Nuclear Engineering and Technology 50 (2018) 582–588

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

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

Original Article

An accident diagnosis algorithm using long short-term memory

Jaemin Yang, Jonghyun Kim*

Department of Nuclear Engineering, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju 501-709, Republic of Korea

A R T I C L E I N F O

Article history:
Received 3 February 2018
Received in revised form
8 March 2018
Accepted 13 March 2018
Available online 28 March 2018

Keywords:
Accident Diagnosis
Long Short-term Memory
Recurrent Neural Network
Softmax

A B S T R A C T

Accident diagnosis is one of the complex tasks for nuclear power plant (NPP) operators. In abnormal or emergency situations, the diagnostic activity of the NPP states is burdensome though necessary. Numerous computer-based methods and operator support systems have been suggested to address this problem. Among them, the recurrent neural network (RNN) has performed well at analyzing time series data. This study proposes an algorithm for accident diagnosis using long short-term memory (LSTM), which is a kind of RNN, which improves the limitation for time reflection. The algorithm consists of preprocessing, the LSTM network, and postprocessing. In the LSTM-based algorithm, preprocessed input variables are calculated to output the accident diagnosis results. The outputs are also postprocessed using softmax to determine the ranking of accident diagnosis results with probabilities. This algorithm was trained using a compact nuclear simulator for several accidents: a loss of coolant accident, a steam generator tube rupture, and a main steam line break. The trained algorithm was also tested to demonstrate the feasibility of diagnosing NPP accidents.

1. Introduction

Diagnosis tasks in nuclear power plants (NPPs) are directly associated with safe and efficient operation. Monitoring and diagnosis of the state of NPPs is typically performed by operators who consider process variables based on operating procedures. To diagnose the status of an NPP, operators use measurements from numerous instruments, such as indicators and alarms.

However, there are several difficulties in the diagnosis of accidents in NPPs. First, the quantity of variables generated by these measurements is too high and can increase the difficulty of and delay in their interpretation. In abnormal or emergency situations, continuous monitoring of the critical safety variables can be a mentally burdensome task for operators because they need to identify possible success paths simultaneously [1]. In addition, even though operating procedures are provided to operators, event diagnosis under emergency conditions is regarded as the most difficult task for operators because of the extremely stressful conditions (e.g., time pressure for performing diagnosis within a limited time and many fast-changing process parameters needed for diagnosis) [2–6]. Moreover, depending on the severity of the accident, there may not be a clear indication of an abnormal state or anomaly in the initial stage [7]. Owing to these factors, diagnostic activities in emergency situations can cause not only a delay in effective response but also serious consequences when selecting an inadequate procedure (i.e., incorrect diagnosis) [8].

To reduce this burden on operators, many operator support systems and diagnostic algorithms have been suggested to help operators diagnose or detect accidents in NPPs. These are generally based on artificial intelligence techniques, such as artificial neural networks (ANNs), fuzzy logic, the hidden Markov model (HMM), and the support vector machine. ANNs are regarded as one of the most relevant approaches because they can deal with pattern recognition problems as well as nonlinear problems that are characteristic of NPP diagnosis tasks. Thus, several studies have applied ANNs to develop algorithms for diagnosis tasks [7,10–15].

Unlike the conventional transient diagnostic methods, the neural network, like the recurrent neural network (RNN), can only cope with the dynamic emergent situation simultaneously. Commonly, the diagnostic methods can be divided into two types according to the timing of data being received as follows: receiving all the data after the transient or receiving the data at intervals during the operation. In the first case, the transient can be identified after the transient is complete, e.g., the plant is shut down. In the second case, if the transient is too fast to be treated, then it will be necessary to use an RNN or several successive sets of samples as inputs to the neural network [11]. The RNN can naturally represent dynamic systems and can capture the dynamic behavior of a

* Corresponding author.
E-mail address: jonghyun.kim@chosun.ac.kr (J. Kim).

https://doi.org/10.1016/j.net.2018.03.010
1738-5733/

system; moreover, it is a powerful network from which the information feature related to the dynamic system in its hidden layer is extracted [16]. However, there are two well-known issues with RNNs: blowing-up and vanishing problems. These issues are the temporal evolution of back-propagated error. Blowing-up may cause the oscillation of weights; whereas, vanishing may lead weights to be zero in most areas. Thus, it can make a prohibitive duration for learning or even may not work at all [17].

Recently, long short-term memory (LSTM) has been suggested to solve these issues [18]. LSTM, which is based on RNN architecture, has been developed as a neural network architecture for processing long temporal sequences of data. LSTM combines fast training with efficient learning on tasks that require sequential short-term memory storage for many time steps during a trial. Because of these advantages, it can be applied to a variety of tasks for varying-length sequential data, such as natural language processing, image captioning, handwriting recognition, and genomic analysis, and achieve state-of-the-art results for problems [19–26].

This article proposes an algorithm for the diagnosis of NPP accidents using LSTM. First, we briefly introduce the LSTM network and RNN. Then, we suggest a diagnosis algorithm to diagnose accidents using the LSTM, which is trained using a compact nuclear simulator (CNS) that is based on a Westinghouse three-loop, 930 MWe pressurized water reactor. Lastly, we test the trained algorithm to demonstrate its effectiveness.

