Authors Information

Authors Name	Email- Address	Affiliation	Address
Ahmad Shaheryar	shaheryar011@yahoo.com	University of Science and Technology Beijing.	USTB, 30 Xue-yuan Road, Haidian District, POB 100083, Beijing , China
Dr. Xu-Cheng Yin	xuchengyin@ustb.edu.cn	University of Science and Technology Beijing.	USTB, 30 Xue-yuan Road, Haidian District, POB 100083, Beijing , China
Dr. Hong-Wei Hao	hongwei.hao@ia.ac.cn	Institute of Automation, Chinese Academy of Sciences	Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun East Road, Beijing 100190, P.R. China
Dr. Hazrat Ali	engr.hazratali@yahoo.com	Department of Electrical Engineering, COMSATS Institute of Information Technology Abbottabad.	Department of Electrical Engineering, COMSATS Institute of Information Technology Abbottabad.Pakistan
Dr. Khalid Iqbal	khalidiqbal@ciit-attock.edu.pk	Department of Computer Science, COMSATS Institute of Information Technology Attock.	COMSATS Institute of Information Technology Near Officers colony, Kamra Road Attock, Pakistan.

TITLE

A Denoising based Auto-Associative model for robust sensor monitoring in Nuclear Power Plant.

ABSTRACT

Sensors health monitoring is essentially important for safe and reliable functioning of safety-critical chemical and nuclear power plants. Auto-associative neural network (AANN) based empirical sensor models have widely been reported in context of calibration monitoring. However, good generalization and effective robustness are the prime issues to be handled carefully during training of such ill-posed data driven models. To address above mentioned issues, several regularization heuristics such as training with jitter, weight decay, cross-validation Bayesian regularization etc. are suggested in literature. Apart from these regularization heuristics; traditional error gradient based supervised learning algorithms, for deep multilayered AANN models, are highly susceptible of being trapped in local optimum; hence constrain the prediction performance. In order to address poor regularization and robust learning issues, here, we propose a denoised auto-associative sensor model (DAASM) based on deep learning framework. The proposed sensor model is composed and regularized under denoising based learning objective. Full DAASM model comprises of multiple hidden layers which are pre-trained greedily in an unsupervised fashion under denoising autoencoder (DAE) architecture. In order to improve robustness, dropout heuristic and domain specific data corruption processes are exercised during unsupervised pre-training phase. The proposed sensor model is trained and tested on sensor data from a PWR type nuclear power plant. Accuracy, auto-sensitivity, Spillover and fault detectability metrics are used for performance assessment and comparison with extensively reported five layer AANN model by Kramer. Sensor fault detection and isolation competency is evaluated by employing residual based sequential probability ratio test (SPRT).

Keywords

Sensor Fault Detection, Calibration Monitoring, Process Monitoring, Nuclear Power Plants, Auto-Associative Neural Network, Sensor Health Monitoring, Auto-encoder Networks, Sensor Validation, Auto-Associative Sensor Models, Denoised Auto-Encoder, Online Monitoring, OLM, Condition Monitoring.

Abbreviations

AANN	Auto-Associative Neural Network.
K-AANN	Kramer proposed Auto-Associative Neural Network
DAASM	Denoised Auto Associative Sensor Model.
NPP	Nuclear Power Plant.
PWR	Pressurized Water Reactor.
S_{Auto}	Auto-Sensitivity.
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.

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 automatic controller, depends on correct plant state reflected by its sensors. "Defense in depth" [1]safety concept for such mission critical processes essentially requires a sensor health monitoring system. Such sensor health monitoring system have 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 Fig. 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 either based on the first principles models (e.g. energy conservation, material balance etc.) or history based data driven models[5]. However, sensor modeling using empirical techniques from statistics and artificial intelligence are an active area of research [6][7].

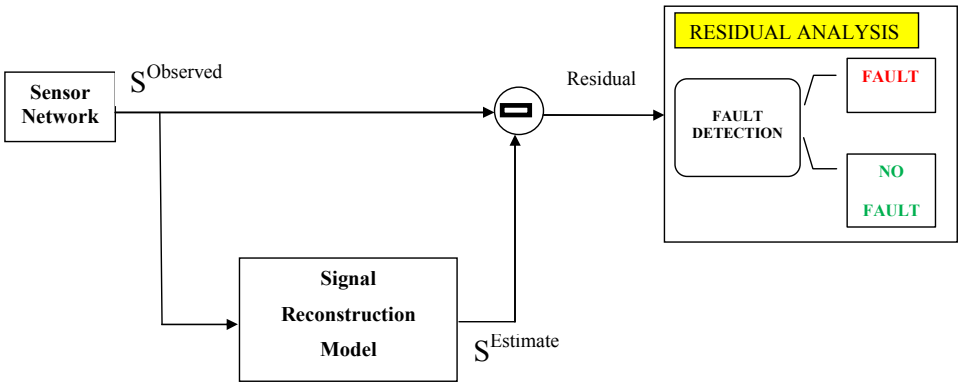


Figure 1: Integrated sensor estimation and fault detection architecture.

[1] Defense in depth safety concept requires mission critical systems to be redundant and diverse in implementation to avoid single mode failure scenerios.

In order to model complex non-linearity in physical process sensors, auto-associative neural network based sensor models had widely been used and reported for calibration monitoring in chemical processes [8][9][10][11]and nuclear power plants [12][13][14][15]. Data driven training procedures for such neural network based sensor models discover the underlying statistical regularities among input sensors from history data and tries to models them by adjusting network parameters. Five layers AANN is one of the earliest auto-associative architectures proposed for sensor and process modeling[8].

In contrast to shallow single layered architectures , these multi-layered Neural architectures have flexibility for modeling complex non-linear functions [16][17].However, harnessing the complexity offered by these deep NN models without over fitting 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 of all these regularization heuristics, the joint learning of multiple hidden layers via back propagation 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 get even more complex and hard due to inherently noisy and co-linearity 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.

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 architecture without being compromised on generalization and robustness of the sensor model. Recent research on greedy layer wise learning approaches [22][23] have been found successful for efficient learning in deep multilayered Neural architectures for image, speech and Natural language processing [24]. So, for a multilayered DAASM model, 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 layer wise pre-training approach. Following the deep learning framework, each layer in the proposed DAASM model is regularized individually through unsupervised pre-training 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 [25][17].
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 exercise during denoising based unsupervised pre-training phase. The proposed corruption process is derived from drift, additive and gross type failure scenarios as elaborated in section 4.2. Robustness to faulty sensor conditions is an implicit process of denoising based unsupervised pre-training phase. Robustness of the proposed DAASM model, 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 layer wise pre-training 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, auto-sensitivity, 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 $\varphi(.)$ as follow:

$$\tilde{S}_{Obs} = \varphi(S_{True})$$

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.

$$f = arg\min_{f} E_{S_{True}} \left\| f\left(\tilde{S}_{Obs}\right) - S_{True} \right\|_2^2 \quad (1)$$

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

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 sub-optimal solutions with poor generalization. Joint learning of multiple hidden layers via back propagation of error gradient inherently suffers from gradient vanishing problem at the earlier layers, hence, constrain 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 sub-networks and use unsupervised learning to minimize input reconstruction error. This technique successfully eliminates the short-comings of the gradient based learning by averting the local minima. Deep learning framework employs a systematic three step training approach as follows.

1. Pre-training 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.

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 pre-training and composition of Deep Auto-associative 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 Fig. 2. Hence, implicitly captures the underlying data generating distribution by exploring robust statistical regularities in input data. A typical DAE architecture, as depicted in Fig. 3, comprise of 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. While, the red filed units in input layer are proxies to clean input units which are randomly selected for corruption under some artificial noise process. $L(x, \hat{x})$ is an empirical loss function to be optimized during training process.

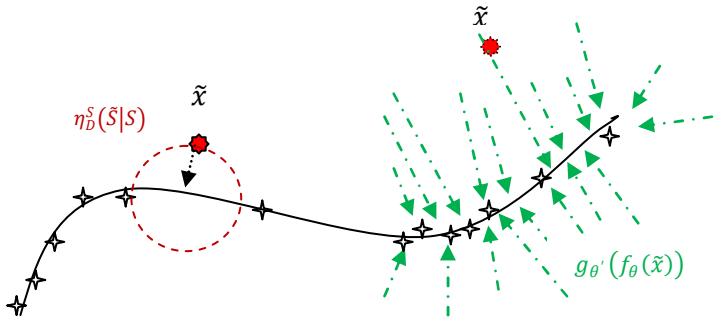


Figure 2: Suppose training data (✦) concentrate near a low-dimensional manifold. Corrupted examples (●) obtained by applying corruption process $\eta_D^S(\tilde{S}|S)$ will lie farther from the manifold. The model learns $g_{\theta'}(f_\theta(\tilde{x}))$ with $P(X|\tilde{X})$ to "project them back"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

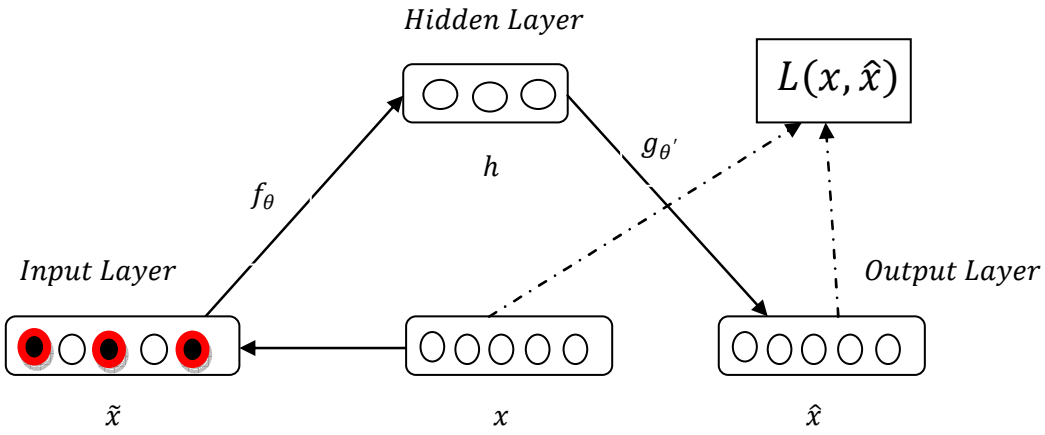


Figure 3: Basic Denoising Auto Encoder (DAE) scheme. An empty circle depicts a single neuron. A filled circle depicts corrupted units in input vector.

