Fault Detection and Diagnosis of Nuclear Power Plant Using Deep Learning Architecture

Ibrahim Ahmed, Gyunyoung Heo*, Moath Kassim
Department of Nuclear Engineering, Kyung Hee University, 1732 Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do 17104, Korea
**Corresponding author: gheo@khu.ac.kr*

1. Introduction

Being a safety-critical system, safety is extremely important for any nuclear power plant (NPP). Therefore, to maintain the safety of NPPs at an acceptable level, preventive measures are necessary to deal with potential issues. During plant operation, faults and failures can occur in sensors, equipment, and processes which can have impact on the performance of the plant. These faults are more prominent in aged NPPs because of their vulnerability to aging-related faults [1]. Hence there is need to monitor the status of the plant during operation. To do this, fault detection and diagnosis (FDD) techniques are developed and used in NPPs. FDD is the process of detecting and identifying unexpected behavior in a system. One of these techniques is data-driven methods, which comprises of artificial neural network (ANN), multivariate state estimation technique (MSET), principal component analysis (PCA), and autoassociative kernel regression (AAKR) [2]. However, the possibility and applicability of the deep learning – the current trend in the field of machine learning, to FDD of NPPs is not yet explored. Therefore, this work seeks to propose and apply deep learning techniques to FDD of NPPs. Deep learning originated from artificial neural network, and it is a branch of machine learning algorithms that use a cascade of many layers of non-linear processing units for feature extraction and transformation. It based on a set of algorithms that attempt to model/learn high level abstractions and hierarchy representations in data. There are various deep learning architectures, which include Restricted Boltzmann Machine (RBM) based deep belief network (DBN), Convolutional Neural Network (CNN), deep Auto-encoders, and Recurrent Neural Network (RNN). Recently, deep learning has been successfully adopted in various areas such as computer vision, automatic speech recognition, natural language processing, audio recognition and bioinformatics [3, 4], where they have been shown to produce state-of-the-art results on various tasks.

To verify the applicability of the proposed deep learning model, we used the NPP simulation data for accident detection and identification. The verified model showed high performance applicability to FDD of NPPs.

2. Methods and Results

In this section the deep learning technique and its algorithms used in this work is briefly described. The deep learning architecture selected for this work is deep belief network (DBN) which is based on the Restricted Boltzmann Machine (RBM) pre-training techniques. At the end of this section, the results of the application of DBN to the FDD of NPPs are presented and discussed.

2.1 Deep Belief Network Architecture

Deep belief networks is a deep neural network that can be constructed by stacking multiple RBMs, where the output of the lth layer (hidden units) is used as the input of the $(l+1)$th layer (visible units). Since DBN structure is similar to the stacked network of the RBM [5], it is necessary to briefly describe the RBM techniques. The RBM model is as shown in Fig. 1, and it consists of two layers, namely visible layer and hidden layer. RBMs are undirected probabilistic graphical models containing a layer of observable variables and a single layer of latent variables. RBMs may be stacked (one on top of the other) to form deeper models. As noted by its name, the connections between the nodes/neurons within each RBM layer (visible layer, v or hidden layer, h) are restricted, that is, it is a bipartite graph with no connections permitted between any variables in the visible layer or between any units in the hidden layer. Both the visible and hidden layers have their respective biases A_i and B_j as shown in Fig. 1.

DBN employs a multilayered architecture which consists of one visible layer and multiple hidden layers. The visible layer of a DBN accepts the input data and transfers the data to the hidden layers in order to complete the learning process [6]. An example of DBN structure that consists of four (4) stacked RBMs is shown in Fig. 2. Each successive layer in the DBN structure follows the same transformation concept and passes the regularity throughout the DBN architecture [5].

