Nuclear Engineering and Technology 53 (2021) 1181–1188

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Abnormal state diagnosis model tolerant to noise in plant data

Ji Hyeon Shin, Jae Min Kim, Seung Jun Lee*

Department of Nuclear Engineering, Ulsan National Institute of Science and Technology, 50, UNIST-gil, Ulsan, 44919, Republic of Korea

ARTICLE INFO

Article history:
Received 3 March 2020
Received in revised form
28 August 2020
Accepted 20 September 2020
Available online 23 September 2020

Keywords:
Accident diagnosis
Nuclear power plant
Abnormal operating procedure
neural network

ABSTRACT

When abnormal events occur in a nuclear power plant, operators must conduct appropriate abnormal operating procedures. It is burdensome though for operators to choose the appropriate procedure considering the numerous main plant parameters and hundreds of alarms that should be judged in a short time. Recently, various research has applied deep-learning algorithms to support this problem by classifying each abnormal condition with high accuracy. Most of these models are trained with simulator data because of a lack of plant data for abnormal states, and as such, developed models may not have tolerance for plant data in actual situations. In this study, two approaches are investigated for a deep-learning model trained with simulator data to overcome the performance degradation caused by noise in actual plant data. First, a preprocessing method using several filters was employed to smooth the test data noise, and second, a data augmentation method was applied to increase the acceptability of the untrained data. Results of this study confirm that the combination of these two approaches can enable high model performance even in the presence of noisy data as in real plants.

1. Introduction

A nuclear power plant (NPP) comprises a large variety of systems such as the reactor coolant system, the main steam system, and so on. Within each system, there are hundreds of components such as pumps, valves, and tubes, and each of these components can exhibit errors such as trip, leakage, or failure that can lead to an abnormal NPP state. If an abnormal state exceeding an allowable range is detected, alarms and plant parameter changes inform operators of the corresponding plant state. In this way, the NPP can be put into an abnormal state for a variety of reasons. When such a state occurs, operators have to conduct the appropriate abnormal operating procedure (AOP) to return the NPP to a normal state [1].

States of an NPP consist of a large number of key parameters and hundreds of alarms. Entry conditions to the AOPs are guided by alarms and symptoms to determine which specific abnormal state the current plant state corresponds to. For example, there are a total of 82 AOPs in the case of the Advanced Power Reactor 1400, with many AOPs including two or more stages. In the event of alarms with corresponding parameter changes, operators have to choose the appropriate AOP based on whether the current plant state satisfies particular entry conditions. To do so, operators must recognize and judge a large amount of information within a given time. Such a task can be burdensome for operators [2,6].

In order to relieve this difficulty in abnormal state diagnosis, many operator support systems have been developed. Most recently, deep learning has been applied to the classification of plant states. Deep learning refers to a type of artificial intelligence algorithm mainly used to classify datasets with neural networks. When applied in the nuclear field, as a first step, models are generally trained with datasets including all plant parameter information that corresponds to each abnormal state. When an abnormality similar to model training occurs, all plant parameter information is used as input data, and the deep-learning model predicts the specific abnormal state based on a trained weight structure. In addition to this type of abnormal state diagnosis, deep learning enables the classification of other various states, such as transient, through the same model algorithm [3–5].

Kwak et al. [7] proposed using convolutional neural networks (CNN) algorithms for crack detection. The CNN is one type of deep-learning algorithm that excels in learning and classifying image features. The above work showed that, after training the model with video frame images, the CNN is able to detect cracks in the reactor with good performance at about 98.3% accuracy to support operators with NPP component inspection. Yang et al. [8] classified five initiating events of an NPP using a long short-term memory deep-learning algorithm, with the trained model classifying the labels with over 80% accuracy. As these studies demonstrate,

* Corresponding author.
E-mail address: sjlee420@unist.ac.kr (S.J. Lee).

https://doi.org/10.1016/j.net.2020.09.025

classification can be attempted using various types of deep-learning algorithms, but performance can differ for each algorithm type depending on the characteristics of the input data to be classified.

The accuracy of any given model is determined not only by the choice of deep-learning algorithm, but also by the preprocessing of the training dataset. There are over 2000 parameters in a plant, which all need to be normalized in various ways for data preprocessing; the range of parameter values varies greatly, such as comparing a binary parameter to a numerical parameter. Because of this, the ranges of some parameters exceed 1000 with various unit sizes. It is thus important to select the important or representative parameters so that the model only considers necessary data for more efficient classification.

The specific type of artificial neural networks selected for these deep-learning models is largely determined by the type and amount of data for training. In order for the model to be trained reliably, the data must vary within the same label, while additionally the amount of data in each label must be equal. Only a small fraction of abnormal states has occurred in NPPs generally designed with safety as a top priority. For this reason, there are no abnormality cases for all the thousands of components, and thus, this presents a challenge to consider abnormal situations. It is impossible to obtain data by artificially generating an abnormal situation that has not occurred in a real NPP. In other words, the amount of existing real plant data corresponding to each abnormal state is currently insufficient and uneven to train deep-learning models. For this reason, most prior research has used simulator data for model training. Simulators are useful to not only produce as much data as desired for abnormal cases, but also to adjust for abnormality degree, ramp time, and so on.

