

Research Article

A Denoising Based Autoassociative Model for Robust Sensor Monitoring in Nuclear Power Plants

Ahmad Shaheryar,¹ Xu-Cheng Yin,¹ Hong-Wei Hao,^{1,2} Hazrat Ali,³ and Khalid Iqbal⁴

¹University of Science and Technology Beijing (USTB), 30 Xueyuan Road, Haidian District, Beijing 100083, China

²Institute of Automation, Chinese Academy of Sciences, University of Science and Technology Beijing, 95 Zhongguancun East Road, Beijing 100190, China

³Department of Electrical Engineering, COMSATS Institute of Information Technology, University Road, Abbottabad 22060, Pakistan

⁴COMSATS Institute of Information Technology near Officers Colony, Kamra Road, Attock 43600, Pakistan

Correspondence should be addressed to Ahmad Shaheryar; shaheryar011@yahoo.com and Xu-Cheng Yin; xuchengyin@ustb.edu.cn

Received 18 November 2015; Accepted 19 January 2016

Academic Editor: Eugenijus Ušpuras

Copyright © 2016 Ahmad Shaheryar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sensors health monitoring is essentially important for reliable functioning of safety-critical chemical and nuclear power plants. Autoassociative neural network (AANN) based empirical sensor models have widely been reported for sensor calibration monitoring. However, such ill-posed data driven models may result in poor generalization and robustness. To address above-mentioned issues, several regularization heuristics such as training with jitter, weight decay, and cross-validation are suggested in literature. Apart from these regularization heuristics, traditional error gradient based supervised learning algorithms for multilayered AANN models are highly susceptible of being trapped in local optimum. In order to address poor regularization and robust learning issues, here, we propose a denoised autoassociative sensor model (DAASM) based on deep learning framework. Proposed DAASM model comprises multiple hidden layers which are pretrained greedily in an unsupervised fashion under denoising autoencoder architecture. In order to improve robustness, dropout heuristic and domain specific data corruption processes are exercised during unsupervised pretraining phase. The proposed sensor model is trained and tested on sensor data from a PWR type nuclear power plant. Accuracy, autosensitivity, spillover, and sequential probability ratio test (SPRT) based fault detectability metrics are used for performance assessment and comparison with extensively reported five-layer AANN model by Kramer.

1. Introduction

From safety and reliability stand point, sensors are one of the critical infrastructures in modern day automatic controlled nuclear power plants [1]. Decision for a control action, either by operator or by automatic controller, depends on correct plant state reflected by its sensors. “Defense in depth” (defense in depth safety concept requires mission critical systems to be redundant and diverse in implementation to avoid single mode failure scenarios) safety concept for such mission critical processes essentially requires a sensor health monitoring system. Such sensor health monitoring system has multifaceted benefits which are just not limited to process safety, reliability, and availability but also in context of cost

benefits from condition based maintenance approach [2, 3]. A typical sensor health monitoring system may include tasks of sensor fault detection, isolation, and value estimation [4]. Basic sensor monitoring architecture comprises two modules as depicted in Figure 1. The first module implements a correlated sensor model which provides analytical estimates for monitored sensor’s values. Residuals values are evaluated by differencing the observed and estimated sensor values and are supplied to residual analysis module for fault hypothesis testing. These correlated sensor models are based on either the first principles models (e.g., energy conservation and material balance) or history based data driven models [5]. However, sensor modeling using empirical techniques from

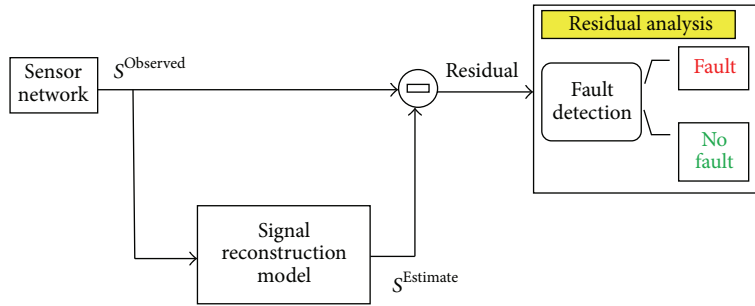


FIGURE 1: Integrated sensor estimation and fault detection architecture.

statistics and artificial intelligence are an active area of research [6, 7].

In order to model complex nonlinearity in physical process sensors, autoassociative neural network based sensor models had widely been used and reported for calibration monitoring in chemical processes [8–11] and nuclear power plants [12–15]. Data driven training procedures for such neural network based sensor models discover the underlying statistical regularities among input sensors from history data and try to model them by adjusting network parameters. Five-layer AANN is one of the earliest autoassociative architectures proposed for sensor and process modeling [8].

In contrast to shallow single layered architectures, these multilayered neural architectures have flexibility for modeling complex nonlinear functions [16, 17]. However, harnessing the complexity offered by these deep NN models without overfitting requires effective regularization techniques. Several heuristics based standard regularization methods are suggested and exercised in literature [18, 19] such as training with jitter (noise), Levenberg-Marquardt training, weight decay, neuron pruning, cross validation, and Bayesian regularization. Despite all these regularization heuristics, the joint learning of multiple hidden layers via backpropagation of error gradient inherently suffers from gradient vanishing problem at the earlier layers [20]. This gradient instability problem restricts the very first hidden layer (closer to input) from fully exploiting the underlying structure in original data distribution. Result is the poor generalization and prediction inconsistency. Problem gets even more complex and hard due to inherent noise and colinearity in sensor data.

Considering the complexity and training difficulty due to gradient instability in five-layer AANN topology, Tan and Mayrovouniotis proposed a shallow network topology of three layers, known as input trained neural network (ITN-network) [21]. However, the modeling flexibility gets compromised by shallow architecture of ITN-network.

The regularization and robustness issues associated with these traditional learning procedures motivate the need for complementary approaches. Contrary to shallow architecture approach by Tan and Mayrovouniotis [21], here, we are interested in preserving the modeling flexibility offered by many layered architectures without being compromised on generalization and robustness of the sensor model. Recent research on greedy layerwise learning approaches [22, 23] has been found successful for efficient learning in deep multilayered

neural architectures for image, speech, and natural language processing [24]. So, for a multilayered DAASM, we proposed to address poor regularization through deep learning framework. Contrary to joint multilayer learning methods for traditional AANN models, the deep learning framework employs greedy layerwise pretraining approach. Following the deep learning framework, each layer in the proposed DAASM is regularized individually through unsupervised pretraining under denoising based learning objective. This denoising based learning is commenced under autoencoder architectures as elaborated in Section 3. It essentially serves several purposes:

- (1) Helps deep models in capturing robust statistical regularities among input sensors.
- (2) Initializes network parameters in basin of attraction with good generalization properties [17, 25].
- (3) Implicitly addresses model's robustness by learning hidden layer mappings which are stable and invariant to perturbation caused by failed sensor states.

Moreover, robustness to failed sensor states is not an automatic property of AANN based sensor models but is primarily essential for fault detection. Consequently, traditional AANN based sensor model requires explicit treatment for robustness against failed sensor states. However, for the case of DAASM, an explicit data corruption process is exercised during denoising based unsupervised pretraining phase. The proposed corruption process is derived from drift, additive, and gross type failure scenarios as elaborated in Section 4.1. Robustness to faulty sensor conditions is an implicit process of denoising based unsupervised pretraining phase. Robustness of the proposed DAASM, against different sensor failure scenarios, is rigorously studied and demonstrated through invariance measurement at multiple hidden layers in the DAASM network (see Section 7). The full DAASM architecture and layerwise pretraining is detailed in Section 4. We will compare the proposed DAASM based sensor model with an extensively reported five-layer AANN based sensor model by Kramer. Both sensor models are trained on sensor data sampled from full power steady operation of a pressurized water reactor. Finally, performance assessment with respect to accuracy, autosensitivity, cross-sensitivity, and fault detectability metrics is conducted under Section 8.

2. Problem Formulation

In context of sensor fault detection application, the purpose of a typical sensor reconstruction model is to estimate correct sensor value from its corrupted observation. The objective is to model relationships among input sensors which are invariant and robust against sensor faults. So, empirical learning for robust sensor relationships can be formulated as sensor denoising problem. However, contrary to the superimposed channel/acquisition noise, the term “denoising” specifically corresponds to the corruption caused by gross, offset, and drift type sensor failures. Under such denoising based learning objective, the empirical sensor model can be forced to learn a function that captures the robust relationships among correlated sensors and is capable of restoring true sensor value from a corrupted version of it.

Let S_{True} and \tilde{S}_{Obs} be the normal and corrupted sensor states related by some corruption process $\varphi(\cdot)$ as follows:

$$\tilde{S}_{\text{Obs}} = \varphi(S_{\text{True}}), \quad (1)$$

where $\varphi : R^n \rightarrow R^n$ is a stochastic corruption caused by an arbitrary type sensor failure. The learning objective for denoising task can be formulated as

$$f = \arg \min_f E_{S_{\text{True}}} \|f(\tilde{S}_{\text{Obs}}) - S_{\text{True}}\|_2^2. \quad (2)$$

Under minimization of above formulation, the objective of empirical learning is to search for f that best approximates φ^{-1} . Furthermore, we will formulate and learn such sensor value estimation and restoration function under neural network based autoassociative model driven by deep learning framework.

2.1. Basic Deep Learning Framework. Neural network research suggests that the composition of several levels of nonlinearity is key to the efficient modeling of complex functions. However, optimization of deep architecture with traditional gradient based supervised learning methods has resulted in suboptimal solutions with poor generalization. Joint learning of multiple hidden layers via backpropagation of error gradient inherently suffers from gradient vanishing problem at the earlier layers and hence constrains the hidden layers from fully exploiting the underlying structure in original data distribution. In 2006, Hinton in his pioneering work proposed a systematic greedy layer by layer training of a deep network. The idea is to divide the training of successive layers of a deep network in the form of small subnetworks and use unsupervised learning to minimize input reconstruction error. This technique successfully eliminates the shortcomings of the gradient based learning by averting the local minima. Deep learning framework employs a systematic three-step training approach as follows:

- (1) Pretraining one layer at a time in a greedy way.
- (2) Using unsupervised learning at each layer in a way that preserves information from the input and disentangles factors of variation.
- (3) Fine-tuning the whole network with respect to the ultimate criterion of interest.

