



Research Article

Incipient fault detection of sensors used in wastewater treatment plants based on deep dropout neural network

Barasha Mali^{1,2}  · Shahedul Haque Laskar¹

Received: 27 August 2020 / Accepted: 18 November 2020 / Published online: 1 December 2020
© Springer Nature Switzerland AG 2020

Abstract

Risk of sudden collapse of any industrial plant increases if small magnitude incipient faults are not detected at an early stage. The paper proposes an optimized Monte Carlo deep dropout neural network (MC-DDNN) to identify incipient faults of sensors installed in wastewater treatment plants using the historical dataset of the plant. Such faults usually remain invisible or are misinterpreted as noise signals due to their small magnitude but can be overcome by the proposed method. MC-DDNN easily identifies the incipient faults of sensors installed in a simulated wastewater treatment benchmark model as well as sensors installed in a real industrial plant. The tabulated results show the estimated probability of incipient fault in terms of percentage probability as detected by the MC-DDNN. The dissolved oxygen (DO) sensor incipient faults in benchmark simulation model (BSM2) are detected with probability ranging from 4.9% to 23.4% and DO, pH and mixed liquor suspended solids (MLSS) sensors of effluent treatment plant (ETP) are detected with probability ranging from 0.07% to 11.43%. This estimated probability of faults indicates the small magnitude of the faults and hence proves that the method is capable of identifying faults at an early stage to issue warnings for early maintenance of the plant.

Keywords Wastewater treatment plant · Sensors · Incipient fault · Neural network · Deep learning · Dropout

1 Introduction

Sensors are the key components of Industry 4.0 which are becoming mandatory for most of the industries. Industry 4.0 or fourth industrial revolution started in 2011 and aims to make industrial processes much more operationally efficient, productive and automated [17, 22]. To achieve these objectives, the main component connecting the real world to the engineered system must give correct information. This key component connecting the real world and engineered system is known as a sensor whose fault may cause the system to compromise in operation and sometimes may cause the whole system to suddenly shut down even risking lives or resulting in irreparable damages [21]. In order to prevent such situations and to achieve increased reliability, timely sensor fault detection

is required [3]. Specifically, if the fault is detected as soon as it starts to appear, preventive maintenance can be made well in advance so as to avoid the sudden collapse of the system. Faults that has just started are known as incipient faults and their occurrence shows gradually and slowly. Also, incipient faults are often confused to be noise signals or uncertain behaviour of the system. The detection, therefore, becomes more elusive than the traditional fault detection methods [19, 24].

Although incipient sensor faults occur commonly both in linear and nonlinear systems and have practical significance, they have received less attention in the past and only from 2015 research focusing incipient sensor faults are seen [15, 30, 33]. Initially, sensor incipient fault detection was popular in aircraft fault detection and maintenance [9, 29] but with increasing demand of effective

✉ Barasha Mali, barashamali@gmail.com | ¹National Institute of Technology Silchar, Assam, India. ²Sant Longowal Institute of Engineering and Technology, Longowal, Punjab, India.



fault diagnosis in industries, sensor fault detection has become popular in rotating machinery [8, 34, 35], train air brake system [26, 31], industrial cyber-physical systems [6], nuclear power plants [10], simulated models of continuous stirred tank reactor and tennessee eastman process [27, 32] and traction systems [12].

Fault detection and diagnosis (FDD) in general is classified into three categories: model based, signal based and data based [13]. The first and the second categories require system experts and at the same time is costly because of which these methods become inconvenient for complex systems. The third category, data-based FDD, does not require previous knowledge of the system and is easy to implement. They are in demand recently and is becoming popular. The data-based techniques have also shown better performance in real-time operation of engineered systems and give minimum false alarm rate as compared to the traditional methods of the [18–20].

These data-based methods can be subclassified into two categories: statistical analysis based and machine learning based [23]. The available incipient fault detection and diagnosis works mostly use statistical analysis and observer-based methods to detect and diagnose sensor incipient faults [7, 18]. But the machine learning-based techniques remain unexplored for sensor incipient fault detection and diagnosis. This motivated the authors to explore the different machine learning data-based techniques for sensor incipient fault detection.

The deep neural network is one of the machine learning techniques which is reliable and effective in indicating incipient faults [13]. It is a supervised learning method which requires labelled data for training. But labelled incipient fault data are practically not available in most of the cases. For such unseen faults, Monte Carlo dropout (MC dropout method) proves to be effective in both detection and diagnosis of incipient faults using uncertainty estimates [13].

Fault detection in sensors of the wastewater treatment plant (WTP) using deep neural network performs effectively for timely and automatic WTP management. [18]. Now, because of the increasing environmental concern WTPs in all industries and municipalities are expected to be fully functional and also updated for automation. In such a situation, if fault detection is done as soon as it occurs, it will save time as well as money. Incipient fault detection is a possible direction to improve the automation of WTPs. In consideration of this, the authors have contributed the following:

1. Propose an optimized MC deep-dropout neural network method to improve the learning of the neural networks and at the same time detect and diagnose the incipient fault of sensors installed in WTPs.

2. The proposed method is applied to sensors in a wastewater treatment benchmark simulated model using the self-generated historical dataset of the same plant.
3. The proposed method is validated in three sensors installed in a petrochemical industry ETP dataset to verify its applicability in a practically operating plant.

The rest of the paper is organized as follows: Sect. 2 discusses the optimized MC deep dropout neural network method and the environments to which the proposed method is applied, and Sect. 3 discusses the circumstances in which the data are collected and how the incipient fault detection is achieved. This is followed by the conclusion in Sect. 4.

2 Methodology

2.1 Dataset description

The proposed incipient fault detection method is implemented in a simulated and a real dataset listed below to verify its application in practical cases.