However, a problem arises in that real plant data can vary depending on data issues such as output noise, outliers, and the offset of transmitters. Fig. 1 below shows an example of variation in flow rates between simulator and plant data for the same abnormal state. While the trends are similar, the plant data contain some noise, while the simulator data do not. It is unknown whether most models trained with simulator data can yield tolerant performance in prediction using plant data containing noise, and accordingly, consideration of the differences between plant and simulator data is required before applying diagnosis models to real NPPs. With respect to the deep-learning model in this work, the diagnostic performance of our gated recurrent unit (GRU) network model deteriorated with increasing levels of general noise in the test data.

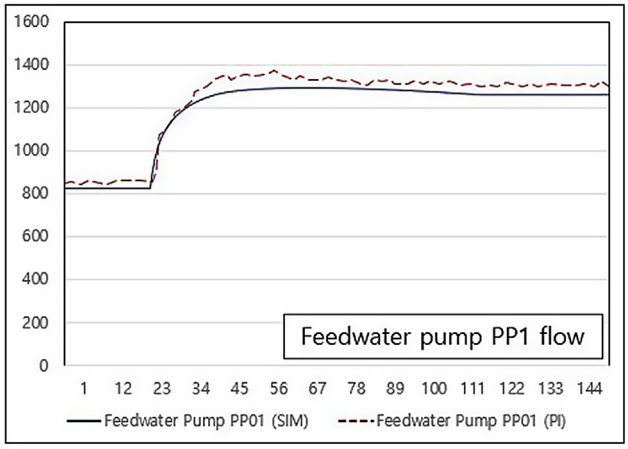

Fig. 1. Flow rate changes in the feedwater pump for plant data (red) and simulator data (blue) in a main feedwater pump trip state [9]. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

In this study, we suggest a way for a deep-learning model trained with simulator data to overcome the performance degradation caused by noise in actual NPP data. To improve the degraded model performance following noise inclusion, two methods were tested: data preprocessing and augmentation. When the test data with noise was preprocessed through smoothing filters, the accuracy of the model moderately increased compared with no preprocessing. Then, by training with noisy data along with existing data, the modified model was able to diagnose untrained test data, showing improved performance.

2. Base model

In [10,11], Kim et al. found that the performance of the principal component analysis (PCA) data preprocessing method showed high accuracy in a GRU model for abnormal state diagnosis in NPPs. GRU networks consist of two gates, the update gate and the reset gate, which determine the information to be considered for the output. This architecture can solve the vanishing gradient problem of recurrent neural networks [12,13]. In addition, as GRU has a recurrent loop structure like a recurrent neural networks, it is suitable to consider time series data. Therefore, as the current study considers parameter information observed over time, the GRU networks was chosen as the algorithm for the base model [14,15].

To determine the effect of noise on the GRU model, this study needed a model with good performance in abnormal state classification using simulator data. However, when we trained rough data for over 1000 parameters in the basis GRU model, the model accuracy was very poor at 13.3%. Therefore, it was necessary to improve the performance of the model by only using the principal components of the data. Generally, PCA is a way to reduce data dimensions while maintaining as much information as possible, and is thus a good tool to solve the curse of high dimensionality by linearly transforming data onto a new axis that preserves the variance of the raw data [16]. With the PCA preprocessing method, the large amount of information from the over 1000 parameters chosen in this study can be concisely expressed. Based on this, among various algorithms and preprocessing methods, the basic model of the current study employed a GRU algorithm and the PCA preprocessing method.

2.1. Basic model training

The training dataset was generated using a generic pressurized water reactor 1400 MWe simulator made by the Western Services Corporation [17]. A total of eight abnormal states were labeled in the generated data: steam generator tube rupture (SGTR), charging water system (CHRG) abnormality, letdown water system (LTDN) abnormality, condensate system (CDS) abnormality, pilot-operated safety relief valve (POSRV) abnormality, circulating water system (CWS) abnormality, main steam isolation valve (MSIV) abnormality, and main steam system (MSS) abnormality. Each state contained 200 datasets that covered about 1004 parameters over 60 s for use in model training.

The data produced above was then used to train the GRU model; Table 1 lists the model hyper-parameters. In data preprocessing, the normalization result of the test data should not be drastically influenced by the noise to be added; therefore, this study judged that normalization that only fits the training data is not suitable for incoming test data. First, all data was normalized using the maximum and minimum values for the general broad ranges each parameter value can have, and the principal components of the data were extracted with 20 components for each data by PCA as

Table 1
Experimental model hyper-parameters.

Input shape	(60, 20)
Nodes in hidden layers	100
Activation function (GRU)	tanh
Recurrent activation function (GRU)	sigmoid
Kernel initializer	Xavier Initialization
Recurrent initializer	Orthogonal
Bias initializer	Zeros
Activation function (Dense)	Softmax
Loss function	Categorial cross-entropy
Optimizer	Adam
Learning rate	0.001
Epoch	100 epochs with early stopping
Batch size	28

mentioned above.

The results of training the GRU model with all preprocessed datasets are as follows. As shown in Fig. 2(a), the model achieved over 90% accuracy for all labels. The confusion matrix in Fig. 2(b) shows that the model classified the validation datasets with good performance. Based on these results, it was determined that the GRU model can effectively diagnose abnormal states, and therefore this research was conducted using this model with a test dataset that includes noise, as described in the next section.