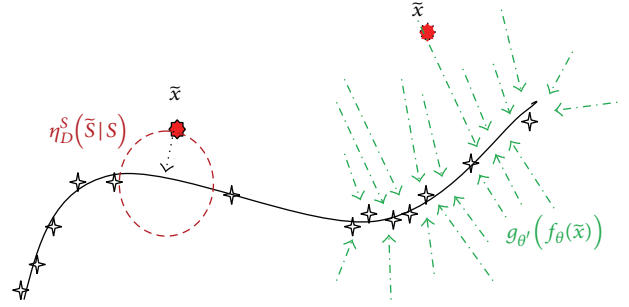


FIGURE 2: Suppose training data (✦) concentrate near a low-dimensional manifold. Corrupted examples (✶) obtained by applying corruption process $\eta_D^S(\tilde{S} | S)$ will lie farther from the manifold. The model learns $g_{\theta'}(f_{\theta}(\tilde{x}))$ with $P(X | \tilde{X})$ to “project them back” onto the manifold.

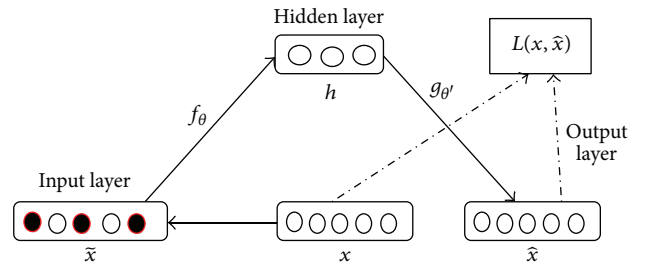


FIGURE 3: Basic denoising autoencoder (DAE) scheme. An empty circle depicts a single neuron. A filled circle depicts corrupted units in input vector.

3. Building Block for DAASM

In relation to empirical modeling approach as formulated in Section 2, denoising autoencoder (DAE) [26] is the most promising building block for pretraining and composition of deep autoassociative sensor model. DAE is a variant of the traditional autoencoder neural network, where learning objective is to reconstruct the original uncorrupted input x from partially corrupted or missing inputs \tilde{x} . Under training criterion of reconstruction error minimization, DAE is forced to conserve information details about the input at its hidden layer mappings. The regularization effect of denoising based learning objective pushes the DAE network towards true manifold underlying the high dimension input data as depicted in Figure 2. Hence, implicitly captures the underlying data generating distribution by exploring robust statistical regularities in input data. A typical DAE architecture, as depicted in Figure 3, comprises an input, output, and a hidden layer. An empty circle depicts a neuron unit. The input layer acts as a proxy layer to the original clean input. Meanwhile, the red filled units in input layer are proxies to clean input units which are randomly selected for corruption under some artificial noise process. $L(x, \tilde{x})$ is an empirical loss function to be optimized during training process.

Let x_i be the original data vector with $i = 1, 2, \dots, N$ elements, while \tilde{x}_i represents the partially corrupted version obtained through corruption process η_D . The encoder and

decoder functions corresponding to DAE in Figure 3 are defined as

$$\begin{aligned} h(\tilde{x}_i) &= f_\theta(\tilde{x}) = \sigma(W\tilde{x}_i + b), \\ \widehat{X}(\tilde{x}_i) &= g_{\theta'}(h) = \sigma(W'h(\tilde{x}_i) + b'). \end{aligned} \quad (3)$$

The encoder function $f_\theta(\tilde{x})$ transforms input data to $h(\tilde{x}_i)$ mapping through a sigmoid type activation function $\sigma(x) = (1 + \exp^{-x})^{-1}$ at hidden layer neurons. $\widehat{X}(\tilde{x}_i)$ is an approximate reconstruction of x obtained through decoder function $g_{\theta'}(h)$ through reverse mapping followed by sigmoid activation at output layer. Meanwhile, $\theta = \{\theta, \theta'\} = \{W, b, W', b'\}$ are the weight and bias parameters corresponding to these encoder and decoder functions.

In relation to sensor reconstruction model as formulated in Section 2, the above-described DAE can be reinterpreted as follows:

$$\begin{aligned} x &\sim S = \{s_i\}_{i=1}^n, \\ \tilde{x} &\sim \tilde{S} = \eta_D((\tilde{s} | s)), \\ \hat{x} &\sim \widehat{S} = \eta_D^{-1}(\tilde{S}). \end{aligned} \quad (4)$$

S are the input sensor values under fault free steady state operation. \tilde{S} is a partially corrupted input which is generated through an artificial corruption process η_D on selected subset in input sensor set $\{s_i\}$. \widehat{S} are the estimated sensor values by reconstruction function learnt on clean and corrupted inputs S and \tilde{S} . Network parameters θ , for DAE, can be learned in an unsupervised setting through minimization of the reconstruction loss in

$$L(x, \hat{x}; \theta) \sim L(S, \widehat{S}; \theta) = \arg \min_{\theta} \sum_{i=1}^N \|S - \widehat{S}\|_2^2. \quad (5)$$

4. DAASM Architecture and Regularization

In order to capture complex nonlinear relationships among input sensors, a multilayered architecture is proposed for denoised autoassociative sensor model (DAASM). Individual layers in network hierarchy are pretrained successively from bottom to top. For a well regularized sensor model, the structure and optimization objective in greedy layerwise pretraining play a crucial role. Two heuristics are applied for robust learning in DAASM as follows:

- (1) Each successive layer in multilayered DAASM assembly is pretrained in an unsupervised fashion under denoising autoencoder (DAE) as elaborated in Section 3.
- (2) To address robustness, data corruption processes for denoising based pretraining task are incorporated with domain specific failure scenarios which are derived from different types of sensor faults. These heuristics serve several purposes:
 - (i) Forcing the DAE output to match the original uncorrupted input data acts as a strong

regularizer. It helps avoid the trivial identity learning especially under overcomplete hidden layer setting.

- (ii) Denoising procedure during pretraining leads to latent representations that are robust to input perturbations.
- (iii) Addition of corrupted data set increases training set size and thus is useful in alleviating overfitting problem.

Full DAASM is learnt in two stages: (1) an unsupervised pretraining phase and (2) a supervised fine-tuning phase. As shown in Figure 4, the pretraining phase follows a hierarchal learning process in which successive DAEs in the stack hierarchy are defined and trained in an unsupervised fashion on the preceding hidden layer activations. Full sensor model is constructed by stacking hidden layers from unsupervised pretrained DAEs followed by a supervised fine-tuning phase. For each DAE in the stack hierarchy, the optimization objective for unsupervised pretraining will remain the same as in relation (5). However, weight decay regularization term is added to the loss function which constrains network complexity by penalizing large weight values. In relation (6), $\{W, W'\}$ are the network weight parameters corresponding to encoder and decoder function, while λ is the weight decay hyperparameter:

$$L(S, \widehat{S}; \theta) = \frac{1}{N} \sum_{i=1}^N \|S - \widehat{S}\|_2^2 + \frac{\lambda}{2} (\|W\|^2 + \|W'\|^2). \quad (6)$$

In a typical DAE architecture, a number of input and output layer neurons are fixed corresponding to input data dimension d ; however, middle layer neuron counts d' can be adjusted according to problem complexity. Literature in deep learning suggests that under complete middle layer ($d' < d$), for DAE architecture, results in dense compressed representation at the middle layer. Such compressed representation has tendency to entangle information (change in a single aspect of the input translates into significant changes in all components of the hidden representation) [27]. This entangling tendency directly affects the cross-sensitivity of sensor reconstruction model especially for the case of gross type sensor failure. Considering that, here, we choose for an overcomplete hidden layer setting ($d' > d$). Under overcomplete setting, denoising based optimization objective acts as a strong regularizer and inherently prevents DAE from learning identity function.

Anticlockwise flow in Figure 4 shows architecture and greedy layer by layer unsupervised pretraining procedure for all hidden layers in DAASM stack. For each hidden layer h_l , a DAE block is shown, in which an encoder function $f_\theta^l(\cdot)$ and a decoder function $g_{\theta'}^l(\cdot)$ are learnt by minimizing the loss function corresponding to fault free reconstruction of the inputs as in relation (6). For the case of first hidden layer h_1 , the corresponding DAE-1 is trained directly on sensor data using $L(S, \widehat{S}, \theta)$ loss function in (6). However, hidden layers h_2 through h_3 are learnt on data from preceding hidden layer activations using recursive relation in (7). So the loss function

