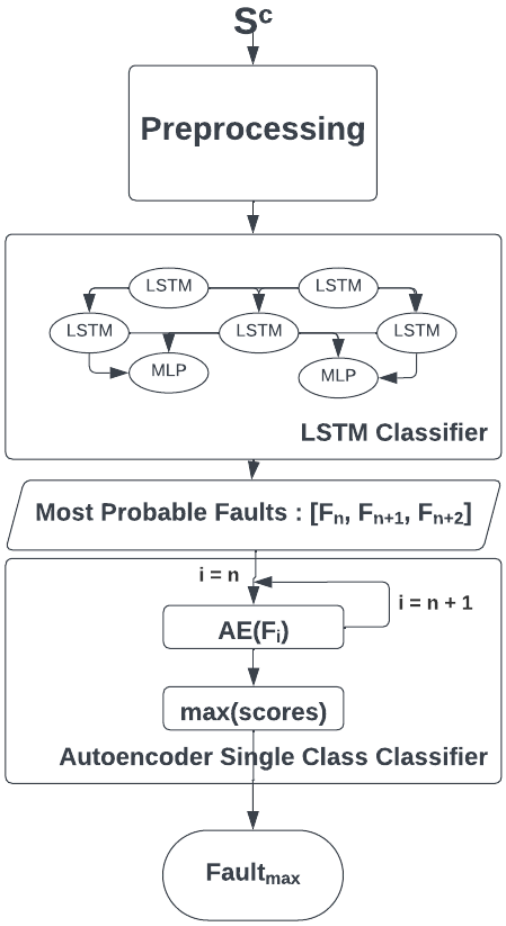

Figure 2: Overview of proposed solution.

As mentioned, our approach uses parameter n, the top number of single-class models, to compare during classification. There is an inherent tradeoff between n and the inference speed of the proposed solution. A smaller n reduces the number of models requiring inference, increasing our framework's speed. At the same time, a small n also emphasizes the LSTM classifier, which is more prone to overfitting than the LSTM AE in noisy scenarios. As a result, it is essential to consider minimizing n while maintaining a high level of robustness.

4.2 Long Short Term Memory Classifier

We propose using the LSTM classifier for multi-class classification in the FDD hierarchical framework. NPP control system, signals, and sensor values rely on time series integration controllers, which produce inherently temporal relations in the data. We choose LSTM to leverage the long-term time series nature of NPP data, which allows the model to learn from the entire time sequence without losing any dependencies.

Multi-class classification can be formulated as a machine learning problem defined by the model $f_{LSTM}(\cdot) : S^c \rightarrow y$, where S^c is the time series input from the NPP, and y is the output probabilities all classes of the training data. In our problem, S^c is represented by a time series window, where each window is composed of t timeslots, and each timeslot contains **F** features. These features represent sensor readings and operational control signals of the different components in a PWR NPP. By examining the last t collected sensor readings, LSTM can learn how the controllers interacted with the data, generating a better classification model for that particular dataset. On the other hand, the output y is an array of probabilities that contains the confidence that S^c belongs to that specific class. We then take the n classes that contain the highest probabilities from the indexes of y. The n classes are then used to obtain the most probable single-class classifiers for further evaluation.

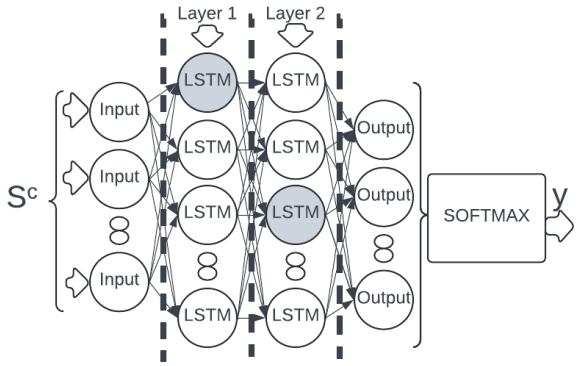

Figure 3: Architecture of proposed LSTM classification model with Softmax activation and gray Dropout cells.

The proposed LSTM architecture is shown in Figure. 3, consists of two hidden LSTM layers, each with a Tanh activation function and a Dropout, symbolized by the gray LSTM cells. The Tanh activation function provides an output within the $[-1, 1]$ range, which fits the scaled data. The dropout layer randomly selects p cells and sets their weights to zero, effectively removing them from the model. Adding dropout to LSTM helps improve the robustness and generalizability of the architecture. The final component we use is the Softmax activation function, which finds the probability that the input belongs to each class in the training dataset, turning the LSTM into a classification model.

4.3 LSTM Autoencoder Model

The second component of the hierarchical solution is the proposed AE. The LSTM AE model can help better understand the temporal relationship of the data in an unsupervised manner. The AE can detect when an operational state differs from the information on which it was trained. However, unlike a supervised model, the AE learns the relationship of the temporal features in the data but not what class it belongs to. All possible models must be compared to select which provides the closest match to support a multi-class classification with AEs. In this case, we can build each AE as a regressor, where we define a model $f^c_{AE}(\cdot) : S^c \rightarrow S^c$, which establishes a regression model for one class (f^c_{AE}) that maps the sample

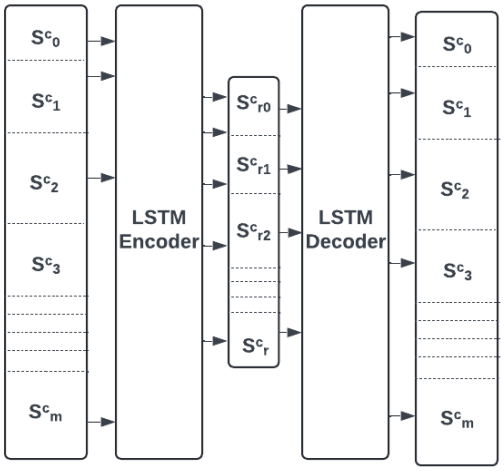

Figure 4: Architecture of proposed LSTM AE where S_m represents the input sample S with m dimensions and S^r the reduced sample space.

space belonging to the class c, S^c to itself. Self-mapping is integral for AE, which relies on changing the data dimensionality before reconstructing it to the original values. Using only one class per model, f^c_{AE} will be able to effectively rebuild samples belonging to class c but have higher errors when rebuilding unknown samples. We can apply this formulation to build one AE model per class for all possible class labels in our dataset.