2. Long short-term memory

This study used the LSTM network for the online diagnosis algorithm for NPP accidents. LSTM is an advanced version of the RNN, which is also an approach of the ANN. The ANN is a statistical learning algorithm used in machine learning, which was inspired by the neural network (i.e., brain) of biology. It is a model that has a problem-solving ability owing to artificial neurons (nodes) that form a network of synaptic connections and change the synaptic bond strength through learning. It can be divided into three paradigms of learning, i.e., supervised learning, unsupervised learning, and reinforcement learning, depending on the particular type of learning task. For supervised learning, LSTM is optimized for the problem by mapping implied data with the correct answers. It is generally used to guess and approximate a veiled function. In other words, supervised learning is appropriate for analyzing tasks such as pattern recognition, regression, and sequential data. Accident diagnosis can be classified as a pattern recognition problem. It is generally known that ANNs show good performance in solving pattern recognition problems. This section presents a short introduction to the RNN and LSTM.

2.1. Recurrent neural network

Although numerous ANNs have been developed, we selected the RNN to model the accident diagnosis algorithm because it performs well in analyzing time series data. In contrast to other ANNs, it assumes that the input and output are not independent of each other, that is, it uses sequential information as input data. The same calculation is applied to every element of a sequence, and the output result is affected by the previous calculation result. According to this assumption, because it uses the same calculation, the structure of the vanilla (i.e., the state consists of a single hidden vector, h) RNN has a circular shape, as shown in Fig. 1. It is a kind of ANN in which the hidden node is connected to the directional edge to form a circular structure. Because of this structure, they all share the same parameters, which is unlike general ANNs with different parameters for each layer. A sequence of vectors, x, is processed by applying recurrence formulas, Equations (1)–(3), at every time step. Fig. 2 shows the internal operational process in a single RNN time step [10–13].

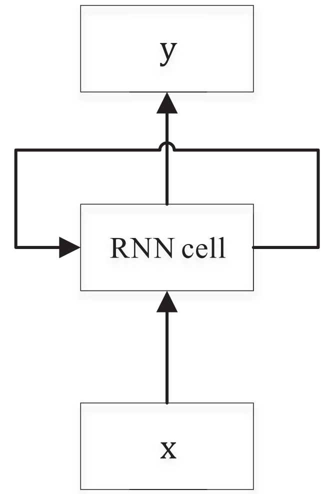


Fig. 1. Structure of vanilla RNN.
RNN, recurrent neural network.

$$h_t = f_W(h_{t-1}, x_t) \quad (1)$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \quad (2)$$

$$y_t = W_{hy}h_t \quad (3)$$

Because of this structure, the same task is applied to every element of a sequence, and the output is affected by the results of previous calculations; thus, it is called “recurrent”. In other words, an RNN has memory information about the results computed thus far, so the previous information can be used to solve the current problem. Therefore, this algorithm is the most appropriate method for solving a series of events or problems.

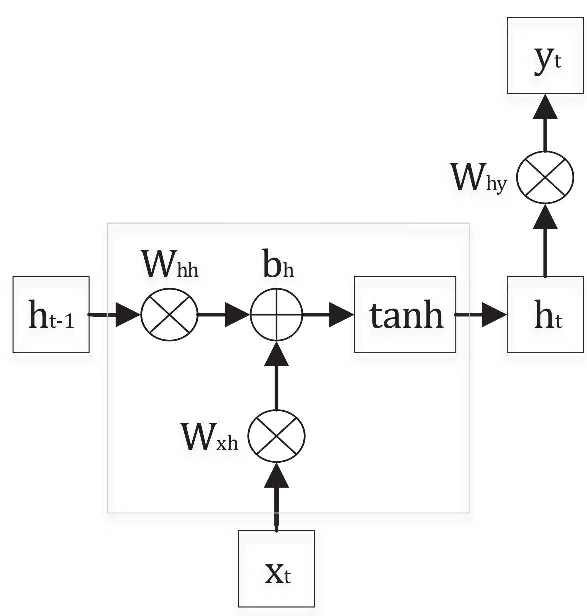


Fig. 2. Internal operational process in a single RNN time step.
RNN, recurrent neural network.

In addition, for training, it does not adjust the weights by merely transferring the errors that occurred at the current point to the lower layer as in conventional error back-propagation, which is the learning algorithm of existing ANNs. The conventional back-propagation updates each weight and bias by going back to the neural network and considering the portion of error attributed to the output stage. However, the RNN learns this from the back-propagation through time algorithm, which delivers the errors occurring at the current point to the past point. In this case, since each layer has the same weight in the RNN, all the derivative errors corresponding to the weight at the same position are added, and the weight is updated by averaging. That is, the error occurring in the current step is learned by propagating the error to the past state.