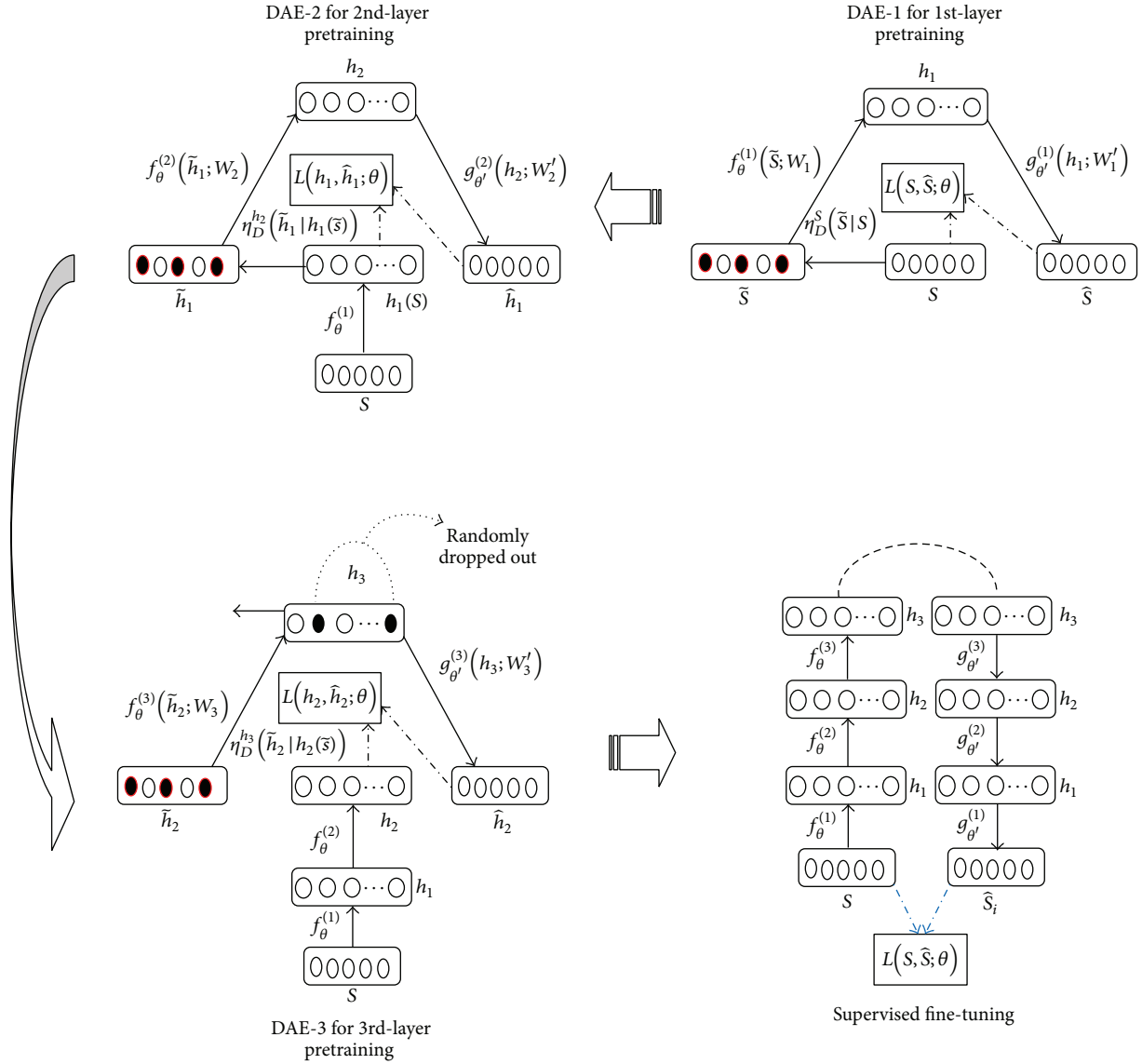


FIGURE 4: DAASM architecture and greedy learning procedure. Greedy layerwise pretraining procedure is depicted by counterclockwise flow in the figure.

corresponding to DAE-1 and DAE-2 can be represented as $L(h_l, \hat{h}_l, \theta)$, where \hat{h} is an approximate reconstruction of h :

$$h^l = f_\theta^l(h_{l-1}; W_l) = \text{sigm}(W^l h^{l-1} + b^l); \quad (7)$$

$$1 \leq l \leq L = 3.$$

$\langle W \rangle$ are the network weights corresponding to encoder part in DAE.

The noise process $\eta_D^S(\tilde{S} | S)$ for DAE-1 corresponds to a salt-and-pepper (SPN) type corruption process, in which a fraction of the input sensor set S (chosen at random for each example) is set to minimum or maximum possible value (typically 0 or 1). The selected noise process models gross type failure scenarios and drives the DAE-1 network to learning invariance against such type of sensor failures. The noise functions $\eta_D^{h_2}(\tilde{h}_1 | h_1(\tilde{S}))$ employ a corruption process in

which $h_1(S)$ and $h_1(\tilde{S})$ from pretrained DAE-1 will be used as the clean and noisy input for DAE-2 pretraining. Finally, an additive Gaussian type corruption process (AGN) : $\tilde{x} | x \sim N(x, \sigma^2)$ is used for DAE-3 noise function $\eta_D^{h_3}(\tilde{h}_2 | h_2(\tilde{S}))$. We will further mathematically formulate and discuss all these corruption processes in detail in Section 4.1.

These pretrained layers will initialize the DAASM network parameters in basin of attractions which have good generalization and robustness property. In order to generate a sensor model that is fairly dependent on all inputs, "Dropout" [28] heuristic is applied on h_3 hidden units during DAE-3 pretraining. Random dropouts make it hard for latent representations at h_3 to get specialized on particular sensors in the input set. Finally, pretrained DAEs are unfolded into a deep autoassociator network with L number of encoder and $L - 1$ decoder cascade as shown in unsupervised fine-tuning

phase in Figure 3. The final network comprises one input layer, one output, and $2L - 1$ hidden layers. The input sensor values flow through encoder cascade $f = f_{\theta}^1 o f_{\theta}^{l-1} o \dots o f_{\theta}^1$ using recursive expression in (7) and a decoder cascade $g = g_{\theta'}^1 o g_{\theta'}^{l+1} o \dots o g_{\theta'}^{L-1}$ using the following equations:

$$\begin{aligned} \hat{h}^L &= h^L, \\ \hat{h}^l &= g_{\theta'}^l(h_{l+1}; W_l') = \text{sigm}(W_l' \hat{h}^{l+1} + b_l'); \\ &1 \leq l \leq L - 1 = 2, \end{aligned} \quad (8)$$

$$\hat{S} = g^0(\hat{h}^1) = W_0' \hat{h}^1 + b_0',$$

where $\langle W', b' \rangle$ are network weights and biases of the decoder part in DAE. The entire network is fine-tuned using a semiheuristic based “*Augmented Efficient Backpropagation Algorithm*,” proposed by Embrechts et al. [29], with following minimization objective:

$$L(S, \hat{S}; \theta) = \frac{1}{N} \sum_{i=1}^N \|S - \hat{S}\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^{2L} \|W_k\|_2^2. \quad (9)$$

A $L - 2$ weight decay term is added to the above loss function for network regularization purpose during fine-tuning phase. To circumvent the overfitting, an early stopping procedure, which uses validation error as proxy for the generalization performance, is used during fine-tuning phase.

4.1. Corruption Process $\eta(\cdot)$ for Invariance. For the case of calibration monitoring, an ideal DAASM should learn encoder and decoder functions which are invariant to failed sensor states. So during DAE based pretraining phase, engineered transformations from prior knowledge about the involved

failure types are imposed on clean input. Different data corruption processes $\eta(\cdot)$ are devised for learning of each successive hidden layer. Denoising based learning objective drives the hidden layer mappings to get invariant against such engineered transformations on input data. It is important to understand that denoising based learning approach does not correct the faulty signal explicitly; rather it seeks to extract statistical structure among input signals which is stable and invariant under faults and hence implicitly estimates correct value for faulty signal. Two failure types are identified and defined as follows:

- (i) Gross sensor failure: it includes catastrophic sensor failures. Salt-and-pepper type corruption process, in which a fraction ν of the input sensor set S (chosen at random for each example) is set to minimum or maximum possible value (typically 0 or 1), is selected for modeling gross type failure scenarios.
- (ii) Miscalibration sensor failure: it includes drift, multiplicative, and outlier type sensor failures and is modeled through isotropic Gaussian noise (GS) : $\tilde{x} | x \sim N(x, \sigma^2)$. Instead of selecting an arbitrarily simple noise distribution, we estimated the distribution of sensor's natural noise and exaggerated it to generate noisy training data.

We propose to distribute the denoising based invariance learning task across multiple hidden layers in the DAASM network. Both gross and miscalibration noise types are equally likely to occur in the input space. Gaussian type corruption process is not suitable for input data space S because of its low denoising efficiency against gross type sensor failures. Contrarily, salt-and-pepper type corruption process covers two extremes of sensors failure range and hence provides an upper bound on perturbation due to minor offset and miscalibration type sensor failures. So, salt-and-pepper type corruption process is devised for DAE-1 pretraining as follows:

$$\eta_D^S(\tilde{S} | S) = \text{SPN} \begin{cases} \tilde{s}_i = s_i, & i \notin j, \\ \tilde{s}_i = k, & i \in j, k = \{0, 1\} \text{ with } \begin{cases} \Pr(0) = \frac{1}{2} \\ \Pr(1) = \frac{1}{2} \end{cases} \\ \bigcup_{i=1}^{i=\nu n} \{j_{i \dots N=\nu n}\}, & j_i = [1, n] = \text{rand}(), \nu = \text{input fraction}, n = \text{input dimension}. \end{cases} \quad (10)$$

Gross type sensor failures usually have high impact on cross-sensitivity and can trigger false alarms in other sensors. Such high cross-sensitivity effect may affect isolation of miscalibration type secondary failures in other sensors. In order to minimize the effect, a corruption procedure in which $h_1(S)$ and $h_1(\tilde{S})$ from pretrained DAE-1 are proposed as the clean and noisy input for DAE-2 pretraining. This corruption method is more natural since it causes next hidden layer mappings to get invariant against cross-sensitivity effects

and network aberrations from previous layer. The corruption process is supposed to improve invariance in h_2 layer mappings against cross-sensitivity effects from gross type sensor failures:

$$\eta_D^{h_2}(\tilde{h}_1 | h_1(\tilde{S})) = \tilde{h}_1^i = h_1^i(\tilde{S}), \quad (11)$$

$$\text{where } \tilde{S} = \eta_D^S(\tilde{S} | S) = \text{SPN}.$$

TABLE 1: List of NPP sensors.

Transmitter ID	Transmitter name	Service	Units	Low range	High range
FF1	Feed flow 1	Feedwater flow	KG/S	0	600
FF2	Feed flow 2	Feedwater flow	KG/S	0	600
SF1	STM flow 1	Steam flow	KG/S	0	600
SF2	STM flow 2	Steam flow	KG/S	0	600
SP1	STM PSR 1	Steam pressure	BARG	0	100
SP2	STM PSR 2	Steam pressure	BARG	0	100
SP3	STM PSR 3	Steam pressure	BARG	0	100
PP1	PZR PSR 1	Pressurizer pressure	BARG	116	170
PP2	PZR PSR 2	Pressurizer pressure	BARG	116	170
PL1	PZR LVL 1	Pressurizer level	%	0	100
PL2	PZR LVL 2	Pressurizer level	%	0	100
SGL1	SG LVL NR 1 range	Steam generator level narrow	%	0	100
SGL2	SG LVL NR 2 range	Steam generator level narrow	%	0	100

Here $h_1^i(s)$ corresponds to hidden layer activations against clean sensors at the input layer, while $h_1^i(\tilde{s})$ corresponds to hidden layer activations against partially faulted sensor set.