Let x_i be the original data vector with i = 1, 2, ..., 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 Fig 3 are defined as.

$$h(\tilde{x}_i) = f_\theta(\tilde{x}) = \sigma(W\tilde{x}_i + b)$$

$$\hat{X}(\tilde{x}_i) = g_{\theta'}(h) = \sigma(W'h(\tilde{x}_i) + b')$$

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. $\hat{X}(\tilde{x}_i)$ is an approximate reconstruction of x obtained through decoder function $g_{\theta'}(h)$ through reverse mapping followed by sigmoid activation at output layer. While, $\theta = \{\theta, \theta'\} = \{W, b, W', b'\}$ are the weight and bias parametres corresponding to these encoder and decoder function.

In relation to sensor reconstruction model as formulated in section 2, the above described DAE can be re- interpreted as follow.

$$x \sim S = \{s_i\}_{i=1}^{n}$$

$$\tilde{x} \sim \tilde{S} = \eta_D((\tilde{s}|s))$$

$$\hat{x} \sim \hat{S} = \eta_D^{-1}(\tilde{S})$$

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 $\{s_i\}$. $\hat{S}$ are the estimated sensor values by reconstruction function learnt on clean and corrupted input S and $\tilde{S}$. Network parameters θ , for DAE, can be learned in an unsupervised setting through minimization of the reconstruction loss in equation 2 as follow.

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

4. DAASM Architecture and Regularization

In order to capture complex nonlinear relationships among input sensors, a multilayered architecture is proposed for Denoised Auto-Associative Sensor Model (DAASM). Individual layers in network hierarchy are pre-trained successively from bottom to top. For a well regularized sensor model, the structure and optimization objective in greedy layer wise pre-training plays a crucial role. Two heuristics are applied for robust learning in DAASM as follow.

1. Each successive layer in multi-layered DAASM assembly is pre-trained in an un-supervised fashion under denoising based auto-encoder (DAE) as elaborated in section 3.
2. To address robustness, data corruption processes for denoising based pre-training task are incorporated with domain specific failure scenarios which are derived from different types of sensor faults. These heuristics serve several purposes.
    - 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 over complete hidden layer setting.
    - Denoising procedure during pre-training leads to latent representations that are robust to input perturbations.
    - Addition of corrupted data set increases training set size and thus useful in alleviating over fitting problem.

Full DAASM model is learnt in two stages 1) an unsupervised pre-training phase. 2) A supervised fine tuning phase. As shown in fig.4 , the pre-training 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 pre-trained DAEs followed by a supervised fine tuning phase. For each DAE in the stack hierarchy, the optimization objective for unsupervised pre-training will remain same as in relation 2. However, weight decay regularization term is added to the loss function which constrains network complexity by penalizing large weight values. In relation 3, $\{W, W'\}$ are the network weight parameters corresponding to encoder and decoder function while λ is the weight decay hyper-parametre.

$$L(S, \hat{S}\,;\,\theta) = \frac{1}{N}\sum_{i=1}^{N}\left\|S - \hat{S}\right\|_2^2 + \frac{\lambda}{2}\left(\|W\|^2 + \|W'\|^2\right) \tag{3}$$

In a typical DAE architecture, number of input and output layer neuron 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 have tendency to entangle information.(i-e. change in a single aspect of the input translates into significant changes in all components of the hidden representation[27](Vincent et al., 2008). 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 over complete hidden layer setting$(d' > d)$. Under over complete setting, denoising based optimization objective acts as a strong regularizer and inherently prevents DAE from learning identity function.

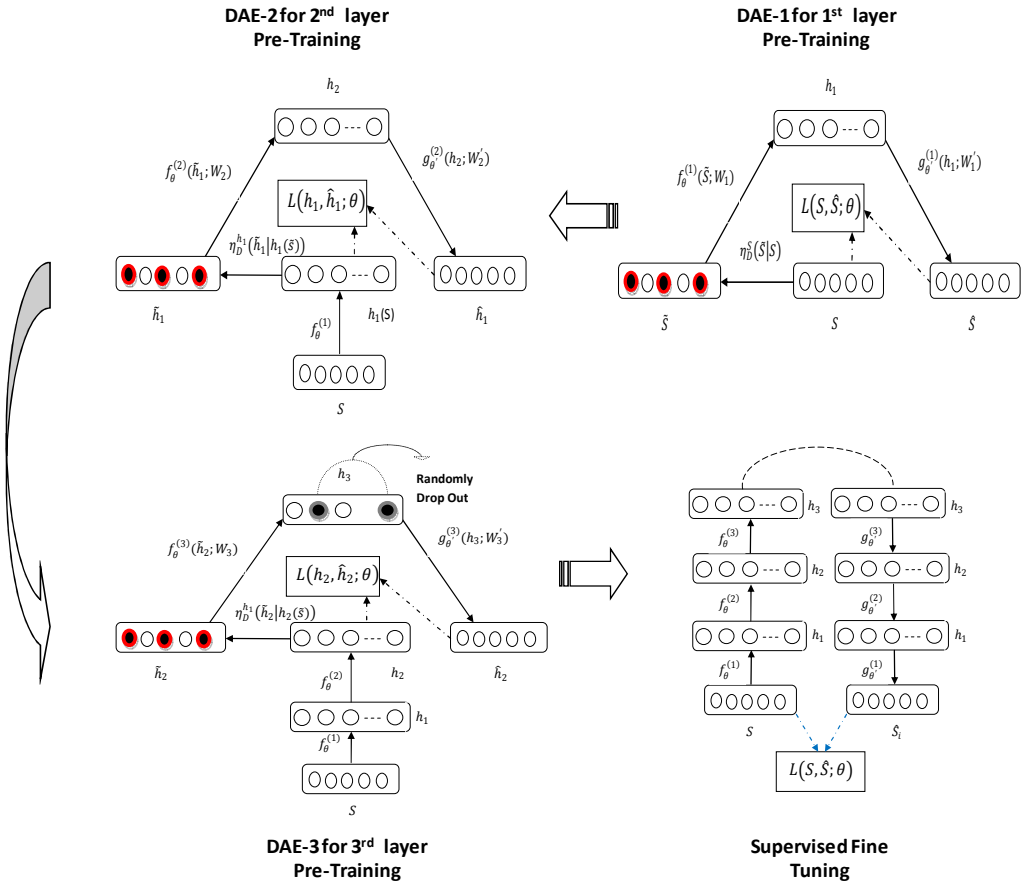


Figure 4: DAASM Architecture and greedy Learning procedure. Greedy layerwise pre-training procedure is depicted by counter clockwise flow in the figure.

Anti-clockwise flow in fig. 4 shows architecture and greedy layer by layer unsupervised pre-training procedure for all hidden layers in DAASM stack. For each hidden layer h_l, a DAE block is shown; in which an encoder function $f_\theta^l(.)$ and a decoder function $g_\theta^l(.)$ are learnt by minimizing the loss function corresponding to fault free reconstruction of the inputs as in relation 3. For the case of first hidden layer h_1, the corresponding DAE-1 is trained directly on sensor data using $L(S, \hat{S}, \theta)$ loss function in equation 3. However, hidden layers h_2 through h_3 are learnt on data from preceding hidden layer activations using recursive relation in eq. 4. So the loss function corresponding to DAE-1 and DAE-2 can be re-presented as $L(h_l, \hat{h}_l, \theta)$ where $\hat{h}$ is an approximate reconstruction of h.

$$h^l = f_\theta^l(h_{l-1}; W_l) = sigm(W^l\, h^{l-1} + b^l)\ ; \qquad 1 \le l \le L = 3 \tag{4}$$

$\langle W \rangle$ 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 $\eta_D^{h_2}(\tilde{h}_1|h_1(\tilde{s}))$ employs a corruption process in which $h_1(S)$ and $h_1(\tilde{S})$ from pre-trained DAE-1 will be used as the clean and noisy input for DAE-2 pre-training. Finally, an additive Gaussian type corruption process $(\mathrm{AGN}): \tilde{\mathrm{x}}|\mathrm{x} \sim \mathrm{N}(\mathrm{x}, \sigma^2)$ 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 under section 4.2.

