2018년 8월

석사학위 논문

Prediction of Major Factors Using Deep Learning in NPP Severe Accidents

조선대학교 대학원

원 자 력 공 학 과

구 영 도

Prediction of Major Factors Using Deep Learning in NPP Severe Accidents

원전 중대사고시 딥러닝을 이용한
원자력발전소 주요 인자 예측

2018년 8월 24일

조선대학교 대학원
원 자 력 공 학 과
구 영 도

Prediction of Major Factors Using Deep Learning in NPP Severe Accidents

지도교수 나 만 균

이 논문을 공학 석사학위신청 논문으로 제출함

2018년 4월

조선대학교 대학원

원 자 력 공 학 과

구 영 도

구영도의 석사학위논문을 인준함

위원장 조선대학교 교수 송 종 순 (인)

위 원 조선대학교 교수 김 진 원 (인)

위 원 조선대학교 교수 나 만 균 (인)

2018년 5월

조선대학교 대학원

CONTENTS

List of Tables

List of Figures

초 록

원전 중대사고시 딥러닝을 이용한 원자력발전소 주요 인자 예측

구 영 도
지도 교수 : 나 만 균
원자력공학과
조선대학교 대학원

원전에서 정상운전 상태 또는 사고 상황에서 원전의 건전성 및 안전을 유지할 수 있는 것은 원전에서 발생하는 신호들을 바탕으로 운전원들과 엔지니어들이 원전의 상태를 정확하게 파악하고 상황에 따른 적절한 조치를 취함으로써 가능해진다. 따라서 원전의 계측 신호를 획득하는 것은 원전의 안전을 보장하기 위해 매우 중요하다. 본 연구에서는 원전에서 발생하는 다양한 신호들 중, 일차 계통의 원자로용기, 가압기, 증기발생기, 배관, 격납건물 등에서 얻어지는 수위, 압력, 온도, 중성자속, 누출 유량, 수소농도 등과 같은 원전의 주요 인자 중 일부를 예측하였다. 이는 원자로냉각재계통(RCS; Reactor Coolant System)이자 일차 계통의 안전을 위해 중대사고 상황에서 원전의 상태 판단 및 비정상상태 완화에 필요한 주요 사고 감시변수들이기 때문이다.

그러나 이러한 주요 인자들은 원전의 중대사고 상황에서 계측기 불능 또는 오작동으로 인하여 신호의 건전성이 보장되지 않을 수 있으며, 이는 작업자들의 부적절한 상태 판단 및 조치로 이어져 노심을 포함한 RCS 및 원전에 심각한 영향을 초래할 수 있을 것이다. 따라서 본 연구에서는 원전에서 발생할 수 있는 사고 중 하나로써 냉각재상실사고(LOCAs; Loss of Coolant Accidents)로 비롯된 중대사고 상황에서 원전에서 발생하는 일부 신호 및 예측된 신호 데이터를 Deep Neural Networks(DNNs)에 적용하여 원전의 주요 인자들을 예측하였다.

DNN은 여러 은닉 계층으로 구성되어 있으며, 그것의 단방향의 weight flow로 인하여 기본적이고 간단한 형태를 가진 딥러닝 기법으로 정의할 수 있다. 또한 DNN 기법은 일반적으로 지도학습 알고리즘을 사용하기 때문에 목표 값 또는 실제 값으로 알려진 데이터가 필요하다. 이에 Modular Accident Analysis Program(MAAP) 코드를 이용하여 세 가지 파단 위치에서 발생한 것으로 가정된 다양한 LOCA 중 일부를 모사한 데이터를 획득하였다.

본 연구의 개발된 DNN 모델에 이러한 사고 모사 데이터를 구성하는 수치로 표현된 일부 신호 및 예측된 신호 데이터를 적용한 결과, 주요 인자들에 대한 딥러닝의 좋은 예측 성능을 확인할 수 있었다. 또한 이전 연구에서 수행되었던 머신러닝 기법을 이용한 주요 인자 예측 연구 결과와 비교하였을 때, 본 연구에서 제안된 DNN 기법이 대부분의 경우에서 상대적으로 더 좋은 성능을 보여주었으나 반대의 경우도 있었다. 결론적으로 이러한 주요 인자들에 대한 예측 결과들을 종합적으로 고려하였을 때, 본 연구에서 사용된 DNN 모델은 원전의 중대사고시 주요 인자들을 예측하는데 있어 기존의 머신러닝 기법보다 더 향상된 성능을 가지고 있다고 판단되며, 향후 원전 및 원전 기기의 상태를 감시, 진단 및 예측하는데 필요한 지원 정보를 제공할 수 있는 딥러닝 기법으로써의 적용가능성을 기대할 수 있을 것으로 판단된다.