Finally, to add robustness against small offset and miscalibration type sensor failures, an isotropic Gaussian

type corruption process is devised for DAE-3 pretraining. The corruption procedure corrupts the h_2 hidden layer mappings, against clean sensors at the input layer as $h_2(h_1(S))$, by employing an isotropic Gaussian noise as follows:

$$\eta_D^{h_2}(\tilde{h}_2 | h_2(S)) = \text{AGN} \begin{cases} \tilde{h}_2^i = h_2^i(h_1^i(S)), & i \notin j \\ \tilde{h}_2^i | h_2^i \sim N(s, \sigma^2 I), & i \in j \\ \bigcup_{i=1}^{i=\nu n} \{j_{i \dots N=\nu n}\}, & j_i = [1, n] = \text{rand}(), \nu = \text{input fraction}, n = \text{total inputs}. \end{cases} \quad (12)$$

Finally, clean input is used for the supervised fine-tuning phase in Figure 4.

5. Data Set Description

Intentionally, for study purposes, we limited the modeling scope of DAASM to full power steady operational state. It is the common state in which NPP operates from one refueling to the next. However, in practice it is not possible for NPP systems to be in perfect steady state. Reactivity induced power perturbations, natural process fluctuations, sensor and controller noises, and so forth, are some of the evident causes for NPP parameter fluctuations and are responsible for steady state dynamics. Considering that the collected data set should be fairly representative of all possible steady state dynamics and noise, the selected sensors are sampled during different time spans of one complete operating cycle. The training data set consists of 6104 samples collected during the first two months of full power reactor operations after refueling cycle. Meanwhile 3260 and 2616 samples are reserved for validation and test data sets, respectively. Five test data sets are used for model's performance evaluation. Each test data set consists of 4360 samples collected during eight-month period after refueling operation. In order to account for fault propagation

phenomenon due to large signal groups, a sensor subset is selected for this study. An engineering sense selection based on physical proximity and functional correlation is used to define the sensor subset for this study. Thirteen transmitters, as listed in Table 1, are selected from various services in nuclear steam supply system of a real PWR type NPP. Figure 5 shows the spatial distribution of the selected sensors.

Starting from postrefueling full power startup, the data set covers approximately one year of selected sensors values. Selected sensors are sampled every 10 seconds for consecutive 12-hour time window. Figure 6 shows data plot from few selected sensors.

6. Model Training

NPP sensor's data are divided into training, test, and validation set. Each sensor data set is scaled in 0.1 to 0.9 ranges by using lower and upper extremities corresponding to individual sensor. However, the values 0 and 1 are explicitly reserved for gross and saturation type sensor failures, respectively. Training data consists of 4320 samples from full power steady state reactor operation. Meanwhile test and validation data are used for sensor model optimization and performance evaluation, respectively. The training setup for DAASM

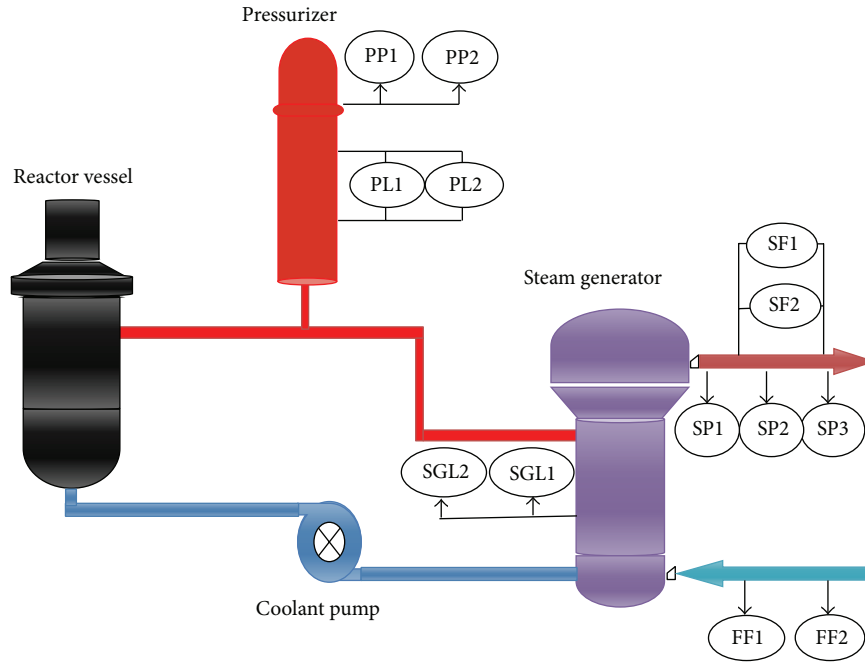


FIGURE 5: Spatial distribution of selected sensor set.

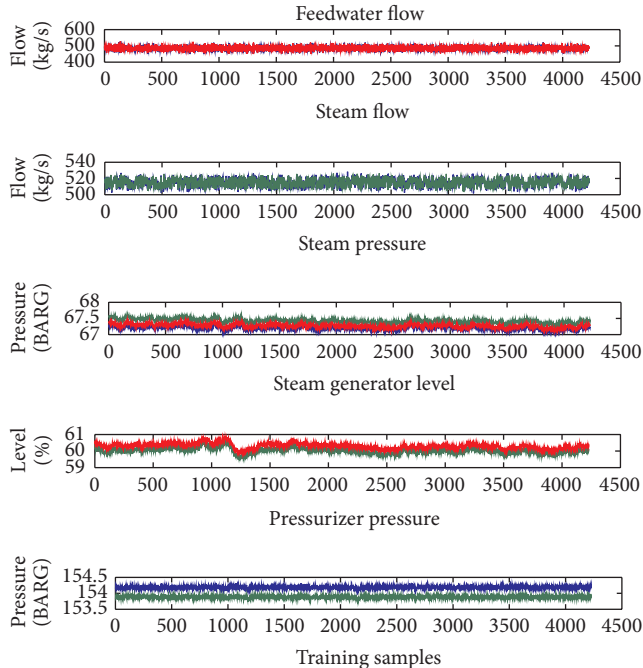


FIGURE 6: Plot of NPP sensors listed in Table 1.

employs two learning stages, an unsupervised learning phase and supervised training phase. DAE based greedy layerwise pretraining of each hidden layer, as described in Section 4, is performed using minibatches from training data set. Stochastic gradient descent based learning algorithm is employed

as suggested in practical training recommendations by [30]. Finally, standard backpropagation algorithm is employed for supervised fine-tuning in fully stacked DAASM in Figure 4. Supervised training is performed using clean sensor input only. The model hyperparameters are set by random grid search method [31]. A summary of the training hyperparameters corresponding to optimum DAASM is shown in Table 2.

7. Invariance Test for Robustness

A layer by layer invariance study is conducted to test the robustness of fully trained DAASM against failed sensor states. Data corruption processes applied during pretraining are essentially meant to learn hidden layer mappings which are stable and invariant to faulty sensor conditions. The following invariance test, for successive hidden layers in final DAASM stack, can provide an insight into the effectiveness of data corruption processes exercised during denoising based pretraining phase. Invariance, for hidden layer mappings h_i , is quantified through mean square error (MSE) between Euclidean (L_2) normalized hidden layer activation $\langle h_i \rangle_n / \|\langle h_i \rangle_n\|_2$ and $\langle \tilde{h}_i \rangle_n / \|\langle \tilde{h}_i \rangle_n\|_2$ against clean and faulty sensors, respectively. Invariance test samples are generated by corrupting randomly selected sensors in input set with varying level of offset failures [5%–50%]. The MSE against each offset level is normalized across hidden layer dimension D_h and number of test samples T_N as shown in (13). Finally these MSE values are normalized with maximal MSE value as in (14). Normalized MSE curves for each successive hidden layer are plotted in Figure 7. Consider

TABLE 2: Summary of DAASM hyperparameters.

Hyperparameter type	Tested hyperparameter values	Successful hyperparameter against optimum model
Pretrained DAE units	3	3
Network architecture	$I_n - E(n, p) - B_n - D(n, p) - o_n$ I_n : input layer neurons B_n : bottleneck layer neurons o_n : output layer neurons $E(n, p)$: encoder cascade $D(n, p)$: decoder cascade n : number of layers p : neurons per layer	$13 - E(2, 20) - 8 - D(2, 20) - 13$
Learning rate for unsupervised pretraining	$\{0.1, 5 \times 10^{-2}, 1 \times 10^{-2}, 5 \times 10^{-3}, 1 \times 10^{-3}, 5 \times 10^{-4}\}$	$[5 \times 10^{-2}, 1 \times 10^{-2}]$
Learning rate for supervised training	Scheduled learning rate based on training error monitoring: $\{0.15, 0.1, 0.005, 0.001, 0.0001\}$	$\{0.1, 0.005, 0.001\}$
Mean pretraining error for each hidden layer	Corresponding to minima observed during cross validation	10^{-4}
Weight decay, λ	$\{10^{-3}, 10^{-4}, 10^{-5}\}$	10^{-3}
Momentum, m	$[0.85, 0.99]$	$[0.95, 0.98]$
Input corruption level, ν	Corrupted input fraction: $\{10\%, 25\%, 30\%, 40\%\}$ Gaussian corruption (% of sensor's nominal value): $\{0.05, 0.10, 0.20, 0.35, 0.50\}$	Input fraction: $[25-35]\%$ Gaussian noise level: $[0.10-0.25]$
Dropout fraction in DAE-3	$\{0.10, 0.20\}$	0.1

$$\text{MSE}(H_l, \% \text{Offset}) = \frac{(1/T_N) \sum_{n=1}^{T_N} (\langle h_i \rangle_n / \|\langle h_i \rangle_n\|_2 - \langle \tilde{h}_i \rangle_n / \|\langle \tilde{h}_i \rangle_n\|_2)^2}{D_h}, \quad (13)$$

$$1 \leq l \leq L = \text{No. of encoder layers} = 3; \% \text{Offset} = 5\%, 10\%, 20\% \dots 50\%,$$

$$\text{MSE}_{\text{Normalized}}(H_l) = \frac{\text{MSE}(H_l, \% \text{Offset})}{\text{MSE}(H_l, \% \text{Max-Offset})}. \quad (14)$$

Layerwise MSE plots, in Figure 7, clearly show that invariance to faulty sensor conditions increases towards higher layers in the network hierarchy. In these plots, lower curves indicate higher level of invariance. To further investigate the effect of increasing invariance on reconstructed sensor values, a sensor model, corresponding to the level “ L ” of each hidden layer, is assembled via encoder and decoder cascade. Robustness of these partial models is quantified through $(1 - S_{\text{Auto}}^i)$. Autosensitivity values S_{Auto}^i (see Section 8.2) are calculated against varying offset failure levels. In Figure 8, layerwise increase in robustness confirms that increased invariance helps in improving overall model's robustness.