2.2. Effect of noise

As seen in Fig. 1, plots of the plant parameters are not smooth due to shifting and noise. In this study, among the various causes of discrepancy between simulator data and plant data, the effect of noise on the GRU model was examined. Because of the confidential nature of actual NPP data, the data in Fig. 1 was referenced in this study. Since all noise types for each instrument in NPPs could not be considered, we used white noise for the instruments, which is an additive noise that does not have an outlier and is not constantly at the same level. For that, a random number having a standard distribution for the x% size of each parameter value was added to the simulator data. In other words, the noise of actual power plant data was replaced with Gaussian noise.

Using the average of each parameter in the learning interval (60 s), the degree of each parameter shift in the plant data from the simulator data can be estimated. In this study, the difference between shifted values was ignored by zeroing using the calculated shifting degree. Then, by calculating the absolute values of the differences between the plant data and the simulator data, the degree of noise in the plant data could be measured. In order to determine the effect of this noise level, test datasets were generated to contain 1–5% Gaussian noise. The accuracy of the model with an increase in noise was found as follows.

As a result of calculation, the plant data had an average noise degree of about 0.0135 compared to the simulator data. Expressed as Gaussian noise, this is about 2.91%, which was subsequently the level of noise applied to the test data in this research in order to confirm model performance before application to a real power plant.

As seen in Fig. 3, the accuracy of the model drastically decreased as the degree of noise in the test data increased. With even just 1% noise, the accuracy dropped from over 99% to under 79%. In other words, a model trained only with simple simulator data may be very poor in terms of stability with the addition of even minor levels of noise to the data. Furthermore, the accuracy of the model tested with data containing 2% and 3% noise, which is similar to the noise level of actual power plant data as described above, was 55.55% and 41.15%, respectively. In practice, based on these results, even if a given model shows a high performance of over 90% with simulator data, its accuracy when applied to real plant data may only be about 40%. It is therefore necessary to mitigate the effect of noise on diagnosis models.

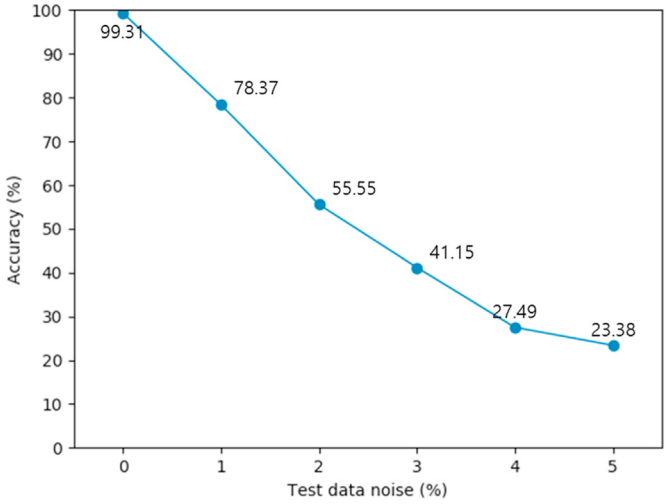

Fig. 3. Model accuracy by degree of noise.

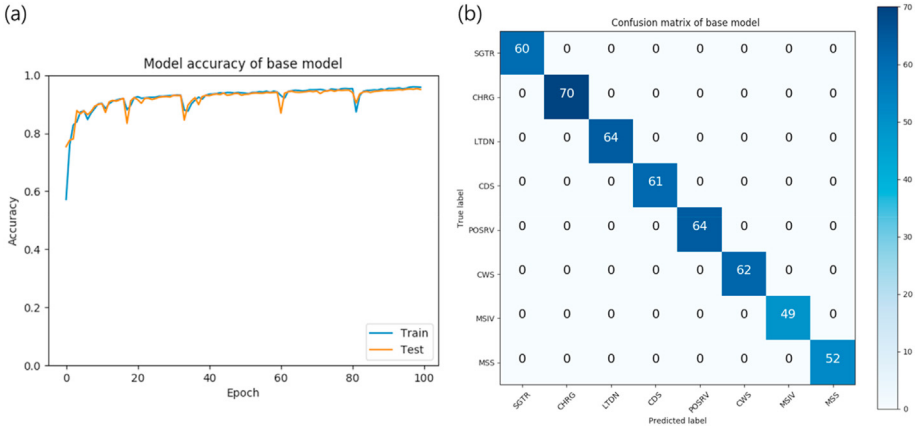

Fig. 2. (a) GRU model accuracy and (b) confusion matrix.

3. Methodology

3.1. Smoothing filters for noise

One method commonly employed to solve noise problems in data is the use of smoothing filters. Noisy data can be smoothed and cleaned by passing it through several kinds of filters [18,19]; this work separately applied four different types of smoothing filters for odd window, the size including the center and sides. The first was the moving average (MA) filter, in which the result of filtering $y_{MA,i}$ can be calculated by Eq. (1) as:

$$y_{MA,i} = \frac{1}{w}\sum_{j=0}^{w-1} x_{i-j+\frac{w-1}{2}} \tag{1a}$$

for some parameter value x_i at time j. Here, w is the window size of the filter that has the same weight at all window points. Window size means number of parameter values in the filter. In order to minimize the difference between filtering result and raw data, the average of all point values in the window is equivalent to the filtering result of the point value located in the center of the window.