These pre-trained 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 pre-training. Random dropouts make it hard for latent representations at h_3 to get specialize on particular sensors in the input set. Finally, pre-trained DAE's are unfolded into a deep auto-associator network with L number of encoder and $L-1$ decoder cascade as shown in unsupervised fine tuning phase in Fig 3. The final network comprise of one input layer, one output and $2L-1$ hidden layers. The input sensor values flow through encoder cascade $f = f_\theta^l \, o \, f_\theta^{l-1}$ o … f_θ^1 using recursive expression in equation 4 and a decoder cascade $g = g_{\theta'}^1, o \, g_{\theta'}^{l+1}$ o … $g_{\theta'}^{L-1}$ using following equations.

$$\hat{h}^L = h^L \tag{5}$$

$$\hat{h}^l = g_{\theta'}^l(h_{l+1}; W_l') = sigm\left(W_l' \, \hat{h}^{l+1} + b_l'\right) \; ; \qquad 1 \le l \le L-1 = 2 \tag{6}$$

$$\hat{S} = g^0\left(\hat{h}^1\right) = \; W_0' \, \hat{h}^1 + b_0' \tag{7}$$

Where $\langle W', b' \rangle$ are network weights and biases of the decoder part in DAE. The entire network is fine tuned using a semi heuristic based **"Augmented Efficient Back-propagation algorithm"** , Proposed by M. J. Embrechts et.al [29] , with following minimization objective:

$$L(S, \hat{S}\,;\, \theta) = \frac{1}{N}\sum_{\mathrm{i}=1}^{\mathrm{N}} \left\| S - \hat{S} \right\|_2^2 + \frac{\lambda}{2}\sum_{\mathrm{k}=1}^{2\mathrm{L}} \|\mathrm{W_k}\|_2^2 \tag{8}$$

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

4.2. Corruption Process $\eta(.)$ for Invariance

For the case of calibration monitoring, an ideal DAASM model should learn encoder and decoder functions which are invariant to failed sensor states. So during DAE based pre-training phase, engineered transformations from prior knowledge about the involved failure types are imposed on clean input. Different data corruption processes $\eta(.)$ are devised for learning of each successive hidden layer.

Denoising based learning objective drives the hidden layer mapings to get invariant against such engineered transformations on input data. It is important to understand that denoising based learning approach do not correct the faulty signal explicitly rather seek to extract statistical structure among input signals which is stable and invariant under faults. Hence implicitly estimates correct value for faulty signal. Two failure types are identified and defined as follow.

- ***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.
- ***Miscalibration Sensor Failure:***it includes drift, multiplicative and outlier type sensor failures and is modeled through isotropic Gaussian noise(GS): $\tilde{\mathrm{x}}|\mathrm{x} \sim \mathrm{N}(\mathrm{x}, \sigma^2)$. 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. Hence, provide 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 pre-training as follow.

$$\eta_D^S(\tilde{S}|S) = SPN\left\{\begin{array}{c} \tilde{s}_i = s_i \;\; where\; i \notin j \\ \tilde{s}_i = k\; where\; i \in j \\ k = \{0,1\}\; with \begin{Bmatrix} \Pr(0) = \frac{1}{2} \\ \Pr(1) = \frac{1}{2} \end{Bmatrix} \\ \cup_{i=1}^{i!=\nu n}\{j_{i\ldots.N=\nu n}\}\; where\; j_i = [1,n] = rand() \\ \nu = \; input\; fraction\; and\; n = input\; dimension \end{array}\right\} \tag{9}$$

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 affects 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 pre-trained DAE-1 are proposed as the clean and noisy input for DAE-2 pre-training. 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.

$$\eta_D^{h_2}(\tilde{h}_1|h_1(\tilde{s})) = \left\{\begin{array}{c} \tilde{h}_1^i = h_1^i(\tilde{S}) \\ where\;\; \tilde{S} = \eta_D^S(\tilde{S}|S) = SPN \end{array}\right\} \tag{10}$$

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 pre-training. The corruption procedure corrupts the h_2 hidden layer mappings, against clean sensors at the input layer as $h_2\big(h_1(S)\big)$,by employing an isotropic gaussian noise as follow.

$$\eta_D^{h_3}\big(\tilde{h}_2\big|h_2(\tilde{S})\big) = AGN\left\{\begin{array}{c} \tilde{h}_2^i = h_2^i\left(h_1^i(S)\right) \;\; where\; i \notin j \\ \tilde{h}_2^i\big|h_2^i \sim N(s,\sigma^2 I)\; where\; i \in j \\ \\ \cup_{i=1}^{i!=vn}\{j_{i\ldots N=vn}\}\; where\; j_i = [1,n] = rand() \\ where\; v = \; input\; fraction\; and\; n = total\; inputs \end{array}\right\} \tag{11}$$

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

5. Data Set Description

Intentionally, for study purposes, we limited the modeling scope of DAASM to full power steady operational state. It's the common state in which NPP operates from one refueling to the next. However, in practice it's not possible for NPP systems to be in perfect steady state. Reactivity induced power perturbations; natural process fluctuations, sensor and controller noises etc. 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. So 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 month of full power reactor operations after refueling cycle. While 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. Fig 5 shows the spatial distribution of the selected sensors.

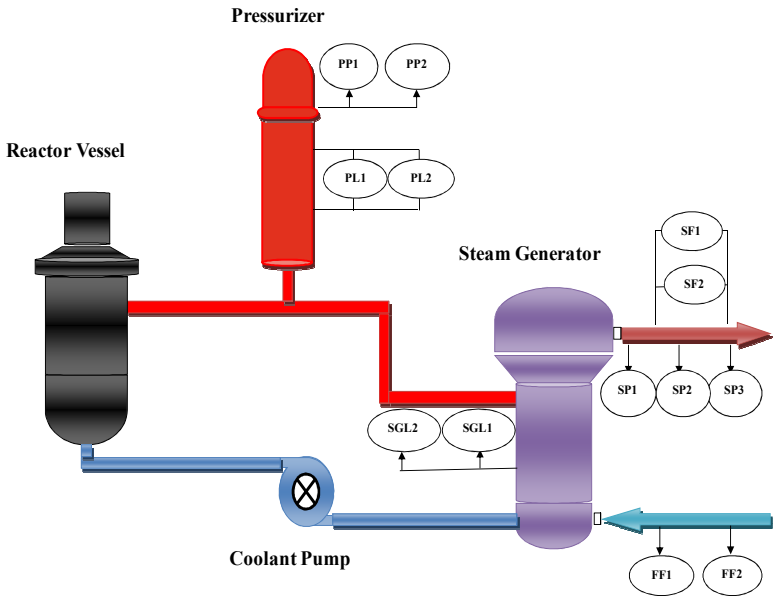


Figure 5: Spatial distribution of selected Sensor Set.

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

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

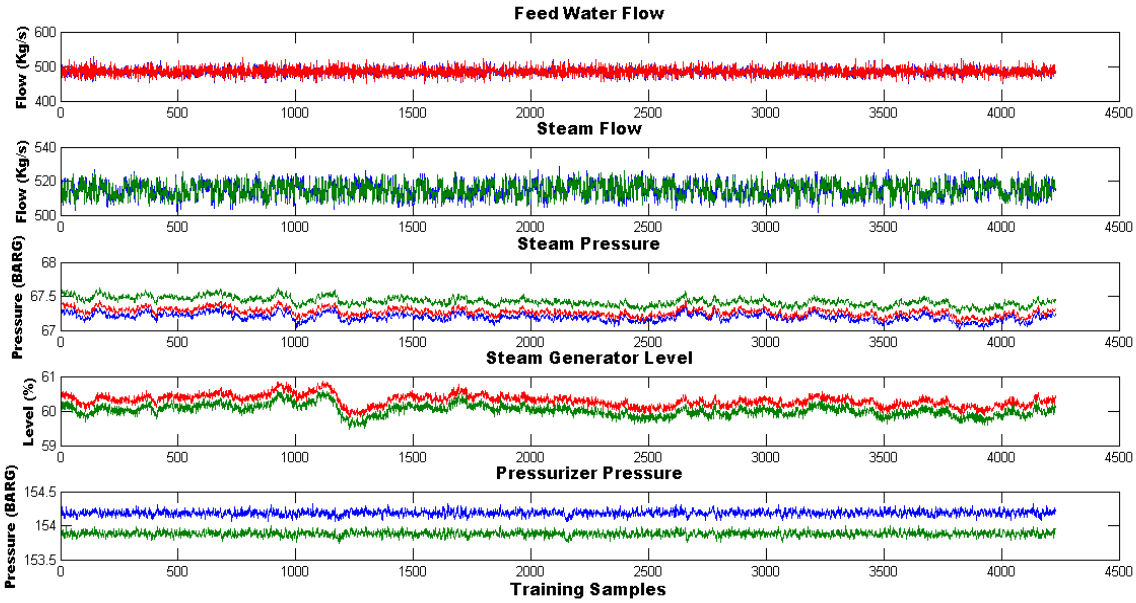


Figure 6: Plot of NPP sensors listed in table 1.

6. Model Training

NPP sensor's data is divided into training, test and validation set. Each sensor data is scaled in 0.1 to 0.9 ranges by using lower and upper extremities corresponding to individual sensor. However, the value 0 and 1 is explicitly reserved for gross and saturation type sensor failures, respectively. Training data consists of 4320 samples from full power steady state reactor operation. While test and validation data is used for sensor model optimization and performance evaluation, respectively. The training setup for DAASM employs two learning stages, an unsupervised learning phase and supervised training phase. DAE based Greedy layer wise pre-training of each hidden layer, as described in section 4, is performed using mini-batches from training data set .Stochastic gradient descent based learning algorithm is employed as suggested in Practical training Recommendations by [30].Finally, standard back propagation algorithm is employed for supervised fine tuning in fully stacked DAASM model in Fig. 4. Supervised training is performed using clean sensor input only. The model hyper parameters are set by random grid search method[31].A summary of the training Hyper –parameters corresponding to optimum DAASM model are summarized in Table 2 .