8. DAASM versus K-AANN Performance Analysis

Here we will assess and compare the performance of DAASM with popular five-layer AANN model originally proposed by Kramer [8]. The K-AANN model is trained with same data

set as used for DAASM and is regularized with Levenberg-Marquardt algorithm. Furthermore, to improve robustness, training with jitter heuristic is employed by introducing a noise of 10% magnitude on clean sensor input. The five-layer topology 13-17-9-17-13 is found to be optimum for K-AANN model. Both DAASM and K-AANN model are compared through accuracy, robustness, spillover, and fault detectability based performance metrics in the following subsections. All performance metrics are calculated against test data set, consisting of 4320 samples from postrefueling full power NPP operations. Performance metric values are reported.

8.1. Accuracy. Mean square error (MSE) of observed and model estimated sensor values, against fault free test data set, is used to quantify accuracy metric as follows:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N (\hat{S}_i - S_i)^2. \quad (15)$$

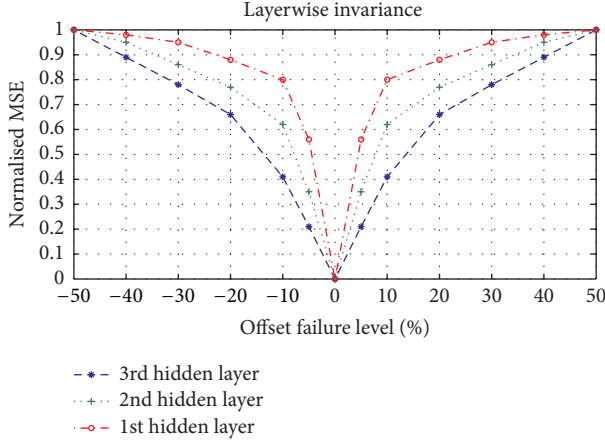


FIGURE 7: Layerwise invariance in DAASM. Lower curves depict higher invariance.

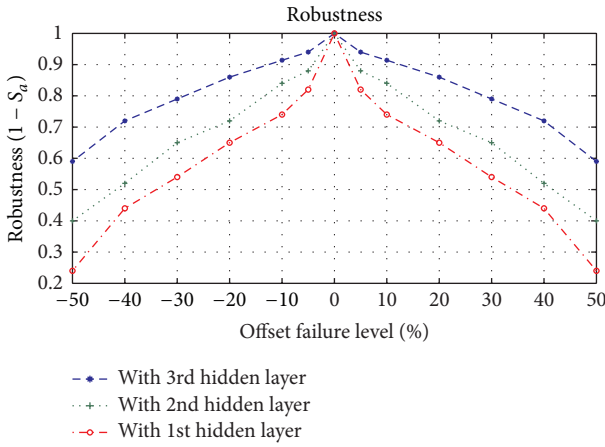


FIGURE 8: Robustness measure (1 – autosensitivity) at multiple hidden layers in DAASM. Higher curves depict high robustness.

The MSE values of all sensors are normalized to their respective span and are presented as percent span in Figure 9. Being an error measure, the lower MSE values by DAASM signify its prediction accuracy.

8.2. Robustness. Robustness is quantified through *autosensitivity* as defined by [32, 33]. It is the measure of model's ability to predict correct sensor values under missing or corrupted sensor states. The measure is averaged over an operating region defined by k samples from test data set as follows:

$$S_{\text{Auto}}^i = \frac{1}{N} \sum_{k=1}^N \left| \frac{\widehat{s}_{ki}^{\text{drift}} - \widehat{s}_{ki}}{s_{ki}^{\text{drift}} - s_{ki}} \right|, \quad (16)$$

where i and k are indexes corresponding to sensors and their respective test samples. s_{ki} is the original sensor value without fault. \widehat{s}_{ki} is the model estimated sensor value against s_{ki} . s_{ki}^{drift} is the drifted/faulted sensor value. $\widehat{s}_{ki}^{\text{drift}}$ is the model estimated sensor value against drifted value s_{ki}^{drift} .

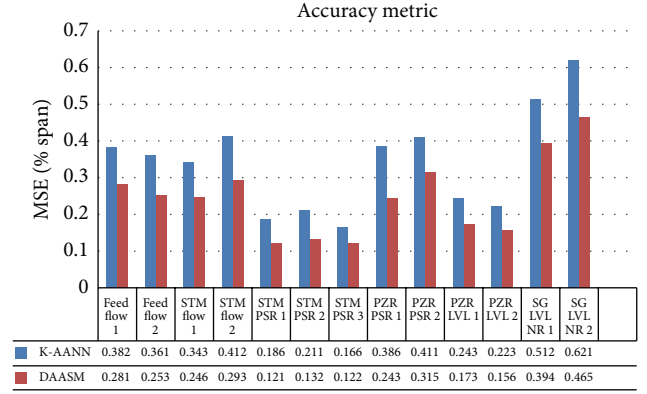


FIGURE 9: MSE depicting DAASM and K-AANN accuracy on each sensor.

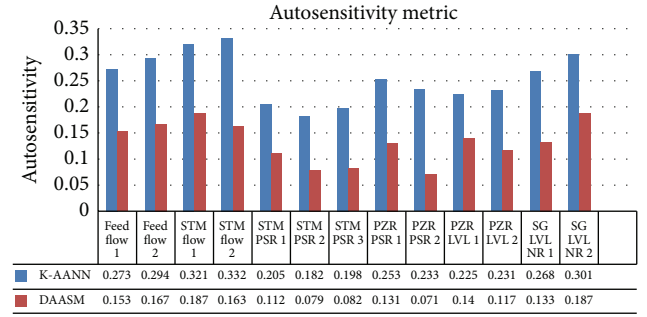


FIGURE 10: Autosensitivity values of individual sensors in both models.

The autosensitivity metric lies in $[0, 1]$ range. For autosensitivity value of one, the model predictions follow the fault with zero residuals; hence no fault can be detected. Smaller autosensitivity values are preferred, which essentially means decreased sensitivity towards small perturbations. Large autosensitivity values may lead to missed alarms due to underestimation of the fault size caused by small residual values. Compared to K-AANN model, in case of DAASM, a significant decrease in autosensitivity values for all sensors is observed. The plot in Figure 10 shows that DAASM is more robust to failed sensor inputs.

To further investigate robustness against large offset failures, both models are evaluated against offset failures in $[5\%–50\%]$ range. For each sensor, samples from test data are corrupted with specific offset level and corresponding autosensitivities are averaged over whole sensor set. Autosensitivity values less than 0.2 are considered as robust. The maximum autosensitivity value of 0.187 is observed in steam flow sensor. The plot in Figure 11 shows that average autosensitivity for both models increases with increasing level of offset failure. However, the autosensitivity curve for DAASM autosensitivity is well below the corresponding K-AANN curve.

8.3. Spillover. Depending upon the size and type of failure, a failed sensor input can cause discrepancy in estimated output for other sensors. The phenomenon is referred to in

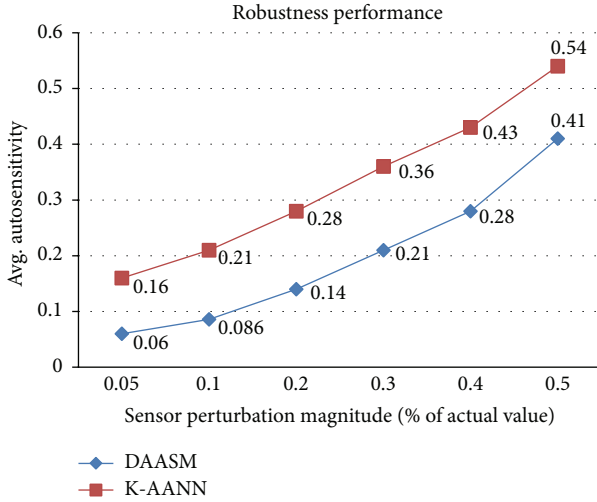


FIGURE 11: Comparison of robustness against increasing offset failure.

literature as “spillover effect” and is quantified through “cross-sensitivity” metric [32]. It quantifies the influence of faulty sensor i on predictions of sensor j as follows:

$$S_{\text{Cross}}^{ji} = \frac{1}{N} \sum_{k=1}^N \left| \frac{\widehat{s}_{kj}^{\text{drift}} - \widehat{s}_{kj}}{s_{ki}^{\text{drift}} - s_{ki}} \right|, \quad (17)$$

$$S_{\text{Cross}}^j = \frac{1}{N-1} \sum_{i=1}^N S_{\text{Cross}}^{ji}, \quad i \neq j, \quad (18)$$

where i and j indexes are used to refer to faulty and nonfaulty sensors, respectively. Meanwhile, k is the index for corresponding test samples. S_{Cross}^{ji} is the cross-sensitivity of sensor j with respect to drift in i th sensor. s_{ki} is the value of i th sensor without any fault. \widehat{s}_{kj} is the model estimated value of j th sensor against s_{ki} . s_{ki}^{drift} is the drifted/faulted value of i th sensor. $\widehat{s}_{kj}^{\text{drift}}$ is the model estimated value of j th sensor against drifted value s_{ki}^{drift} .

The highly distributed representation of the input in neural network based sensor models has pronounced effect on the cross-sensitivity performance. Cross-sensitivity metric value lies in $[0, 1]$ range. High value of cross-sensitivity may set off false alarms in other sensors, provided the residual values overshoot the fault detectability threshold in other sensors. So, minimum cross-sensitivity value is desired for a robust model. The plot in Figure 12 shows that the cross-sensitivity for DAASM is reduced by a large factor as compared to K-AANN model.

