Received January 14, 2022, accepted January 29, 2022, date of publication February 7, 2022, date of current version February 16, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3149772

Fault Detection and Isolation of a Pressurized Water Reactor Based on Neural Network and K-Nearest Neighbor

AMINE NAIMI[1], JIAMEI DENG[1], (Senior Member, IEEE), S. R. SHIMJITH[2], AND A. JOHN ARUL[3]

[1]School of Built Environment, Engineering, and Computing, Leeds Beckett University, Leeds LS1 3HE, U.K.
[2]Bhabha Atomic Research Centre, Homi Bhabha National Institute, Mumbai 400085, India
[3]Indira Gandhi Centre for Atomic Research, Kalpakkam 603102, India

Corresponding authors: Amine Naimi (amine.naimi01@gmail.com) and Jiamei Deng (j.deng@leedsbeckett.ac.uk)

This work was supported by the Engineering and Physical Sciences Research Council under Grant EP/R021961/1.

ABSTRACT Nuclear power plants (NPPs) are complex dynamic systems with multiple sensors and actuators. The presence of faults in the actuators and sensors can deteriorate the system's performance and cause serious safety issues. This calls for the development of fault detection and diagnosis systems for detection and isolation of such faults. In this study, fault detection and diagnosis (FDD) based on neural networks (NN) and K-nearest neighbour (KNN) algorithm is applied to a pressurized water reactor (PWR). Fault detection is first determined based on the NN. Second, the KNN algorithm is used to classify the faults. The proposed approach is capable of classifying a variety of actuator faults, sensor faults, and multiple simultaneous actuator and sensor faults. A set of simulation results is provided to demonstrate the accuracy of the FDD method. The classifier performance is further compared with other machine learning techniques.

INDEX TERMS Fault classification, fault detection, K-nearest neighbor (KNN), neural networks (NNs), nuclear power plants (NPPs), pressurized water reactor (PWR).

I. INTRODUCTION

Nuclear power plants (NPPs) play a key role in reducing greenhouse gas emissions. However, the safety of their operation remains a significant concern. Nuclear plants are complex dynamic systems with many actuators and sensors. Because of their vital role in NPPs, any fault in actuators and sensors can degrade the system's performance and cause serious safety issues. Therefore, particular attention should be paid to the detection and diagnosis of such problems to prevent their degradation, which can lead to catastrophic damage. To achieve this, model-based fault detection and diagnosis (FDD) is applied for NPPs [1]. This approach uses a mathematical model to describe the normal behavior of the plant. Faults in the process are detected and isolated by comparing the system's behavior with the fault-free model. However, difficulty in obtaining exact and accurate models of NPPs puts forth hurdles in practical applications of model based FDD techniques. As opposed to model-based approaches, data-driven methods do not rely on explicit knowledge about the process. Instead, they use the data acquired from the process to construct an empirical model. Recent studies have developed data-driven approaches for monitoring NPPs. For instance, the support vector machine (SVM) algorithm has been used for FDD of NPPs [2], [3]. The simple and flexible structure of principal component analysis (PCA) has garnered widespread interest in the past decade. For instance, Farhan *et al.* applied data-driven techniques based on PCA along with Fisher discriminant analysis for a control rod withdrawal fault and an external reactivity insertion fault [4]. In another study, an improved PCA was employed for FDD of sensors in an NPP [5]. Another approach to the diagnosis of faults in an NPP was proposed in [6] that used data acquired from a full-scope simulator for a kernel PCA. More recently, the PCA approach was used with multivariate contribution plots (MCP) [7].

Another alternative approach is a neural network (NN), which is a network of neurons that learns complex functions through a series of non-linear transformations.

The associate editor coordinating the review of this manuscript and approving it for publication was Dazhong Ma.

They have been successfully employed for complex classification tasks such as image recognition [8], speech recognition [9], and system identification [10]–[12]. NNs have also been used to address the fault diagnosis problem in NPPs [13]. A convolutional NN model was developed for abnormality diagnosis in an NPP in [13]. Three types of sensor fault signals were simulated in [14] using a modified ensemble empirical mode decomposition and probabilistic NN. In [14], a distributed fault diagnosis approach was proposed that was based on a fuzzy NN and data fusion, and the efficiency of the diagnostic approach was improved in [15]. In a more recent work [16], fault diagnosis performance was tested via the comparison of a radial basis network and the Elman NN.

The K-nearest neighbour (KNN) algorithm is a non-parametric classification method that can be used for classification, regression, and pattern recognition problems [17]. The KNN algorithm is simple and easy to implement. The purpose of KNN classification is to categorize data points based on the classification of their neighbours, where K represents the number of nearest neighbors considered for the determination of the object class. Although KNN has been successful with fault detection in industrial processes [18]–[20], there are only a few studies on this approach with fault diagnosis and identification. In this study, NN and KNN are applied for the first time for the detection and classification of single and multiple simultaneous sensor and actuator faults in a pressurized water reactor (PWR). With this framework, faults are first detected using an NN approach, and then the KNN method is used to classify them. Compared to existing techniques, the KNN approach is more accurate and uses less computational time.

Studies in the existing body of literature have primarily focused on faults that affect either the sensor or the actuator but not both. Moreover, most studies assume that only a single fault is injected into the system at a time. A realistic study should establish an FDD for both actuator and sensor failures and should consider the injection of multiple simultaneous faults. The proposed FDD in [5] is capable of detecting and isolating multiple simultaneous fault but it is limited only to sensor faults. In [21], a simple case of sensor and actuator faults was studied by the development of a fault detection technique based on an NN to determine the presence of saturation faults in the actuator and bias faults in the power and temperature sensors of a PWR. The present study can be considered as an extension of [21] by examining drift faults in sensors and offset faults in the actuator. Furthermore, the injection of multiple simultaneously sensor and actuator faults is studied in addition to the single faults. The KNN algorithm is used to perform the classification of faults. The main contributions of this paper are as follows:

- The NN technique and KNN algorithm are employed for fault detection and classification in a PWR.
- Actuator offset, actuator saturation, sensor bias, and sensor drift faults are studied.
- Both single faults and multiple simultaneous actuator and sensor faults are considered.
- The proposed classifier is compared to other machine learning techniques.