The second filter was the triangular moving average (TMA) filter, in which the $y_{TMA,i}$ filtering result is given as the average of the MA filter, as follows in Eq. (2).

$$\text{If, } y_{MA,i} = \frac{1}{w}\sum_{j=0}^{w-1} x_{i-j+\frac{w-1}{2}} \tag{1b}$$

$$\text{then, } y_{TMA,i} = \frac{1}{w}\sum_{k=0}^{w-1} y_{MA,i-k+\frac{w-1}{2}} \tag{2}$$

When calculating Eq. (2), the weights of the window points create a triangle. In other words, the further away from the point to be filtered, that is the middle point of the window section, the lower the weight.

The third filter was the Gaussian filter, in which each point in the window has a weight corresponding to the G value in a Gaussian distribution. For the following one-dimensional Gaussian distribution equation (Eq. (3)), the σ value of the Gaussian distribution is determined according to the window size w:

$$G(j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{j^2}{2\sigma^2}} \tag{3}$$

Similar to the MA filter, the filtering result $y_{G,i}$ can be calculated as an average of values multiplied by each weight and parameter value in the window, as follows in Eq. (4).

$$y_{G,i} = \frac{1}{w}\sum_{j=-\frac{w-1}{2}}^{\frac{w-1}{2}} G(j)\cdot x_{i+j} \tag{4}$$

The last filter used was the Savitzky–Golay filter, which has a coefficient of local least-square polynomial fit as the weight for each point in the window [20]. While applying various other types of filters such as finite impulse response or Kalman filters may be possible, this work preprocesses the noisy data using the above four filters that are most commonly used for general noise. Fig. 4 plots the results of smoothing the noisy data with the four filters.

It should be noted that one important aspect in the smoothing filters is the window size, which refers in each above equation to the total number of points, including the surrounding values, that are referenced to filter out the corresponding point values with noise. The degree of smoothing depends on the window size: the larger the window of the filter, the less noise in the data trends by preprocessing. However, an excessive window size may affect the trend characteristics of the data by excessive smoothing. Therefore, it is necessary to select an appropriate window size. When we tested each smoothing filter with window sizes of 5 s, 7 s, and 9 s with a simple classification model, the model showed better noise data classification with the 9 s filtering window size. For longer window sizes, any data points that change after the 10 s window from changes in NPP abnormalities are reflected as one data point; we judged that this was not suitable for our diagnosis model using data of short length, 60 s. For this reason, the window size in each filter was set to 9 s in this work.

3.2. Data augmentation

Image classification can be conducted with a deep-learning algorithm like a CNN. However, even images with the same label can be easily mislabeled by issues such as a deformation of the shape, a darker image, different angles of the same object, cropping, and so on. While models should be able to classify deformed or altered images with the same label, it is difficult for a trained model to make good judgments about untrained data. Similarly, for the classification of time series data, model diagnosis may become inaccurate for data with increased complexity; above, for example, we confirmed that the diagnostic performance of the GRU model was poor for simulator data containing noise. Alternatively stated, when a model classifies new data, the greater the variance of the shape of the new data from the training data, the harder it will be to make the right judgment.

In image classification, limited amounts of training data can be transformed in various ways, such as left/right inversion or brightness adjustment, to make the images as diverse as possible [20]. Used to increase the amount of training data, the data augmentation process allows a given model to be prepared for new data that differs from its limited training data. In this way, when the amount of data per label is insufficient to classify various types of images, data augmentation is a way to inflate the raw data to improve the performance of the model [21–24]. Similar to this approach, time-series data can be transformed into various data by adding noise, inverting, or changing the scale of the data values [25]. Fig. 5 shows an example of how one time-series data can be augmented into four different data in various ways.

This study applied similar data augmentation to simulator data for training the GRU model [26]. Simulator data is not wide enough to classify plant data with shifted differences or other variation, and it is also assumed that differences between plant and simulator data cannot be known by any other means. Because the shape of plant data is unknown, this study used data augmentation to add random values to the data values, i.e. augmented the data with noise. Gaussian noise, the most common form, was added to the simulator data to double the quantity of the training data. With the use of such augmented data for training, it was necessary to verify that the model was able to be well trained and that the trained model could classify not only the trained data but also untrained noisy data. Which particular augmented data type that best covers the noise of plant data should also be determined.

3.3. Experiment setup

Preprocessing with a smoothing filter is one approach to reduce the degree of noise in test data and represents Method 1 in this work. This step was performed before the other preprocessing steps, namely normalization and PCA. Method 2 involved data augmentation, in which the limited amount of training data was