1. Simulated wastewater treatment plant model known as benchmark simulation model 2 (BSM2).
2. Effluent treatment plant of Guwahati Oil Refinery.

2.1.1 Benchmark simulation model 2

BSM2 proposed by COST action group 624 and IWA Task group is a popular wastewater treatment benchmark model [2].

The model considers the plant-wide set-up of the WTP as shown in Fig. 1. It includes primary, secondary and tertiary treatment representations. Activated sludge reactors also known as biological reactors represent the secondary treatment of BSM2 and consist of two anoxic tanks followed by three aerobic tanks. The sensors connected to this part of the plant are explored for experiments in this paperwork.

The sensors used in BSM2 are classified into six classes based on the response time. This response time is a combination of the delay time and rise/fall time as per ISO 2003 norm. The dataset acquired for our experiment is from Class A sensors. Such sensors have 1 minute response time and are very close to ideal sensors. The different parameters obtained from the model after simulation give the value as sensed by ideal sensors. Specifically, the dataset is generated by introducing different severity level faults in terms of percentage fault in the ideal oxygen sensors of the activated sludge part of BSM2.

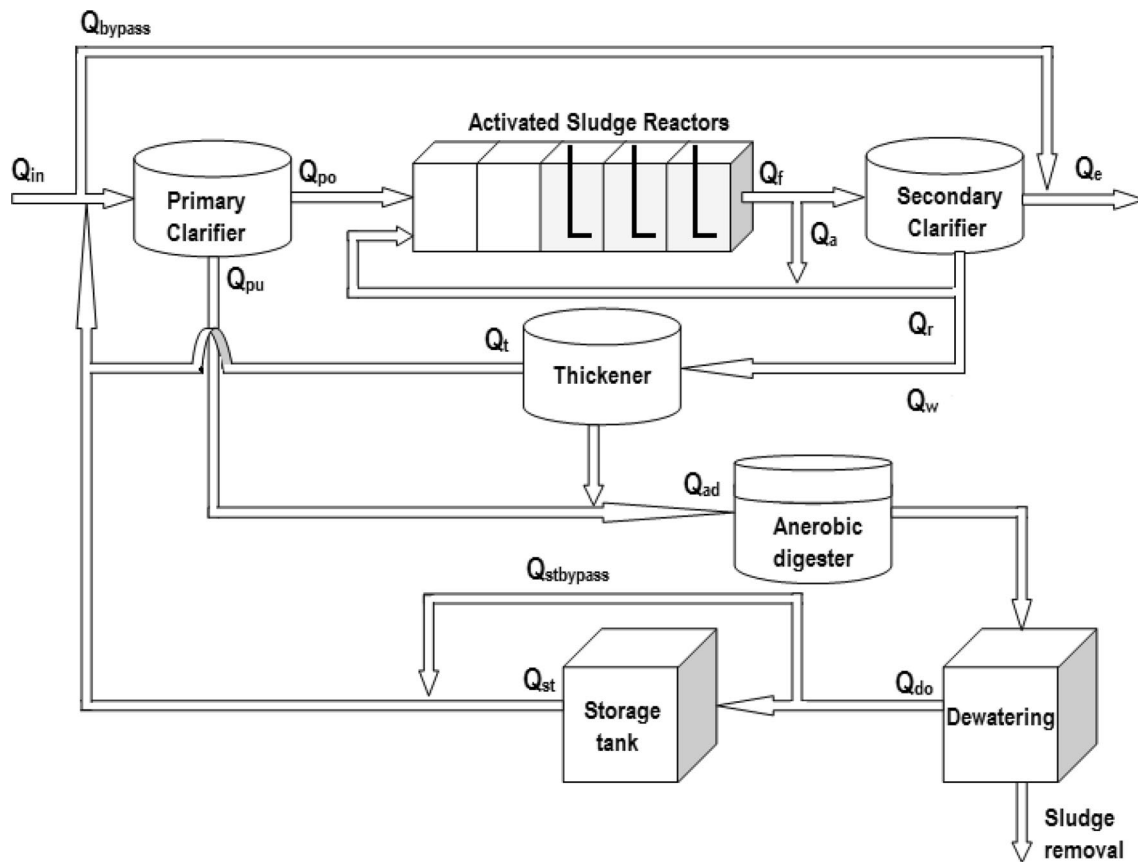


Fig. 1 Plant layout of benchmark simulation model 2

2.1.2 Guwahati Oil Refinery data

Guwahati Oil Refinery is IndianOil's first Refinery and is operational since 1962. Since the waste treatment is crucial in oil industries, treatment is done in a treatment plant known as effluent treatment plant (ETP). The treatment plant has a number of complex instruments connected together and is remotely controlled. Different sensors installed in the facility provide the raw data to the controlling unit.

The control therefore depends on the readings from these sensors, and their fault detection becomes important. The earlier we detect the fault the better. And the plant operators can perform timely maintenance to avoid any sudden failure of the whole set-up. Therefore, the proposed incipient fault detection is applied to the three sensors installed in the effluent treatment plant. The past sensor readings are collected for two years at a sampling rate of 1 day from IOCL Guwahati, and size of the acquired dataset is increased by linear interpolation. This modified dataset is used for incipient fault detection of the sensors in the present working condition.

Training data for incipient fault are generated by introducing faults in the dataset because the available sensor data are from an errorless functional sensor installed in the plant. For the experiments, the available data are normalized and considered as the reference.

2.2 Proposed approach

The proposed method is summarized in Fig. 2. The MC-DDNN is applied to a single sensor error data from BSM2 to find the network training epoch and dropout for the assumed dataset. The learning rate (α) is the measure of change of weight per time and is assumed to be 0.01 so that the output easily converges to the desired prediction [16]. The optimized parameters of the network are then used to predict the incipient fault of multiple sensors in the BSM2 and a petrochemical industry ETP.