I. Introduction

Nuclear power plants (NPPs) are composed of various facilities and equipment for the safety and power generation. The NPPs operating in Republic of Korea are mostly pressurized water reactor (PWR). The PWR types are largely classified into the primary system (or reactor coolant system (RCS)), which typically contains the reactor coolant passing through the reactor core, and the secondary system which is not directly exposed to these reactor coolant with radioactivity. In addition, the design lifetime for most of the NPPs with the above characteristics is generally more than 40 years. For these reasons, therefore, it is obvious that maintaining the integrity of diverse facilities and systems is essential for safe long-term operation (LTO) for the NPPs. The operators keep the safety by monitoring and diagnosing the NPPs based on a variety of instrumentation signals from the NPPs.

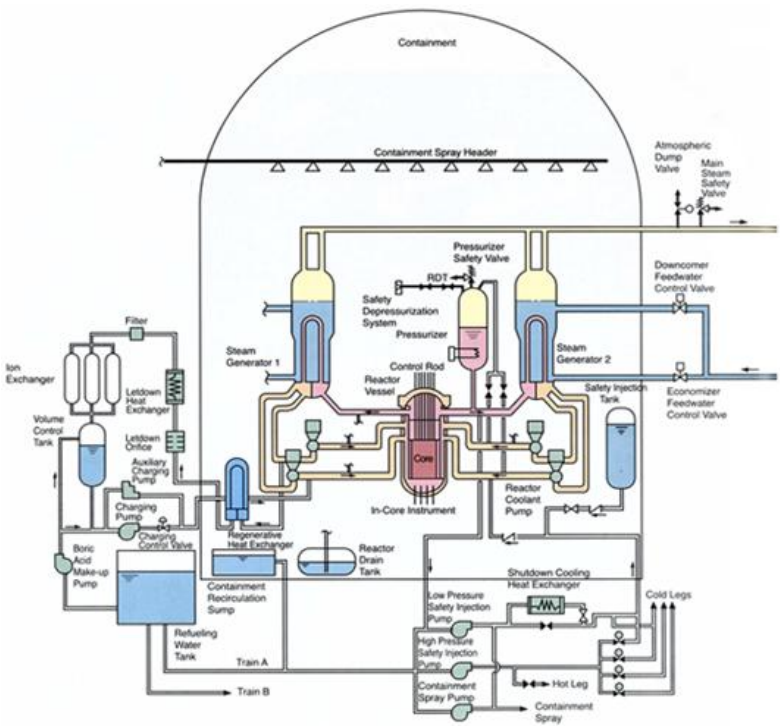


Fig. 1. OPR 1000 schematic diagram [1]

Accordingly, acquiring the various instrumentation signals arisen from the NPP systems is critical to guarantee the safety of the NPPs under the normal operation condition or the accident circumstances. Among the systems in the NPPs, keeping the safety of the primary system is the traditionally main concern owing to the characteristics of the NPPs. Thus, many instrumentation signals such as water level, pressure, temperature, flow rate, and hydrogen concentration from the reactor vessel (RV), pressurizer, steam generator (S/G), pipes, containment, and so on are considered as safety-critical major factors for the NPPs (refer to Fig. 1). That is, it is possible to properly control and take necessary actions depending on the situations by diagnosing the NPP states employing these monitoring variables of the NPPs by the operators.

However, the integrity of these safety-related instrumentation signals can not be secured due to instrument inability and its unreliability under the severe accident circumstances. For instance, the RV water level can not be accurately measured by heated junction thermocouple (HJTC) in the severe accidents. Eventually, the abnormal NPP states can be worse by the improper decision by the operators under such circumstances. In this study, therefore, in an effort to provide supporting information to the operators, a part of major factors of the NPPs were predicted applying other signals of the NPPs to a deep learning method under the severe accident circumstances when the integrity of instruments can not be ensured.

Deep neural networks (DNNs) [2,3], as a deep learning method used in this thesis, are defined as multi-layer neural networks with effective techniques. Additionally, since the training of the DNNs is commonly performed using the supervised learning algorithm, the data known as actual or desired result are necessary. However, there are rarely the actual NPP accident data. Therefore, the modular accident analysis program (MAAP) [4], which is the severe accident analysis code for the PWR and the boiling water reactor (BWR), was used to gain the accident simulation data applied to the DNN model. These accident data from the MAAP code are comprised of the behaviors of the simulated instrumentation signals for the NPP parameters, which are numerically expressed.