The rest of this paper is organized as follows: Section II provides a description of the PWR process. Section III presents the data collection for FDD. The two classification methods are described in Section IV, and Section V describes the efficacy of the proposed technique. Finally, conclusions are drawn in Section VI.

II. PRESSURIZED WATER REACTOR

The PWR model used in this study can be found in the literatue [22]. The PWR mathematical model assumes a point kinetics equation coupled with six delayed neutron groups and a lumped thermal hydraulic model. The dynamic model is described in (1) through (5) [22].

$$\frac{dP}{dt} = \frac{\rho_t - \sum_{i=1}^{6} \beta_i}{\Lambda} P + \sum_{i=1}^{6} \lambda_i C_i \tag{1}$$

$$\frac{dC_i}{dt} = \frac{\beta_i}{\Lambda} P - \lambda_i C_i, \quad i = 1, 2, \ldots 6 \tag{2}$$

$$\frac{dT_f}{dt} = H_f P - \frac{1}{\tau_f}\left(T_f - T_{c1}\right) \tag{3}$$

$$\frac{dT_{c1}}{dt} = H_c P + \frac{1}{\tau_c}\left(T_f - T_{c1}\right) - \frac{2}{\tau_r}\left(T_{c1} - T_{cin}\right) \tag{4}$$

$$\frac{dT_{c2}}{dt} = H_c P + \frac{1}{\tau_c}\left(T_f - T_{c1}\right) - \frac{2}{\tau_r}\left(T_{c2} - T_{c1}\right) \tag{5}$$

where P is the neutronic power, Λ is the prompt neutron generation time, and C_i, λ_i, and β_i are the delayed neutron precursors' concentration, decay constant, and fraction of delayed neutrons, respectively; H_f and H_c denote the proportionality constants; τ_f, τ_c, and τ_r denote the time constants; T_f, T_{c1}, and T_{c2} are the temperatures of fuel, coolant nodes 1 and 2, respectively.

III. DATA COLLECTION FOR FDD

This section discusses the application of the NN to detect faults in the actuator and sensors of the previously described PWR plant. The PWR plant is assumed to be controlled by a robust PID controller that is carefully tuned and operating in the range of 80%–100% full power.

A. TYPES OF FAULTS

Six single faults and two simultaneous faults are considered in this study, as shown in Table 1. The types of faults considered in this study included bias, drift, actuator offset, and actuator saturation faults, which are described as follows:

Bias fault. This is one of the most common faults in sensors, corresponding to a constant offset added to the sensor output, which may be caused by inappropriate calibration or physical changes in the sensor [23]. Bias failures are a common fault in NPPs, and their maintenance can be costly [1]. In this study, the bias fault is injected into the power and temperature sensors at a certain time.

Drift fault. This consists of a time-varying offset [24]. The drift fault is difficult to detect because the drifting amplitude is initially low [25], therefore it is important to have a performant sensor drift FDD. Drift faults are common in NPPs and can cause power reduction [1]. As with the bias fault, the drift fault is injected into the power and temperature sensors at a certain time.

Actuator saturation fault. This is when the actuator (control rod system) exceeds a set saturation value. This phenomenon inevitably must be considered because of physical limitations that, in practice, can led to important deterioration of the system [26].

Actuator offset fault. This corresponds to an offset added to the control rod system at a certain time. This failure can occur because of design/ manufacturing defects in the actuator [27].

B. RESIDUAL GENERATION WITH NEURAL NETWORK

The NN is used to detect faults in the PWR, with two NNs trained to represent the power and temperature of the original (non-faulty) power plants. Thus, when the nuclear plant presents faults, a residual is generated between the faulty and non-faulty NPPs. The data generated for training these two NNs are described in detail in [21]. Both networks are trained independently to adopt the behaviour of the closed-loop process during normal operation. One NN is dedicated to learning the dynamics of the power, whereas the other is dedicated to the reactor temperature. Both NNs used in this study are two-layer feedforward networks as shown in Fig. 1. They have a tanh activation function in the hidden layer (layer 1) and a linear function in the output layer (layer 2). The Levenberg–Marquardt optimization is selected to train the networks. This algorithm uses an approximation of Newton's method rather than the gradient descent method. The best NNs obtained for the power has the following parameters: five neurons in the hidden layer, 999 training epochs, and a mean square error (MSE) of $5, 9.10^{-4}$. The best NNs obtained for the temperature has the following parameters: six neurons in the hidden layer, 1000 training epochs, and an MSE of 0.10225.

The proposed fault detection method is based on the scheme shown in Fig.2. The input P_{dem} corresponds to power demand and the outputs P and T correspond to the power and temperature measures, respectively. The residuals result from the measurement error between the sensor measures and NN estimations, where e_1 corresponds to the measurement error between the measured power and the power estimated by neural network 1 and e_2 corresponds to the measurement error between the measured temperature and the temperature estimated by neural network 2. Threshold alarms are defined to detect the faults, which is when a residual value is greater than the alarm threshold.

TABLE 1. Descriptions of faults.

Fault ID	Process variable	Type
Fault 1	Reactivity	Actuator offset fault (AOF) .
Fault 2	Reactivity	Actuator saturation fault (ASF).
Fault 3	Power	Power sensor bias fault (PSBF).
Fault 4	Power	Power sensor drift fault (PSDF).
Fault 5	Temperature	Temperature sensor bias fault (TSBF).
Fault 6	Temperature	Temperature sensor drift fault (TSDF).
Fault 7	Reactivity/ Power	Actuator saturation fault (ASF) + Power sensor bias fault (PSBF) .
Fault 8	Reactivity/ Power	Actuator saturation fault (ASF) + Power sensor drift fault (PSDF).