Table 2: Summary of DAASM Hyper parameters

Hyper parameter Type	Tested Hyper-parameter values	Successful Hyper parameter against Optimum model
Pre-trained 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 Pre-training.	$\{0.1, 5\times10^{-2}, 1\times10^{-2}, 5\times10^{-3}$ $1\times10^{-3}, 5\times10^{-4}\}$	$[5\times10^{-2}, 1\times10^{-2}]$
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 pre-Training error for each hidden layers	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

7. Invariance Test for Robustness

A layer by layer invariance study is conducted to test the robustness of fully trained DAASM model against failed sensor states. Data corruption processes applied during pre-training 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 pre-training phase. Invariance, for hidden layer mappings h_l, is quantified through mean square error (MSE) between Euclidean (L2) normalized hidden layer activation $\frac{\langle h_i \rangle_n}{\|\langle h_i \rangle_n\|_2}$ and $\frac{\langle \tilde{h}_i \rangle_n}{\|\langle \tilde{h}_i \rangle_n\|_2}$ 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 $\mathrm{D_h}$ and number of test samples $\mathrm{T_N}$ as shown in equation 12. Finally these MSE values are normalized with maximal MSE value as in equation 13. Normalized MSE curves for each successive hidden layer are plotted in fig. 7.

$$MSE(H_l, \%Offset) = \frac{1}{T_N}\sum_{n=1}^{T_N}\left(\frac{\langle h_i\rangle_n}{\|\langle h_i\rangle n\|_2} - \frac{\langle \tilde{h}_i\rangle_n}{\|\langle \tilde{h}_i\rangle n\|_2}\right)^2 \Big/ D_h \tag{12}$$

$$1 \le l \le L = No.of\ encoder\ layers = 3; \quad and\ \ \%offset = 5\%, 10\%, 20\% \ldots 50\%$$

$$MSE_{Normalized}(H_l) = \frac{MSE(H_l, \%Offset)}{MSE(H_l, \%Max-offset)} \tag{13}$$

Layer wise MSE plots, in fig. 7, clearly shows that invariance to faulty sensor conditions increases toward 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$(1 - \mathrm{S^i_{Auto}})$. Auto-sensitivity values $\mathrm{S^i_{Auto}}$ (see section 8.2) are calculated against varying offset failure levels. In Fig. 8, layer wise increase in robustness confirms that increased invariance helps in improving overall model's robustness.

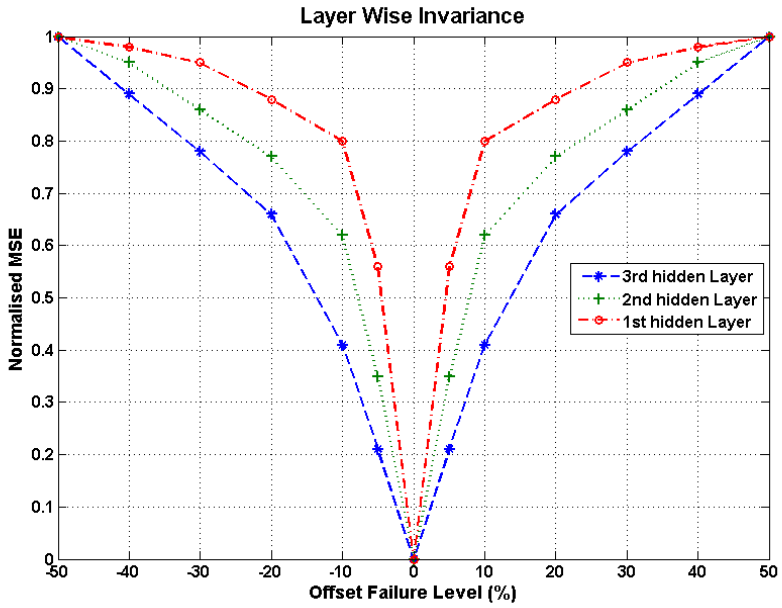


Figure 7: Layer-wise invariance in DAASM. Lower curves depict higher invariance.

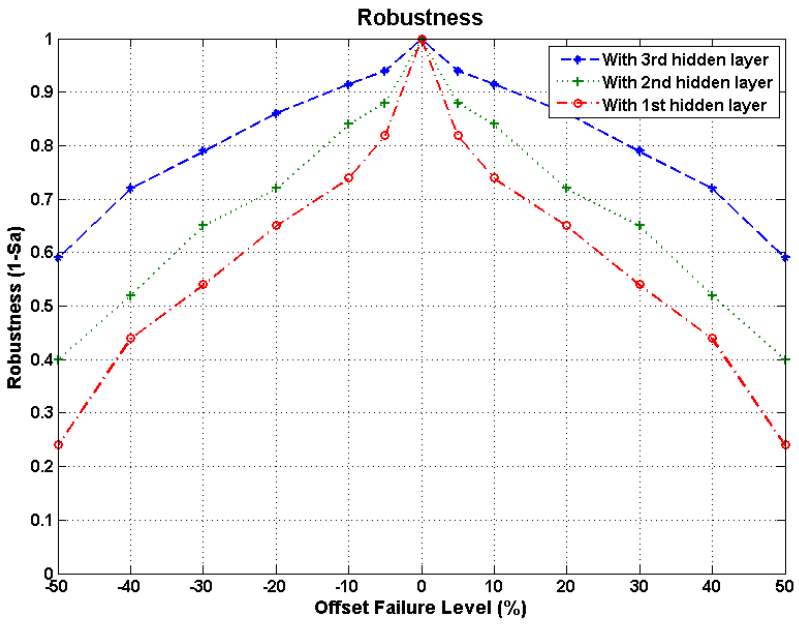


Figure 8: Robustness measure (1-autoSensitivity) at multiple hidden layers in DAASM. Higher curves depict high robustness.

8. DAASM Vs K-AANN Performance Analysis.

Here we will assess and compare the performance of DAASM model 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. Further to improve robustness, training with jitter heuristic is employed by introducing a noise of 10% magnitude on clean sensor input. The following five layer topology 13-17-9-17-13 is found optimum for k-AANN model. Both DAASM and K-AANN models 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 post refueling full power NPP operations. Performance metric values are reported in Table A.1 in appendix.

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.

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{S}_i - S_i\right)^2 \tag{14}$$

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

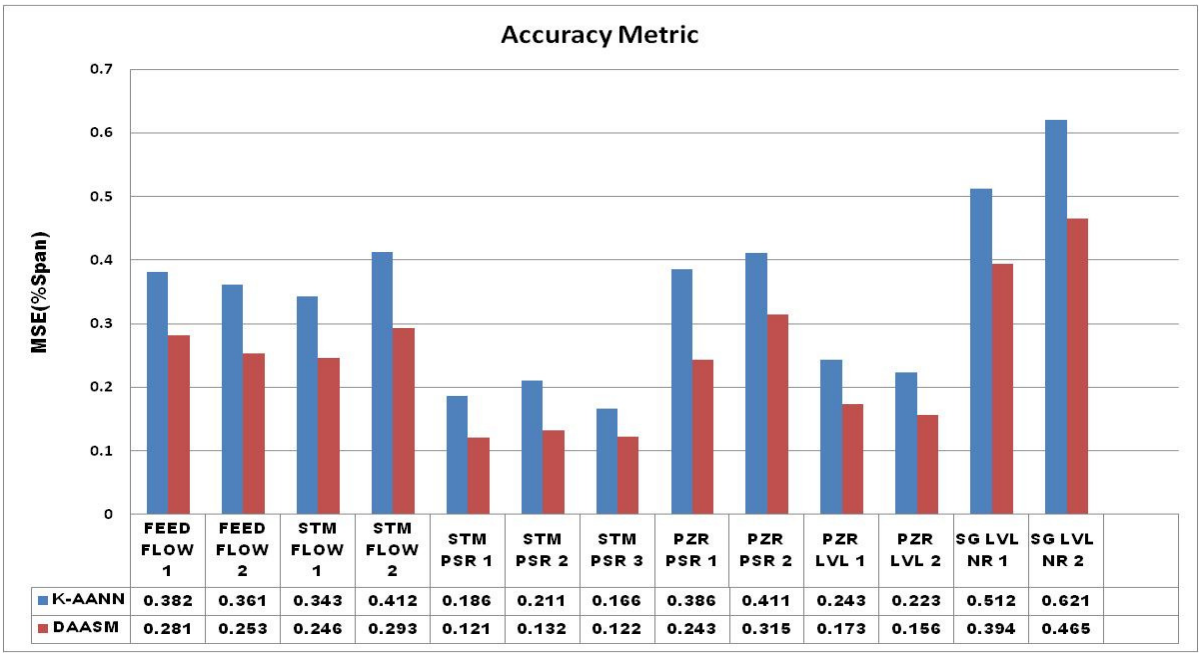


Figure 9: MSE depicting DAASM and K-AANN accuracy on each sensor.

8.2 Robustness

Robustness is quantified through ***Auto-sensitivity*** 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 follow.

$$S_{Auto}^{i} = \frac{1}{N}\sum_{k=1}^{N}\left|\frac{\hat{s}_{ki}^{drift}-\hat{s}_{ki}}{s_{ki}^{drift}-s_{ki}}\right| \tag{15}$$

Where

iAnd k are indexes corresponding to sensors and their respective test samples,.

s_{ki} is the original sensor value without fault.