Because of these characteristics from back-propagation through time, for the NPP field, the RNN is used for a wide range of time series data analysis, such as system health management, fault or anomaly detection, and accident diagnosis. The neuro-expert system was proposed for the combination of an RNN and multilayer perceptron with simple rules. To detect anomalies in the early stages and to give alerts about occurring wrong signals, neural networks, such as the RNN or multilayer perceptron, are applied. In parallel with neural networks, diagnosing based on the alert information and inference of the cause are performed through the expert system. It can also be applied to analyze dynamic cases solely (e.g., a high-temperature gas cooling reactor, bearing damage) [9,16,27,28].

The original RNN tracks past values back in time. However, too much back-propagation over a long period causes a vanishing gradient problem owing to the weight being multiplied repeatedly in the process of learning so far into the past. The concept of a gradient is simply a measure of the rate of change of y with variations in x. By applying this to a neural network, the relationship between all weights and errors of the neural network can be obtained; that is, changing the value of a neural network allows the resulting error change to be determined. If the gradient cannot be obtained accurately, the relationship between the measure and the error is not clear; thus, learning cannot be achieved properly. During the back-tracking of the RNN in time, because the neural network consists of multiplication operations, multiplying a very small value several times ultimately results in a large value (i.e., a vanishing gradient, blowing up), like compound interest charged by banks [17].

2.2. Long short-term memory

We propose LSTM for sequence learning to deal with the RNN for this vanishing gradient problem. LSTM is a neural network architecture based on the RNN for processing long temporal sequences of data. Sequential data can also be dealt with by other sequence models such as Markov models, conditional random fields, and Kalman filters. However, only LSTM is equipped to learn long-range dependencies. It may be difficult to say that LSTM has a different structure from an RNN, but it uses a different equation to calculate the hidden state. LSTM uses a structure called a memory cell instead of an RNN neuron. It combines fast training with efficient learning on the tasks, which require sequential short-term memory storage for many time steps during a trial. LSTM can learn to bridge minimal time lags in excess of 1,000 discrete time steps by enforcing special units, which are called memory cells. It determines whether the previous memory value should be altered and calculates the value to be stored in the current memory based on the current state and the input value of the memory cell. This structure is highly effective in storing long sequences. In addition, alternative models (i.e., Markov models, conditional random fields, and Kalman filters) require domain knowledge or feature engineering, offering less chance for unexpected discovery, whereas LSTM can learn representations and discover unforeseen structures [9,29,30].

As with other LSTM models, in this study, each LSTM cell adjusts the output value using the input gate, forgetting gate, and output gate while maintaining the cell state. Information in the cell state is unchanged, and information can be added or deleted through each gate. In addition, since the operation of each gate is composed of an addition operation attached to the cell state, it can avoid the vanishing gradient problem.

The input gate determines the capacity of the input value. The forgetting gate determines the degree to which the previous cell state is forgotten, and the output gate determines how much to output. Equation (4), denoted by g, represents the input node and has a *tanh* activation function denoted by ϕ; Equations (5)–(7) represent the gates denoted by i, f, and o, respectively; σ represents a sigmoid function. Fig. 3 shows the architecture of the LSTM cell applied in this study.

$$g_l^{(t)} = \phi\left(W_g \cdot [h_{t-1}, x_t] + b_l^g\right) \tag{4}$$

$$i_l^{(t)} = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_l^i\right) \tag{5}$$

$$f_l^{(t)} = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_l^f\right) \tag{6}$$

$$o_l^{(t)} = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_l^o\right) \tag{7}$$

As shown in Fig. 3, the LSTM unit consists of a cell with several gates attached. These gates update the layers of memory cells $h_l^{(t)}$, where $h_{l-1}^{(t)}$ represents the previous layer at the same sequence step (i.e., a previous LSTM layer) and $h_l^{(t-1)}$ means the same layer at the previous sequence step.

This study applies the conventional LSTM structure. To design the optimized LSTM network without change of the LSTM unit, it is necessary to determine the proper hyperparameters to model the algorithm, such as the number of input sequences or hidden layers. This is because the purpose of learning through the neural network is to determine the weight and bias values that minimize the cost function. However, to obtain the expected level without the overfitting problem, the optimization of hyperparameters is required.

The length of input sequence can be a kind of hyperparameters that represents the temporal length of the input data that LSTM

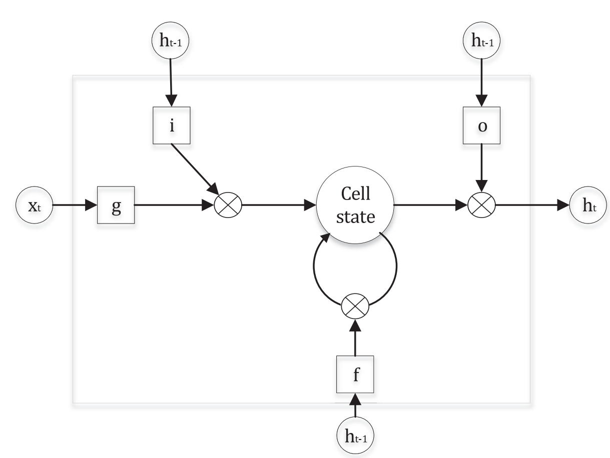


Fig. 3. Architecture of the LSTM cell.
LSTM, long short-term memory.

uses to compute the output. The performance of the network changes according to the length of the input sequence. In addition, the hidden layer is also the hyperparameter that is needed to transform the inputs into a useable form for the output layer. Basically, each layer in the neural network obtains the input for analysis farther from the original raw data, which is closer to the goal. Therefore, the performance of the model can be influenced by the number of hidden layers.