As shown in Figure. 4, our AE architecture consists of a single LSTM hidden layer in the encoder and decoder using Tanh activation. We take our input S^c with timeslots t and features F and apply an LSTM layer operation. The layer will capture the temporal relationships of the model in a flattened array S^c_r, with F_r features. We then use a time-distributed LSTM layer to reconstruct the original S^c from the reduced features S^c_r. We evaluate the reconstruction error of the model by comparing the output $f_{AE}(S^c)$ with the input S^c. The mean square error(MSE) is used as the primary comparison metric, described in Equation 1.

$$MSE(X^p, X^t) = \frac{1}{N}\sum_{i=0}^{N-1}(X^p_i - X^t_i)^2 \tag{1}$$

In the previous equation, N is the number of input and reconstructed output features. We build LSTM AE models of the same architecture for each class in the dataset. Afterward, when an unknown sample requires classification, each model must be evaluated, and the class corresponding to the smallest reconstruction error is taken.

4.4 FDD Framework Algorithm

This section describes the proposed FDD hierarchical framework in detail. We show our solution in algorithm 1, using the previously trained multi-class classifier F_{LSTM} and the group of single-class classifiers $\mathcal{F}_{AE}$. The algorithm's input is the sample S^c, and the output consists of the predicted condition of the plant p.

The first step of our algorithm is to use F_{LSTM} to find the probabilities that the sample S^c belongs to each of the possible p^c, which we denote as the probability array y, as shown in line 1. After that, we sort all of our probabilities in decreasing order, which we store in the class array c_{all}. We then select the top n classes c_n from the decrementally sorted c_{all} in lines 2 and 3. Lines 4 to 6 obtained the pre-trained corresponding single-class classifiers F^n_{AE} and initialized the predicted class p_c and maximum score $loss_{max}$. The loop from lines 7 to 11 evaluates each model in F^n_{AE} and calculates the criteria for class selection. The for loop is conducted for each n classifier, and the one with the best metric is selected. As shown, the model evaluation stage is not stopped early to guarantee the complete robustness of the unsupervised step. As a result, the model goes through all possible n classes to remove any significant bias from the multi-class classifier. In our proposed hierarchical solution, we define F_{lstm} as the LSTM multi-class-classifier and each F_{AE} as the unsupervised LSTM AE models, and the criteria $\mathcal{L}$ as the MSE loss defined in Equation 1.

Also, specific considerations must be taken to implement the hierarchical framework into existing NPP safety systems. We propose a centralized environment where the framework can access all the sensors' data for integration. Centralizing the FDD system allows us to meet the higher computing necessities DNN models require adequately. They also have access to a complete view of the NPP state instead of fewer sensors. Another real-world implementation consideration is time window generation, as individual sensors are expected to provide the most recent reading at each control step. A historical database of at least size t is required to store the last t samples. The database is then used to build the time window for the obtained sensor values. This centralized approach lets us contain the historical data of all sensors, allowing us to analyze the NPP in its entirety.

Algorithm 1: FDD Framework Algorithm

Data: Classification model F_{lstm}, input sample s^c and set of identification models $\mathcal{F}^n_{AE}$
Result: Corresponding fault label p

1 $y = F_{lstm}(s^c)$
2 $c_{all} = SORT(-1 * y_c)$
3 $c_n = c_{all}[0...n]$
4 $F^n_{AE} = \mathcal{F}_{AE}[c_0...c_n]$
5 $p_c = None$
6 $loss_{max} = inf$
7 **for** F^c_{AE} *in* F^n_{AE} **do**
8 $loss = \mathcal{L}(s^c, F^c_{AE}(S^c))$
9 **if** $loss_{max} > loss$ **then**
10 $p = i$
11 $loss_{max} = loss$
12 return p

5 Evaluation

In the following section, we examine the performance of our model compared to nondeep learning approaches using and ignoring time series features as well as a single-model baseline. We also explore a study investigating the impact of adding single-class classifiers to the framework on accuracy and runtime.

5.1 Non-Deep Learning Approaches

We consider two hierarchical approaches to the DNN framework as baselines for evaluation. The first approach is a time series nondeep learning method (TS) to explore the impact of DNN techniques on the model's accuracy in noisy and non-noisy scenarios. The TS approach uses the same architecture described in algorithm 1. However, instead of using time windows and LSTM-based models, the TS extracts the rate of change of each feature within the time windows and appends it to the last sample. The TS feature extraction is done by finding the line that best fits the t window for each feature and obtaining the respective slope. A linear regression model calculates the line of best fit, and the slope is extracted from that model. The slope is then appended to the last entry in the time window. Each feature in the sample will have a distinct slope and be appended accordingly until all features have temporal information.