Dropout is a regularization technique used in deep neural networks to increase the generalization capacity of deep neural networks and improve the overfitting problem. Nodes in each layer of the deep neural network are dropped randomly during training to induce randomization in the

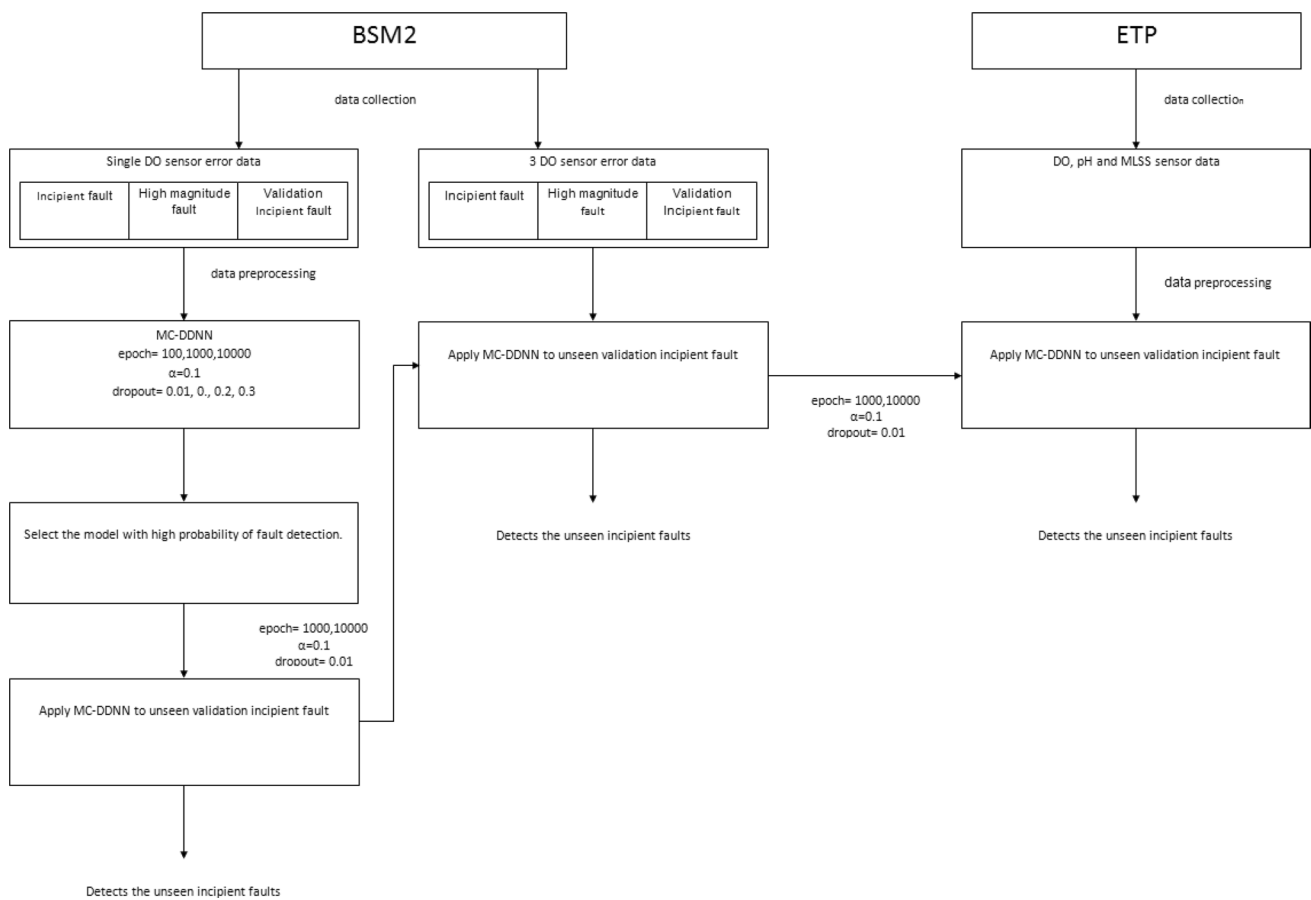


Fig. 2 Schematic of the proposed method

network and increase the number of neural networks [1, 13, 28].

Figure 3 shows the structure of a dropout implementation to the deep neural network after dropping out some of the nodes from two hidden layers. The nodes marked with a cross are dropped out of the network. Also, the connections with the dropped node are deactivated during this dropout. Therefore, the contribution of these dropout nodes withdraws temporally for both forward and backward pass during the training of the deep neural network.

Using the probabilistic approach, a random output variable after passing through the pre-activation of a node has an output [14]

$$v = \sum_{i=1}^n w_i * \delta_i * x_i + b * \delta_b \tag{1}$$

where δ_i and δ_b are the gating variables which decide the exclusion of the node from that particular training session. δ here is assumed to be Bernoulli random variable. It deletes the node inputs and its connecting weights with probability $P(\delta_i = 0) = q_i$ and selects the remaining nodes

with probability $P(\delta_i = 1) = p_i$. δ in dropout calculations is assumed to be independent of each other, the connecting weights and the activity of the nodes. w and b are the learning parameters [5].