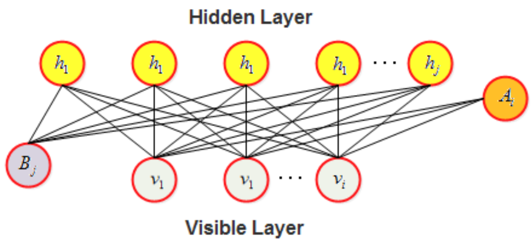

Fig. 1. Structure of a restricted Boltzmann machine.

DBN can be trained in a greedy layer-wise unsupervised way without the label data using RBM training techniques. After pre-training, the parameters of this deep architecture can then be further fine-tuned with respect to labels of the training data.

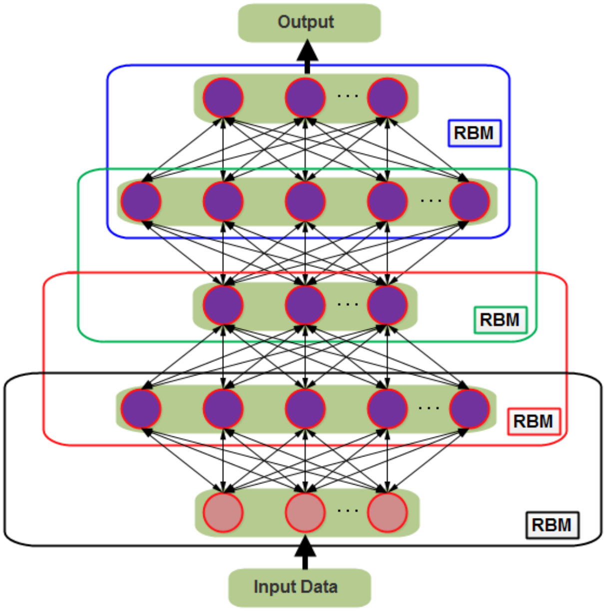

Fig. 2. An architecture of deep belief network.

An RBM is an energy-based function and a special type of Markov random field, which can be trained through contrastive divergence using Gibbs sampling.

Given an RBM model shown in Fig.1 with *m* number of visible (*v*) units and *n* number of hidden (*h*) units and parameters $\varphi(W, B, A)$, the energy function of joint configuration is given as [6]

$$E(\mathbf{v}, \mathbf{h}|\varphi) = -\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij} v_i h_j - \sum_{i=1}^{m} B_i v_i - \sum_{i=1}^{n} A_j h_j \quad (1)$$

The joint distribution over all the units is calculated based on the energy function as

$$P(\mathbf{v}, \mathbf{h}|\varphi) = \frac{e^{-(E(\mathbf{v},\mathbf{h}|\varphi))}}{\sum_{\mathrm{h,v}} e^{-(E(\mathbf{v},\mathbf{h}|\varphi))}}, \quad (2)$$

where the denominator of eqn.(2) is a normalization factor. From eqns. (1) and (2), the procedure for training an RBM is developed. The synaptic weights between the visible and hidden layer of the RBM can be determined iteratively during an RBM training as follows:

For a given training data, the states of neurons or neuron activation probabilities in the RBM hidden layer are determined through transforming the states of neurons in the visible layer with corresponding synaptic weights and the biases of hidden layer neurons with a conditional probability distribution function

$$P(h_j = 1|v) = \sigma\left(B_j + \sum_{i=1}^{m} w_{ij} v_i\right) \quad (3)$$

where $\sigma(x) = 1/(1 + e^{-x})$ is a sigmoid activation function. We then compute the positive phase, $\langle v_i h_j \rangle$.
The visible layer is then reconstructed by transforming the states of neurons in the hidden layer with corresponding synaptic weights and the biases of visible layer with a conditional probability distribution function

$$P(v_i = 1|h) = \sigma\left(A_i + \sum_{j=1}^{n} w_{ij} h_j\right) \quad (4)$$

The results of the reconstruction of the visible units are again used to determine the neuron activation probability of the hidden units using eqn. (3), and then compute the negative phase, $\langle v_i h_j \rangle$.
Each weight w_{ij} is then updated using

$$\Delta w_{ij} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{reconstruction} \quad (5)$$

We then repeatedly update each weights using eqn. (5) for every training example until a specified number of epochs is reached.