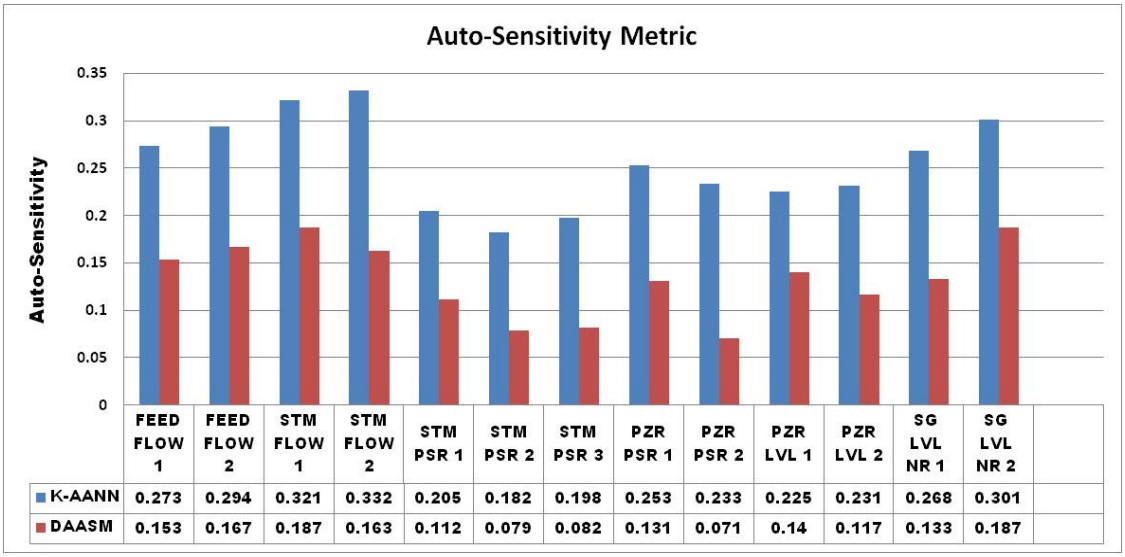
$\hat{s}_{ki}$ is the model estimated sensor value against s_{ki} .

s_{ki}^{drift} is the drifted/faulted sensor value .

$\hat{s}_{ki}^{drift}$ is the model estimated sensor value against drifted value s_{ki}^{drift} .

The auto-sensitivity metric lie in [0,1] range. For auto-sensitivity value of one, the model predictions follow the fault with zero residuals; hence no fault can be detected. Smaller auto-sensitivity values are preferred, which essentially means decreased sensitivity towards small perturbations. Large auto-sensitivity values may lead to missed alarms due to under-estimation of the fault size caused by small

residual values. Compared to k-AANN model, in case of DAASM model, a significant decrease in auto-sensitivity values for all sensors is observed. The plot in fig 10 shows that DAASM is more robust to failed sensor inputs.



	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
K-AANN	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
DAASM	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

Figure 10:Auto-sensitivity values of individual sensors in both models.

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 auto-sensitivities are averaged over whole sensor set. Auto-sensitivity values less than 0.2 are considered as robust. The maximum auto-sensitivity value of 0.187 is observed in steam flow sensor. The plot in fig. 11 shows that average auto-sensitivity for both models increases with increasing level of offset failure. However, the auto-sensitivity curve for DAASM auto-sensitivity is well below than the corresponding K-AANN curve.

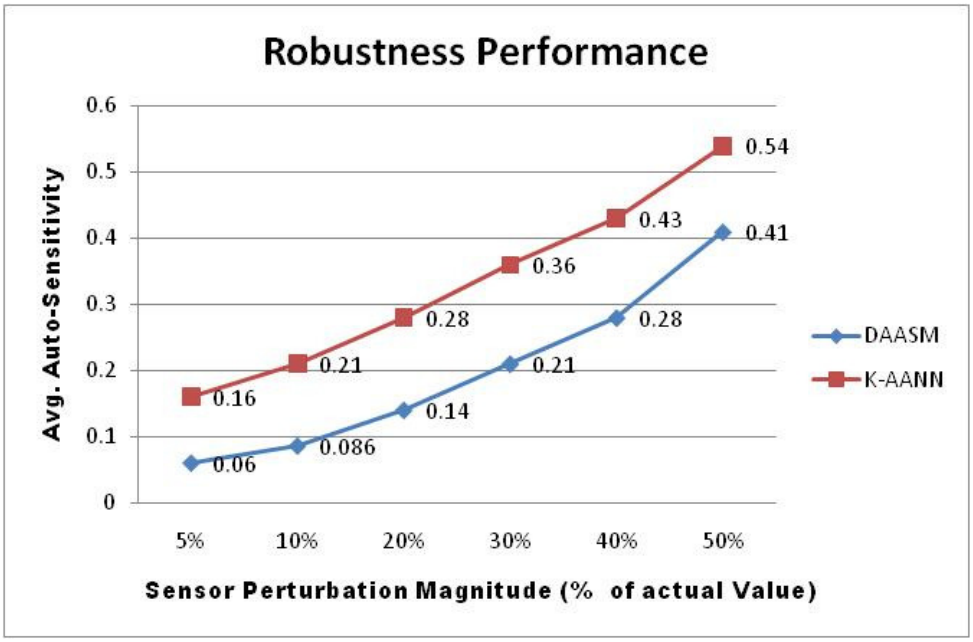


Figure 11: Comparison of robustness against increasing offset failure.

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 in 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 follow.

$$S_{Cross}^{ji} = \frac{1}{N}\sum_{k=1}^{N}\left|\frac{\hat{s}_{kj}^{drift} - \hat{s}_{kj}}{s_{ki}^{drift} - s_{ki}}\right| \tag{12}$$

$$S_{Cross}^{j} = \frac{1}{N-1}\sum_{i=1}^{N} S_{Cross}^{ji} \quad and\ i \neq j \tag{13}$$

Where, $i\ and\ j$ indexes are used to refer faulty and non-faulty sensors, respectively. While, k is the index for corresponding test samples.

S_{Cross}^{ji} is the cross-sensitivity of sensor j w.r.t drift in i^{th} sensor.

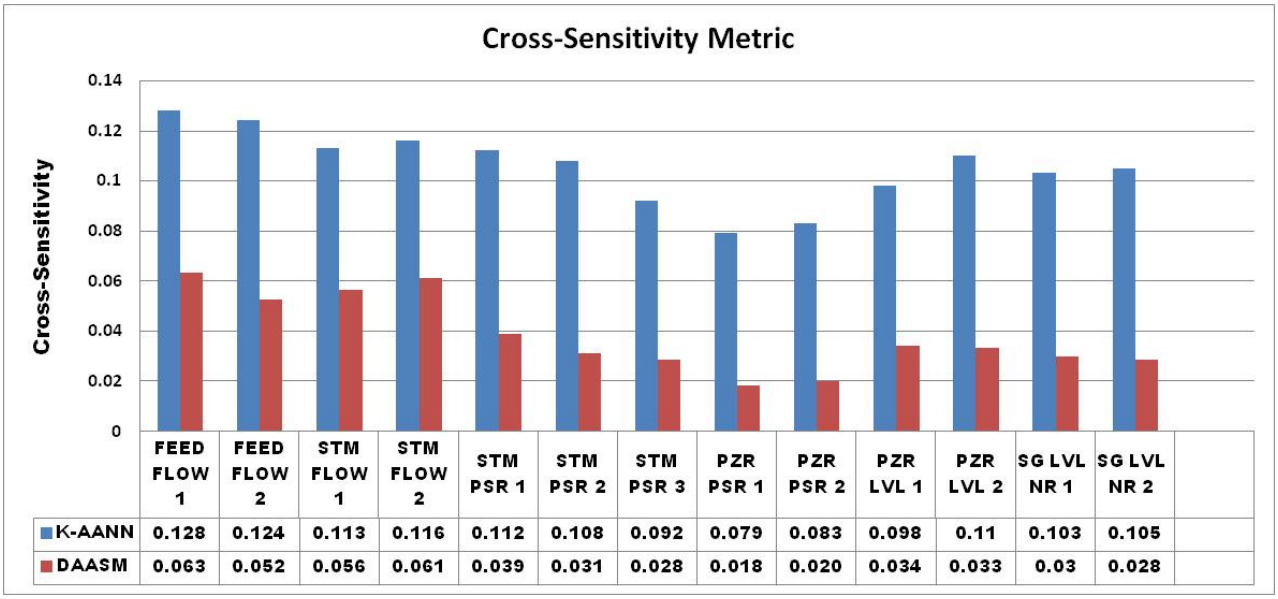
s_{ki} is the value of i^{th} sensor without any fault.

$\hat{s}_{kj}$ is the model estimated value of j^{th} sensor against s_{ki} .

s_{ki}^{drift} is the drifted/faulted value of i^{th} sensor.

$\hat{s}_{kj}^{drift}$ is the Model estimated value of j[th]sensor against drifted value s_{ki}^{drift} .

The cross-sensitivity affect is more eminent for neural network based sensor models because of the highly distributed representation of input at hidden layer mappings. 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 fig. 12 shows that the cross-sensitivity for DAASM model is reduced by a large factor as compared to k-AANN model.



	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
K-AANN	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
DAASM	0.063	0.052	0.056	0.061	0.039	0.031	0.028	0.018	0.020	0.034	0.033	0.03	0.028

Figure 12: Cross-Sensitivity values of individual sensors in Both Models.

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

$$Avg.Cross\ Sensitivity(K\%) = \frac{\sum_{j=1}^{N} S_{cross(K\%)}^{j}}{N} \tag{14}$$

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

The cross-sensitivity values $S_{cross(K)}^{j}$, against K% offset failure level ,are calculated using equation 13. fig. 13 shows the average cross-sensitivity plot for both models. Small cross-sensitivities are observed in DAASM model 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 need to be relaxed to avoid false alarms and isolate the faulty sensor. For k-AANN model, the spillover effect beyond 15% offset failure significantly deteriorates model's robustness.