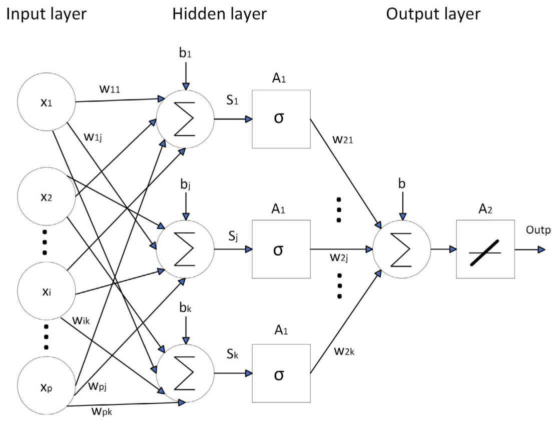

FIGURE 1. Feed-forward neural network for fault detection: $x_i(i \ldots P)$—inputs, $w_{ij}(i \ldots p, j \ldots k)$—weights from input to hidden layer, $b_j(j \ldots k)$—biases of the neurons in the hidden layer, A_1—activation function in the hidden layer, $w_2j(j \ldots K)$—weighting functions from hidden to output layer, b-bias of the output neuron, A_2—linear activation function of the output layer, Y—output.

C. DATA GENERATION FOR FAULT CLASSIFICATION

The fault classification problem is solved using the data of power, temperature, and the two residuals (e_1 and e_2). To create a meaningful representation of the data, four data points are collected in each of the eight fault cases for 100 simulation runs and are sampled at a frequency of 1000 Hz. The four data measurements in the presence of bias-type faults in the power sensor are shown in Fig. 3–6. The variation of the reactor power is shown in Fig. 3. The variation of the temperature is shown in Fig. 4. The variations of the two residuals e_1 and e_2 are shown in Fig. 5 and Fig. 6, respectively. Only five simulation runs are presented here. The sensor bias fault was injected randomly and for each simulation run. For instance, a bias fault occurs at 161 seconds in the first simulation. The fault effect is more obvious for the temperature, whereas it is

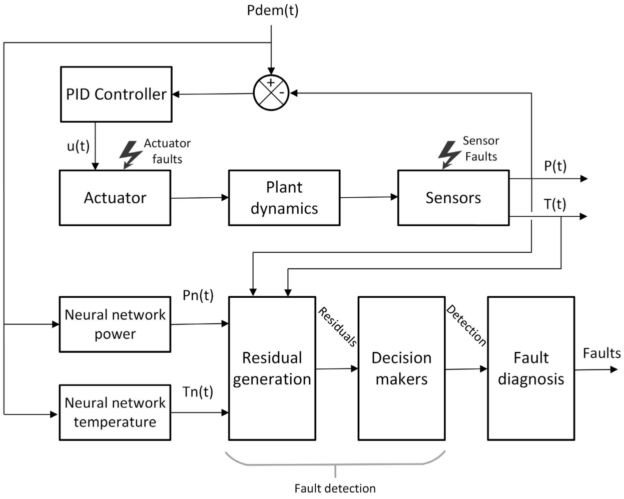

FIGURE 2. Block diagram of the fault detection approach.

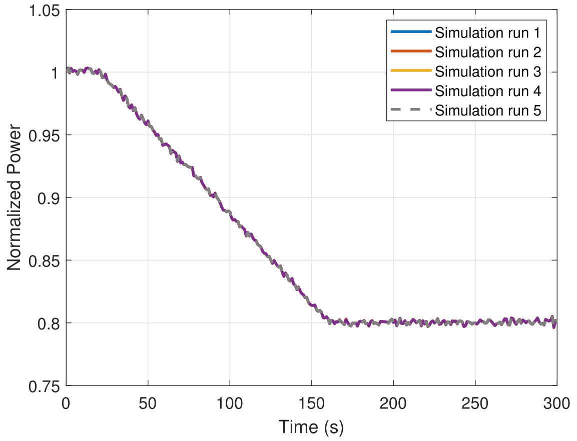

FIGURE 3. Reactor power in the presence of bias-type faults in the power sensor.

less noticeable for the power, which is controlled by the PID controller.

IV. FAULT CLASSIFICATION METHODS

Three classification algorithms are analysed for this study, which are the KNN, SVM and NN classifiers. The SVM and NN classifiers are employed to benchmark the performance of the KNN classifier.

A. NN-BASED CLASSIFICATION

The standard NN that is used for classification purposes is characterized by a two-layer feedforward network. They have a sigmoid function in the hidden layer and a softmax transfer function in the output layer. The NN structure is shown in Fig. 7. The hidden layers transform the input data into higher representations by using the nonlinear transformations:

$$h = \sigma(\omega x + b) \tag{6}$$

where x and h are the input vectors and hidden representations, respectively; b are the biases of the neurons in the hidden layer; w are the weights from the input to the hidden layer; and σ is a sigmoid activation function. The transformation

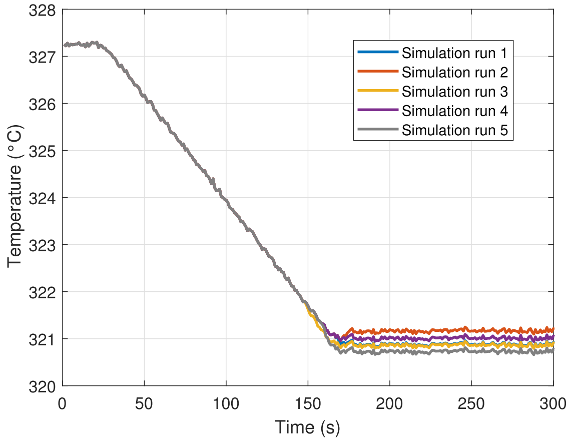

FIGURE 4. Reactor temperature in the presence of bias-type faults in the power sensor.

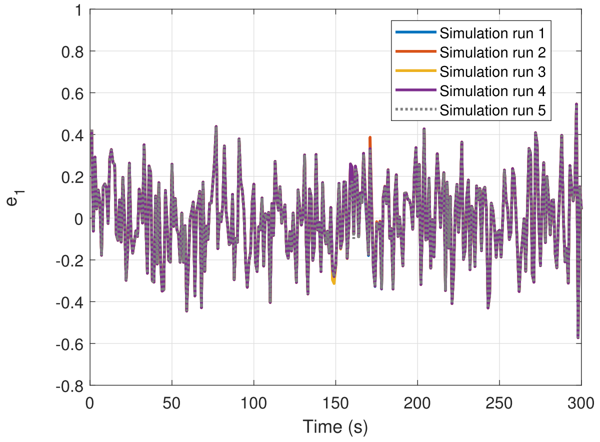

FIGURE 5. e_1 in the presence of bias-type faults in the power sensor.