The above described procedures are applied to each RBM in the DBN. After training all the RBMs in an unsupervised manner using only the data inputs, the DBN is then fine-tuning using the training data with output label examples in a supervised manner.

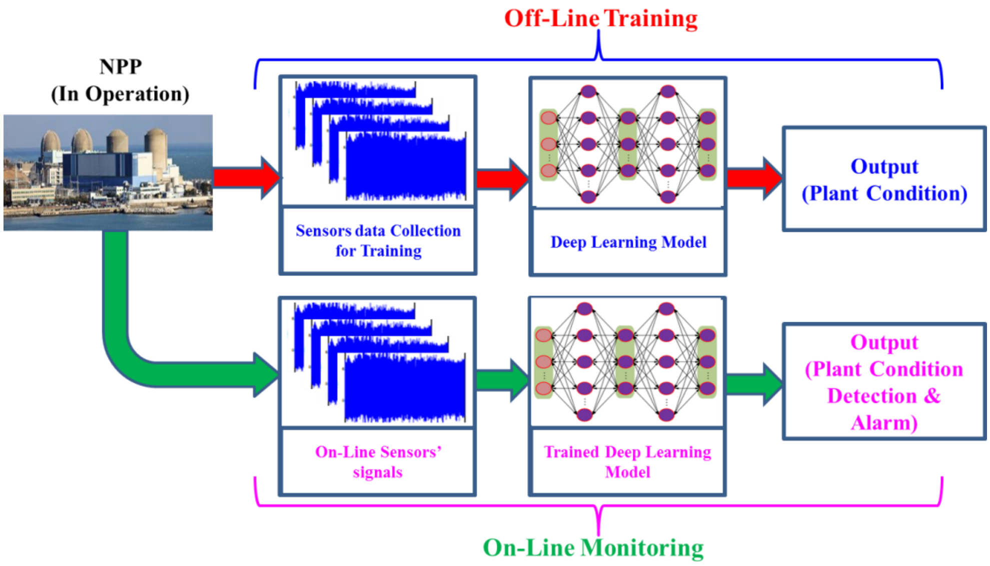

Fig. 3. Proposed deep learning model framework for fault detection and diagnosis of NPP.

2.2 Application of DBN to FDD of NPPs

In order to demonstrate the applicability of deep learning to FDD in NPP, we first developed a model framework shown in Fig. 3. As shown in Fig. 3, the training and test data for the model is collected from the NPP during plant operation which is used to train the model off-line. After the model is built, trained and tested, it can now be applied to NPP for on-line monitoring to monitor plant condition and alert the operator through alarm if a fault is detected. It is important to note that, the regression is used at the top/output layer of the DBN in case of predictions and inferences while the softmax function (eqn. (6)), is used in case of classification during fine-tuning with back-propagation. The softmax squashed a K-dimensional vector $\mathbf{x}$ of arbitrary real values to a K-dimensional vector $p(\mathbf{x})$ of real values in the range (0, 1) that add up to 1. The probability for the jth class given a sample input vector $\mathbf{x}$ to the output layer and a weighting vector $\mathbf{w}$ connected to the jth neuron of the output layer is calculated as

$$p(y = j|\mathbf{x}) = \frac{e^{\mathbf{w}_j^T\mathbf{x}}}{\sum_{i=1}^{K} e^{\mathbf{w}_i^T\mathbf{x}}} . \qquad (6)$$

The proposed deep learning model is applied to the accidents detection, identification and classification in NPPs. The focus of this research work is that, the NPP operator should be able to know at every time instance, the condition of the NPP. The proposed model should be able to tell the operator that the plant condition is either normal or not. If the abnormality occurs as a result of an accident, the proposed model should be able to tell the operator the kind of accident. However, in order to achieve this purpose, there is need to train a deep learning with all the possible accidents in NPP.