For the nontime series (NTS) approach, we do not consider any temporal features for the training or testing data. Instead, we use the most recent collected sample without considering any previous dependencies. We implement NTS to evaluate the requirement of a time series approach for NPP FDD. Additionally, we assess the TS and NTS algorithms by examining how increasing the n classes can improve the accuracy and robustness of the proposed solution. The architecture used for TS and NTS in algorithm 1 replace the F_{LSTM} model trained on all 18 classes and each of the 18 F_{AE} single class classification models with nondeep learning approaches.

The ML models used for TS and NTS in algorithm 1 are the random forest classifier replacing F_{lstm}. RF is an ensemble model that works on nontime series data samples, using several decision tree estimators on what class a sample belongs to. Each tree makes an independent decision and uses a consensus algorithm to reach a final answer. Decision trees work by updating weights to allow each node to make the decision that maximizes the entropy of the class division. The ensemble portion of the model will enable it to reduce overfitting by using independent and volatile decision trees that learn independently. The Gaussian Mixture Model (GMM) is used instead of the AE for the single-class classifiers. GMM is a probabilistic clustering algorithm that identifies different distribution clusters that fit the data. GMM uses a standard distribution group that best describes the dataset and considers the distributions as clusters. Any sample significantly outside the GMM clusters can be viewed as an outlier. As a result, the decision criteria in GMM also change from distance loss to a probabilistic metric such as confidence. We propose using the logarithmic likelihood as the criteria for GMM comparisons, which uses the log of the probability that the test sample belongs to any of the distributions defined by the clustering model. We can, therefore, evaluate which GMM best fits that sample using the negative log-likelihood metric.

5.2 Data Description

We test and compare our proposed framework to nondeep learning methods using the synthetic dataset by Qi et al., [34] collected from the PCTRAN NPP simulator. The dataset contains time series data corresponding to the NPP simulation in one of 18 conditions. Additionally, as this dataset contains simulated values, it does not consider any physical sensor noise in the readings collected from the simulator, which requires the adding noise to evaluate model transferability and robustness. The subset of the dataset used consists of over 4000 samples and 98 different features. To improve the class balance of the dataset, we reduced the number of complete simulations used to one per fault. PCTRAN simulates the operation of an NPP given specific input conditions. These conditions are manipulated to reproduce 17 possible faults in a time series format. However, as PCTRAN is a simulator, it does not consider physical noise in the system. Additionally, we compare our architecture to a baseline using the Fully Connected Neural Network (FCNN) proposed by Li et al. [29] to assess the performance of the hierarchical solution against the state-of-the-art ML models, which we train and test using the same data processing as the NTS method. The baseline only uses the FCNN classifier to select the best class without using the hierarchical architecture introduced in algorithm 1.

Before testing, we apply preprocessing on the collected dataset [34] to transform the data into a time window format. The first step of preprocessing is to generate time windows from the dataset. We consider each prediction sample made of the past t timeslots. We then implement a rolling approach for the time window generation, producing an overlap of $t - 1$ timeslots per window. The data is then scaled into the range $[0, 1]$ to normalize the results for MSE. For the TS model, we then fit a linear regression model of each feature for each time series window. The rate of change of each feature is extracted and appended to the final timeslot in the window, which is then used as our sample. For all experiments in this paper, we set t to 10. Furthermore, we use the hyperparameters on TABLE 1 to build our models and each baseline. Each hyperparameter was found experimentally to provide the highest accuracy. We chose smaller model sizes to reduce the possibility of overfitting. We build our deep learning models using the TensorFlow Keras library. All nondeep learning models are built using the sci-kit learn library. Numpy and Pandas libraries in Python are also used for data preprocessing.

5.3 Evaluation Without Noise

To evaluate the overall performance of our proposed solution, we explore the accuracy of our DNN hierarchical approach against nondeep learning methods and a single-model baseline. We first evaluate the overall performance of the model against the baselines. Additionally, we consider the change in accuracy of all three hierarchical architectures if the number of single-class classifiers used increases. This evaluation ignores noise in the system to measure which models provide the best results in an optimal scenario. However, as previously mentioned, noise is typically present in NPPs, and it is difficult to remove it thoroughly. Therefore, a clean data evaluation can help showcase the performance in optimal scenarios but is insufficient for NPPs and cyber-physical system security.

Table 1: Model Hyper-parameters for each tested DNN and nondeep learning model.

Model	Hyper-parameters
LSTM	n_layers = 2, Neurons = [128, 128], Dropout = 0.2, epoch=50, batch_size=64
AE	n_layers_encoder = 1, Neurons_encoder = [256], n_layers_decoder=1, Neurons_decoder = [256], epoch=50, batch_size=64
RFC	n_estimators=100, max_depth=100
GMM	n_components=3, metric=log_likelihood
FCNN	n_layers=4, Neurons = [90, 50, 20, 18], epoch = 100, activation= [Relu, Relu, Relu, Softmax] batch_size=64, learning_rate = 0.0015,

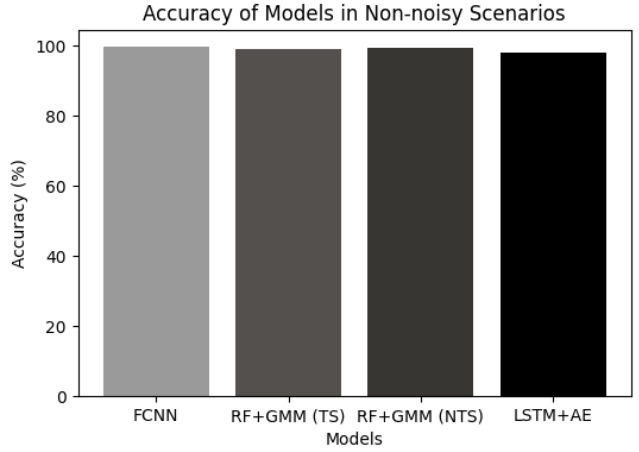

Figure 5: Comparison between TS, NTS, and Deep Learning model architectures in non-noisy scenarios.