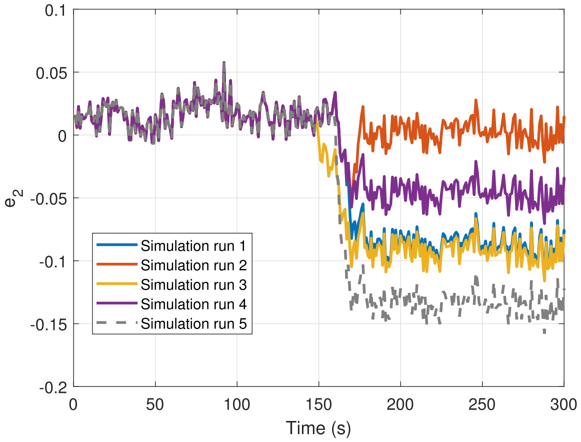

FIGURE 6. e_2 in the presence of bias-type faults in the power sensor.

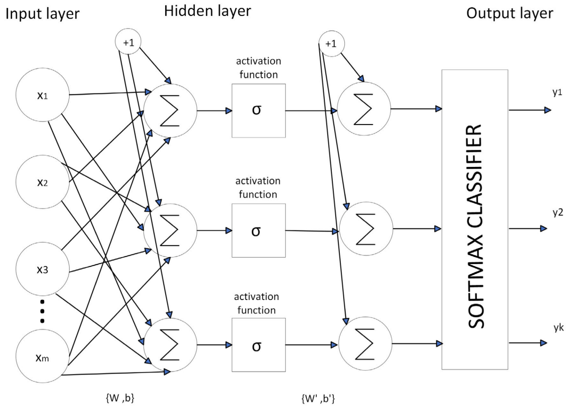

FIGURE 7. Feed-forward neural network for fault classification, where x_i ($i \ldots m$)—input, Y—output.

of (7) without the activation function is applied to the output of the last hidden layer, as follows:

$$h' = \omega' h + b' \tag{7}$$

where h' represents the last hidden layer, b' represents the biases of the output neuron, and w' is the weighting function from the hidden to the output layer. The softmax function is then used to calculate each output neuron. In this study, the neural network is trained using scaled conjugate gradient back propagation. The goal of training the network is to maximize its accuracy, which can be defined as follows [28]:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{8}$$

B. KNN-BASED CLASSIFICATION

The KNN method is a simple non-parametric classification method. Despite the simplicity of the algorithm, it is known to perform well. Furthermore, it is an important benchmark method [29], [30]. KNN performs the classification task based on the similarity index considering the distance measure; k corresponds to the integer value that is mostly lying within the range [3-10]. It is advisable to choose an odd value of k to obtain a clear prediction. Among all the input classes that are stored in the algorithm, the class decision selection is predicted based on the majority votes given by the neighboring points correspondingly nearer to the class. Distance is a key word in this algorithm. Distance measurements are used to measure the distance between individuals in a space. The Euclidean distance is the most common distance measurement method. For example, if x and y are two points in the Euclidean space and it is assumed that x $= (x_1, x_2, x_3, x_4, \ldots, x_n)$ and y $= (y_1, y_2, y_3, y_4, \ldots, y_n)$, then, the Euclidean distance of line segment xs can be expressed as follows [31]:

$$\begin{aligned} Dist(x, y) &= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2} \\ &= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \end{aligned} \tag{9}$$

C. SVM CLASSIFIER

The SVM is a machine learning algorithm based on structural risk minimization and statistical learning that is used for data classification and regression [32]. The SVM method has been successfully employed for various applications separating data into two or more classes. The aim of using the SVM is to find an optimal hyperplane that separates data points of one class from those of another class. An optimal hyperplane is defined as one that maximizes the margin of separation between two classes. Thus, the SVM is basically employed to address classes that are linearly separable. For nonlinear cases, the classifier may not perform well. Hence, kernel functions are used for nonlinear transformation. A kernel function turns a nonlinearly separable object into a linearly separable one by mapping it into a higher dimensional feature space. Common kernel functions include the linear kernel, polynomial kernel, and Gaussian radial basis function kernel [33].

D. TRAINING PROCESS

1) DATA PREPROCESSING

The data collected in Section III are used for training the three classifiers. Before training the classifiers, it is necessary to transform the raw measured data into a form that can be input to the learning classifiers. To succeed with this, the data are transformed through preprocessing steps. First, the data are normalized between 0 and 1. The data are then re-sampled to 100 Hz because the datasets are too large for a personal computer to accommodate. Finally, the data are reshaped as a vector matrix.

2) NEURAL NETWORK

For the purpose of training the NN classifier, the data collected are sorted into a training set (50%), validation set (25%) and testing set (25%). Validation, and testing sets are used to avoid overfitting and to check the generalization properties, respectively. The NN is trained using a scaled conjugate gradient for 1000 training epochs. The number of neurons in the hidden layer is increased to five because that is where the largest improvement is achieved. The number of neurons in the output layer is fixed to eight as it corresponds to the number of elements in the target vector.

3) KNN

For the KNN classifier, six different methods available in MATLAB are trained, namely the fine KNN, medium KNN, coarse KNN, cosine KNN, and cubic KNN. Table 2 presents the definition of each classifier. In this paper, a fivefold cross validation is performed to avoid overfitting. In this approach, the data are divided into five folds, of which four folds are used for training and one is used for testing. This operation is repeated five times in such a manner that each fold is used for testing exactly once. The average test error is obtained by averaging all five folds. Among the six KNN classifiers, the one presenting the best performance is found to be the weighted KNN. The classification accuracy of the six classifiers are detailed in the appendix (Table 8). For the rest of the paper, the KNN model considered is the weighted one.