The spillover effect, against particular level of offset failure in $[5\% - 50\%]$ range, is averaged over all sensors as follows:

$$\text{Avg. Cross Sensitivity (K\%)} = \frac{\sum_{j=1}^N S_{\text{Cross}}^j(K\%) }{N}, \quad (19)$$

offset failure level $K\% = 5\%, 10\%, 20\% \dots 50\%$.

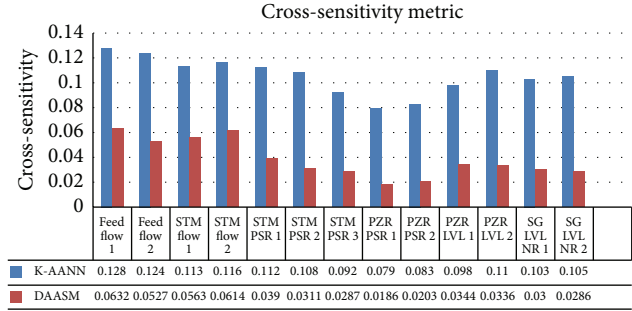


FIGURE 12: Cross-sensitivity values of individual sensors in both models.

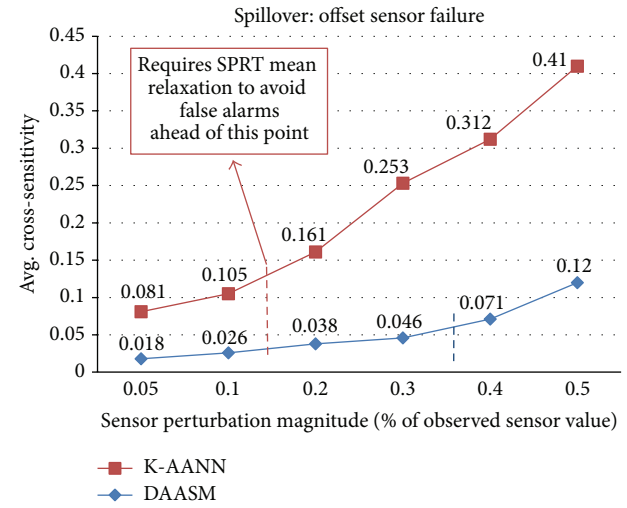


FIGURE 13: Comparison of spillover effects against increasing offset failure.

The cross-sensitivity values $S_{\text{Cross}}^j(K)$, against $K\%$ offset failure level, are calculated using (18). Figure 13 shows the average cross-sensitivity plot for both models. Small cross-sensitivities are observed in DAASM which effectively avoided false alarms in other channels without relaxing the SPRT faulted mean value up to an offset failure of 35–40% in any channel. However, for the case of offset noise larger than 35%, SPRT mean needs to be relaxed to avoid false alarms and isolate the faulty sensor. However, Robustness of K-AANN model deteriorates significantly due to spillover effect beyond 15% offset failure.

Similarly, gross failure scenarios corresponding to two extremities of sensor range can cause severe Spillover effect. To study robustness against gross type failure scenario, a subset of input sensors is simultaneously failed with gross high or low value and average cross-sensitivity of remaining sensor set is calculated using relation (19). Plot in Figure 14 shows that average cross-sensitivity of K-AANN model increases drastically beyond 10% gross failure. However, DAASM resulted in a very nominal spillover, even in case of multiple sensor failure. The DAASM effectively managed simultaneous gross high or low failures in 25% of total sensor set as compared to 10% in case of K-AANN.

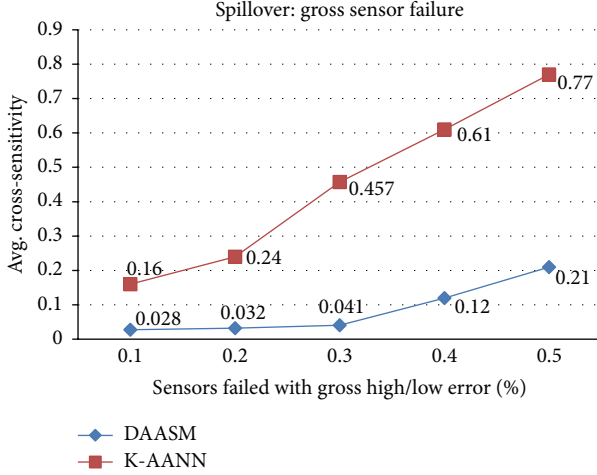


FIGURE 14: Comparison of spillover effect against simultaneous gross high/low failure in multiple sensors.

8.4. Fault Detectability. Fault detectability metric measures the smallest fault that can be detected by integrated sensor estimation and fault detection module as shown in Figure 1 [32]. The detectability metric is measured as percentage of sensor span $D = M/\text{Span}$, where value M corresponds to minimum detectable fault. Minimum fault detectability limit, for each sensor, is quantified through statistical based sequential probability ratio test (SPRT) by Wald [34]. SPRT test is carried out to detect if the residual being generated from normal distribution $N(\mu_1, \sigma^2)$ or $N(\mu_0, \sigma^2)$ as defined for faulty and fault free sensor operations, respectively [35]. Calibration failures are reflected in the mean parameter of residual's distribution. The SPRT procedure is applied to detect changes in the mean of residual's distribution. The application of SPRT requires setting of following parameters value [36]:

μ_0 : normal mode residual mean.

σ^2 : normal mode residual variance.

μ_1 : expected offset in residual mean in abnormal mode.

α : false alarm probability.

β : missed alarm probability.

Under normal mode, the residuals from observed and model estimated sensor values behave as a white Gaussian noise with mean $\mu_0 = 0$. The residual variance σ^2 is estimated for each sensor under normal operating conditions and remained fixed. The false alarm α and missed alarm β probabilities are set to be 0.001 and 0.01, respectively. In order to determine minimum fault detectability limit, a numerical procedure is opted which searches for minimum expected offset μ_1 in the interval $\{\mu_1 : [\sigma-3\sigma]\}$, provided the constraint on missed and false alarm rate holds. σ is the standard deviation corresponding to residual variance of particular sensor. The plot in Figure 15 shows the detectability metric for each sensor. The plot in Figure 15 shows that DAASM can detect

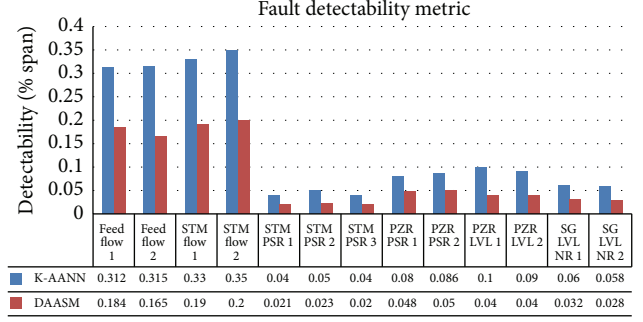


FIGURE 15: Comparison of fault detectability metrics.

faults which are two times smaller in magnitude than those detectable by K-AANN model.

Improvement in fault detectability metric for DAASM can be attributed to observed improvement in model robustness, as suggested by the following relation:

$$\frac{r_i}{\Delta S_i^{\text{drift}}} = (1 - S_{\text{Auto}}^i). \quad (20)$$

The term $r_i/\Delta S_i^{\text{drift}}$ measures the ratio of observed residual to actual sensor drift in terms of autosensitivity. For highly robust model, this ratio reduces to one which means residual reflects the actual drift and results in high fault detectability. Contrarily, ratio value close to zero means that the prediction is following the input and results in poor fault detectability.

8.4.1. SPRT Based Fault Detectability Test. Sequential probability ratio [34, 36] based fault hypothesis test is applied to residual sequence $\{R_i\} = r_{1(t_1)}, r_{1(t_2)} \dots r_{n(t_n)}$ generated by relation $R_i(t_i) = S^{\text{Obs}}(t_i) - S^{\text{Est}}(t_i)$ at time t_i , where $S^{\text{Obs}}(t_i)$ and $S^{\text{Est}}(t_i)$ are the actual and model predicted sensor values, respectively. The SPRT procedure analyzes whether the residual sequence is more likely to be generated from a probability distribution that belongs to normal mode hypothesis H_0 or abnormal mode hypothesis H_1 by using likelihood ratio as follows:

$$L_n = \exp^{[-(1/2\sigma^2)(\sum_{i=1}^n \mu_1(\mu_1 - 2r_i))]} \quad (21)$$

For fault free sensor values, the normal mode hypothesis H_0 is approximated by Gaussian distribution with mean $\mu_0 = 0$ and variance σ^2 . Abnormal mode hypothesis H_1 is approximated with mean $\mu_1 > \mu_0$ using the same variance σ^2 . The SPRT index for the positive mean test is finally obtained by taking logarithm of the likelihood ratio in (21) as follows [35]:

$$\begin{aligned} \text{SPRT} &= \ln(L_n) = -\frac{1}{2\sigma^2} \left[\sum_{i=1}^n \mu_1(\mu_1 - 2r_i) \right] \\ &= \frac{\mu_1}{\sigma^2} \sum_{i=1}^n \left(r_i - \frac{\mu_1}{2} \right). \end{aligned} \quad (22)$$

Pressurizer pressure sensor, sampled at a frequency of 10 seconds, is used as a test signal to validate the fault detectability