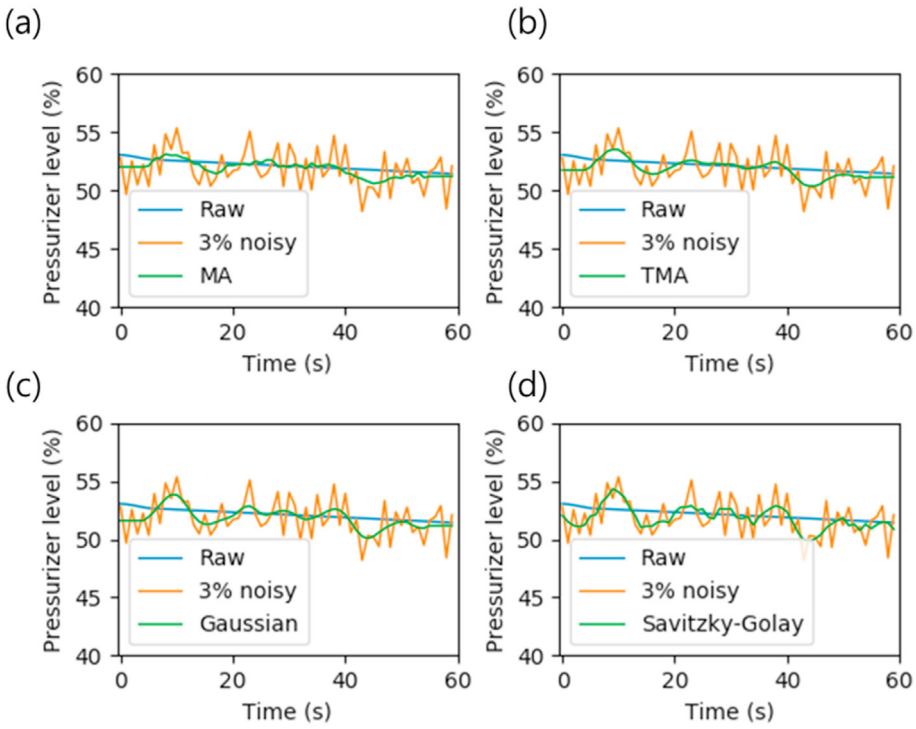

Fig. 4. Preprocessing results with (a) MA filter, (b) TMA filter, (c) Gaussian filter, and (d) Savitsky–Golay filter.

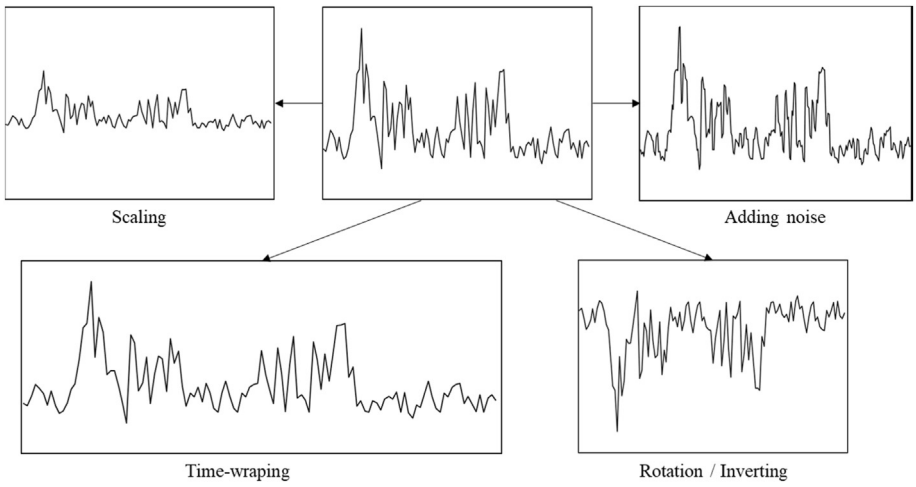

Fig. 5. Example of data augmentation [20].

increased in quantity to improve the performance of classification of untrained test data with noise. Augmentation was conducted between the training data generation and preprocessing steps. After testing these two methods separately, this study then combined them into Method 3 with optimized features for better results. An overview of each method is shown in Fig. 6.

4. Results

4.1. Data preprocessing (Method 1)

The first method used test datasets with 1–5% noise preprocessed through four kinds of smoothing filters. The window size of all filters was 9 s. The test datasets were predicted by a base GRU model trained with raw simulator data.

Table 2 shows the noise degrees as calculated by the noise measurement method in Section 2.2. The mean results are given for the four filters for test datasets with 1–5% noise added. With the MA filter, which gave the overall largest decreases, the noise degrees lowered to 0.93% and 2.24% for the 1% and 5% raw data noise degrees. As can be seen from the table, the noise degrees of all the data preprocessed with the filters were reduced.

For the 1–5% noisy data preprocessed with smoothing filters, the overall model accuracy improved by an average of 11.5%, from each improvement in Fig. 7. The MA filter, which gave data the most similar to the raw data (as discussed in the previous paragraph, also see Fig. 4(a)), showed the highest performance improvement. However, the performance improvement with this method was