4) SVM

For the SVM classifier, six different methods available in MATLAB are trained: linear SVM, quadratic SVM, cubic SVM, fine Gaussian SVM, medium Gaussian SVM, and coarse Gaussian SVM. The definition of each classifier is presented in Table 2. As KNN, the performance of SVM is also evaluated using the fivefold cross-validation. Among the six SVM classifiers, the fine Gaussian SVM is found to be the most performant. The classification accuracy of the six classifiers are provided in the appendix (Table 8). For the remainder of the paper, the SVM model considered is the fine Gaussian SVM.

V. FAULT CLASSIFICATION RESULTS

The simulation is performed to test the performance of the three classifiers: the KNN, NN, and SVM classifiers. The three classifiers are trained to classify the eight fault cases (Table 1). Fig. 8 shows the training performance of the NN. It can be seen that the training performance of the NN is good.

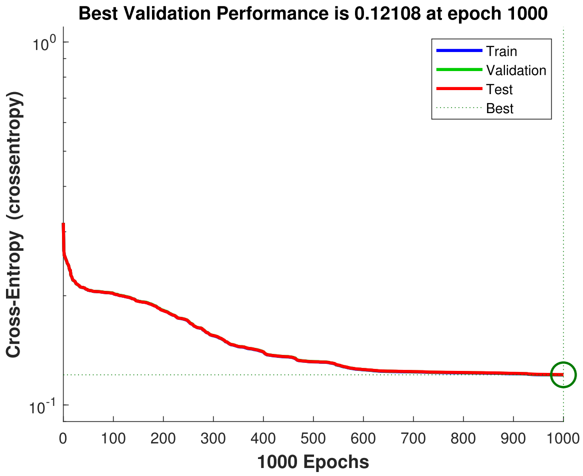

FIGURE 8. Mean square error performance of the network.

The performances of the three classifiers are evaluated with confusion matrix tables. The basic statistical results of the confusion matrix can be extended to the following three indicators: accuracy, precision, and recall, and the indicators are calculated as follows [31]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

TABLE 2. Classifiers from MATLAB machine learning toolbox [34].

Classification algorithms	Classifier types	Description (from Matlab)
SVM	Linear SVM	Proceeds to a linear separation between classes, using the linear kernel.
	Quadratic SVM	uses the quadratic kernel.
	Cubic SVM	Uses the cubic kernel.
	Fine Gaussian SVM	Makes finely-detailed distinctions between classes using the Gaussian kernel with the kernel scale set to $\sqrt{P}/4$ where P is the number of predictors.
	Medium Gaussian SVM	Makes fewer distinctions than a Fine Gaussian SVM, using the Gaussian kernel with the kernel scale set to $\sqrt{P}$ where P is the number of predictors.
	Coarse Gaussian SVM	Makes coarse distinctions between the classes using the Gaussian kernel with the kernel scale set to $\sqrt{P} \times 4$ where P is the number of predictors.
KNN	Fine KNN	makes finely detailed distinctions between classes, with K=1.
	Medium KNN	Makes fewer distinctions than a Fine KNN, with K=10.
	Coarse KNN	Makes coarse distinction between classes, with K=100.
	Cosine KNN	Uses a cosine distance metric, with K=10.
	Cubic KNN	Uses a cubic distance metric, with K=10.
	Weighted KNN	Uses a weighted distance metric, with K=10.

where TP and TN are true positive and true negative, respectively; FP and FN are, respectively, false positive and negative.

Table 3-5 show the confusion matrices obtained for the different classifiers, with the diagonal elements representing the test samples that are correctly classified and the off-diagonal elements representing those that are wrongly classified. The rows of the tables correspond to all faults in the true class. The columns denote the predicted class. Table 3 shows the confusion matrix of the NN classifier. Only faults that belong to F2 or F5 are correctly classified. Conversely, the other six faults have poor classification accuracy, notably those

TABLE 3. Confusion matrix of the dataset using NN classifier (test set).

	F1	F2	F3	F4	F5	F6	F7	F8
Fault 1 (F1)	44.9%	0%	0%	42.6%	0.3%	0%	5.7%	6.5%
Fault 2 (F2)	1.2%	71.6%	4.7%	1.2%	3.8%	4.3%	11.5%	1.7%
Fault 3 (F3)	1.8%	10.1%	46.7%	1.7%	15.1%	20.2%	3.7%	0.7%
Fault 4 (F4)	35.4%	0%	0%	36.5%	0%	0%	15.2%	12.8%
Fault 5 (F5)	0%	0%	1.7%	0%	88.9%	0%	4.8%	1.5%
Fault 6 (F6)	2.2%	3%	29.4%	2.2%	9.3%	52%	1.3%	0.2%
Fault 7 (F7)	6.7%	1%	0.2%	7.9%	0%	0%	61.3%	22.4%
Fault 8 (F8)	2.9%	0%	0%	2.7%	0%	0%	23.6%	70.7%

TABLE 4. Confusion matrix of the dataset using SVM classifer (validation set).

	F1	F2	F3	F4	F5	F6	F7	F8
Fault 1 (F1)	60.2%	3.4%	5.1%	29.5%	0%	1.8%	0%	0%
Fault 2 (F2)	7.7%	81.5%	1.9%	7.8%	0%	1%	0.1%	0%
Fault 3 (F3)	24.1%	3.3%	48.4%	22.7%	0.1%	1.2%	0.1%	0%
Fault 4 (F4)	54.5%	3.4%	8.9%	31.6%	0%	1.7%	0%	0.0%
Fault 5 (F5)	12.9%	2.9%	6.9%	12.6%	64%	0.2%	0.5%	0.1%
Fault 6 (F6)	12.3%	1.9%	3.6%	7.4%	0%	73.8%	1%	0%
Fault 7 (F7)	0.3%	5.3%	1.8%	0.6%	1.7%	5.5%	82.1%	2.7%
Fault 8 (F8)	0%	0.7%	0.7%	0%	0.5%	0.6%	16.4%	81.1%

TABLE 5. Confusion matrix of the dataset using KNN classifier (validation set).