Having developed the framework, the simulation data is collected from plant simulator model. To collect the data, several plant simulation conditions are performed. Firstly, the data is collected for normal operation condition which are fault-free normal data. Then, various accident simulation scenarios such as loss of coolant accident (LOCA), steam generator tube rupture (SGTR), and steam line break (SLB) inside and outside of containment (SLB-IC, SLB-OC) building are performed and data is collected for each accident conditions. The data are collected from the sensors variables in the simulator. The variables that are sensitive to all the plant conditions are selected. As such, 23 sensor variables that best described the plant conditions are selected for training and testing of the model. The selection criteria for the 23 sensor variables are as follows:

- ✓ Among any redundant sensor variables, one is selected since they all have the same data values. Training the model with all the redundant variables included may degrade the model performance.
- ✓ For all the accident scenarios simulated, some particular sensor variables did not change. That is, they remained constant and the accident conditions have not impact on them. Those sensor variables data are eliminated from the training data.

A softmax classifier function given in eqn. (6) is used at the output layer of the Model for this purpose. The number of the neurons at the output of the Deep network softmax classifier is equal to the number of accident scenarios plus normal condition. That is, with 23 sensor variables, the deep network will have 23 set of inputs vector and 5 set of outputs if the number of accident scenarios is 4 plus normal operating condition. As described earlier, softmax operates based on probability distribution (between 0 & 1) and assigned a probability to each of its output based on the input to the network. The output that is related to the applied input to the network has the highest probability. With this approach, the operator will be able to know if the accident occur or not and what type of accident it is if occurred.

Having selected the data, the DBN is trained with the following parameters: 3 numbers of hidden layers, 100 numbers of neuron per each hidden layer, learning rate of 0.1 with momentum of 0.9, and 10,000 epochs.

2.3 Results

For the five plant conditions simulated and learned by the proposed deep learning model, the obtained accuracy after built, trained, and tested the model, for each of the plant conditions is shown in Table I. The model was able to classify a set of data points of 23 sensor variables, and detected the plant conditions for all the data except in the case of STGR, which has the accuracy of 93.75%. The overall accuracy of the model is obtained to be 98.9%, with error rate of 1.1%.

Table I: Classification Accuracy

Plant condition	Normal	LOCA	SGTR	SLB-IC	SLB-OC
Accuracy (%)	100	100	93.75	100	100

Table II: Confusion Matrix

		Predicted				
		Normal	LOCA	SGTR	SLB-IC	SLB-OC
Actual Class	Normal	25	0	0	0	0
	LOCA	0	16	0	0	0
	SGTR	1	0	15	0	0
	SLB-IC	0	0	0	16	0
	SLB-OC	0	0	0	0	16

Table II shows the confusion matrix for the 89 samples of the data points associated with the five plant conditions which allows visualization of the performance of the developed deep learning model. All the correct predictions are located in the diagonal of the matrix, so it is easy to visually inspect the matrix for errors, since they are represented by values outside the diagonal. As can be seen on the matrix, out of the 16 actual samples of the data points corrupted as a result of the SGTR accident, the model predicted that one is normal, which is the only error in the classification of the model. We can see that the model was able to distinguish between the learned plant conditions pretty well.

The probability distributions for each class of the plant conditions are shown in Fig. 4. All the plant conditions are 100% predicted except in one of the cases of SGTR where the classified output of the model erroneously predicted *normal* with probability of 51.1% instead of SGTR accident in which the model assigned a probability of 48.9%.