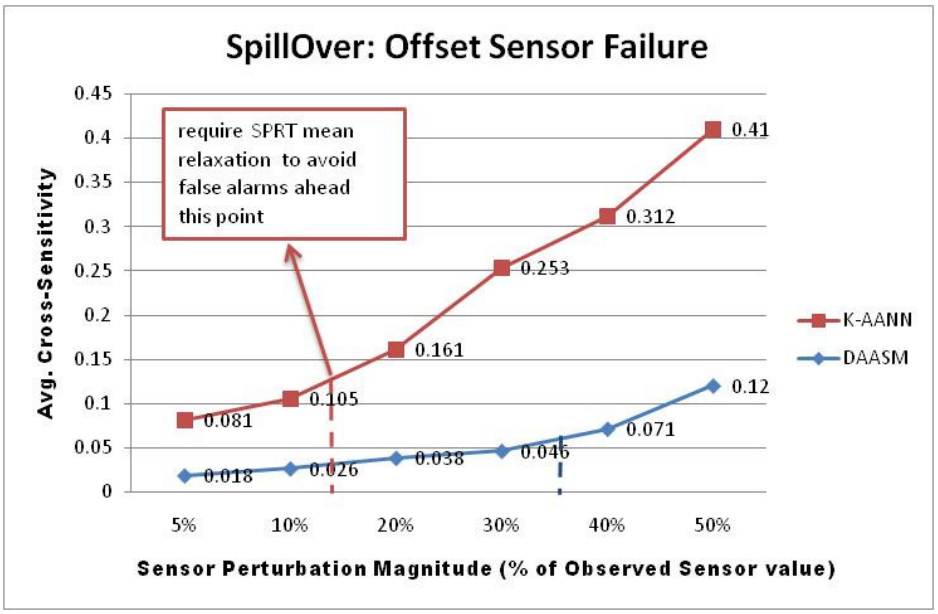


Figure 13: Comparison of spillover effects against increasing offset failure.

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 14. Plot in fig. 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 model effectively managed simultaneous gross high or low failures in 25% of total sensor set as compared to 10% in case of k-AANN.

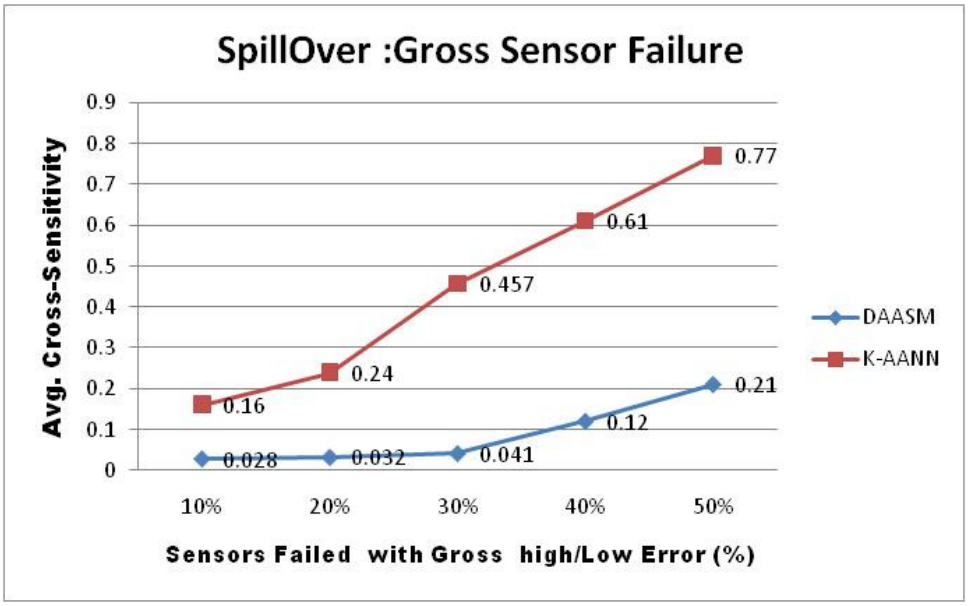


Figure 14: Comparison of spillover effect against simultaneous gross high/low failure in multiple sensors.

8.4 Fault Detectibility

Fault Detectability metric measures the smallest fault that can be detected by integrated sensor estimation and fault detection module as shown in Fig. 1 [32].The detectability metric is measured as percentage of sensor span D=$\frac{M}{Span}$, 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].

$$\mu_0 : normal\ mode\ residual\ mean$$

$$\sigma^2 : normal\ mode\ residual\ variance$$

$$\mu_1 : expected\ offset\ in\ residual\ mean\ in\ abnormal\ mode$$

$$\alpha : false\ alarm\ probability$$

$$\beta : Missed\ alarm\ probability$$

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 fix. 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 that searches for minimum expected offset μ_1 in the interval {μ_1: [σ-3σ]}, 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 fig. 15 shows the detectability metric for each sensor. The plot in fig. 15 shows that DAASM model can detect faults which are two times smaller in magnitude than those detectable by K-AANN model.

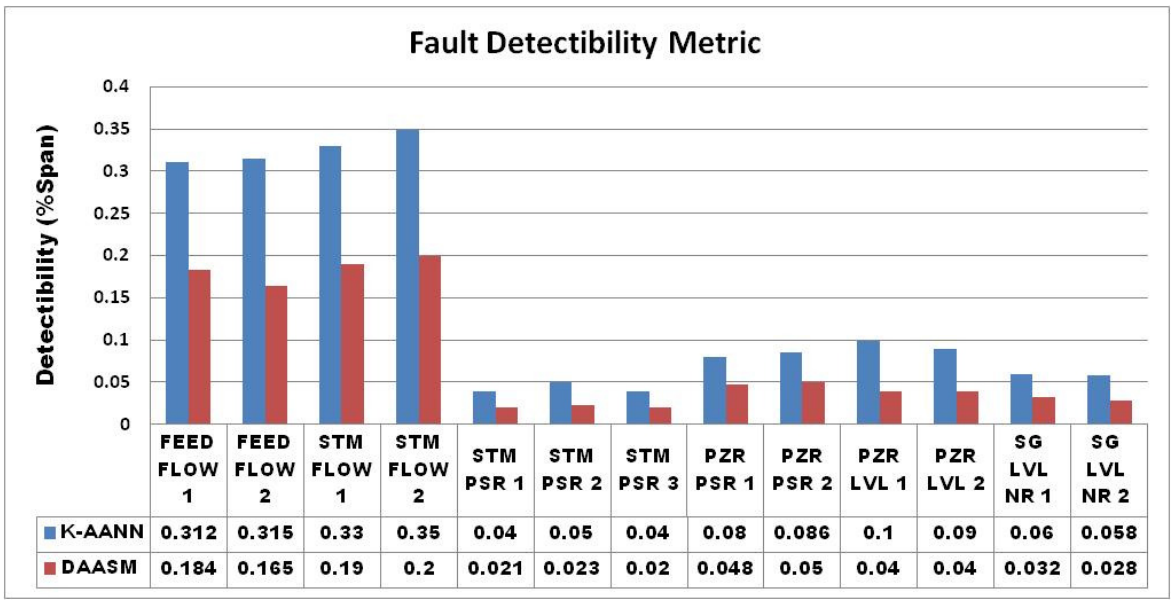


Figure 15: Comparison of fault detectability metric.

Improvement in fault detectibility metric for DAASM model can be attributed to observed improvement in model robustness ; as suggested by following relation.

$$\frac{r_i}{\Delta S_i^{drift}} = (1 - S_{Auto}^i) \tag{15}$$

The term $\frac{r_i}{\Delta S_i^{drift}}$ measures the ratio of observed residual to actual sensor drift in terms of auto-sensitivity. 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 Detectibility Test

Sequential probability ratio[34][36] based fault hypothesis test is applied to residual sequence $\{R_i\} = r_{1(t_1)}, r_{1(t_1)} \ldots r_{n(t_n)}$ generated by relation $\mathrm{R_i(t_i)} = \mathrm{S^{Obs}(t_i)} - \mathrm{S^{Est}(t_i)}$ at time t_i. Where $\mathrm{S^{Obs}(t_i)}$ and $\mathrm{S^{Est}(t_i)}$ 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_O or abnormal mode hypothesis H_1 by using likelihood ratio as follow.

$$L_n = exp\left[-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n} \mu_1(\mu_1 - 2r_i)\right]\right] \tag{16}$$

For fault free sensor values, the normal mode hypotheses H_0 is approximated by gaussian distribution with mean μ_0=0 and varianceσ^2. Abnormal mode hypothesis H_1 is approximated with mean $\mu_1 > \mu_0$ using same variance σ^2 . The SPRT index for the positive mean test is finally obtained by taking logarithm of the likelihood ratio in equation 16 as follow [35].

$$SPRT = \ln(L_n) = -\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}\mu_1(\mu_1 - 2r_i)\right] = \frac{\mu_1}{\sigma^2}\sum_{i=1}^{n}\left(r_i - \frac{\mu_1}{2}\right) \quad (17)$$

Pressurizer pressure sensor, sampled at a frequency of 10 second, is used as a test signal to validate the fault detectability 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 plot in Fig. 16 shows drifted and estimated pressure signal from DAASM and k-AANN model, 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 and DAASM models. The hypothesis H1 and H0 corresponds 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 hour since the drift inception, shows that DAASM model is more sensitive to small drifts. On the other hand, SPRT index on k-AANN based sensor estimates registered the same drift at 3800thsample 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.