	F1	F2	F3	F4	F5	F6	F7	F8
Fault 1 (F1)	91.8%	0.1%	1.3%	6.4%	0.3%	0.1%	0%	0%
Fault 2 (F2)	16.0%	82.2%	0.5%	0.1%	0%	0%	1%	0.1%
Fault 3 (F3)	1.5%	0.4%	80.5%	15.4%	1.5%	0.3%	0.4%	0.1%
Fault 4 (F4)	5.5%	0.1%	24.9%	69%	0.4%	0.2%	0%	0.0%
Fault 5 (F5)	7.7%	0.3%	3.2%	0.3%	87.5%	0%	0.9%	0.1%
Fault 6 (F6)	0%	0.1%	0.1%	0.3%	0%	98.4%	0.6%	0.4%
Fault 7 (F7)	0%	4.6%	1.1%	0.1%	1.2%	1.3%	86%	5.7%
Fault 8 (F8)	0%	0.6%	0.2%	0%	0.2%	3.1%	9%	86.9%

that belong to F1, F3 or F4. Of the faults belonging to F1, 44.9% are correctly classified, whereas 42.6% are incorrectly classified as F4. Thus, F1 and F4 have an equal chance of being the classification outcome when F1 occurs. There is a similar problem for faults belonging to F4; 36.5% are correctly classified as F4, and 35.4% are misclassified as F1. Faults belonging to F3 have 46.7% correctly classified and 10.1%, 15.1%, and 20.2% incorrectly classified as F2, F5, and F6, respectively. These facts demonstrate that the NN classifier is imprecise and unreliable. Table 4 is the confusion matrix of the SVM classifier. In general, the results are more or less correct, except that there are two fault modes that are poorly classified (those belonging to F3 or F4). Of the faults belonging to F3, 24.1% are misclassified as F1, and 22.7% are misclassified as F4. Faults belonging to F4 have only 31.6% correctly classified and 54.5% incorrectly classified. This means that faults belonging to F4 have a greater chance of being wrongly classified than of being correctly classified. The overall 65.3% accuracy of SVM shows that it has better overall accuracy than the NN classifier (57.5%). However, the SVM classification accuracy performed poorly in one fault mode (F4). Table 5 is the confusion matrix with the KNN classifier. It is worth noting that the KNN classifier has better accuracy than the others. The overall accuracy is correct, but it can be seen that faults belonging to F4 are the least accurate. They have 69% correctly classified and 24.9% incorrectly classified as F3. The KNN classifier presents an overall accuracy of 85.3%, as compared to 68.5% and 57.5% for the SVM and NN classifiers, respectively, meaning that KNN is undoubtedly a better performer than the others.

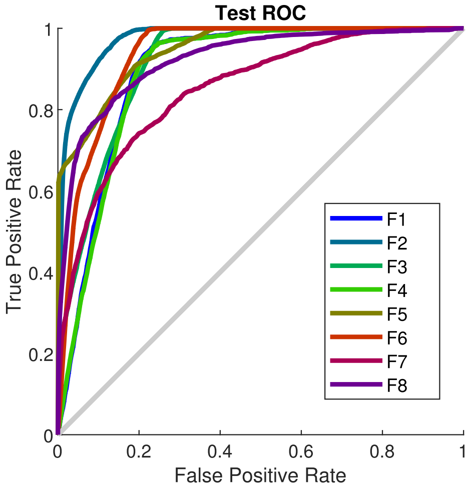

FIGURE 9. ROC curves of the dataset using NN classifier.

TABLE 6. AUC values of the classifers.

Fault type in PWR	ANN classifier	KNN classifier	SVM classifier
Fault 1	0.88	0.97	0.86
Fault 2	0.92	0.90	0.98
Fault 3	0.85	0.96	0.88
Fault 4	0.85	0.94	0.82
Fault 5	0.90	0.95	0.91
Fault 6	0.88	1	0.96
Fault 7	0.82	0.97	0.97
Fault 8	0.90	0.97	0.93

TABLE 7. Performance comparison of the classifiers.

Methods	Average AUC	Average accuracy	Execution time
KNN	0.95	85.3%	3.7×10^1 s
SVM	0.91	63.3%	4.5×10^4 s
NN	0.87	57.5%	1.15×10^3 s

Receiver operating characteristic (ROC) curves are also used to analyse the performance of the classifiers. By definition, an ROC curve shows the true positive rate versus the false positive rate for different thresholds of the classifier output. This approach is then used to visualize the classification performance under different decision thresholds; therefore, it is a good tool for evaluating the performance of an algorithm. The ROC curves generated to evaluate the NN classifier are shown in Fig. 9. From ROC curve shapes, it can be said that NN is a reasonably accurate algorithm for the eight faults because the curves are all away from the diagonal. For the sake of brevity, the ROC curves for KNN and SVM are not provided. Instead, the area under curve (AUC) is used as a summary of the ROC curve and describes how much the curve is stretched toward the upper left corner of the diagonal [35]. The AUC for the different classifiers is summarized in Table 6, and the overall performance is given in Table 7. The AUC measurements reveal good classification accuracy for the three classifiers. Nevertheless, the KNN classifier has far better classification performance, with an average AUC of 0.95. The SVM and NN classifiers present an average AUC of 0.91 and 0.87, respectively. In addition to being accurate, the KNN classifier has the lowest computational time with a training time of 3.7×10^1 seconds, as compared to 1.15×10^3 seconds and 4.5×10^4 seconds for the NN and SVM classifiers, respectively.

VI. CONCLUSION

This paper proposes a (FDD) method using NN and KNN approaches, where NN is used for fault detection and KNN is used for fault classification. In the diagnosis process, the data obtained from the PWR reactor are used to train the classifiers. Actuator faults and some types of sensor faults, such as drift and bias faults, are employed to test the classifiers. Moreover, the injection of multiple simultaneous faults is considered. The proposed classification method demonstrates good performance and can be used effectively for the diagnosis of PWR faults. Eight different faults have been successfully classified.