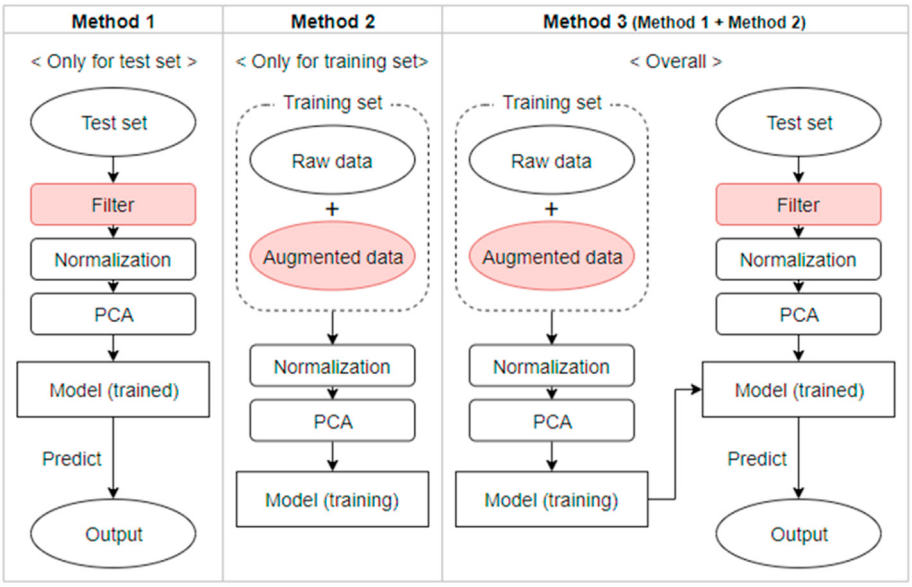

Fig. 6. Overview of experimental approach.

Table 2
Noise degree after preprocessing using each filter.

No filter	1%	2%	3%	4%	5%
Moving avg.	0.93%	1.25%	1.57%	1.92%	2.24%
Triangular moving avg.	0.93%	1.27%	1.64%	2.03%	2.37%
Gaussian	0.97%	1.38%	1.79%	2.22%	2.63%
Savitzky–Golay	0.80%	1.31%	1.85%	2.37%	2.89%

slight when compared to the preprocessing load required by each filter, and furthermore, although the noise degree of all parameters was reduced, errors different from real parameter tendencies may occur by the window size of the filter.

4.2. Data augmentation (Method 2)

This second method verified the performance of the model trained with noisy data together with existing training data. In other words, the model was trained with twice as much training data as the model in Method 1. Table 3 shows the details of the data augmentation for each training dataset.

Table 4 shows the accuracy of classification for test datasets with 0–5% noise added into each model trained with augmented training datasets 1 to 3.

Results showed a high accuracy for test data with noise similar to that of the augmented training data. In addition, it showed a high accuracy for test data without noise; that is, it performed well for

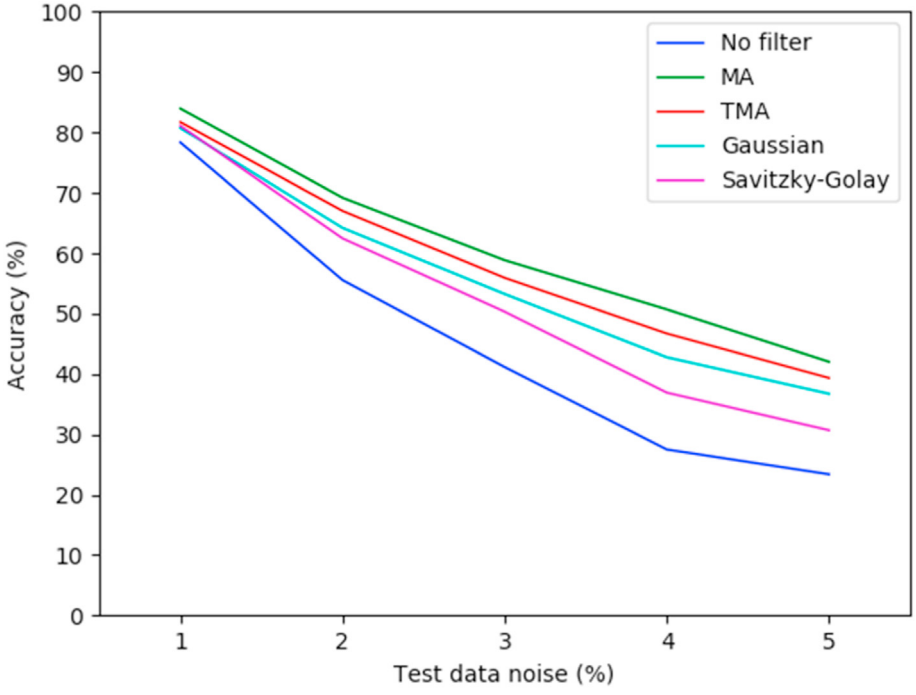

Fig. 7. Model accuracy following data preprocessing.

Table 3
Data augmentation.

Name	Augmented type	Augmented data	Total data of each label
Raw dataset	None	None	200
Training dataset 1	Added 1% noise	200	400
Training dataset 2	Added 2% noise	200	400
Training dataset 3	Added 3% noise	200	400

Table 4
Model accuracy following training data augmentation.

Test data noise (%) Training dataset	1%	2%	3%	4%	5%
None	78.37%	55.55%	41.15%	27.49%	23.38%
Training dataset 1	97.19%	78.37%	63.78%	53.12%	44.89%
Training dataset 2	87.90%	93.95%	76.06%	66.21%	58.10%
Training dataset 3	85.85%	83.98%	92.08%	72.07%	66.02%

the trained level of data. Even for untrained levels of noise, the test data with 5% noise showed a good improvement in accuracy, as seen with the 66% level of accuracy. In other words, it can be seen that data augmentation for training can improve classification performance even for untrained data. On average over the 1–5% noisy test data, this method resulted in a 29.45% accuracy improvement, which was better than the previous filtering approach.

4.3. Data preprocessing and augmentation (Method 3)

The third method was to apply Methods 1 and 2 simultaneously. Such a combination was thought to be advantageous as data augmentation improves the tolerance of the model itself to noise contained in the test data, while preprocessing with filters allows the model to classify plant states more easily by lowering the noise degree of the test data.