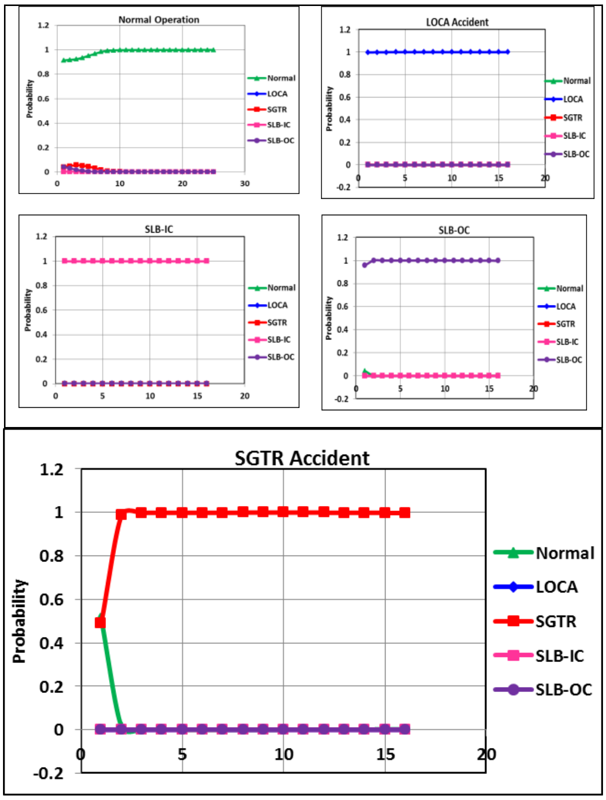

Fig. 4. Plot of probabilities predicted for each class of the plant conditions.

3. Conclusions

Several data-driven methods has been developed and applied to detect faults and monitor the NPPs sensors and equipment. However, the applicability of deep learning, which is the current trend in the field of machine learning, has not been explored. In this work, we proposed, showed, demonstrated, and verified the deep learning architecture for fault detection and diagnosis of NPPs. The selected architecture in this work is deep belief network which is based on the restricted Boltzmann machine. To verify the proposed model, NPP simulation data is collected and used to train the model. Several plant accident simulations with normal operation are performed, and data is collected for each plant conditions. The proposed model gave the overall performance detection and classification accuracy of 98.9%, with error rate of 1.1% which is enough to monitor the status of the plant condition.

Conclusively, the proposed deep learning techniques in this work shows that the deep learning can effectively be used to monitor and detect the plant condition at every point in time in NPPs. One of the major advantages of this deep learning technique is that, it can be used to model highly complicated and complex non-linear feature with high level of abstractions, which is good for a complex systems like NPPs. The further study of this research and its future direction is to extend the deep learning applications to time-series data as well as to verify the applicability of other deep learning techniques, compare their results with each other and with the current data-driven techniques, and select the best model for fault detection and diagnosis of NPPs. In order to achieve this, two major future directions are defined:

- ✓ The development of the time-series recurrent neural network applications to FDD of NPPs.
- ✓ The development of the integrated clustering algorithms with deep learning techniques for robust plant fault detection, diagnosis and prognosis.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIP) (Grant Number: NRF-2016M2B2A9A02945090).

REFERENCES

[1] International Atomic Energy Agency, Management of Life Cycle and Aging at Nuclear Power Plants: Improved I&C Maintenance, IAEA-TECDOC-1402, Vienna, 2004.

[2] J. Ma, J. Jiang, Applications of Fault Detection and Diagnosis Methods in Nuclear Power Plants: A Review, Progress in Nuclear Energy, Vol.53, pp. 255-266, 2011.

[3] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in Proceedings of the 25th international conference on Machine learning, 2008, pp. 160–167.

[4] A-r. Mohamed, G. E. Dahl, G. E. Hinton, Acoustic Modeling Using Deep Belief Networks, *IEEE Trans. on Audio, Speech, and Language Processing*, Vol.20, no.1, pp. 14-22, 2012.

[5] G. E. Hinton, A Practical Guide to Training Restricted Boltzmann Machines, Univ. of Toronto, Toronto, ON, Canada, Tech. Rep. 2010-003, 2010.

[6] G. E. Hinton, S. Osindero, YW. Teh, A Fast Learning Algorithm for Deep Belief Nets, Neural Computation, Vol.18, no.7, pp. 1527–1554. 2006.