The KNN classifier is also compared to two other machine learning algorithms (NN and SVM). The performance of the KNN classifier is better than the other classifiers in identifying the single and multiple simultanous sensor and actuator faults. A detailed analysis has been performed to compare the performance of the classifiers by computing ROC curves, AUC measures, and confusion matrices. It has been observed that the KNN has the highest AUC average than the NN and SVM classifiers. The confusion matrices confirm the outstanding performance of the KNN classifier over the other techniques. The KNN classifier has indeed an overall accuracy of 85.3%, as compared to 68.5% and 57.5% for the SVM and NN classifiers, respectively. The KNN algorithm is then doubtessly a better performer than the other techniques. SVM has a better overall accuracy than the NN classifier. However, the SVM classifier is defective as it fails in classifying the fault F4; the fault F4 has more chance of being wrongly classified as F1 (54.5%) than of being correctly classified (31.6%). The NN classifier performs poorly in classifying faults belonging to F1 and F4 but it is not classified as defective. In addition to providing better classification accuracy, the KNN is also found to be less computationally expensive in comparison to the NN and SVM methods.

The simple architecture of the proposed KNN algorithm allows easy implementation. But before that the integration of the proposed classifier in a real power plant must follow several stages of verification and validation (V&V), review, and approvals [36], [37]. Future work will investigate an ensemble classifier that combines NN and KNN to improve the efficiency and accuracy of the detection and diagnosis of faults in a PWR.

APPENDIX

Table 8 presents the classification accuracy of the KNN and SVM classifiers.

TABLE 8. Performance of the SVM and KNN classifiers.

Machine learning method	Subgroup	Average accuracy (%)	Training time (s)
SVM	Linear SVM	22.1	1.215×10^5
	Quadratic SVM	34.4	1.497×10^5
	Cubic SVM	19.9	1.309×10^5
	Fine Gaussian SVM	65.5	4.550×10^4.
	Medium Gaussian SVM	62.3	4.052×10^4
	Coarse Gaussian SVM	57.9	5.365×10^4
KNN	Fine KNN	82.6	2.037×10^1
	Medium KNN	84.7	2.255×10^1
	Coarse KNN	77.6	7.812×10^1
	Cosine KNN	74.3	2.935×10^2
	Cubic KNN	84.7	4.152×10^1
	Weighted KNN	85.3	3.700×10^1

REFERENCES

[1] J. Ma and J. Jiang, “Applications of fault detection and diagnosis methods in nuclear power plants: A review,” *Prog. Nucl. Energy*, vol. 53, no. 3, pp. 255–266, Apr. 2011.

[2] A. Ayodeji and Y.-K. Liu, “Support vector ensemble for incipient fault diagnosis in nuclear plant components,” *Nucl. Eng. Technol.*, vol. 50, no. 8, pp. 1306–1313, Dec. 2018.

[3] H. Wang, M.-J. Peng, J. W. Hines, G.-Y. Zheng, Y.-K. Liu, and B. R. Upadhyaya, “A hybrid fault diagnosis methodology with support vector machine and improved particle swarm optimization for nuclear power plants,” *ISA Trans.*, vol. 95, pp. 358–371, Dec. 2019.

[4] F. Jamil, M. Abid, I. Haq, A. Q. Khan, and M. Iqbal, “Fault diagnosis of Pakistan research reactor-2 with data-driven techniques,” *Ann. Nucl. Energy*, vol. 90, pp. 433–440, Apr. 2016.

[5] W. Li, M. Peng, and Q. Wang, “Improved PCA method for sensor fault detection and isolation in a nuclear power plant,” *Nucl. Eng. Technol.*, vol. 51, no. 1, pp. 146–154, Feb. 2019.

[6] Y. Yao, J. Wang, M. Xie, L. Hu, and J. Wang, “A new approach for fault diagnosis with full-scope simulator based on state information imaging in nuclear power plant,” *Ann. Nucl. Energy*, vol. 141, Jun. 2020, Art. no. 107274.

[7] L. M. Elshenawy, M. A. Halawa, T. A. Mahmoud, H. A. Awad, and M. I. Abdo, “Unsupervised machine learning techniques for fault detection and diagnosis in nuclear power plants,” *Prog. Nucl. Energy*, vol. 142, Dec. 2021, Art. no. 103990.

[8] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *arXiv:1409.1556*.

[9] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[10] A. Naimi, J. Deng, A. Abdulrahman, V. Vajpayee, V. Becerra, and N. Bausch, “Dynamic neural network-based system identification of a pressurized water reactor,” in *Proc. 8th Int. Conf. Control, Mechatronics Autom. (ICCMA)*, 2020, pp. 100–104.

[11] A. Naimi, J. Deng, S. Shimjith, and A. J. Arul, “Dynamic neural network-based feedback linearization control of a pressurized water reactor,” in *Proc. 13th Int. Conf. Develop. eSyst. Eng. (DeSE)*, 2020, pp. 228–232.

[12] M. Boroushaki, M. B. Ghofrani, C. Lucas, and M. J. Yazdanpanah, “Identification and control of a nuclear reactor core (VVER) using recurrent neural networks and fuzzy systems,” *IEEE Trans. Nucl. Sci.*, vol. 50, no. 1, pp. 159–174, Feb. 2003.

[13] G. Lee, S. J. Lee, and C. Lee, "A convolutional neural network model for abnormality diagnosis in a nuclear power plant," *Appl. Soft Comput.*, vol. 99, Feb. 2021, Art. no. 106874.

[14] L. Yong-kuo, P. Min-jun, X. Chun-li, and D. Ya-xin, "Research and design of distributed fault diagnosis system in nuclear power plant," *Prog. Nucl. Energy*, vol. 68, pp. 97–110, Sep. 2013.

[15] Y.-K. Liu, C.-L. Xie, M.-J. Peng, and S.-H. Ling, "Improvement of fault diagnosis efficiency in nuclear power plants using hybrid intelligence approach," *Prog. Nucl. Energy*, vol. 76, pp. 122–136, Sep. 2014.

[16] A. Ayodeji, Y.-K. Liu, and H. Xia, "Knowledge base operator support system for nuclear power plant fault diagnosis," *Prog. Nucl. Energy*, vol. 105, pp. 42–50, May 2018.