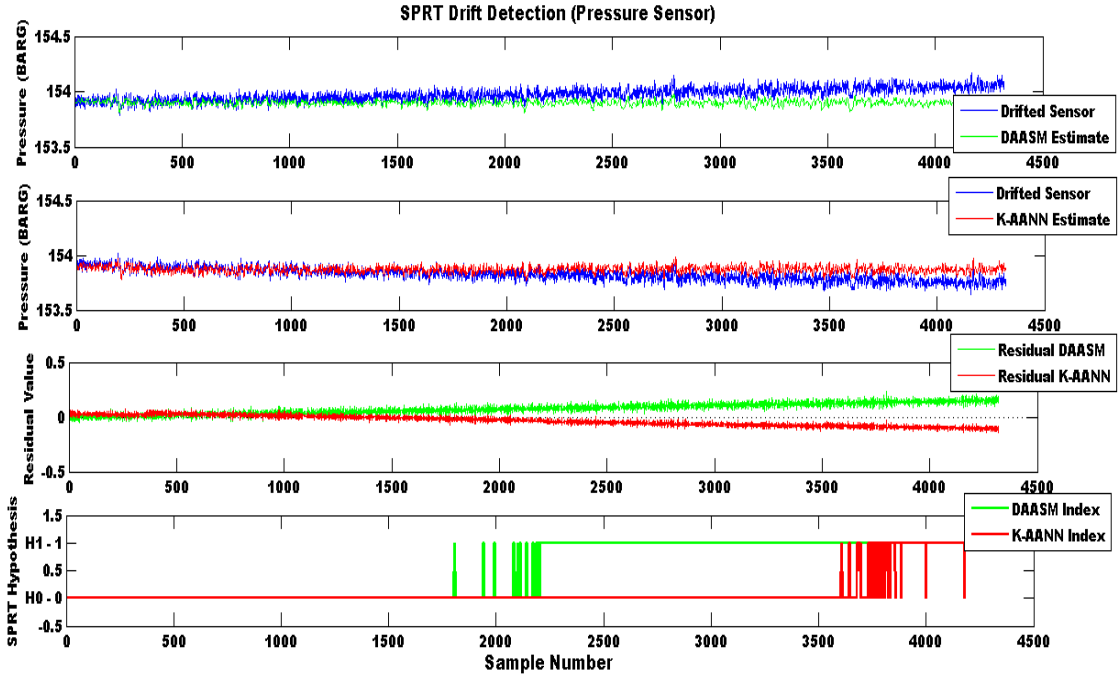


Figure 16: SPRT based fault detection in pressurizer pressure sensor.

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. The k-AANN model failed to register a very small drift up to 0.1% in steam flow (STM FLOW1) channel. A small drift up to 0.1 BARG is detected in test set 5 of pressurizer pressure channel. However, in case of drift type sensor failures, fault detection lag for DAASM model was on average 0.5 times smaller in comparison with k-AANN model. Plots in fig. 17 through 21 show the estimated sensor values ,from both models, on five test data sets of few selected channels.

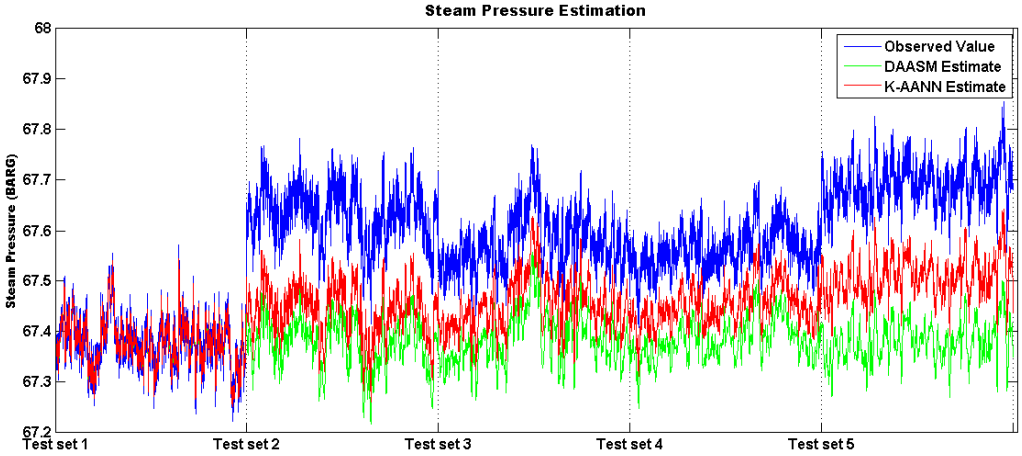


Figure 17: Steam pressure estimates against offset failures up to 0.3 (BARG)

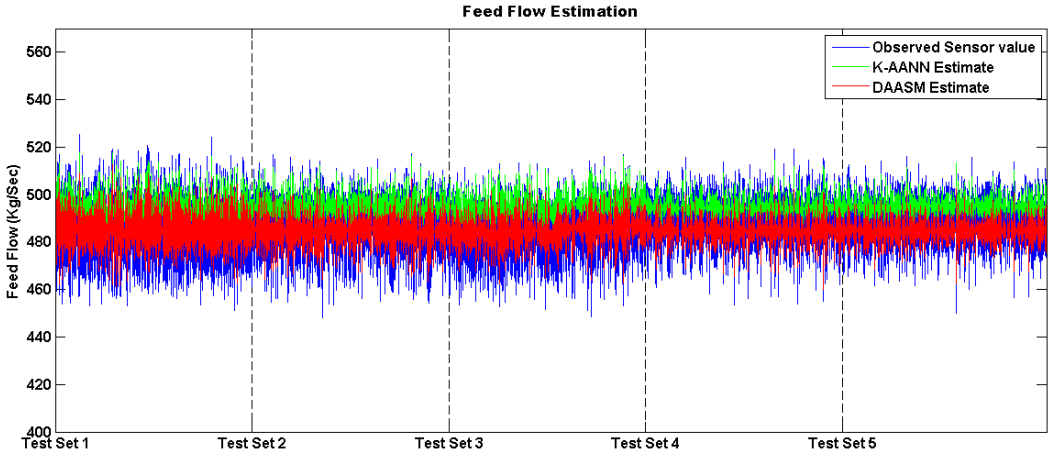


Figure 18: Feed flow estimates.

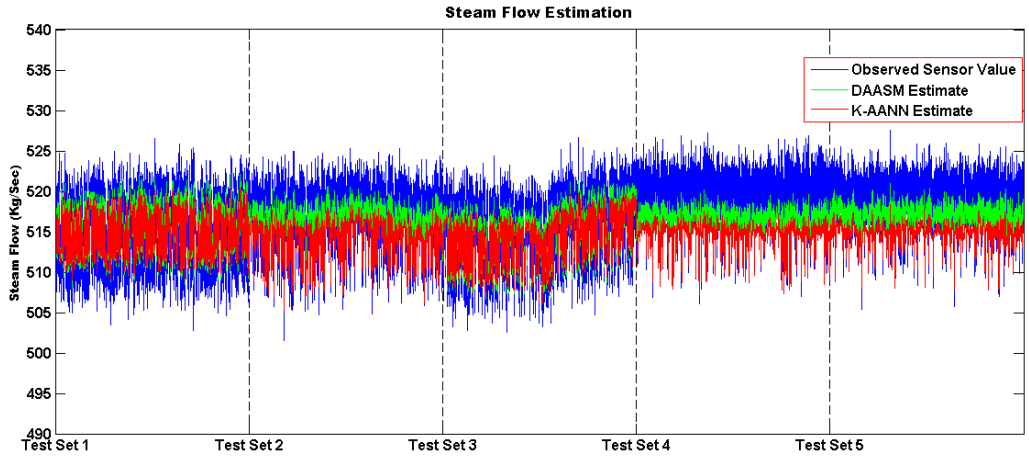


Figure 19: Steam flow estimates against drift failure up to 0.1%

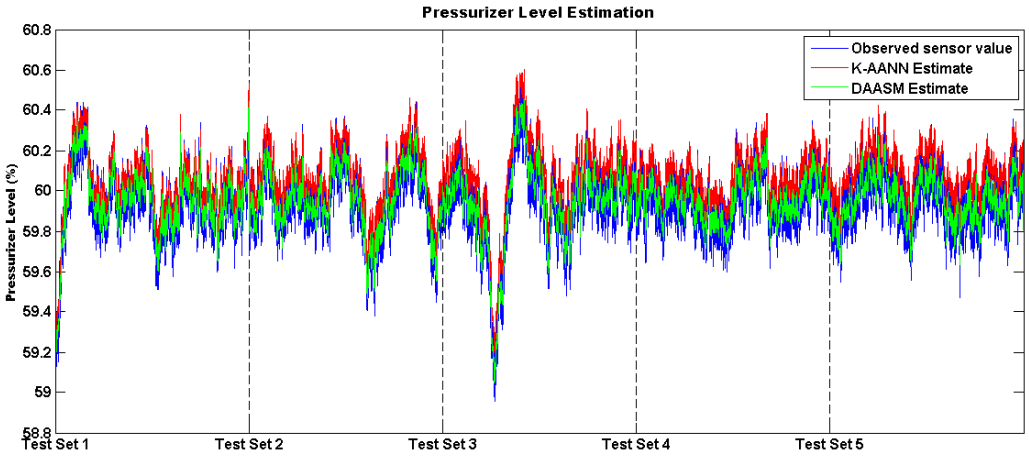


Figure 20: Pressurizer level estimates.