We first evaluate the overall performance of the TS, NTS, DNN, and FCNN models under no noise. As shown in Figure. 5, the performance of all three hierarchical frameworks, given that we use a top n class set to 5, is relatively equal. The figure shows that nondeep learning approaches achieve a relative performance of around 99%, while the deep learning hierarchical framework has an accuracy of 97.5%. The FCNN baseline we use obtains an accuracy of approximately 99% under non-noisy conditions. We can show that while the four models have relatively similar performance in terms of accuracy, the deep learning framework shows reduced performance by around 1.5%. The decrease could be a result of an insufficient training dataset. If more samples are added to the training data, the accuracy should increase, but this could lead to overfitting. Overall, the performance results of all three approaches and the baseline are not significantly different in optimal scenarios.

We then examine the difference in accuracy as each of the three framework's top number of classes changes. In particular, we explore how increasing the n from 1, only using the multi-class classifier, to 18, where only the single-class classifiers are being used, changes the prediction results. Figure. 6 shows that for all three architectures, the performance is consistent with the conclusions

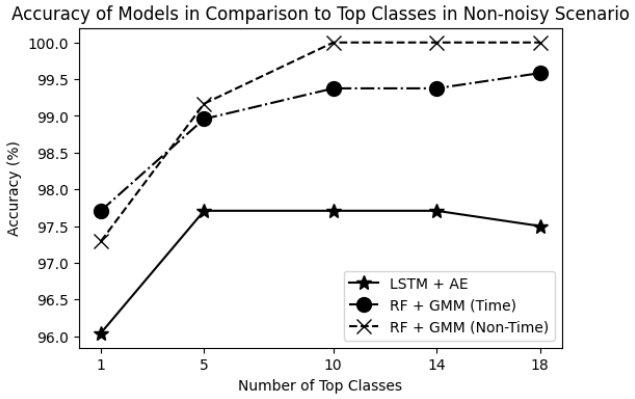

Figure 6: Comparison between TS, NTS, and deep learning architectures in non-noisy scenarios with changing top number of classes.

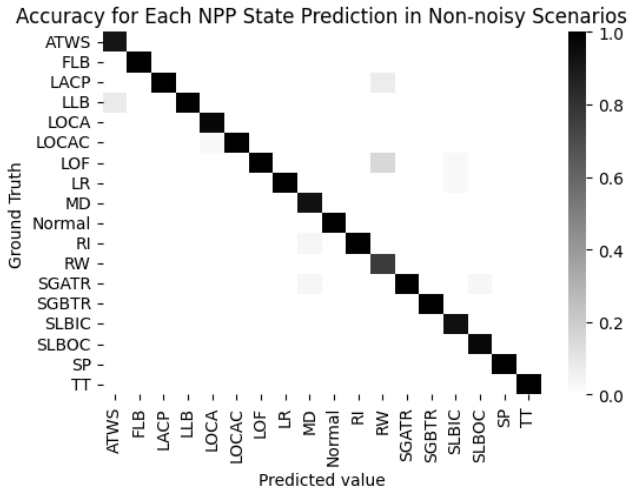

Figure 7: Comparison of classification accuracy of different class labels for deep learning model using 5 top classes.

obtained from Figure. 5, where the deep learning approach is 1 to 2 percent less accurate than the nondeep learning approach at all number of classes. It can also be seen that regardless of the models used, increasing the number of classes also improves the accuracy of the overall prediction. We can notice an increase in the performance of the deep learning model of 1.5% by increasing n from 1 to 5. A similar increase from 97.5% to 99% is observed for the TS-based RF and GMM hierarchical architectures. The evaluation on n shows that adding the single class classifiers improves classification accuracy in non-noisy data.

We then evaluate the performance of the DNN model per class label to explore if the model shows decreased performance under specific faults. We use the LSTM and AE models with 5 top classes selected and examine the accuracy of each of the individual 18 labels. Figure. 7 showcases the per-label performance of our architecture to evaluate what faults provide the lowest identification accuracy. This

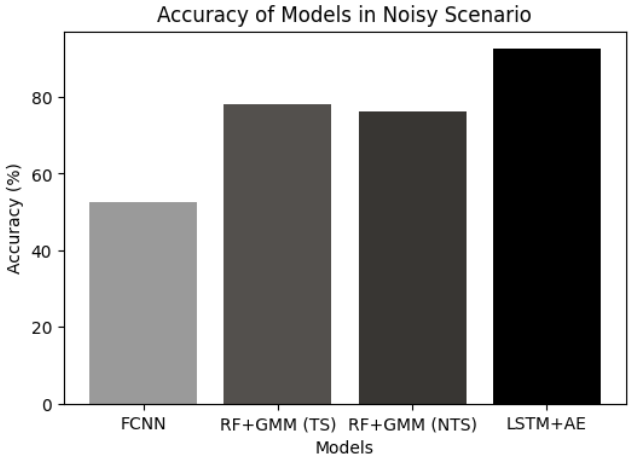

Figure 8: Comparison of TS, NTS, and Deep Learning model architectures using top 5 classes under 3% noise.