This random output after passing through the activation function $f(\cdot)$ becomes

$$y = f(v) = f\left(\sum_{i=1}^n w_i * \delta_i * x_i + b * \delta_b\right) \tag{2}$$

$$\implies y = f(v) = f(W * \delta_i * x + b * \delta_b) \tag{3}$$

Therefore, the deep dropout neural network in Fig. 3 has the following outputs at the hidden layer 1.

$$v_1 = W_1 * \delta_1 * x + b_1 = W_1 * h_0 + b_1 * \delta_{b1} \tag{4}$$

$$h_1 = g(v_1) = g(W_1 * \delta_1 * h_0 + b_1 * \delta_{b1}) \tag{5}$$

Similarly, the outputs at hidden layer 2 are

$$v_2 = W_2 * \delta_2 * h_1 + b_2 * \delta_{b2} \tag{6}$$

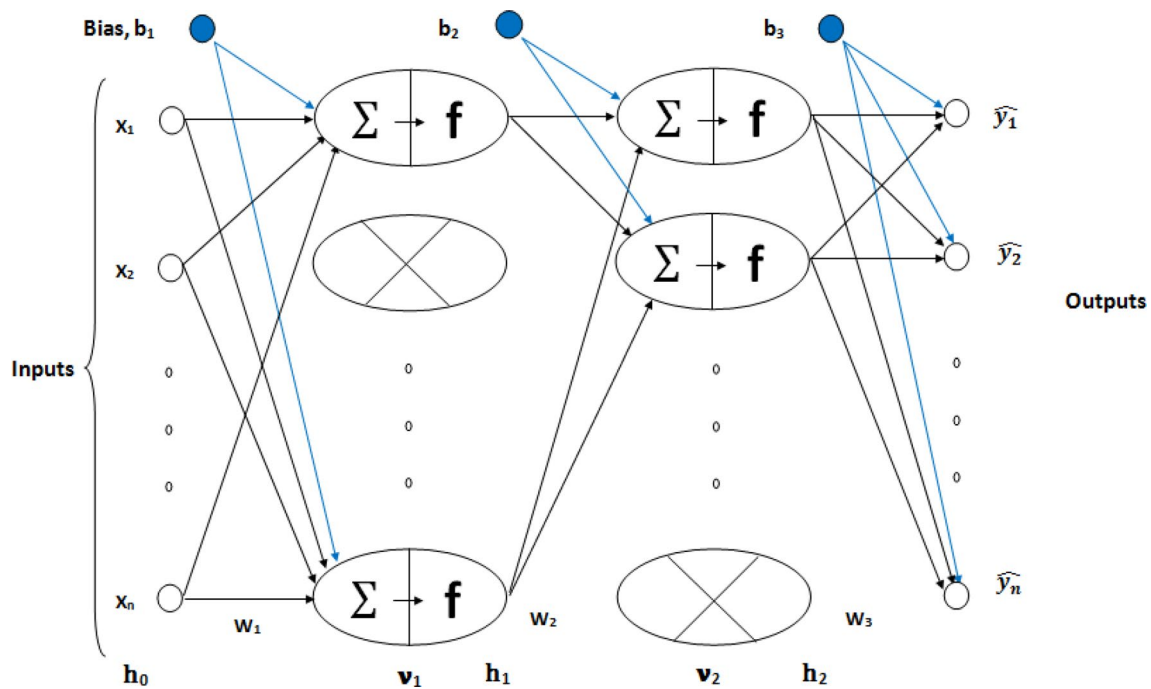


Fig. 3 Dropout representation

$$h_2 = g(v_2) = g(W_2 * \delta_2 * h_1 + b_2 * \delta_{b2}) \tag{7}$$

Estimated output after passing through the deep dropout neural network is obtained at the output layer which is the third layer in this case. The output is given by

$$v_3 = W_3 * h_2 + b_3 \tag{8}$$

$$h_L = O(v_3) = O(W_3 * h_2 + b_3) = \hat{y}_i \tag{9}$$

The output vector \hat{y}_i is a function of x_i similar to the deep neural network without dropout. Inclusion of dropout in the network gives the generalized output as

$$\hat{y}_i = f(x_i) = O(W_3 * g(W_2 * \delta_2 * g(W_1 * \delta_1 * x + b_1 * \delta_{b1}) + b_2 * \delta_{b2}) + b_3) \tag{10}$$

where $x \in R^n$, $W_i \in R^{n*n}$, $\delta_i \in R^n$ and $b_i \in R^n$

In this technique, randomly dropping of nodes from the network during training will force the other activated nodes to learn the information about the input–output relation of the feed in data. This happens again and again, and the network generalizes in a better way without memorizing the pattern. And each training dataset experiences independent randomization.

The estimation during test time is, however, deterministic. All the nodes are present along with the connections, and the weights are adjusted according to the dropout ratio as $W * \delta_i$. A dataset is represented by a

constant input vector, and the expectation of all the available nodes is considered for the output which is given by

$$E(h_3) = \hat{y}_i = O(W_3 * g(W_2 * p_2 * g(W_1 * p_1 * x + b_1 * p_{b1}) + b_2 * p_{b2}) + b_3) \tag{11}$$

So, one set of test data will give the same output label each time it is passed through the trained deep neural network. The output is the average of an ensemble of deep neural networks. Therefore, the network gives better accuracy using this ensemble concept.

Monte Carlo dropout technique introduces the same concept of inducing randomness in the test time in addition to the training time [11, 13]. In this method, one set of test data will give different outputs each time it is passed through the trained network based on the selected nodes. So the random or the estimated outputs are considered as probabilistic distribution and interpreted as Bayesian interpretation. This is represented as a percent probability of the estimates.

The proposed approach optimized MC deep-dropout neural network (optimized MC-DDNN) detects and diagnoses the sensor incipient fault efficiently with two methods augmented to the deep neural network: cross-entropy and dropout. The cross-entropy function optimizes the network’s learning capability and the dropout to handle the overfitting of the network.