In this thesis, these two or three simulated signal data including the estimated signal data were applied to the DNN model to check its prediction performance for the major factors of the NPPs. In addition, the prediction result of the proposed DNN model was

compared with that of the machine learning model for predicting the NPP factors of the previous studies [5]-[7] to check the applicability of the DNNs to the NPP fields. Consequently, the result of this thesis can be a case study trying to apply a deep learning to the NPP fields in the future.

II. Artificial Intelligence Methods with Neural Network Framework

The deep learning used for predicting the major factors of the NPPs is the subset of the machine learning which is a subset of artificial intelligence (AI) (refer to Fig. 2). Briefly, the concept of AI is one of the technologies that is able to carry out specific tasks such as classification or prediction by the computers as much as or better than humans can. Although the AI in this level is called "Narrow AI" [8], various AI methods applied to a wide range of the industrial fields have shown their good performances in the domains of voice recognition, visual object recognition, regression analysis, and so on [2].

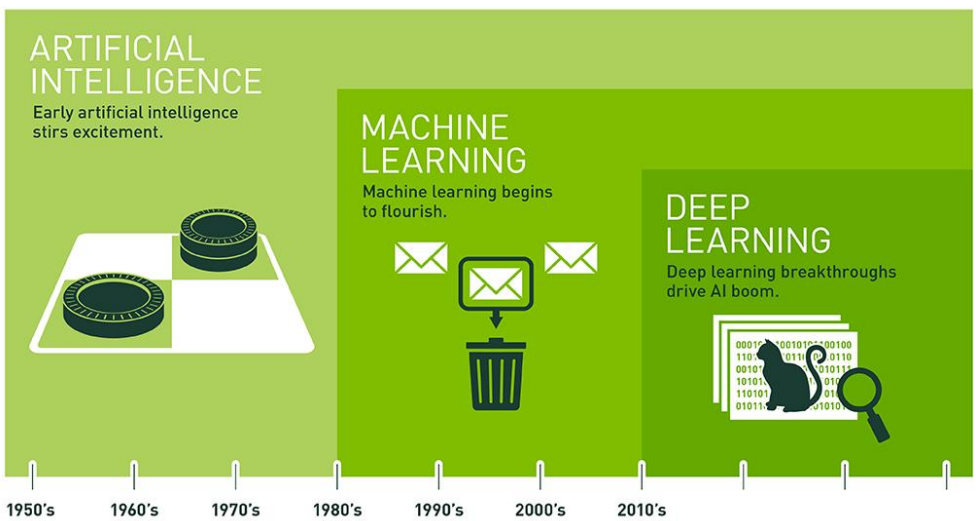


Fig. 2. Machine learning and deep learning of AI [8]

The interesting performance of the AI methods in such domains was able to be made by development of the machine learning algorithms such as representative support vector machines [9,10] (SVMs) and several methods based on artificial neural networks (ANNs), and by constantly upgraded computer hardware and some proposed techniques for effective calculation. In the nuclear fields, accordingly, many studies on NPP state monitoring and

diagnosis [11]-[13], prognostics and health management (PHM) [14] for the equipment, sensor signal validation [15], smart sensing [16], and so on have been carried out using the AI methods.

In a part of an effort to apply the AI methods to the NPPs, in this thesis, the DNN model was used to predict the major factors of the NPPs and its results were compared with the results using the cascaded fuzzy neural network (CFNN) model of the previous studies [17]-[19]. As all the AI models utilized for NPP factor prediction in this thesis are on the basis of the neural network structures which are inspired by interconnections between the neurons of the human brain, the DNNs and fuzzy neural networks (FNNs), which are single module of the CFNNs, have a system that the data are continuously transferred from its first layer to the next layer, and finally the outputs are computed in the last layer. That is, these methods are trained and optimized through the networks using the data and the learning algorithm.

However, the efficacy of each AI method with a neural network framework differs depending on the feature inherent in the method [20]. For instance, shortly, the DNNs consist of multiple hidden layers between the input layer and the output layer to learn representations using the outputs from every single node [2] while the basic idea of the FNNs is a fuzzy inference system (FIS) embodied in the neural networks. In other words, the FNNs are comprised of the specific processing layers [18,21].