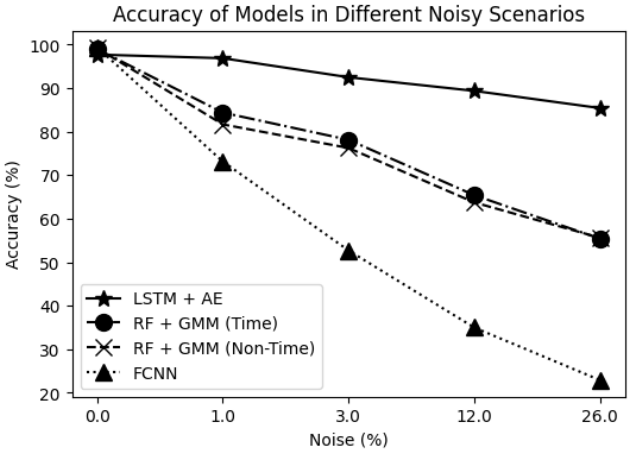

Figure 9: Comparison between TS, NTS, and Deep Learning model architectures using top 5 classes under various noisy scenarios.

evaluation shows that our model accurately identifies most faults correctly. Our model shows high accuracy for most operational conditions, such as the normal label, which has an almost perfect score. On the other hand, the rod withdrawal (RW) fault is predicted more commonly than it is present, showing a trend towards RW as a default class. Two cases that show misclassification as RW are the loss of flow (LOF) and Loss of AC Power (LACP). The proposed hierarchical solution is sufficient in non-noisy optimal scenarios as the DNN correctly classifies most faults.

5.4 Evaluation with Noise

As previously shown, all four methods perform similarly to each other under the optimal scenario. However, as explained in section 3.3, NPPs, as with most cyber-physical systems, are prone to noise in sensor measurements and data transmission. The presence of noise requires highly robust architectures capable of working within a variable error of measurements. The plant's safety systems also need the same robustness as controllers, which must be able to handle noisy data. To showcase the robustness of our proposed solution, we evaluated our DNN hierarchical framework with the TS, NTS, and FCNN architectures as baselines when the sensor data has added noise.

In the case of NPPs, different systems can cause different noise levels, either from the physical component or electromagnetic interference on the NPP sensors. In addition to sensor errors, transmission lines, and communication systems can add more noise to collected data. In this study, we consider the noise present in the NPP to be Gaussian white noise, defined as an additional signal to the clean sensor values. Gaussian white noise provides randomized obfuscation to the original signal and is widely used due to its prevalence in nature, which we add using Equation 2.

$$X_n^i(t) = X^i(t) + n(t) \tag{2}$$

In this equation, $X_n(t)$ represents the clean signal obtained from a sensor, i, in the NPP system. The combined noise, $n(t)$, is a Gaussian distribution with a mean of 0 and a standard deviation adjusted during testing. We then evaluate the impact of the noise by considering the change in the standard deviation from the non-noisy data to the noisy data in terms of percentage change. The bigger the standard deviation of $n(t)$, the more impact it will have on the initial signal. This approach allows us to test our proposed architecture's robustness in multiple noisy scenarios where an NPP could be active. We evaluate our model in 5 noisy scenarios, with 0, 0.05, 0.1, 0.2, and 0.3 standard deviations. These standard deviations correspond to a noise ratio of 0%, 1%, 3%, 12%, and 26% of the total signal, respectively.

For this evaluation, we first explore the performance of the DNN framework compared to the TS and NTS-based hierarchical architectures and the FCNN model; Figure 8 showcases the results when considering a noise of approximately 3%, and the top classes are set to 5 for each architecture. We can see from the previous figure that, unlike in the optimal scenario, adding a small amount of noise to the system causes the accuracy to decrease. Still, the DNN stays significantly more accurate than the nondeep learning approaches. In this case, the performance of the deep learning framework is approximately 10% better than the TS and NTS architectures and 40% more than the FCNN baseline. At the same time, it can be demonstrated that the TS model performs better than the NTS model, showing that adding the temporal feature to the dataset improves the performance by around 2% when the number of top classes is set to 5. Figure 8 shows that all three hierarchical frameworks are more robust than the FCNN baseline. However, accuracy is improved further when hierarchical architecture is used alongside DNN.

Figure. 9 showcases the performance of all three models under different noise amounts. In particular, varying the noise when the number of top classes remains fixed is explored in Figure 9. Overall, we can show that deep learning in a hierarchical framework is

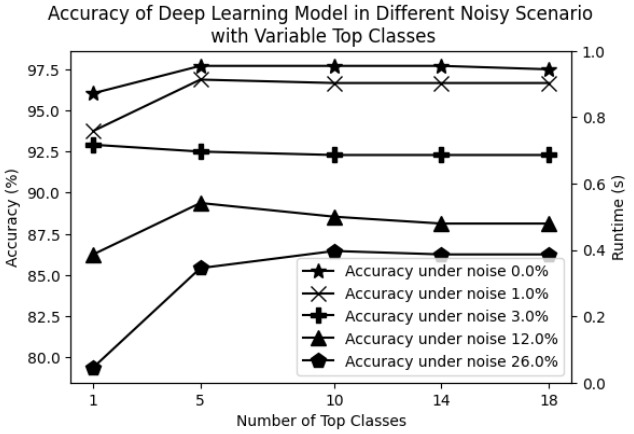

Figure 10: Comparison of Deep Learning hierarchical architecture under various noisy scenarios and with different top classes.