Algorithm 1 Algorithm for training optimized MC-DDNN

Input: x_1, x_2, \dots, x_n
Output: y_1, y_2, \dots, y_n
Initialisation :
 1: Initialize Weights matrix, W with random values
 2: Initialize learning rate, α
LOOP Process
 3: **for** $epoch = 1$ to 10000 **do**
 4: Calculate the neural network weighted sum, $v = \sum_{i=1}^n w_i * x_i + b$ and output, \hat{y}
 5: Calculate the error, $e = y - \hat{y}$
 6: Calculate the back propagated error of the output node, $\delta_n = e$,
 7: Propagate δ backward and calculate the δ till the $(n - 1)_{th}$ layer
 8: $e_n = W^T * \delta$
 9: $\delta_n = f(v_n)e_n$
 10: Update the weights using
 11: $\Delta w_{ij} = \alpha \delta x_i$
 12: $w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$
 13: Repeat steps 4 to 12 for all the input-output data sets
 14: Repeat 4 to 13 till the cost function is minimized and the network is trained as desired
 15: **end for**

Algorithm 1 and flowchart in Fig. 4 summarize the proposed optimized MC-DDNN. The network consists of one input layer, three hidden layers and one output layer. The number of nodes in each layer is given in Table 1. A number of nodes in the input and output layers are selected based on the number of input features and output classes, respectively. A number of hidden layers are selected randomly to improve the learning rule of the deep neural network. Sigmoid activation function is used in all the input, hidden and output nodes of the network.

The proposed network is initially trained and tested with the same dataset. After that unseen incipient faults are tested to check the network’s efficiency if an unseen fault is applied as input. This is a supervised learning method where the weights are adjusted and updated as per the input–output characteristic of the data. In the process of updating the weights, the network tries to minimize the cross-entropy function to optimize the learning of the network. Cross-entropy function is, therefore, the cost function and depends on the actual output and the estimated output. It is given by

$$J = f(y, \hat{y}) = \sum_{i=1}^n \{-y \ln(\hat{y}) - (1 - y) \ln(1 - \hat{y})\} \tag{12}$$

where n is the number of output nodes, y is the actual output from the training data, and \hat{y} is the estimated output at the output nodes of the network.

During training and testing, $p\%$ of the nodes in each hidden layer is randomly assigned zero value or

dropped out. This dropout generalizes the data in a better way.

3 Experimental results and discussion

3.1 Case A

A wastewater treatment simulation case study using BSM2 is used for our first experiment. It is performed in two stages as explained below:

1. dataset generation by introducing faults in sensors in the simulation model for two conditions: first single sensor fault condition and second multiple sensors fault condition
2. detecting and diagnosing incipient faults using the optimized MC-DDNN for both the single and multiple sensor fault conditions.

Table 1 Number of nodes in each layer of the optimized MC-DDNN

Layer	Number of nodes (single sensor fault)	Number of nodes (multiple sensor faults)
Input layer	35,040	105,120
Hidden layer 1	1400	1400
Hidden layer 2	1400	1400
Hidden layer 3	1400	1400
Output layer	3	4

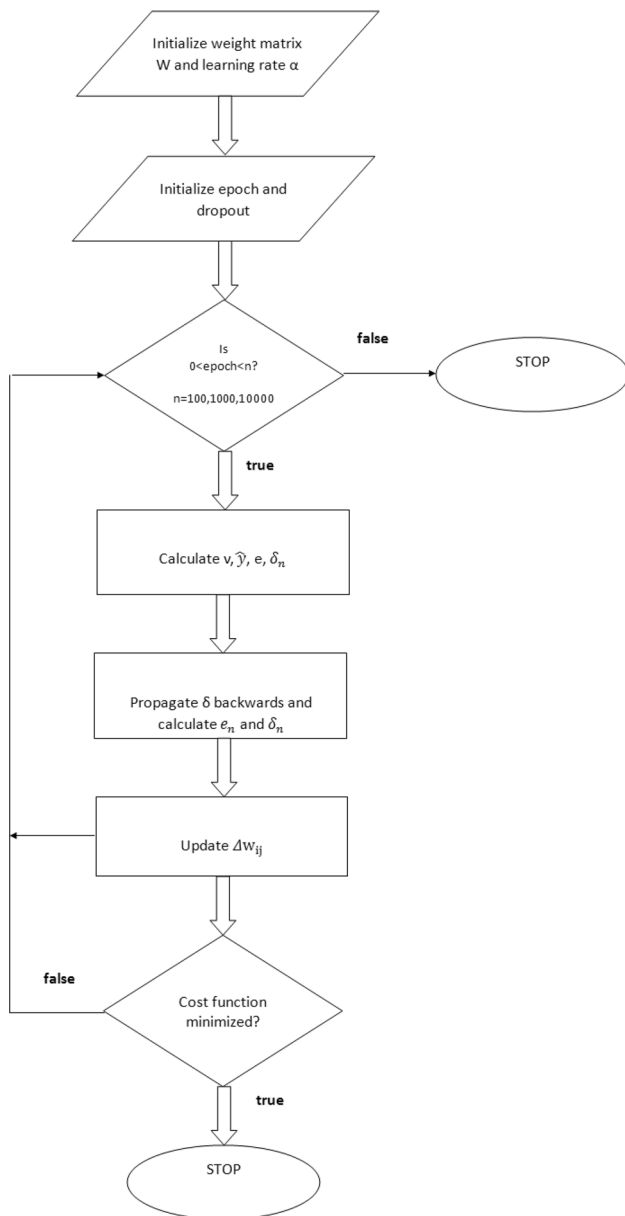


Fig. 4 Flowchart of the proposed method

First, the BSM2 model is simulated in MATLAB by introducing two different types of fault: incipient fault and abrupt high magnitude fault in the DO sensors inside the model. For incipient fault, we have introduced 5%, 10%, 15% and 20% fault in the DO sensor connected to the fourth tank of the biological reactors. The DO sensor data are stored for the different conditions at 15 minutes sampling time. Similarly, for high magnitude faults, we have introduced 80%, 85%, 90% and 95% faults in the same DO sensor and simulated the model to generate the DO sensor data. The DO sensor considered here is a Class A sensor as defined in the BSM2 description. Thus,