Four approaches (manual search, grid search, random search, and Bayesian optimization) are widely used in hyperparameter optimization. The manual search is a method of estimating optimal parameters and observing the results based on the designer's intuition or experience. In a large frame, the grid search has a big difference from the manual search and is conceptually similar; however, it is analyzed using a priori knowledge, and the scope of the hyperparameter is determined. Then, we set the point at a certain interval in the range and test the points individually to determine the optimal value. Following this, based on the estimated optimal values, the new optimum value is searched by subdividing it. Like the grid search, the random search uses a priori knowledge to determine the range of hyperparameters. Then, instead of searching at regular intervals, an operation to find the optimal value proceeds. This does not seem to be different from a grid search, but if the result must be produced within a specific time frame, a random search tends perform better [31]. Since the basic principle of Bayesian optimization uses prior knowledge, the key to this method is based on determining the direction of the next search after creating a statistical model based on the results of the experiments thus far. It tends to find optimal values within a shorter time than using random or grid search [32]. Unfortunately, there is no golden rule thus far, and much of it depends on the experience and intuition of the designer.

3. Development of accident diagnosis algorithm using LSTM

This chapter introduces the accident diagnosis algorithm based on LSTM. The accident diagnosis is performed with the NPP data sets through the trained classifier. During the training stage, the classifier is trained on the basis of training data sets with answer-labeled data, which has a specific pattern for each accident. After sufficient training, it is validated with the test data set and then used for real cases. Fig. 4 shows an overview of the process for accident diagnosis using LSTM.

To model the algorithm, a desktop computer with the following hardware configurations is used: NVIDIA GeForce GTX 1080 8 GB GPU, Intel 4.00 GHz CPU, Samsung 850 PRO 512 GB MZ-7KE512B SSD, and 24 GB memory. In case of software, Python 3.5.3 is used for coding language, which is one of the most popular computer languages for machine learning and deep learning. The libraries developed to model the algorithm for machine and deep learning (e.g., tensorflow and scikit-learn) were used.

3.1. LSTM network model for accident diagnosis

The model for accident diagnosis is designed for multilabel classification because diagnoses may not be mutually exclusive. To predict an accident, the trend of such a sequence of variables is needed as inputs. Thus, a many-to-one structure is applied to design the model. Fig. 5 shows a simple LSTM model for multilabel classification, which is the base model applied in this study.

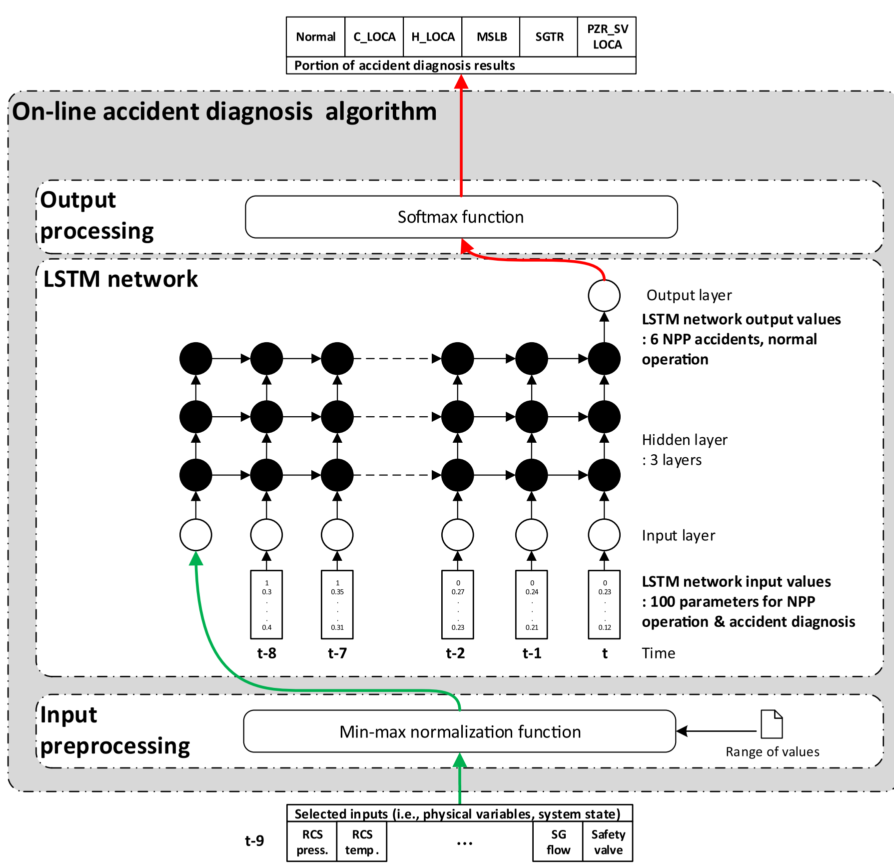


Fig. 4. Overview of process of accident diagnosis.
LSTM, long short-term memory; LOCA, loss of coolant accident; MSLB, main steam line break; NPP, nuclear power plant; SGTR, steam generator tube rupture.