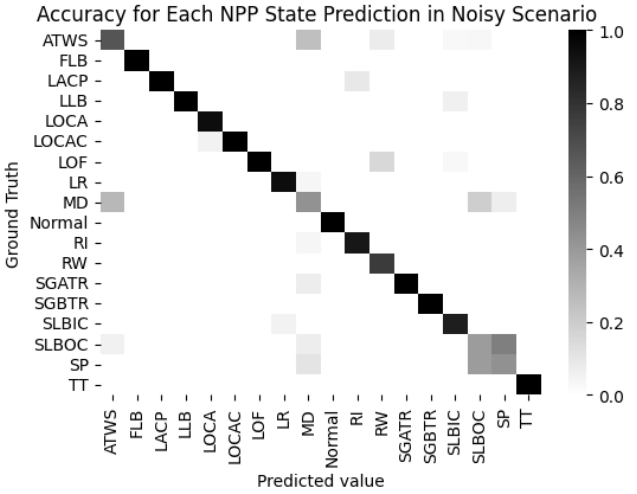

Figure 11: Comparison of classification accuracy of different class labels for deep learning model using 5 top classes in 3% noise.

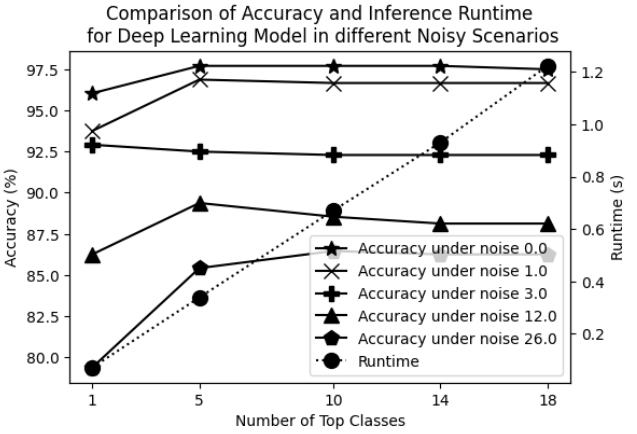

Figure 12: Comparison between accuracy and runtime between different noise scenarios and classes.

significantly more robust than non-deep learning architectures. In Figure. 9, it can be seen that even when the noise level reaches up to 26%, the total accuracy of the DNN does not fall below 80%. On the other hand, the other two approaches' performance drops to below 60% when the noise signal is significant. The FCNN showcases the worst performance with an accuracy of 39% under 26% noise. Additionally, the TS model shows improved performance over the NTS architecture and the FCNN baseline when the noise is above 1% and below 26%, which signifies that in low noise, the temporal features improve the overall accuracy of the framework.

We then explore how changing the number of top classes affects the performance of the proposed deep learning architecture. Figure. 10 showcases the performance of the LSTM and AE hierarchical architecture in the presence of variable noise from 0% to approximately 26% of the total signal. It can be seen that in most noisy scenarios, adding more top classes will increase performance. However, adding more top classes also shows diminishing returns, as when significant classes are added, the accuracy stops improving. When the noise level is 0%, the plateau is reached at a top class of 5, while when the noise is 26%, the prediction stops improving after 10 top classes. In the case of a noise level such as 3%, the system's accuracy does not significantly improve with more classes. Instead, it slightly decreases from 92% to 91%; this can be attributed to the distribution of the Gaussian noise at this level. As a result of the randomness in a Gaussian distribution, it can be possible that the AE shows decreased performance in between very similar classes. In most evaluated scenarios, there is no significant increase in accuracy after 5 top classes, regardless of noise. As a result, we can consider that a top number of classes of 5 provides the most optimal performance for our dataset.

Similarly to before, the accuracy of each label prediction for the LSTM AE architecture is examined under noise to evaluate how it changes from the non-noisy data. Figure. 11 showcases that adding noise lowers the accuracy of multiple faults unevenly. The performance of some labels, such as LOCAC, does not show a significant change. Additionally, labels that the model had issues classifying in non-noisy data, such as RW, show further decreased accuracy under noise. However, the faults of moderator dilution (MD), steam line break outside of containment (SLBOC), and a spark presence (SP) all see increased errors, not present in clean data; in particular, SP and SLBOC are commonly and incorrectly misclassified amongst each other. As a result, noise decreases the accuracy of our model for specific classes more than for others.

5.5 Runtime Evaluation of Hierarchical Framework

One of the primary considerations in the hierarchical approach is using multiple single-class classifiers to provide additional robustness in the prediction. Using multiple single-class classifiers increases the detection accuracy, especially under a noisy scenario.

However, one of our proposed architecture's significant downsides is the hierarchical framework's inference runtime. As shown in Figure. 12, the runtime represented by the dashed line increases linearly from 0 to 1.2 seconds per sample as you increase the number of top classes from 1 to 18. Each single-class classifier takes a set amount of time for inference, which tends to be the same across all classes. As a result, the total inference time should grow linearly as each classifier is added, making the time complexity of the fault detection $O(n)$. For all experiments carried out, the inference time of the multi-class classification model is negligible and constant.

Two solutions are proposed to limit the architecture runtime of the proposed hierarchical solution. The first uses distributed computing to reduce inference by evaluating multiple models in parallel before comparing. A distributed approach requires increased computational capacity compared to running each single-class classifier in series, effectively multiplying the computational need by n. The other solution is to reduce the number of classes that need to be examined by an individual model. The approach taken in this paper is to use a multi-class classifier in a hierarchical framework to reduce the n sample size for efficient model evaluation and to generate a more accurate prediction. As shown in Figure. 12 with the addition of the multi-class classifier, the framework under most noise levels does not significantly improve after 5 top classes. At 5 top classes, the runtime is approximately 0.35 seconds, considerably lower than the 1.2 seconds runtime when using all 18 single class classification models instead of the proposed hierarchical framework. Therefore, by using the proposed structure, we can improve the robustness and accuracy of the model while reducing the total runtime for inference.