Table 2 Estimated value of single sensor incipient faults in BSM2 using optimized MC-DDNN at different dropout rates for 100 trainings (in % probability)

Actual data	Estimated with 0.1 dropout	Estimated with 0.2 dropout	Estimated with 0.3 dropout
<i>100 epochs</i>			
Ref	96.26	53.19	54.18
I_{05}	99.9	44.17	42.28
I_{10}	95.64	58.00	44.21
I_{15}	100	45.11	48.21
I_{20}	99.99	27.99	41.82
HM_{80}	100	20.26	54.76
HM_{85}	100	42.84	59.39
HM_{90}	100	29.92	52.04
HM_{95}	100	45.46	53.74

Table 3 Estimated value of single sensor incipient faults in BSM2 using optimized MC-DDNN at different dropout rates for 1000 and 10,000 trainings (in % probability)

Actual data	Estimated with 0.1 dropout	Estimated with 0.2 dropout	Estimated with 0.3 dropout
<i>1000 epochs</i>			
Ref	100	79.73	100
I_{05}	100	0	0.02
I_{10}	100	0.03	0.06
I_{15}	100	0.14	0.04
I_{20}	100	0	99.98
HM_{80}	100	100	99.97
HM_{85}	100	100	99.98
HM_{90}	100	99.73	99.91
HM_{95}	99.92	99.99	0
<i>10,000 epochs</i>			
Ref	16.40	1.0000	100
I_{05}	64.81	0.0001	0.08
I_{10}	38.54	0.0001	0.01
I_{15}	65.89	0.0001	0.04
I_{20}	59.40	0.0001	0.15
HM_{80}	99.06	0.9999	99.96
HM_{85}	79.84	1.0000	99.97
HM_{90}	89.53	1.0000	99.98
HM_{95}	81.03	1.0000	99.94

8 training and validation datasets having 35, 040 observations are generated.

Next to identify the incipient fault, a deep neural network with one input layer, three hidden layers and one output layers are used. The deep neural network is trained for 100 times with $\alpha = 0.01$, and dropout rate varied from 0.1 to 0.3. Here, dropout rates 0.1, 0.2 and 0.3

Table 4 Estimated value of unseen single sensor incipient faults using MC-DNN soft sensor in BSM2 (in % probability)

Actual data	Estimated with 0.1 dropout (100epochs)	Estimated with 0.1 dropout (1000 epochs)
l_{02}	0.50	0.03
l_{04}	98.95	100
l_{06}	100	99.94
l_{08}	99.85	99.99

represent 10%, 20% and 30% dropout, respectively. This is done to obtain the best possible training to identify the incipient faults.

The data of the reference and all the eight fault conditions are normalized and squared before applying to the network. The training data have three output classes: no fault, incipient fault and high magnitude fault. After training, the learning parameters are fixed which along with the deep neural network estimates the classes of the validation data.

The network has estimated faults as in Table 2. Estimated fault is expressed in percentage probability. The optimized MC-DDNN network with dropout 0.1, 0.2 and 0.3 predicts the incipient and abrupt fault correctly with varying probability. For 0.1 dropout, the faults are predicted with high probability, but with 0.2 and 0.3 dropout, the probability percentage is less. The training is therefore increased to 1000 and 10,000 epochs to teach the learning parameters more specifically. The prediction results in Table 3 show that for 0.1 dropout, the faults are predicted correctly with recall 61.53% for both epochs. But with 0.2 dropout, two cases of incipient faults confuse with the no fault condition giving recall of 23.07% and 61.53% and for 0.3 dropout, one high magnitude fault is falsely predicted as no fault condition with recall of 0% and 61.53%. This dropout can be interpreted as an inversely proportional relation to fault information while training the model.

Therefore, the learning parameters of the 0.1 dropout network trained for 100 and 1000 epochs are used to estimate the unseen faults of the same plant. The results in Table 4 show that the optimized MC-DDNN soft sensor estimates the unseen incipient faults correctly even if such small magnitude faults are not available in the training data.

Now since in real plants multiple sensors are involved, we have assumed a situation of multiple sensor fault of the same system. Here, the optimized MC-DDNN soft sensor is used to predict the incipient fault and at the same time identify the faulty sensor.

The dataset is obtained from BSM2 considering 3 DO sensors connected to the third, fourth and fifth aerobic tanks of the biological treatment process and consists of 4 training and validation datasets having 35,040

Table 5 Estimated values of unseen multi-sensor incipient fault using MC-DDNN for three DO sensors connected to BSM2 with 0.1 dropout (in % probability)

Actual data	Estimated fault in DO Sensor 1	Estimated fault in DO Sensor 2	Estimated fault in DO Sensor 3
<i>100epochs</i>			
l_{02}	9.83	13.81	59.91
l_{04}	0.03	1.98	78.80
l_{06}	0.03	9.01	70.60
l_{08}	0.00	0.09	59.51
<i>1000epochs</i>			
l_{02}	11.75	4.90	12.98
l_{04}	11.08	11.50	23.04
l_{06}	11.08	11.51	23.04
l_{08}	11.08	11.51	23.04

observations. Similar process as above is used to acquire the data for no fault, four sets of incipient faults (5%, 10%, 15%, and 20% fault) and four sets of high magnitude faults (80%, 85%, 90% and 95% faults). Faults are introduced in all the three DO sensors separately.

The network is trained for 100 and 1000 epochs with dropout rate 0.1, and the learning parameters are obtained. The number of training epochs and dropout rates is selected based on the performance in the first part of this experiment.