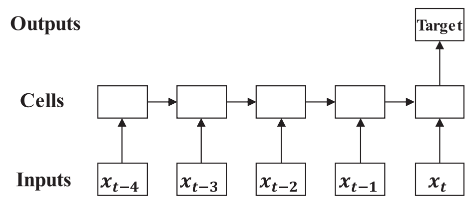


Fig. 5. Simple LSTM model for multilabel classification.
LSTM, long short-term memory.

According to the specific number of NPP input data sequences, the model can diagnose the plant state by recognizing the pattern (i.e., the NPP trend).

3.2. Preprocessing of input variables

Preprocessing of the input values is applied to the LSTM input layer. For the purpose of training the LSTM network, all the input values in the network should be scaled by normalizing each value from the raw NPP data. This is because normalization can help reduce the chance of getting stuck in local minima (i.e., not global minima among the several minimum points of error during the learning process) owing to different scales of variables (e.g., reactor coolant system (RCS) average temperature: 300°C, Valve State: 0 or 1). The min–max scaling method is applied to adjust the input values. The minimum and maximum values are determined within the collected data (i.e., not actual minimum or maximum of plant variables). Normalization using the min–max scaling method performs a linear transformation on the raw data, and via the following Equation (8), the data are scaled to range from zero to one.

$$X_{\text{norm}} = (X - X_{min})/(X_{max} - X_{min}) \tag{8}$$

3.3. Postprocessing of output variables

As a postprocessing for the output of the LSTM network, the softmax function layer shown in Fig. 6 is used to determine the ranking of accident probability. The softmax function is an activation function commonly used in the output layer of the deep learning model; it aims to classify more than three classes [33]. Therefore, this study applies the softmax function for postprocessing because there are six classes in the training. Softmax is a function that exponentially increases the importance through an exponential function; it also increases the deviation between the values and then normalizes. It normalizes the input value to the output value between zero and one via the following Equation (9), and the sum of the output values is always one. Fig. 7 shows an example of an application of the softmax layer to transform output

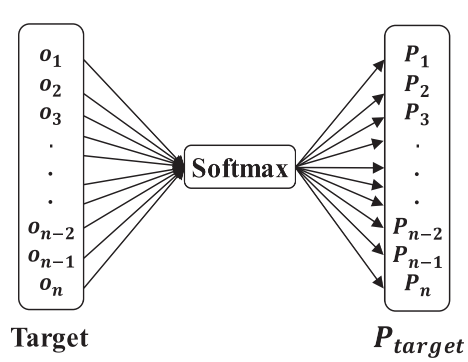


Fig. 6. Example of softmax function layer.

values to probabilities. Even if it is transformed, the magnitude relation of each output value does not change, and the output from softmax can be analyzed in terms of probability [34,35]; thus, it enables stochastic analysis for multilabel classification.

$$S(y_i) = e^{y_i} \Big/ \sum e^{y_i} \tag{9}$$

3.4. Training of the LSTM network

The network is trained and implemented using the CNS, which implements the Westinghouse 3-loop, 930 MWe pressurized water reactor. It was originally developed by the Korea Atomic Energy Research Institute. Fig. 8 shows the LSTM model for multilabel classification that is applied in this study. First, a total of 168 parameters are selected based on emergency operating procedures, critical safety functions, and the importance for the control of NPP operation; finally, input preprocessing is used to select 100 parameters. A total of 112 scenarios with 122,609 data sets (i.e., 122,609 s of data including 100 plant parameter values in each time step) are used for training, as shown in Table 1. The scenarios include automatic actuation of systems and components and manual actuations followed by procedures. The learning rate and number of iteration sets are 0.005 and 3,000, respectively.

3.5. Optimization of network

To optimize the model, we used the manual search method, changing the hyperparameters and selecting proper input variables. Table 2 shows an accuracy comparison of the results of the different structures of networks. Input sequence lengths of five and 10 and two or three hidden layers are tested. To evaluate the performance of

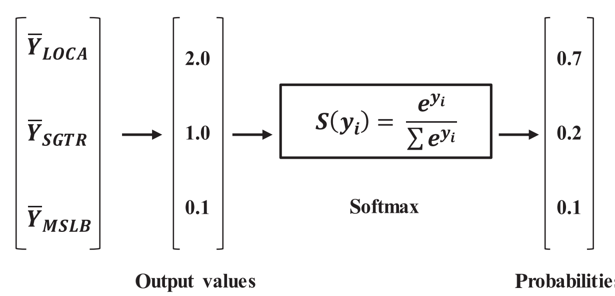


Fig. 7. Example of transformation from outputs to probabilities.
LOCA, loss of coolant accident; MSLB, main steam line break; SGTR, steam generator tube rupture.