A. DNN of Deep Learning Methods

1. Deep Neural Networks

The DNNs used in this thesis can be defined as the neural networks which consist of the deep hidden layers and nodes with the effective computational techniques. Moreover, the DNNs can be considered as a simpler method than other well-known deep learning methods such as recurrent neural networks (RNNs) and long-short term memory (LSTM) [2,3] owing to its straightforward weight propagation from activation functions of the

nodes. In the DNNs, specifically, the calculated weights of each node in the first layer using the inputs x_j, as expressed in Fig. 3, are repeatedly transferred forward into the nodes in the next layer and updated. After then, the predicted values $\hat{y}$ are finally calculated in the output layer (described in Fig. 4). The DNNs are usually called as deep feedforward networks (DFNs) or feedforward neural networks (FFNNs) [3] by its specific weight flow through the networks. The hypothesis of the DNNs is generally defined as follows:

$$H(x) = w_{ij}x_j = \boldsymbol{Wx} \tag{1}$$

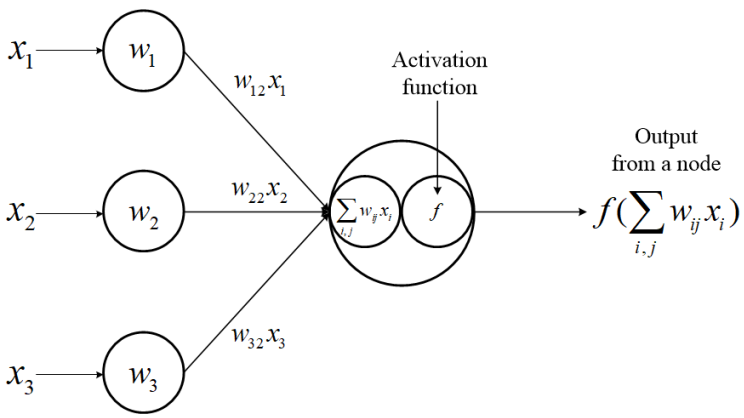


Fig. 3. Single artificial neuron of the DNNs

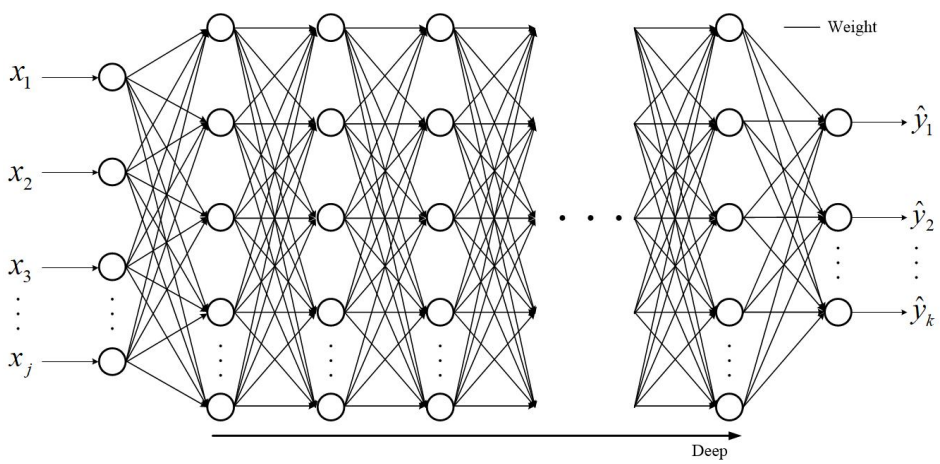


Fig. 4. Deep neural networks

It is noted that the main aspect of the DNNs is that the model is trained via a basic ANN framework using the general-purpose learning algorithms with the data, not by specific processing layers designed by users [2]. Training and optimization of the DNNs are commonly made by the supervised learning algorithms such as back-propagation [22,23] and gradient descent [2,23], which are the most common form of a machine learning algorithm [2]. Concretely, the cost function that measures the error between the predicted output by feedforward training and the target value is calculated first, and then the value of the cost function is propagated backward from the output layer to the initial hidden layer, as indicated in Fig. 5. The gradient is computed to update the weight w_{ij}. The training and optimization process by forward- and back- propagation and calculating the gradient at each current point is usually iterated until the optimal w_{ij} at the global minima, as the lowest point of the cost function, is found out (refer to Fig. 6). In this thesis, root mean square (RMS) error was used as the cost function for the bowl-shaped convex function defined as Eq. (2). Eq. (3) denotes the revised weight after back-propagation.

$$Cost(\boldsymbol{W}) = E = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2} \tag{2}$$

where N is the number of the sample data.

$$w_{ij}^{new} = w_{ij}^{old} - \alpha\frac{dE}{dw_{ij}} \tag{3}$$

where the learning rate α ($0<\alpha<1$) has a role to control a step size of the gradient descent (described in Fig. 6).