With noisy data augmentation, the model learns with the same label for data that are different from the raw data. Like this, when raw data and the data with a large noise difference are trained together, overfitting may occur because of the increasing complexity of the training data. Therefore, an appropriate training epoch number is required. In this study, in order to prevent overfitting, which means focusing on training data only, model training was stopped when the loss values for the validation dataset did not further decrease after a certain number of iterations. In Table 5, model tuning was additionally performed by adjusting the node number and dropout degree to obtain the best improvement by preventing overfitting. By reducing the node number in the GRU model layer, the model capacity was adjusted, which reduced the complexity of the artificial neural networks. In addition, some of the networks were dropped so that model did not focus on specific neurons.

The training datasets 1 to 3 in this case were the same as in Method 2 (Table 3). As shown in Table 5, results showed high accuracy overall, higher than using either of the other methods.

Table 5
Model accuracy following model tuning for performance improvement.

Model		Hyperparameter		Test data noise (%)				
Training dataset	Filter	Nodes	Dropout	1%	2%	3%	4%	5%
Training dataset 1	Moving average Filter	100	0	90.77%	86.91%	84.35%	78.18%	72.63%
		100	0.3	91.08%	87.09%	83.92%	77.31%	71.70%
		100	0.5	90.02%	86.35%	84.48%	77.87%	72.82%
		50	0	90.09%	87.59%	83.42%	77.37%	71.13%
		50	0.3	91.33%	87.66%	85.10%	77.43%	70.95%
		50	0.5	91.46%	88.09%	85.16%	77.74%	71.95%
		30	0	91.40%	87.84%	83.60%	76.62%	72.26%
		30	0.3	91.15%	86.91%	83.10%	77.49%	71.38%
		30	0.5	91.40%	87.22%	82.86%	77.49%	71.13%
Training dataset 2	Moving average Filter	100	0	84.29%	79.24%	77.37%	76.25%	75.12%
		100	0.3	87.53%	81.86%	82.23%	79.05%	77.93%
		100	0.5	86.60%	81.48%	80.49%	78.62%	77.56%
		50	0	89.84%	87.22%	83.17%	79.68%	76.62%
		50	0.3	85.91%	83.85%	82.54%	79.05%	77.06%
		50	0.5	86.66%	83.48%	81.61%	78.68%	77.00%
		30	0	87.97%	85.47%	83.73%	80.86%	77.68%
		30	0.3	87.34%	84.10%	83.92%	80.05%	76.93%
		30	0.5	85.04%	81.23%	83.04%	79.30%	77.43%
Training dataset 3	Moving average Filter	100	0	84.66%	79.47%	78.62%	76.93%	75.56%
		100	0.3	86.41%	80.49%	79.99%	76.37%	75.00%
		100	0.5	84.54%	80.36%	79.30%	76.25%	74.81%
		50	0	88.40%	83.73%	81.23%	76.31%	72.94%
		50	0.3	89.96%	85.10%	82.92%	79.49%	77.31%
		50	0.5	86.91%	83.85%	82.23%	79.86%	77.93%
		30	0	88.28%	84.73%	84.04%	79.43%	77.87%
		30	0.3	89.46%	86.41%	84.98%	79.93%	77.81%
		30	0.5	88.84%	85.41%	84.41%	80.55%	78.93%

Notably, over 70% accuracy was achieved for the 5% noisy data, compared to the 23.38% accuracy of the base model.

Table 3 shows that the MA filter in Method 1 was able to smoothen the 1% and 3% noisy data to 0.97%–1.57%, respectively. Results of Method 2 showed the highest accuracy for training dataset 1 (added 1% noisy data) for low levels of noise. Therefore, as seen in Table 5, the combination applying of the MA filter for preprocessing and 1% noisy data augmentation gave the best result for the 1% to 3% noisy test data.

However, the third combination result, with 3% noisy data augmentation, showed a low accuracy. Table 5 shows that data augmentation with 3% noise resulted in relatively lower performance improvement for the test data with 1%–2% noise. This indicates that data augmentation with unnecessary or excessive noise lowers the accuracy improvement compared to appropriate noisy data augmentation when testing low-noise data through the MA filter.

Consequently, an appropriate combination of the approaches is needed through an analysis of plant noise levels. Previous measurements inferred that plant data would have about 2.91% noise when compared to simulator data with Gaussian noise; accordingly, the combination of the MA filter with 1% noise data augmentation and the model with 0.5 dropout using 50 nodes, which consistently performed the best at 2%–3% noise levels, should be used when applying the model to real plants.

5. Discussion and conclusion

In this study, we proposed a method for diagnosing abnormal states from actual NPP data using the developed GRU model. The focus is on reducing the difference between the training data and the actual data to be diagnosed. First, the noise contained in the actual data, which is the data to be diagnosed, was preprocessed by smoothing. Then, by adding example data including noise to the training data, model robustness against actual data noise was effectively improved.

Using these two methods simultaneously, this study was able to calibrate the trained model to make the test data smoother and to be more tolerant to noise. An appropriate combination increased model performance as compared with either individual method, to achieve over 70% accuracy for a wide range of noise in the test data. Specifically, the combination using training data with 1% noise added to the raw data and preprocessing the test data with the moving average filter showed the best performance in the overall noise range. The proposed model is expected to help recently developed deep-learning models overcome the noise level of real NPPs.