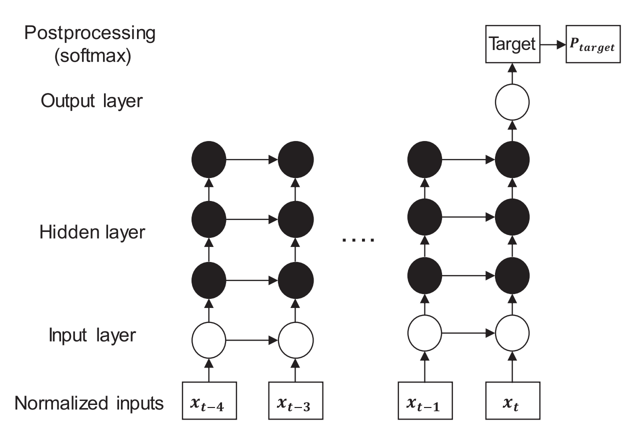


Fig. 8. LSTM model for multilabel classification.
LSTM, long short-term memory.

Table 1
Scenarios used for network training.

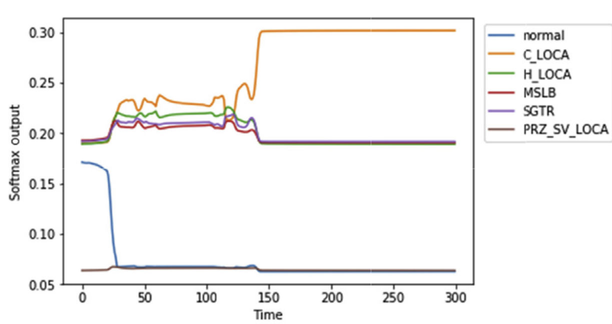
Initiating events	Number
Cold leg loss of coolant accident (LOCA)	29
Hot leg LOCA	29
PZR safety valve failed-to-open	5
Steam generator tube rupture	17
Main steam line break	32
Total	112

Table 2
Accuracy comparison between networks.

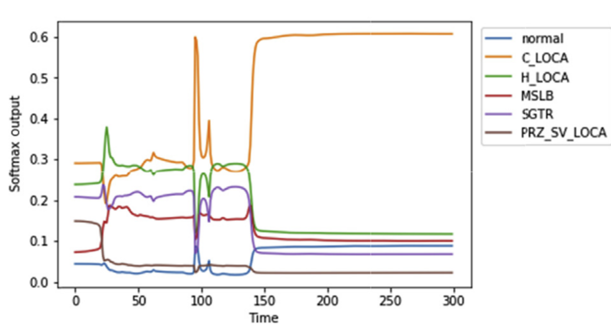
No	Sequence	Layer	Accuracy	
			168 inputs	100 inputs
1	5	2	0.609	0.839
2	10	2	0.784	0.854
3	5	3	0.620	0.833
4	10	3	0.795	0.859

the networks, we consider the accuracy of the diagnosis results. The accuracy is defined as Equation (10). We only consider the accuracy as an optimization parameter. This is because the training and validation data cannot be false positive or false negative if false positive or false negative data are not made artificially. Based on the performance comparison, the optimal LSTM network with an input sequence length of 10 and three layers is selected.

$$Accuracy = \frac{Correct\ results}{Diagnosis\ results} \tag{10}$$



(A) 10 cm^2 LOCA in Loop 1 cold-leg



(B) 100 cm^2 LOCA in Loop 1 cold-leg

Fig. 9. Accident diagnosis results of LOCA.
LOCA, loss of coolant accident; MSLB, main steam line break; SGTR, steam generator tube rupture.

4. Test

We tested the proposed algorithm with three scenarios, loss of coolant accident (LOCA), steam generator tube rupture (SGTR), and main steam line break (MSLB), which were not used in the training session. Fig. 9 shows the test results for LOCA with sizes of 10 and 100 cm^2 in Loop 1 cold-leg. Each line represents the accident or normal state of the NPP. The malfunction was injected at 30 s for every test scenario. The X-axis and Y-axis represent the time and diagnosed result from the model with postprocessing, respectively. The graphical results show that the accident was diagnosed continuously (i.e., the oscillation range is under 0.02) after approximately 150 s.

Figs. 10 and 11 also show the diagnosis correctly right after the malfunction is injected (i.e., 30 s) with the proposed algorithm for SGTR and MSLB accidents. The reason the diagnosis of LOCA takes longer time than SGTR and MSLB is that three different LOCAs were trained in the LSTM network (i.e., cold-leg LOCA, hot-leg LOCA, pressurizer safety valve LOCA). Although the location is different in those LOCAs, the plant behaviors are similar and then the LSTM takes a slightly long time to produce the steady result. In addition, the diagnoses for small and large LOCAs take a similar length of time. However, the large LOCA result shows a more distinguished softmax output from the other accidents than the small LOCA result does.