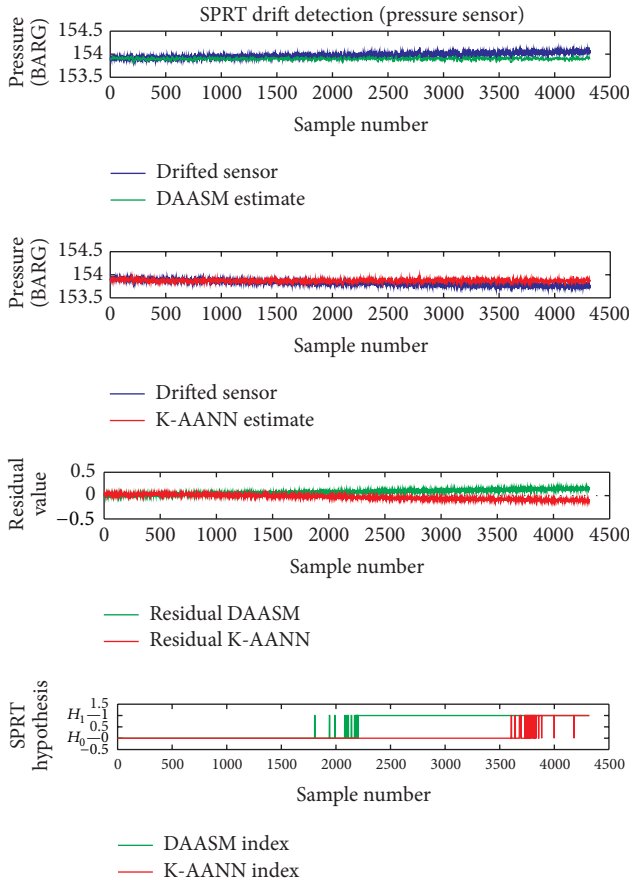


FIGURE 16: SPRT based fault detection in pressurizer pressure sensor.

performance. Two drift faults, at the rate of $+0.01\%/hour$ and $-0.01\%/hour$, are introduced in the test signal for DAASM and K-AANN model’s assessment, respectively. The first and second plots in Figure 16 show drifted and estimated pressure signal from DAASM and K-AANN models, respectively. Third plot shows residual values generated by differencing the drifted and estimated signals from both models. The final plot shows SPRT index values against residuals from K-AANN model and DAASM. The hypotheses H_1 and H_0 correspond to positive and negative fault acceptance, respectively. From SPRT index plot, successful early detection of the sensor drift at 2200th sample, with lag of 6.11 hours since the drift inception, shows that DAASM is more sensitive to small drifts. On the other hand, SPRT index on K-AANN based sensor estimates registered the same drift at 3800th sample with a lag of almost 10.55 hours. The result shows that DAASM is more robust in terms of early fault detection with low false and missed alarm rates.

Finally, both models are tested against five test data sets. Each test set consists of 3630 samples corresponding to different months of full power reactor operation. Both models successfully detected an offset failure of 0.12–0.3 BARG in all steam pressure channels and a drift type failure up to 2.85% in steam generator level (Figure 22). The K-AANN model failed to register a very small drift up to 0.1% in steam flow (STM flow 1) channel. A small drift up to 0.1 BARG is detected in

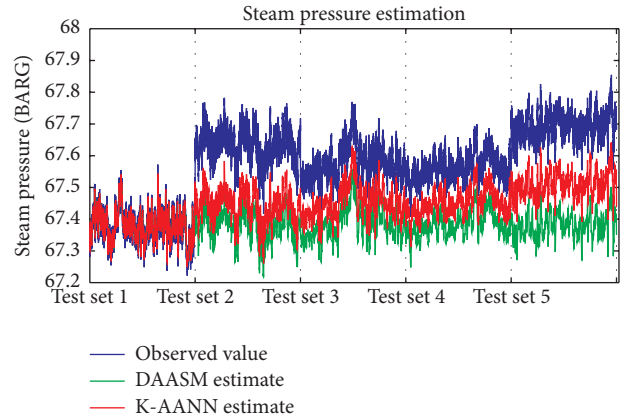


FIGURE 17: Steam pressure estimates against offset failures up to 0.3 (BARG).

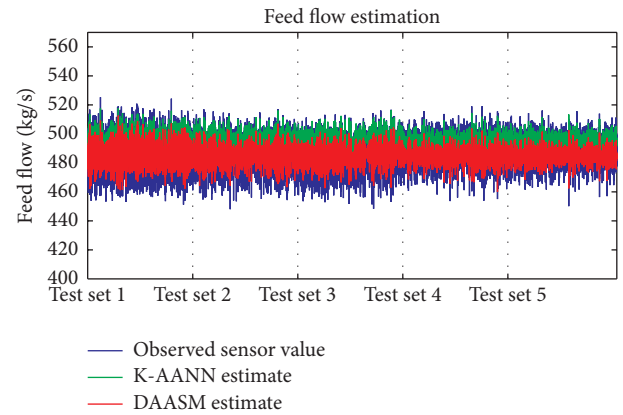


FIGURE 18: Feed flow estimates.

test set 5 of pressurizer pressure channel. However, in case of drift type sensor failures, fault detection lag for DAASM was on average 0.5 times smaller in comparison with K-AANN model. Plots in Figures 17–21 show the estimated sensor values, from both models, on five test data sets of few selected channels.

9. Conclusion

This paper presented a neural network based denoised autoassociative sensor model (DAASM) for empirical sensor modeling. The proposed sensor model is trained to generate a monitoring system for sensor fault detection in nuclear power plants. Multilayer AANN based sensor models may result in suboptimal solutions due to poor regularization by traditional backpropagation based joint multilayer learning procedures. So a complementary deep learning approach, based on greedy layerwise unsupervised pretraining, is employed for effective regularization in the proposed multilayer DAASM. Autoencoder architecture is used for denoising based unsupervised pretraining and regularization of individual layers in the network hierarchy. To address robustness against perturbations in input sensors, data corruption processes exercised during unsupervised pretraining

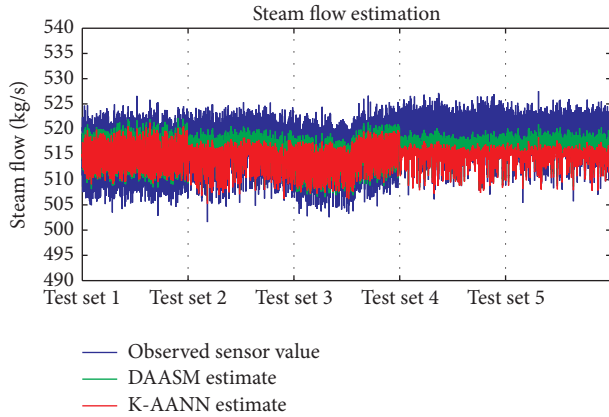


FIGURE 19: Steam flow estimates against drift failure up to 0.1%.

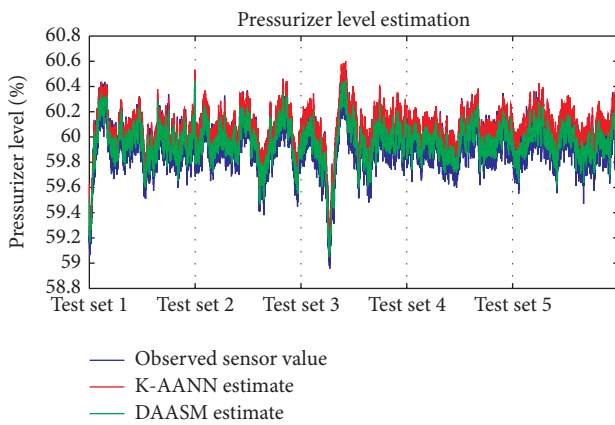


FIGURE 20: Pressurizer level estimates.

phase were based on prior knowledge about different failure scenarios. Results from invariance tests showed that the proposed data corruption schemes were beneficial in learning latent representations at hidden layers and were invariant to multiple levels of perturbation in input sensors. Consequently, these pretrained hidden layers worked as well regularized perturbation filters with increased invariance towards sensor faults. It is also observed that sensitivity against sensor faults decreased significantly towards higher layers in full DAASM assembly. In a practical context of sensor monitoring in nuclear power plants, the proposed model proved its robustness against gross type simultaneous sensor failures. It also showed significant improvement in all performance metrics when compared with popular and widely used five-layered AANN model by Kramer. Moreover, time lag in small drift's detection is significantly reduced. The overall results suggest that greedy layerwise pretraining technique, in combination with domain specific corruption processes, provides a viable framework for effective regularization and robustness in such deep multilayered autoassociative sensor validation models.

Appendix

See Table 3.

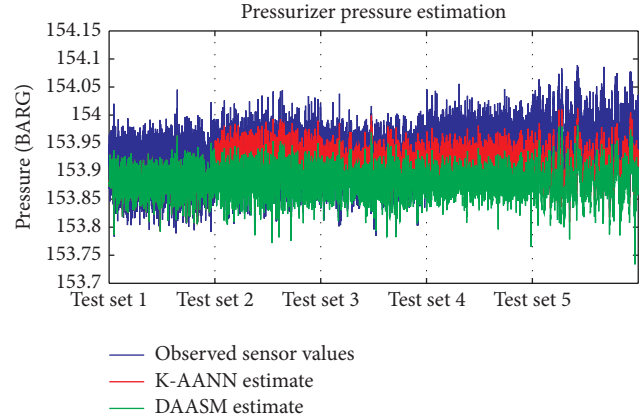


FIGURE 21: Pressurizer pressure estimates against drift failure up to 0.1%.

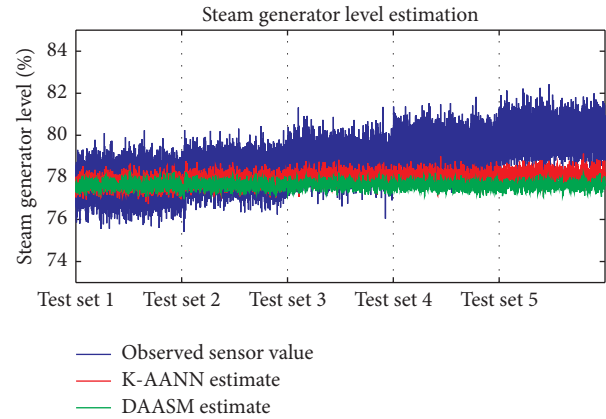


FIGURE 22: Steam generator level estimates against growing drift failure up to 2.87%.

Abbreviations

AANN:	Autoassociative neural network
K-AANN:	Kramer proposed Autoassociative neural network
DAASM:	Denoised autoassociative sensor model
NPP:	Nuclear power plant
PWR:	Pressurized water reactor
S_{Auto} :	Autosensitivity
S_{Cross} :	Cross-sensitivity
DAE:	Denoising autoencoder
S :	Observed sensor value
\hat{S} :	Model predicted sensor value
\tilde{S} :	Corrupted sensor value
SPN:	Salt-and-pepper noise
AGN:	Additive Gaussian noise.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

TABLE 3: Results of DAASM and K-AANN performance metrics.