While the method proposed in this study showed good performance, certain issues remain to be solved before application of the deep-learning model to actual NPPs. It should be noted that there exist other kinds of differences between simulator and plant data in addition to noise, such as shifted values by different zero points, offset of a transmitter, or outliers, which represent characteristics of real data for which future considerations will be essential. In addition, this study only conducted the classification of 10 abnormal state labels. However, as there are 82 AOPs in the APR-1400, for example, real NPPs clearly require greater classification of abnormalities. Therefore, the proposed method needs to be revalidated for deep-learning models extended to the classification of the entire range of AOPs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No.NRF-2018M2B2B1065653), and also by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20171510102040).

References

[1] S.J. Lee, P.H. Seong, Development of automated operating procedure system using fuzzy colored petri nets for nuclear power plants, Ann. Nucl. Energy 31 (8) (2004) 849–869.
[2] S.J. Lee, P.H. Seong, A dynamic neural network based accident diagnosis advisory system for nuclear power plants, Prog. Nucl. Energy 46 (3–4) (2005) 268–281.
[3] S.G. Vinod, A.K. Babar, H.S. Kushwaha, V.V. Raj, Symptom based diagnostic system for nuclear power plant operations using artificial neural networks, Reliab. Eng. Syst. Saf. 82 (1) (2003) 33–40.
[4] T.V. Santosh, G. Vinod, R.K. Saraf, A.K. Ghosh, H.S. Kushwaha, Application of artificial neural networks to nuclear power plant transient diagnosis, Reliab. Eng. Syst. Saf. 92 (10) (2007) 1468–1472.
[5] K. Mo, S.J. Lee, P.H. Seong, A dynamic neural network aggregation model for transient diagnosis in nuclear power plants, Prog. Nucl. Energy 49 (3) (2007) 262–272.
[6] S.J. Lee, P.H. Seong, Development of an integrated decision support system to aid cognitive activities of operators, Nuclear Engineering and Technology 39 (6) (2007) 703–716.
[7] J.T. Kwak, S.M. Hewitt, Nuclear architecture analysis of prostate cancer via convolutional neural networks, IEEE Access 5 (2017) 18526–18533.
[8] J. Yang, J. Kim, An accident diagnosis algorithm using long short-term memory, Nuclear Engineering and Technology 50 (4) (2018) 582–588.
[9] Y.G. Kim, S.M. Choi, J.S. Moon, Development of Convolutional Neural Networks Diagnose Abnormal Status in Nuclear Power Plant Operation. KNS2019, Korean Nuclear Society, 2019.
[10] J.M. Kim, G. Lee, S. Hong, S.J. Lee, February. Input data dimensionality reduction of abnormality diagnosis model for nuclear power plants, in: International Conference on Intelligent Human Systems Integration, Springer, Cham, 2019, pp. 504–509.
[11] J.M. Kim, G.M. Lee, S. Hong, S.J. Lee, Abnormal State Identifying Algorithm Using Recurrent Neural Network. KNS-2018, Korean Nuclear Society, 2018.
[12] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Network. 5 (2) (1994) 157–166.
[13] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation, 2014 arXiv preprint arXiv: 1406.1078.
[14] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, 2014 arXiv preprint arXiv: 1412.3555.
[15] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, J. Wang, Machine health monitoring using local feature-based gated recurrent unit networks, IEEE Trans. Ind. Electron. 65 (2) (2017) 1539–1548.
[16] I. Jolliffe, Principal Component Analysis, second ed., Springer, 2002, pp. 1–59.
[17] 3KEYMASTER Simulator, Western Service Corporation, Frederick, MD, USA, 2013.
[18] W.C. Jhee, J.K. Lee, Performance of neural networks in managerial forecasting, Intell. Syst. Account. Finance Manag. 2 (1) (1993) 55–71.
[19] C.N. Babu, B.E. Reddy, A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data, Appl. Soft Comput. 23 (2014) 27–38.
[20] A. Savitzky, M.J.E. Golay, Anal. Chem. 36 (1964) 1627–1639.
[21] K.M. Rashid, J. Louis, Times-series data augmentation and deep learning for construction equipment activity recognition, Adv. Eng. Inf. 42 (2019) 100944.
[22] S. Zheng, Y. Song, T. Leung, I. Goodfellow, Improving the robustness of deep neural networks via stability training, in: Proceedings of the Ieee Conference on Computer Vision and Pattern Recognition, 2016, pp. 4480–4488.
[23] K. Kamycki, T. Kapuscinski, M. Oszust, Data augmentation with suboptimal warping for time-series classification, Sensors 20 (1) (2020) 98.
[24] X. Li, W. Zhang, Q. Ding, J.Q. Sun, Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation, J. Intell. Manuf. 31 (2) (2020) 433–452.
[25] A. Gómez-Ríos, S. Tabik, J. Luengo, A.S.M. Shihavuddin, B. Krawczyk, F. Herrera, Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation, Expert Syst. Appl. 118 (2019) 315–328.
[26] X.Y. Bi, B. Li, W.L. Lu, X.Z. Zhou, Daily runoff forecasting based on data-augmented neural network model, J. Hydroinf. 22 (4) (2020) 900–915.