5. Discussion

In the test scenarios, the accident diagnosis with the proposed algorithm performs well at trained accidents. It provides stable, distinct results from injection of malfunction after about 120 s in the LOCA and a few seconds in the SGTR and MSLB. In case of a LOCA, the result of the algorithm seems to be unstable in the early stage of the accident as shown in Fig. 9, mainly because different

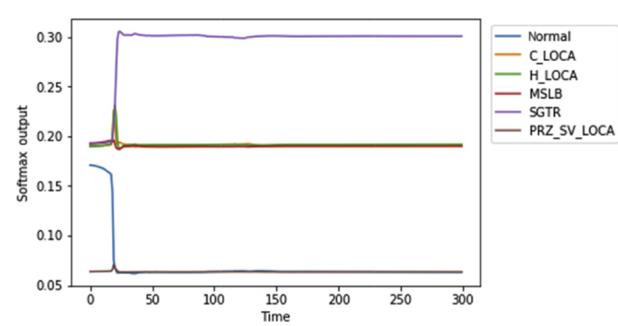


Fig. 10. Accident diagnosis results of 40 cm^2 SGTR in Loop 2.
LOCA, loss of coolant accident; MSLB, main steam line break; SGTR, steam generator tube rupture.

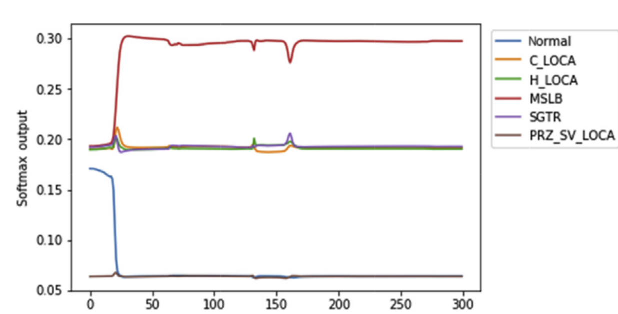


Fig. 11. Accident diagnosis results of 200 cm^2 MSLB in Loop 2.
LOCA, loss of coolant accident; MSLB, main steam line break; SGTR, steam generator tube rupture.

locations of the LOCA have been trained in the LSTM network, i.e., hot-leg LOCA, cold-leg LOCA, and pressurizer safety valve LOCA. The symptoms of these LOCAs are similar so the algorithm takes a relatively longer time to provide a stable diagnosis result.

Compared to the other methods such as in studies by Şeker et al. and Kwon et al. [9,36] which use Elman's RNN and HMM, the proposed algorithm performs better in some ways. In comparison with Elman's RNN [9], the proposed algorithm can diagnose more accidents, while the Elman's RNN only detects whether or not it is a transient. In addition, in comparison with HMM [36], this algorithm can perform the diagnosis of accidents at every time step. Thus, it can be applied to the flexible online diagnosis.

There is also room for improvement of the performance by adjusting the hyperparameters. Owing to the limitation of the computer we used, i.e., a desktop computer with the Intel Core i7-6700k CPU 4.00 GHz and an NVIDIA GeForce GTX 1080 8 GB GPU, we applied three hidden layers as well as a 10 input sequence length. Because this configuration is not a perfect optimization, a computer with better specifications would produce a better performance.

The proposed algorithm does not include the ability to diagnose unknown or untrained accidents. However, if this algorithm is implemented as an actual operator support system in the NPP, this capability is an important feature of the system because it can define the scope of diagnosed accidents or a limitation of the system. Thus, this algorithm needs to be improved to include how to react to unknown or untrained situations.

6. Conclusion

We proposed an accident diagnosis algorithm that uses LSTM, which is a technique of ANNs. The algorithm includes preprocessing, an LSTM network, and postprocessing using softmax. The algorithm was also trained for a few accidents: LOCA, SGTR, and MSLB. A CNS was used to produce the training data. Lastly, the trained algorithm was tested to demonstrate the feasibility of the proposed algorithm. The test results showed that it can diagnose accidents in a stable, distinctive way. We expect that this algorithm can be used as an online diagnostic function in operator support systems, e.g., a fault diagnostic system or an autonomous control system.

Conflict of interest

There is no conflict of interest.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2016R1A5A1013919) and the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant, funded by the Korean Government (MOTIE) (Grant No. 20164030201290).