This trained network then identifies the unseen faults of very small magnitude as in Table 5. It is observed that 100 epochs trained MC-DDNN estimates the incipient fault with random probability for all the three sensor faults. But 1000 epochs trained MC-DDNN estimates and identifies the incipient faults in all the three sensors with probability ranging from 4.9% to 23.4% which can be interpreted as the percentage probability of fault occurrence in the respective sensors.

This indicates that fault has occurred in very small magnitude and will alert the plant operators. The proposed MC-DDNN therefore detects and identifies the location of occurrence of incipient fault if multiple sensor fault occurs in the plant.

3.2 Case B

The second dataset is from the effluent treatment plant of Guwahati Oil Refinery. The experiment is performed in three stages:

1. dataset collection from the industry and applying linear interpolation data augmentation to increase the data size
2. introducing multiple sensor faults in the data to generate the training dataset for the network

Table 6 Reference values for DO, pH and MLSS sensors in ETP

Evaluation parameter	Value at normal operating condition
DO	0–2 mg/l
pH	6–8.5
MLSS	3000 mg/l

Table 7 Estimated values of unseen multi-sensor incipient faults using MC-DDNN for DO, pH and MLSS sensors used in ETP (in % probability)

Actual data	Estimated fault in DO Sensor	Estimated fault in pH Sensor	Estimated fault in MLSS Sensor
<i>100epochs</i>			
I_{02}	1.04	14.71	1.87
I_{04}	0.71	0.54	0.04
I_{06}	0.71	0.54	0.05
I_{08}	0.71	0.54	0.05
<i>1000epochs</i>			
I_{02}	11.43	0.07	5.03
I_{04}	1.24	0.09	8.79
I_{06}	1.25	0.09	8.79
I_{08}	1.25	0.09	8.80

3. detecting and diagnosing incipient faults using the optimized MC-DDNN for multiple sensor fault condition.

Three important sensors installed in the plant read the parameters dissolved oxygen (DO), pH and MLSS and are considered for the experiment. Dissolved oxygen and MLSS sensor installed at the aeration tank and pH sensor installed at the output display the value which is remotely stored at sampling rate 1 sample/day.

Two years recorded data of these three sensors are used for verifying the applicability of the proposed optimized MC-DDNN to practically operating industries. The 3 training and validation datasets having 732 observations are increased by linear interpolation and then normalized for developing an optimized and trained MC-DDNN model to detect incipient fault in the DO, pH and MLSS sensors of the plant.

The no fault values of DO, pH and MLSS sensors are shown in Table 6. Faults in DO, pH and MLSS sensors connected to a treatment plant mostly occur due to fouling. Fouling causes drift and bias faults in the sensors [25]. To get a more practical insight of the sensor faults, bias is introduced in the no fault dataset as [4]

$$X_f = X + \beta + noise \quad (13)$$

where X_f is the faulty sensor data, X is the data acquired from the sensor in no fault condition, β is a constant offset value. β is a linear function of the correct sensor data.

The optimized MC-DLSS network is trained using this faulty sensor data for 100 epochs and 1000 epochs and dropout 0.1. The trained network is then used to estimate and identify the unseen sensor incipient fault. The incipient faults of 2%, 4%, 6% and 8% are validated here. It is observed that the probability percentage of estimated fault in DO sensor and pH sensor is very small, but it can still be interpreted as an faulty condition and hence alert the plant operators that there is a probability of incipient fault in the specified sensor. The MLSS sensor fault is indicated with a probability of 5.03% to 8.80% which correctly indicates the very small magnitude of the sensor fault. Table 7 shows the predicted probability of the detected faults in terms of percentage which overcome the challenge of detecting very small magnitude fault. The small magnitude incipient fault is sometimes detected as fault and sometimes detected as noise, but since the MC-DDNN network works as an ensemble of models, this is combined to detect such faults with less percentage probability instead of misinterpreting as noise. Therefore, the proposed method detects and identifies the sensor incipient faults in industrial sensors too.

4 Conclusions

The optimized MC-DDNN detects and identifies sensor incipient faults in a BSM2 simulation set-up considering both single sensor fault and multi-sensor fault scenarios. In the unseen single sensor fault condition, both 100 epochs and 1000 epochs trained networks with 0.1 dropout detect and identify the sensor incipient faults satisfactory. But in the unseen multi-sensor fault condition, the 1000 epochs trained networks with 0.1 dropout give better detection results. The fault magnitude can be interpreted by the estimated probability percentage in this case. The optimized MC-DDNN is also applied to a real-time industrial ETP dataset of Guwahati Oil Refinery where monitoring is very important. The unseen multi-sensor fault condition is detected and identified using both 100epochs 1000epochs trained networks with 0.1 dropout and fault magnitude can be interpreted by the estimated probability percentage. Because of the very small magnitude of the incipient faults and the small size of training data, the incipient fault estimated probability is also small. This can be improved by decreasing the sampling time of the sensors or increasing the training data size so that the MC-DDNN can have better training. Hence, the proposed optimized MC-DDNN proves its worth in the automation of real industrial plants and can be applied to nonlinear industries for fault detection and diagnosis.