Performance metric	Model type		Feed flow 1	Feed flow 2	STM flow 1	STM flow 2	STM PSR 1	STM PSR 2	STM PSR 3	PZR PSR 1	PZR PSR 2	PZR LVL 1	PZR LVL 2	SG LVL NR 1	SG LVL NR 2
	AANN	Sp-DAANN													
Accuracy (%) span)	✓		0.382	0.361	0.343	0.412	0.186	0.211	0.166	0.386	0.411	0.243	0.223	0.512	0.621
		✓	0.281	0.253	0.246	0.293	0.121	0.132	0.122	0.243	0.315	0.173	0.156	0.394	0.465
			0.273	0.294	0.321	0.332	0.205	0.182	0.198	0.253	0.233	0.225	0.231	0.268	0.301
Autosensitivity	✓		0.153	0.167	0.187	0.163	0.112	0.079	0.082	0.131	0.071	0.14	0.117	0.133	0.187
		✓	0.128	0.124	0.113	0.116	0.112	0.108	0.092	0.079	0.083	0.098	0.11	0.103	0.105
			0.0632	0.0527	0.0563	0.0614	0.039	0.0311	0.0287	0.0186	0.0203	0.0344	0.0336	0.03	0.0286
Cross-sensitivity	✓		0.312	0.315	0.33	0.35	0.04	0.05	0.04	0.08	0.086	0.1	0.09	0.06	0.058
		✓	0.184	0.165	0.19	0.2	0.021	0.023	0.02	0.048	0.05	0.04	0.04	0.032	0.028

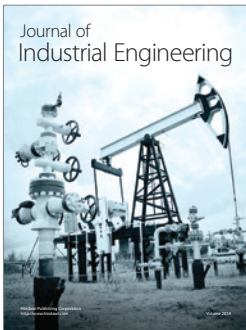
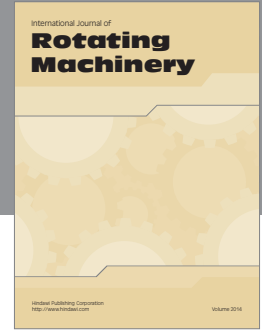
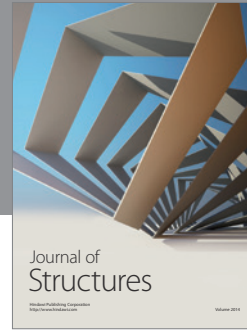
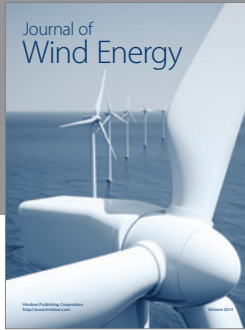
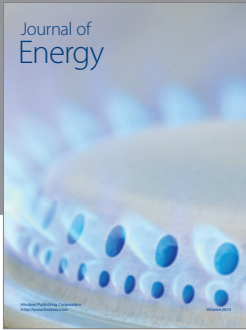
Acknowledgments

The authors acknowledge the support of Mr. José Galindo Rodríguez affiliated to TECNATOM Inc., Spain, and Mr. Chad Painter (Director of Nuclear Power Plant Simulator Development and Training Program at International Atomic Energy Agency) for providing necessary tools and data to conduct this research.

References

- [1] N. Mehranbod, M. Soroush, M. Piovoso, and B. A. Ogunnaike, "Probabilistic model for sensor fault detection and identification," *AIChE Journal*, vol. 49, no. 7, pp. 1787–1802, 2003.
- [2] J. W. Hines and E. Davis, "Lessons learned from the U.S. nuclear power plant on-line monitoring programs," *Progress in Nuclear Energy*, vol. 46, no. 3-4, pp. 176–189, 2005.
- [3] EPRI, "On-line monitoring cost benefit guide," Final Report 1006777, EPRI, Palo Alto, Calif, USA, 2003.
- [4] S. J. Qin and W. Li, "Detection, identification, and reconstruction of faulty sensors with maximized sensitivity," *AIChE Journal*, vol. 45, no. 9, pp. 1963–1976, 1999.
- [5] J. Ma and J. Jiang, "Applications of fault detection and diagnosis methods in nuclear power plants: a review," *Progress in Nuclear Energy*, vol. 53, no. 3, pp. 255–266, 2011.
- [6] G. Y. Heo, "Condition monitoring using empirical models: technical review and prospects for nuclear applications," *Nuclear Engineering and Technology*, vol. 40, no. 1, pp. 49–68, 2008.
- [7] J. B. Coble, R. M. Meyer, P. Ramuhalli et al., "A review of sensor calibration monitoring for calibration interval extension in nuclear power plants," Tech. Rep. PNNL-21687, Pacific Northwest National Laboratory, Richland, Wash, USA, 2012.
- [8] M. A. Kramer, "Autoassociative neural networks," *Computers and Chemical Engineering*, vol. 16, no. 4, pp. 313–328, 1992.
- [9] X. Xu, J. W. Hines, and R. E. Uhrig, "On-line sensor calibration monitoring and fault detection for chemical processes," in *Proceedings of the Maintenance and Reliability Conference (MARCON '98)*, pp. 12–14, Knoxville, Tenn, USA, May 1998.
- [10] M. Hamidreza, S. Mehdi, J.-R. Hooshang, and N. Aliakbar, "Reconstruction based approach to sensor fault diagnosis using auto-associative neural networks," *Journal of Central South University*, vol. 21, no. 6, pp. 2273–2281, 2014.
- [11] U. Thissen, W. J. Melssen, and L. M. C. Buydens, "Nonlinear process monitoring using bottle-neck neural networks," *Analytica Chimica Acta*, vol. 446, no. 1-2, pp. 371–383, 2001.
- [12] D. J. Wrest, J. W. Hines, and R. E. Uhrig, "Instrument surveillance and calibration verification through plant wide monitoring using autoassociative neural networks," in *Proceedings of the American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies*, University Park, Pa, USA, May 1996.
- [13] J. W. Hines, R. E. Uhrig, and D. J. Wrest, "Use of autoassociative neural networks for signal validation," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 21, no. 2, pp. 143–154, 1998.
- [14] P. F. Fantoni, M. I. Hoffmann, R. Shankar, and E. L. Davis, "On-line monitoring of instrument channel performance in nuclear power plant using PEANO," *Progress in Nuclear Energy*, vol. 43, no. 1–4, pp. 83–89, 2003.
- [15] M. Marseguerra and A. Zoia, "The AutoAssociative Neural Network in signal analysis: II. Application to on-line monitoring of a simulated BWR component," *Annals of Nuclear Energy*, vol. 32, no. 11, pp. 1207–1223, 2005.
- [16] M. S. Ikbal, H. Misra, and B. Yegnanarayana, "Analysis of autoassociative mapping neural networks," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '99)*, vol. 5, pp. 3025–3029, IEEE, Washington, DC, USA, July 1999, Proceedings (Cat. no. 99CH36339).
- [17] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–27, 2009.
- [18] J. W. Hines, A. V. Gribok, I. Attieh, and R. E. Uhrig, "Regularization methods for inferential sensing in nuclear power plants," in *Fuzzy Systems and Soft Computing in Nuclear Engineering*, D. Ruan, Ed., vol. 38 of *Studies in Fuzziness and Soft Computing*, pp. 285–314, Physica, 2000.
- [19] A. V. Gribok, J. W. Hines, A. Urmanov, and R. E. Uhrig, "Heuristic, systematic, and informational regularization for process monitoring," *International Journal of Intelligent Systems*, vol. 17, no. 8, pp. 723–749, 2002.
- [20] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS '10)*, vol. 9, pp. 249–256, Sardinia, Italy, May 2010.
- [21] S. Tan and M. L. Mayrovouniotis, "Reducing data dimensionality through optimizing neural network inputs," *AIChE Journal*, vol. 41, no. 6, pp. 1471–1480, 1995.
- [22] Y. Bengio and P. Lamblin, "Greedy layer-wise training of deep networks," in *Proceedings of the Advances in Neural Information Processing Systems 19 (NIPS '07)*, pp. 153–160, September 2007.
- [23] H. Larochelle, Y. Bengio, J. Lourador, and P. Lamblin, "Exploring strategies for training deep neural networks," *The Journal of Machine Learning Research*, vol. 10, pp. 1–40, 2009.
- [24] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing [exploratory DSP]," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2011.
- [25] D. Erhan, A. Courville, and P. Vincent, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.
- [26] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*, pp. 1096–1103, July 2008.
- [27] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.
- [28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [29] M. J. Embrechts, B. J. Hargis, and J. D. Linton, "An augmented efficient backpropagation training strategy for deep autoassociative neural networks," in *Proceedings of the 18th European Symposium on Artificial Neural Networks—Computational Intelligence and Machine Learning (ESANN '10)*, vol. no, pp. 28–30, April 2010.
- [30] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, vol. 7700 of *Lecture Notes in Computer Science*, pp. 437–478, Springer, Berlin, Germany, 2nd edition, 2012.

- [31] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
- [32] J. W. Hines and D. R. Garvey, "Development and application of fault detectability performance metrics for instrument calibration verification and anomaly detection," *Journal of Pattern Recognition Research*, vol. 1, no. 1, pp. 2–15, 2006.
- [33] A. Usynin and J. W. Hines, "On-line monitoring robustness measures and comparisons," in *Proceedings of the International Atomic Energy Agency Technical Meeting on "Increasing Instrument Calibration Interval through On-Line Calibration Technology"*, OECD Halden Reactor Project, Halden, Norway, September 2004.
- [34] A. Wald, "Sequential tests of statistical hypotheses," *Annals of Mathematical Statistics*, vol. 16, no. 2, pp. 117–186, 1945.
- [35] F. Di Maio, P. Baraldi, E. Zio, and R. Seraoui, "Fault detection in nuclear power plants components by a combination of statistical methods," *IEEE Transactions on Reliability*, vol. 62, no. 4, pp. 833–845, 2013.
- [36] S. Cheng and M. Pecht, "Using cross-validation for model parameter selection of sequential probability ratio test," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8467–8473, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