References

[1] O. Maruo, et al., HSI for Monitoring the Critical Safety Functions Status Tree of a NPP, International Nuclear Atlantic Conference, Recife, PE, Brazil, Nov. 24~29, 2013.
[2] J.K. Vaurio, Safety-related decision making at a nuclear power plant, Nucl. Eng. Des. 185 (1998) 335–345.
[3] T. Kontogiannis, Stress and operator decision making in coping with emergencies, Int. J. Hum. Comput. Inter. 45 (1996) 75–104.
[4] David D. Woods, Coping with complexity: the psychology of human behaviour in complex systems, in: Tasks, Errors, and Mental Models, Taylor & Francis, Inc, 1988, pp. 128–148.
[5] David Meister, Cognitive behavior of nuclear reactor operators, Int. J. Ind. Ergonom. 16.2 (1995) 109–122.
[6] Nuclear Energy Agency, Critical Operator Actions: Human Reliability Modeling and Data Issues, 1998. NEA/CSNI/R(98)1.
[7] T.V. Santosh, et al., Application of artificial neural networks to nuclear power plant transient diagnosis, Reliab. Eng. Syst. Saf. 92 (10) (2007) 1468–1472.
[8] Jinkyun Park, Wondea Jung, A study on the systematic framework to develop effective diagnosis procedures of nuclear power plants, Reliab. Eng. Syst. Saf. 84.3 (2004) 319–335.
[9] Serhat Şeker, Emine Ayaz, Erdinç Türkcan, Elman's recurrent neural network applications to condition monitoring in nuclear power plant and rotating machinery, Eng. Appl. Artifi. Intell. 16 (7) (2003) 647–656.
[10] Moshkbar-Bakhshayesh Khalil, Mohammad B. Ghofrani, Transient identification in nuclear power plants: a review, Progr. Nucl. Energy 67 (2013) 23–32.
[11] Robert E. Uhrig, J. Hines, Computational intelligence in nuclear engineering, Nucl. Eng. Technol. 37.2 (2005) 127–138.
[12] P.F. Fantoni, A. Mazzola, A pattern recognition-artificial neural networks based model for signal validation in nuclear power plants, Ann. Nucl. Energy 23 (13) (1996) 1069–1076.
[13] Mark J. Embrechts, Sándor Benedek, Hybrid identification of nuclear power plant transients with artificial neural networks, IEEE Trans. Ind. Electron. 51 (3) (2004) 686–693.
[14] Seung Jun Lee, Poong Hyun Seong, A dynamic neural network based accident diagnosis advisory system for nuclear power plants, Progr. Nucl. Energy 46 (3–4) (2005) 268–281.
[15] Kun Mo, Seung Jun Lee, Poong Hyun Seong, A dynamic neural network aggregation model for transient diagnosis in nuclear power plants, Prog. Nucl. Energy 49.3 (2007) 262–272.
[16] K. Nabeshima, et al., On-line neuro-expert monitoring system for borssele nuclear power plant, Progr. Nucl. Energy 43 (1–4) (2003) 397–404.
[17] Sepp Hochreiter, Jurgen Schmidhuber, Bridging long time lags by weight guessing and Long Short-Term Memory, Spatiotemporal Models Biol. Artif. Syst. 37 (1996) 65–72.
[18] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[19] Michael Auli, Michel Galley, Chris Quirk, Geoffrey Zweig, Joint language and translation modeling with recurrent neural networks, in: Empirical Methods in Natural Language Processing (EMNPL), vol. 3, 2013.
[20] Ilya Sutskever, Oriol Vinyals, Quoc VV. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems (NIPS), vol. 27, 2014, pp. 3104–3112.
[21] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan, Show and tell: a neural image caption generator, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3156–3164.
[22] Andrej Karpathy, Li Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3128–3137.
[23] Marcus Liwicki, Alex Graves, Horst Bunke, Jurgen Schmidhuber, A novel approach to ön-line handwriting recognition based on bidirectional long short-term memory networks, in: Proceedings of the Ninth International Conference on Document Analysis and Recognition, vol. 1, 2007, pp. 367–371.
[24] Gianluca Pollastri, Darisz Przybylski, Burkhard Rost, Pierre Baldi, Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles, Proteins Struct. Funct. Bioinform. 47 (2002) 228–235.
[25] Jiri Vohradsky, Neural network model of gene expression, FASEB J. 15 (3) (2001) 846–854.
[26] Rui Xu, Donald Wunsch II, Ronald Frank, Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization, IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB) (2007) 681–692.
[27] Muhammad Subekti, Kazuhiko Kudo, Kunihiko Nabeshima, The development of anomaly diagnosis method using neuro-expert for PWR monitoring system, in: International Conference on Advanced in Nuclear Science and Engineering (ICANSE2007), 2006.
[28] Abiodun Ayodeji, Yong-kuo Liu, Hong Xia, Knowledge base operator support system for nuclear power plant fault diagnosis, Progr. Nucl. Energy 105 (2018) 42–50.
[29] D. Monner, J.A. Reggia, A generalized LSTM-like training algorithm for second-order recurrent neural networks, Neural Netw. 25 (2012) 70–83.
[30] C. Lipton, et al., learning to diagnose with LSTM recurrent neural networks, in: International Conference on Learning Representation, Caribe Hilton, San Juan, Puerto Rico, May 2~4, 2016.
[31] James Bergstra, Yoshua Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. (2012) 281–305.
[32] Jasper Snoek, Hugo Larochelle, Ryan P. Adams, Practical bayesian optimization of machine learning algorithms, in: Advances in Neural Information Processing Systems, 2012, pp. 2951–2959.
[33] Christopher M. Bishop, Machine learning and pattern recognition, in: Information Science and Statistics, Springer, Heidelberg, 2006.
[34] Yaguo Lei, et al., An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data, IEEE Trans. Ind. Electron. 63 (5) (2016) 3137–3147.
[35] Siqin Tao, et al., Bearing fault diagnosis method based on stacked autoencoder and softmax regression, in: Control Conference (CCC), 2015 34th Chinese, IEEE, 2015.
[36] Kee-Choon Kwon, Jin-Hyung Kim, Accident identification in nuclear power plants using hidden Markov models, Eng. Appl. Artif. Intell. 12 (4) (1999) 491–501.