In this thesis, training of the DNN model stopped when the *Cost*(***W***) of Eq. (2) was converged at the global minima or the maximum number of epochs was reached. An epoch denotes the number of a single DNN training process through the networks using the learning data set and the validation data set.

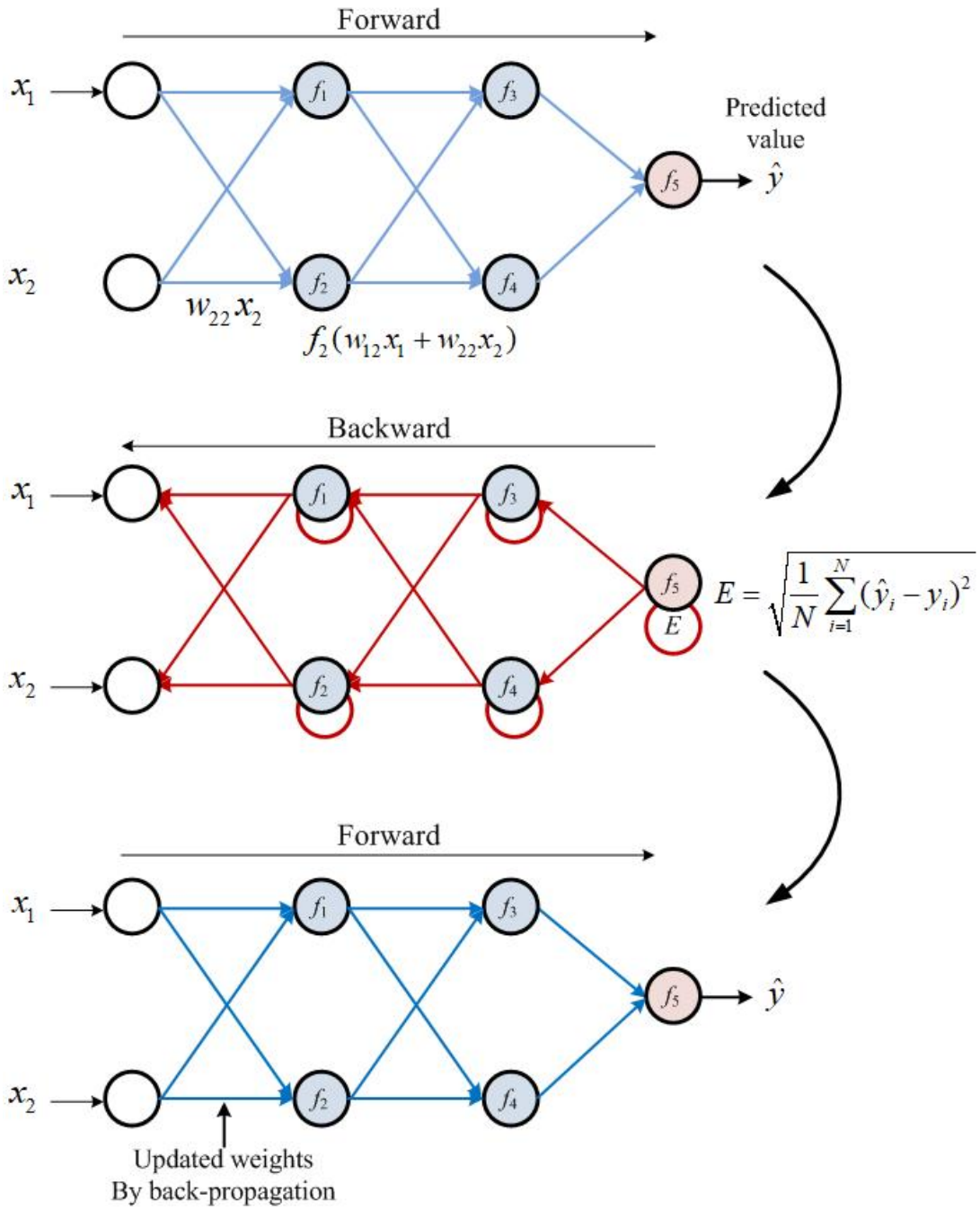


Fig. 5. An learning and optimization procedure by forward- and back- propagation

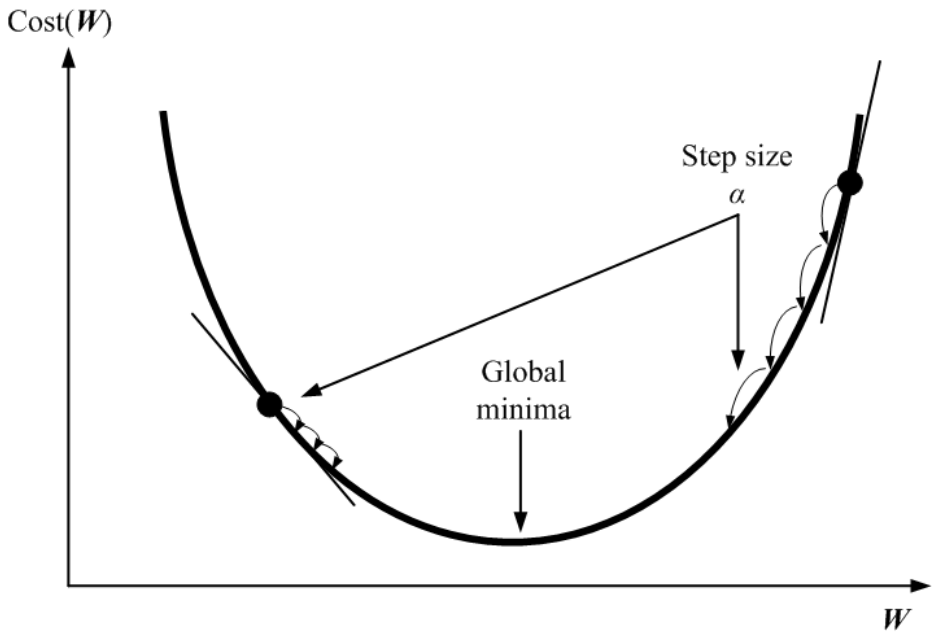


Fig. 6. Global minima in the cost function

2. Hyper-parameters of the DNNs

Since there are specific hyper-parameters influential in the performance of the machine learning including the deep learning, it is essential to find the optimal ones to establish a powerful model. To reach the global minima, namely, the AI methods are able to be optimized