6 Conclusion

NPPs have seen continuous growth in operation over recent years due to the necessity of power generation alternatives to coal power plants. Within NPPs, FDD is critical for ensuring safe and consistent operation. As a result, recent research and fields have extensively explored advances toward FDD. This work proposes a novel hierarchical framework for FDD on NPPs by examining a hybrid supervised and unsupervised approach to learning individual faults. This paper additionally proposes using deep learning ML techniques to improve the framework's robustness, aiming for better accuracy in noisy scenarios. This approach uses an LSTM-based classification model to reduce the inference runtime and enhance the generalizability of our framework. At the same time, we propose an LSTM-based AE single-class classifier for improved robustness. This unsupervised approach leverages the error difference between reconstructed samples of different classes to select the class with the lowest error. This work examines our framework over several evaluations, comparing individual models with both time series and non-time series traditional machine learning frameworks such as GMM and RF in a clean and noisy data environment. Additionally, this work compares the proposed framework to existing FCNN models to evaluate the performance of the solution. One of the primary considerations of our work is to build a highly robust framework that can provide high accuracy under the noisy conditions of NPPs. The proposed framework showcases a performance similar to non-deep learning approaches and a single-model baseline in non-noisy data. However, it demonstrates significant improvements as noise is added to the system. While we can show that the proposed hierarchical approach with DNN has increased performance under different conditions, when noise is small, the performance increase is much smaller. A more comprehensive training of the model using a large non-simulated dataset would improve the model in optimal situations. Additionally, NPP data is rare and difficult to access and can be prone to contain more drastic differences than in simulation. In the future, we will explore techniques to minimize the effects of faults in NPPs and evaluate the framework on data collected from real NPP systems.

7 Acknowledgement

This research was supported in part by the National Security Agency (NSA) under award H98230-22-1-0327, and the Department of Energy (DOE) under award DE-NA0004016.

References

[1] M.D. Mathew. Nuclear energy: A pathway towards mitigation of global warming. *Progress in Nuclear Energy*, 143:104080, 2022.
[2] Clarion Energy Content Directors. Nuclear power could generate 25 percent of global electricity by 2050, Aug 2021.
[3] Iea. Nuclear.
[4] Sirazam Sadekin, Sayma Zaman, Mahjabin Mahfuz, and Rashid Sarkar. Nuclear power as foundation of a clean energy future: A review. *Energy Procedia*, 160:513–518, 2019. 2nd International Conference on Energy and Power, ICEP2018, 13–15 December 2018, Sydney, Australia.
[5] Katsumi Hirose. 2011 fukushima dai-ichi nuclear power plant accident: summary of regional radioactive deposition monitoring results. *Journal of Environmental Radioactivity*, 111:13–17, 2012. Environmental Impacts of the Fukushima Accident (Part I).
[6] Eujeong Choi, Jeong-Gon Ha, Deagi Hahm, and Min Kyu Kim. A review of multihazard risk assessment: Progress, potential, and challenges in the application to nuclear power plants. *International Journal of Disaster Risk Reduction*, 53:101933, 2021.
[7] Taotao Zhou, Mohammad Modarres, and Enrique López Droguett. Multi-unit nuclear power plant probabilistic risk assessment: A comprehensive survey. *Reliability Engineering & System Safety*, 213:107782, 2021.
[8] Chao Lu, Jiafei Lyu, Liming Zhang, Aicheng Gong, Yipeng Fan, Jiangpeng Yan, and Xiu Li. Nuclear power plants with artificial intelligence in industry 4.0 era: Top-level design and current applications—a systemic review. *IEEE Access*, 8:194315–194332, 2020.
[9] Yuchen Jiang, Shen Yin, and Okyay Kaynak. Performance supervised plant-wide process monitoring in industry 4.0: A roadmap. *IEEE Open Journal of the Industrial Electronics Society*, 2:21–35, 2021.
[10] Morteza Ghobakhloo. Industry 4.0, digitization, and opportunities for sustainability. *Journal of Cleaner Production*, 252:119869, 2020.
[11] Lamiaa M. Elshenawy, Mohamed A. Halawa, Tarek A. Mahmoud, Hamdi. A. Awad, and Mohamed I. Abdo. Unsupervised machine learning techniques for fault detection and diagnosis in nuclear power plants. *Progress in Nuclear Energy*, 142:103990, 2021.
[12] A. Deleplace, V. Atamuradov, A. Allali, J. Pellé, R. Plana, and G. Alleaume. Ensemble learning-based fault detection in nuclear power plant screen cleaners. *IFAC-PapersOnLine*, 53(2):10354–10359, 2020. 21st IFAC World Congress.
[13] Hang Wang, Min jun Peng, Yue Yu, Hanan Saeed, Cheng ming Hao, and Yong kuo Liu. Fault identification and diagnosis based on kpca and similarity clustering for nuclear power plants. *Annals of Nuclear Energy*, 150:107786, 2021.
[14] Ting-Han Lin and Shun-Chi Wu. Sensor fault detection, isolation and reconstruction in nuclear power plants. *Annals of Nuclear Energy*, 126:398–409, 2019.
[15] Peng Yue, Faming Fang, Peng Xu, Hongyun Xie, Qizhi Duan, Jiaping Lin, and Liyang Xie. Noise resistant steam generator water level reconstruction for nuclear power plant based on deep residual shrinkage network. *Annals of Nuclear Energy*, 193:110038, 2023.
[16] Guang Hu, Taotao Zhou, and Qianfeng Liu. Data-driven machine learning for fault detection and diagnosis in nuclear power plants: A review. *Frontiers in Energy Research*, 9, 2021.
[17] Xianping Zhong and Heng Ban. Crack fault diagnosis of rotating machine in nuclear power plant based on ensemble learning. *Annals of Nuclear Energy*, 168:108909, 2022.
[18] Mohammad Mahdi Bejani and Mehdi Ghatee. A systematic review on overfitting control in shallow and deep neural networks. *Artificial Intelligence Review*,

54(8):6391–6438, 2021.