[17] N. P. Belfiore and I. J. Rudas, "Applications of computational intelligence to mechanical engineering," in *Proc. IEEE 15th Int. Symp. Comput. Intell. Inform. (CINTI)*, Nov. 2014, pp. 351–368.

[18] Q. P. He and J. Wang, "Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 20, no. 4, pp. 345–354, Nov. 2007.

[19] Q. P. He and J. Wang, "Principal component based k-nearest-neighbor rule for semiconductor process fault detection," in *Proc. Amer. Control Conf.*, Jun. 2008, pp. 1606–1611.

[20] G. Verdier and A. Ferreira, "Fault detection with an adaptive distance for the k-nearest neighbors rule," in *Proc. Int. Conf. Comput. Ind. Eng.*, Jul. 2009, pp. 1273–1278.

[21] S. Banerjee, J. Deng, C. Gorse, V. Vajpayee, V. Becerra, and S. Shimjith, "ANN based sensor and actuator fault detection in nuclear reactors," in *Proc. 8th Int. Conf. Control, Mechatronics Autom. (ICCMA)*, 2020, pp. 88–94.

[22] V. Vajpayee, V. Becerra, N. Bausch, J. Deng, S. R. Shimjith, and A. J. Arul, "Dynamic modelling, simulation, and control design of a pressurized water-type nuclear power plant," *Nucl. Eng. Design*, vol. 370, Dec. 2020, Art. no. 110901.

[23] Y. Zhong, W. Zhang, Y. Zhang, J. Zuo, and H. Zhan, "Sensor fault detection and diagnosis for an unmanned quadrotor helicopter," *J. Intell. Robotic Syst.*, vol. 96, nos. 3–4, pp. 555–572, Dec. 2019.

[24] S. Cho, "A study for detection of drift in sensor measurements," Ph.D. dissertation, School Graduate Postdoctoral Stud., The Univ. Western, London, ON, Canada, 2012.

[25] L. Gao, D. Li, L. Yao, and Y. Gao, "Sensor drift fault diagnosis for chiller system using deep recurrent canonical correlation analysis and k-nearest neighbor classifier," *ISA Trans.*, May 2021.

[26] J. Hui and J. Yuan, "Disturbance observer-based robust backstepping load-following control for MHTGRs with actuator saturation and disturbances," *Nucl. Eng. Technol.*, vol. 53, no. 11, pp. 3685–3693, Nov. 2021.

[27] J. Chandrasekar and D. S. Bernstein, "Setpoint tracking with actuator offset and sensor bias—Probing the limits of integral control," *IEEE Control Syst. Mag.*, vol. 27, no. 1, pp. 61–68, Feb. 2007.

[28] S. Heo and J. H. Lee, "Fault detection and classification using artificial neural networks," *IFAC-PapersOnLine*, vol. 51, no. 18, pp. 470–475, 2018.

[29] M. Farshad and J. Sadeh, "Accurate single-phase fault-location method for transmission lines based on k-nearest neighbor algorithm using one-end voltage," *IEEE Trans. Power Del.*, vol. 27, no. 4, pp. 2360–2367, Oct. 2012.

[30] H. Frigui and P. Gader, "Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors and a possibilistic k-nearest neighbor classifier," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 1, pp. 185–199, Feb. 2008.

[31] Z. Chen, L. J. Zhou, X. D. Li, J. N. Zhang, and W. J. Huo, "The Lao text classification method based on KNN," *Proc. Comput. Sci.*, vol. 166, pp. 523–528, Jan. 2020.

[32] R. Gholami and N. Fakhari, "Support vector machine: Principles, parameters, and applications," in *Handbook of Neural Computation*. Amsterdam, The Netherlands: Elsevier, 2017, pp. 515–535.

[33] M. Z. Ali, M. N. S. K. Shabbir, X. Liang, Y. Zhang, and T. Hu, "Machine learning-based fault diagnosis for single- and multi-faults in induction motors using measured stator currents and vibration signals," *IEEE Trans. Ind. Appl.*, vol. 55, no. 3, pp. 2378–2391, May 2019.

[34] *Statistics and Machine Learning Toolbox User's Guide*, Mathwork, Natick, MA, USA, 2017.

[35] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[36] A. H. Khan, Z. H. Khan, and Z. Weiguo, "Model-based verification and validation of safety-critical embedded real-time systems: Formation and tools," in *Embedded and Real Time System Development: A Software Engineering Perspective*. Berlin, Germany: Springer, 2014, pp. 153–183.

[37] N. Wall and A. Kossilov, "Verification and validation of software related to nuclear power plant instrumentation and control," Int. At. Energy Agency, Vienna, Austria, Tech. Rep. 384, 1999. [Online]. Available: https://www.iaea.org/publications/5718/verification-and-validation-of-software-related-tonuclear-power-plant-instrumentation-and-control

AMINE NAIMI is currently a Researcher with the School of Built Environment, Engineering and Computing, Leeds Beckett University. His research interests include control systems, system identification, and artificial intelligence.

JIAMEI DENG (Senior Member, IEEE) received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 1988, the M.S. degree from the Shanghai Institute of Mechanical Engineering, Shanghai, China, and the Ph.D. degree from Reading University, U.K., in 2005. She is currently a Professor in artificial intelligence and energy with Leeds Beckett University with emphasis on control systems design, predictive models, data analytics, and power plant safety monitoring.

S. R. SHIMJITH received the B.E. degree from Mangalore University, India, in 1995, and the Ph.D. degree from the Indian Institute of Technology Bombay, Mumbai, India, in 2012. He is currently with the Bhabha Atomic Research Centre, India, and is also a Professor with the Homi Bhabha National Institute, India. His research interests include mathematical modeling, analysis, simulation, and control of nuclear reactors and power plants.

A. JOHN ARUL received the M.Sc. degree in physics from Bharathidasan University, in 1988, and the M.S. degree in software systems from the Birla Institute of Technology, in 1999, and the Ph.D. degree from the Indian Institute of Technology Bombay, Mumbai, India, in 2015. He is currently a Researcher in nuclear engineering, risk and reliability analysis with the Indira Gandhi Centre for Atomic Research, Kalpakkam, India. His current research interest includes the application of artificial intelligence techniques to reactor safety and resilience.