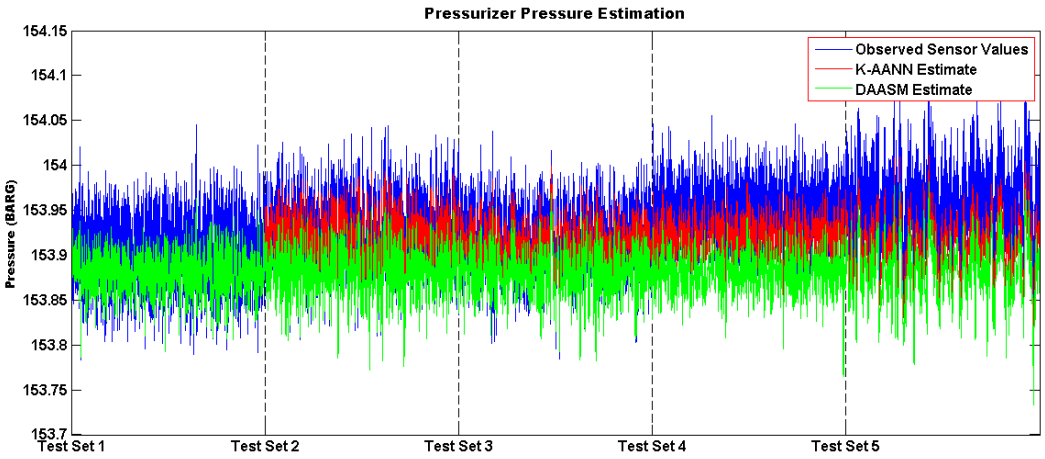


Figure 21: Pressurizer pressure estimates against drift failure up to 0.1%

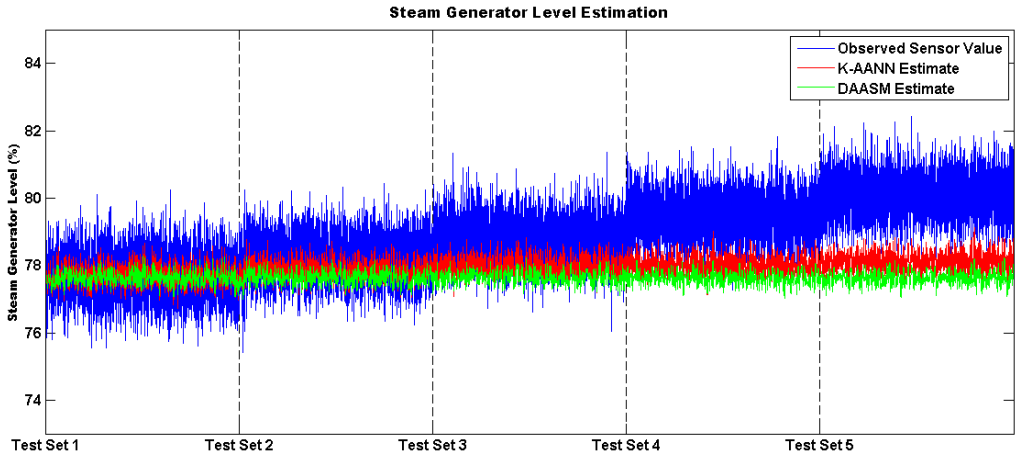


Figure 22: Steam generator level estimates against growing drift failure up to 2.87%.

CONCLUSION

This paper presented a neural network based Denoised Auto-Associative 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. Multi-layer AANN based sensor models may result in suboptimal solutions due to poor regularization by traditional back propagation based joint multi-layer learning procedures. So a complementary deep learning approach, based on greedy layer wise unsupervised pre-training is employed for effective regularization in the proposed multi-layer DAASM model. Auto-encoder architecture is used for denoising based unsupervised pre-training and regularization of individual layers in the network hierarchy. To address robustness against perturbations in input sensors; data corruption processes exercised during unsupervised pre-training 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 pre-trained 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 layer wise pre-training technique, in combination with domain specific corruption processes, provides a viable framework for effective regularization and robustness in such deep multi-layered auto-associative sensor validation models.

Acknowledgement

Authors acknowledges the support of Mr. José Galindo Rodríguez affiliated to TECNATOM Inc. Spain and Mr. Chad Painter(Director Nuclear Power Plant Simulator development and training program at International Atomic Energy Agency) for providing necessary tools and data to conduct this research.

Conflict of Interest Statement

Authors declare that there is no conflict of interest regarding the publication of this article.

APPENDIX A

Table A. 1: 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
Auto-Sensitivity	☑		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
		☑	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
Cross-Sensitivity	☑		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	0.00286
SPRT Detectibility	☑		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

References

[1] N. Mehranbod, “Probabilistic model for sensor fault detection and identification,” *AIChE J.*, vol. 49, no. 7, p. 124, 2002.

[2] J. Hines and E. Davis, “Lessons learned from the US nuclear power plant on-line monitoring programs,” *Prog. Nucl. Energy*, Vol. 46, No. 3-4, pp. 176-189, 2005.

[3] EPRI (2003), On-Line Monitoring Cost Benefit Guide, Final Report, EPRI, Palo Alto,CA: 1006777.

[4] S. J. Qin and W. Li, “Detection, identification, and reconstruction of faulty sensors with maximized sensitivity,” *AIChE J.*, vol. 45, no. 9, pp. 1963–1976, 1999.

[5] J. Ma and J. Jiang, “Progress in Nuclear Energy Applications of fault detection and diagnosis methods in nuclear power plants : A review,” *Prog. Nucl. 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,” *Nucl. Eng. Technol.*, vol. 40, no. 1, pp. 49–68, 2008.

[7] J.B. Coble , R.M. Meyer, P. Ramuhalli, L.J. Bond, H. Hashemian, B. Shumaker, and D.S. Cummins, “A Review of Sensor Calibration Monitoring For Calibration Interval Extension In Nuclear Power Plants,” PNNL---21687, Pacific Northwest National Laboratory, Richland, WA., August 2012.

[8] M. A. Kramer, “Autoassociative neural networks,” *Comput. Chem. Eng.*, vol. 16, no. 4, pp. 313–328, 1992.

[9] X. Xu, J. W. Hines, and R. E. Uhrig, “Sensor "On-Line Sensor Calibration Monitoring and Fault Detection for Chemical Processes,” in *Proc. Maintenance and Reliability Conference (MARCON 98)*, Knoxville, TN, May 12-14, 1998.

[10] M. Hamidreza, S. Mehdi, J.-R. Hooshang, and N. Aliakbar, “Reconstruction based approach to sensor fault diagnosis using auto-associative neural networks,” *J. Cent. South Univ.*, 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,” *Anal. Chim. Acta*, vol. 446, no. 1–2, pp. 369–381, 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 Proc. *of 1996 American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies*, University Park, PA, May 6-9, 1996.

[13] J. W. Hines, R. E. Uhrig, and D. J. Wrest, “Use of autoassociative neural networks for signal validation,” *J. Intell. Robot. Syst.*, 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,” *Prog. Nucl. Energy*, vol. 43, no. 1–4, pp. 83–89, Jan. 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,” *Ann. Nucl. Energy*, vol. 32, no. 11, pp. 1207–1223, Jul. 2005.

[16] M. S. Ikbal, H. Misra, and B. Yegnanarayana, “Analysis of autoassociative mapping neural networks,” in *IJCNN’99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339)*, 1999, vol. 5, pp. 3025–3029.

[17] Y. Bengio, “Learning Deep Architectures for AI,” *Found. Trends® Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[18] J. W. Hines, A. Gribok, I. Attieh, and R. Uhrig, “Regularization Methods for Inferential Sensing in Nuclear Power Plants,” in *Fuzzy Systems and Soft Computing in Nuclear Engineering SE - 13*, vol. 38, D. Ruan, Ed. Physica-Verlag HD, 2000, pp. 285–314.

[19] A. V. Gribok, J. W. Hines, A. Urmanov, and R. E. Uhrig, “Heuristic, systematic, and informational regularization for process monitoring,” *Int. J. Intell. Syst.*, vol. 17, no. 8, pp. 723–749, 2002.

[20] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. of AISTATS 2010,* vol. 9, pp. 249–256, 2010.

[21] S. Tan and M. L. Mayrovouniotis, “Reducing data dimensionality through optimizing neural network inputs,” *AIChE J.*, vol. 41, no. 6, pp. 1471–1480, 1995.

[22] Y. Bengio and P. Lamblin, “Greedy layer-wise training of deep networks,” *Adv. in Neural Information Processing Systems 19* , pp. 153–160, 2007.

[23] H. Larochelle, H. Larochelle, Y. Bengio, Y. Bengio, J. Lourador, J. Lourador, P. Lamblin, and P. Lamblin, “Exploring Strategies for Training Deep Neural Networks,” *J. Mach. Learn. Res.*, vol. 10, pp. 1–40, 2009.

[24] D. Yu and L. Deng, “Deep Learning and Its Applications to Signal and Information Processing [Exploratory DSP],” *Signal Process. Mag. IEEE*, vol. 28, no. 1, pp. 145–154, 2011.

[25] D. Erhan, A. Courville, and P. Vincent, “Why Does Unsupervised Pre-training Help Deep Learning ?,” *J. Mach. Learn. Res.*, 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,” *Proc. 25th Int. Conf. Mach. Learn. - ICML ’08*, pp. 1096–1103, 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,” *J. Mach. Learn. Res.*, vol. 11, no. 3, pp. 3371–3408, 2010.

[28] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv: 1207.0580*, pp. 1–18, 2012.

[29] M. J. Embrechts, B. J. Hargis, and J. D. Linton, “An Augmented Efficient Backpropagation Training Strategy for Deep Autoassociative Neural Networks,” *Comput. Intell.*, no. April, pp. 28–30, 2010.

[30] Y. Bengio, “Practical Recommendations for Gradient-Based Training of Deep Architectures,” *in Neural Networks: Tricks of the Trade (2nd ed.)*, vol. 7700 of Lecture Notes in Computer Science, pp. 437–478,Springer, Berlin, Germany, 2012.

[31] J. Bergstra and Y. Bengio, “Random Search for Hyper-Parameter Optimization,” *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.

[32] J. W. Hines, “Development and Application of Fault Detectability Performance Metrics for Instrument Calibration Verification and Anomaly Detection,” *Pattern Recognit.*, vol. 1, pp. 2–15, 2006.

[33] A. Usynin, J. W. Hines, “On-Line Monitoring Robustness Measures and Comparisons,” International Atomic Energy Agency Technical Meeting on "*Increasing instrument calibration interval through on-line calibration technology*", OECD Halden Reactor Project, Halden, Norway, 27th-29th September 2004.

[34] A. Wald , “Sequential tests of statistical hypotheses,” *Ann. Math. Stat.*, vol. 16, no. 2, pp. 117–186, 1945.

[35] F. Di Maio, P. Baraldi, E. Zio, S. Member, 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 Syst. Appl.*, vol. 39, no. 9, pp. 8467–8473, 2012.