[19] Mohammadreza Ghorvei, Mohammadreza Kavianpour, Mohammad TH Beheshti, and Amin Ramezani. An unsupervised bearing fault diagnosis based on deep subdomain adaptation under noise and variable load condition. *Measurement Science and Technology*, 33(2):025901, dec 2021.

[20] Himanshukumar R Patel and Vipul A Shah. Fault detection and diagnosis methods in power generation plants-the indian power generation sector perspective: an introductory review. *PDPU Journal of Energy and Management*, 2(2):31–49, 2018.

[21] Simon Sjögren. Anomaly detection with machine learning methods at forsmark. (23011), 2023.

[22] Eugenio Brusa, Luca Cibrario, Cristiana Delprete, and Luigi Gianpio Di Maggio. Explainable ai for machine fault diagnosis: Understanding features’ contribution in machine learning models for industrial condition monitoring. *Applied Sciences*, 13(4), 2023.

[23] Jianping Ma and Jin Jiang. Applications of fault detection and diagnosis methods in nuclear power plants: A review. *Progress in Nuclear Energy*, 53(3):255–266, 2011.

[24] K C Gross, R M Singer, S W Wegerich, J P Herzog, R VanAlstine, and F Bockhorst. Application of a model-based fault detection system to nuclear plant signals. Technical report, Argonne National Lab.(ANL), Argonne, IL (United States), 5 1997.

[25] Azzedine Boukerche, Lining Zheng, and Omar Alfandi. Outlier detection: Methods, models, and classification. *ACM Comput. Surv.*, 53(3), jun 2020.

[26] Gonçalo Jesus, António Casimiro, and Anabela Oliveira. Using machine learning for dependable outlier detection in environmental monitoring systems. *ACM Trans. Cyber-Phys. Syst.*, 5(3), jul 2021.

[27] Zhuo Lv, Li Di, Cen Chen, Bo Zhang, and Nuannuan Li. A fast density peak clustering method for power data security detection based on local outlier factors. *Processes*, 11(7), 2023.

[28] Keke Huang, Haofei Wen, Chunhua Yang, Weihua Gui, and Shiyan Hu. Outlier detection for process monitoring in industrial cyber-physical systems. *IEEE Transactions on Automation Science and Engineering*, 19(3):2487–2498, 2022.

[29] Jiangkuan Li and Meng Lin. Research on robustness of five typical data-driven fault diagnosis models for nuclear power plants. *Annals of Nuclear Energy*, 165:108639, 2022.

[30] Gensheng Qian and Jingquan Liu. Fault diagnosis based on gated recurrent unit network with attention mechanism and transfer learning under few samples in nuclear power plants. *Progress in Nuclear Energy*, 155:104502, 2023.

[31] Bing Liu, Jichong Lei, Jinsen Xie, and Jianliang Zhou. Development and validation of a nuclear power plant fault diagnosis system based on deep learning. *Energies*, 15(22), 2022.

[32] Swetha Rajkumar and Jayaprasanth Devakumar. Lstm based data driven fault detection and isolation in small modular reactors. *The Scientific Temper*, 14(01):206–210, 2023.

[33] Ankit Ghosh, Purbita Kole, and Alok Kole. Fault detection an d diagnosis of various mechanical components in a nuclea r power plant combining noise analysis and machin e learning techniques. *International Research Journal of Engineering and Technology*, 2021.

[34] 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.

[35] Ahmad Shaheryar, Xu-Cheng Yin, Hong-Wei Hao, Zahid Mahmood, and Adnan OM Abuassba. Selection of optimal denoising-based regularization hyperparameters for performance improvement in a sensor validation model. *Artificial Intelligence Review*, 50:341–382, 2018.

[36] Xianping Zhong, Fei Wang, and Heng Ban. Development of a plug-and-play anti-noise module for fault diagnosis of rotating machines in nuclear power plants. *Progress in Nuclear Energy*, 151:104344, 2022.

[37] V. Kortov and Yu. Ustyantsev. Chernobyl accident: Causes, consequences and problems of radiation measurements. *Radiation Measurements*, 55:12–16, 2013. 7th International Workshop on Ionizing Radiation Monitoring.

[38] R.D. Blevins. Flow-induced vibration in nuclear reactors: A review. *Progress in Nuclear Energy*, 4(1):25–49, 1979.

[39] Israel Abraham Alarcón-Sánchez, Roberto Linares-y Miranda, Luis Hector Hernández-Gómez, Yunuén López-Grijalba, Alejandra Armenta-Molina, Laura Guadalupe Carbajal-Figueroa, and Luis Alberto Arenas-Magos. Review of electromagnetic compatibility on digital systems of nuclear power plants. *Engineering Design Applications III: Structures, Materials and Processes*, pages 103–113, 2020.

[40] Raveendra K. Rao Abdullah Kadri and Jin Jiang. Low-power chirp spread spectrum signals for wireless communication within nuclear power plants. *Nuclear Technology*, 166(2):156–169, 2009.

[41] Gerrit Bode, Simon Thul, Marc Baranski, and Dirk Müller. Real-world application of machine-learning-based fault detection trained with experimental data. *Energy*, 198:117323, 2020.