Acknowledgements The authors acknowledge the support of Dr. I Santin, Prof. Ulf Jeppson and his team for the BSM2 and IOCL Guwahati for allowing them to access the ETP data.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

- Aggarwal CC (2018) Neural networks and deep learning, vol 10. Springer, Berlin, pp 978–983
- Alex J, Benedetti L, Copp J, Gernaey K, Jeppsson U, Nopens I, Pons M, Rosen C, Steyer J, Vanrolleghem P (2008) Benchmark simulation model no. 2 (bsm2). Report by the IWA Taskgroup on benchmarking of control strategies for WWTPs pp 1–99
- Allahverdi F, Ramezani A, Forouzanfar M (2020) Sensor fault detection and isolation for a class of uncertain nonlinear system using sliding mode observers. *Automatika* 61(2):219–228
- Balaban E, Saxena A, Bansal P, Goebel KF, Curran S (2009) Modeling, detection, and disambiguation of sensor faults for aerospace applications. *IEEE Sens J* 9(12):1907–1917
- Baldi P, Sadowski P (2014) The dropout learning algorithm. *Artif Intell* 210:78–122
- Chen H, Wu J, Jiang B, Chen W (2020) A modified neighborhood preserving embedding-based incipient fault detection with applications to small-scale cyber-physical systems. *ISA Trans* 104:175–183
- Cheng H, Wu J, Liu Y, Huang D (2019) A novel fault identification and root-causality analysis of incipient faults with applications to wastewater treatment processes. *Chemom Intell Lab Syst* 188:24–36
- Cheng Z, Gao M, Liang X, Liu L (2020) Incipient fault detection for the planetary gearbox in rotorcraft based on a statistical metric of the analog tachometer signal. *Measurement* 151:107069
- Dong Y (2019) Implementing deep learning for comprehensive aircraft icing and actuator/sensor fault detection/identification. *Eng Appl Artif Intell* 83:28–44
- Gautam S, Tamboli PK, Roy K, Patankar VH, Duttagupta SP (2019) Sensors incipient fault detection and isolation of nuclear power plant using extended kalman filter and kullback-leibler divergence. *ISA Trans* 92:180–190
- Goodfellow I, Bengio Y, Courville A (2016) Deep Learning, vol 1, no 2. MIT Press, Cambridge
- Ji H, Huang K, Zhou D (2019) Incipient sensor fault isolation based on augmented mahalanobis distance. *Control Eng Pract* 86:144–154
- Jin B, Li D, Srinivasan S, Ng SK, Poolla K, Sangiovanni-Vincentelli A (2019) Detecting and diagnosing incipient building faults using uncertainty information from deep neural networks. In: 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE, pp 1–8
- Khapra MM (2018) Deep learning. NPTEL video lecture, www.nptel.in
- Kiasi F, Prakash J, Shah S (2015) Detection and diagnosis of incipient faults in sensors of an Ili system using a modified glr-based approach. *J Process Control* 33:77–89
- Kim P (2017) Deep learning. In: MATLAB deep learning: with machine learning, neural networks and artificial intelligence, 1st edn. Apress, pp 1–120. ISBN-13 (pbk): 978-1-4842-2844-9, ISBN-13 (electronic): 978-1-4842-2845-6. <https://doi.org/10.1007/978-1-4842-2845-6>
- Lu Y (2017) Industry 4.0: A survey on technologies, applications and open research issues. *J Indus Inf Integr* 6:1–10
- Mamandipoor B, Majd M, Sheikhalishahi S, Modena C, Osmani V (2020) Monitoring and detecting faults in wastewater treatment plants using deep learning. *Environ Monit Assess* 192(2):148
- Mao W, Chen J, Liang X, Zhang X (2019) A new online detection approach for rolling bearing incipient fault via self-adaptive deep feature matching. *IEEE Trans Instrum Meas* 69(2):443–456
- Mao W, Ding L, Tian S, Liang X (2020) Online detection for bearing incipient fault based on deep transfer learning. *Measurement* 152:107278
- Mehranbod N, Soroush M, Panjapornpon C (2005) A method of sensor fault detection and identification. *J Process Control* 15(3):321–339
- Munikoti S, Das L, Natarajan B, Srinivasan B (2019) Data-driven approaches for diagnosis of incipient faults in dc motors. *IEEE Trans Indus Inform* 15(9):5299–5308
- Nor NM, Hassan CRC, Hussain MA (2019) A review of data-driven fault detection and diagnosis methods: applications in chemical process systems. *Rev Chem Eng* 63(4):513–553
- Ren L, Xu Z, Yan X (2010) Single-sensor incipient fault detection. *IEEE Sens J* 11(9):2102–2107
- Samuelsson O, Björk A, Zambrano J, Carlsson B (2018) Fault signatures and bias progression in dissolved oxygen sensors. *Water Sci Technol* 78(5):1034–1044
- Sang J, Guo T, Zhang J, Zhou D, Chen M, Tai X (2020) Incipient fault detection for air brake system of high-speed trains. *IEEE Trans Control Syst Technol*. <https://doi.org/10.1109/TCST.2020.3027673>
- Shang J, Zhou D, Chen M, Ji H, Zhang H (2019) Incipient sensor fault diagnosis in multimode processes using conditionally independent bayesian learning based recursive transformed component statistical analysis. *J Process Control* 77:7–19
- Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(1):1929–1958
- Wei Y, Wu D, Terpenney J (2020) Robust incipient fault detection of complex systems using data fusion. *IEEE Trans Instrum Meas* 69(12):9526–9534
- Wu Y, Jiang B, Lu N, Yang H, Zhou Y (2017) Multiple incipient sensor faults diagnosis with application to high-speed railway traction devices. *ISA Trans* 67:183–192
- Wu Y, Jiang B, Wang Y (2020) Incipient winding fault detection and diagnosis for squirrel-cage induction motors equipped on crh trains. *ISA Trans* 99:488–495
- Zhang C, Peng K, Dong J (2020a) An incipient fault detection and self-learning identification method based on robust svdd and rbm-pnn. *J Process Control* 85:173–183
- Zhang K, Jiang B, Yan XG, Mao Z (2016) Sliding mode observer based incipient sensor fault detection with application to high-speed railway traction device. *ISA Trans* 63:49–59
- Zhang W, Li X, Ding Q (2019) Deep residual learning-based fault diagnosis method for rotating machinery. *ISA Trans* 95:295–305
- Zhang W, Li X, Jia XD, Ma H, Luo Z, Li X (2020b) Machinery fault diagnosis with imbalanced data using deep generative adversarial networks. *Measurement* 152:107377

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.