using these key parameters. The hyper-parameters for the DNNs are the number of hidden layers and nodes, activation functions, generalization techniques, cost functions, learning rate α, weight initialization, batch normalization, and so on, which are also usually considered in other deep learning methods.

Among them, increasing the number of its hidden layers and nodes and applying more data are a well-known basic way to improve the performance of the DNNs (refer to Fig. 7). However, the DNNs with excessively deep hidden layers can be vulnerable to the vanishing gradient or the overfitting problem which is caused by increase in the complexity of the DNNs [24].

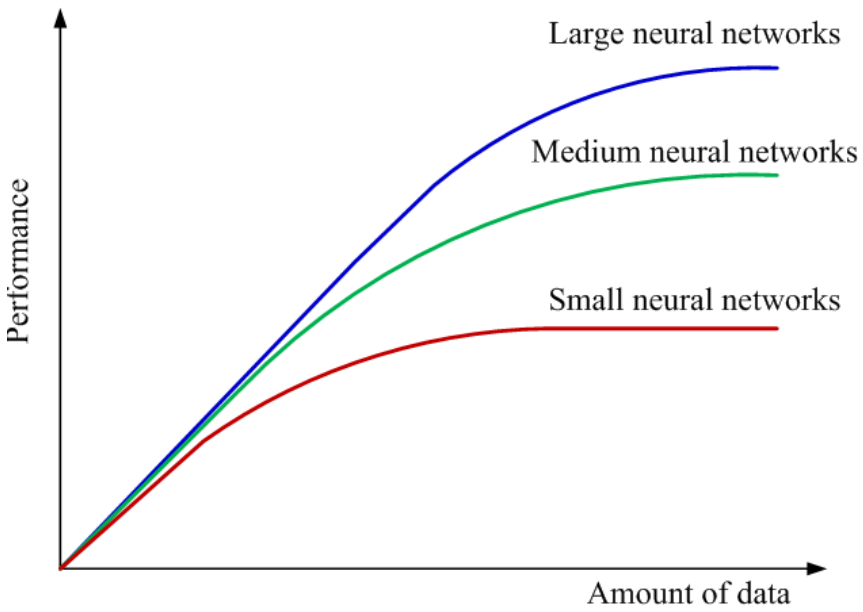


Fig. 7. Comparison of performance according to the neural network scale and the amount of data [25]

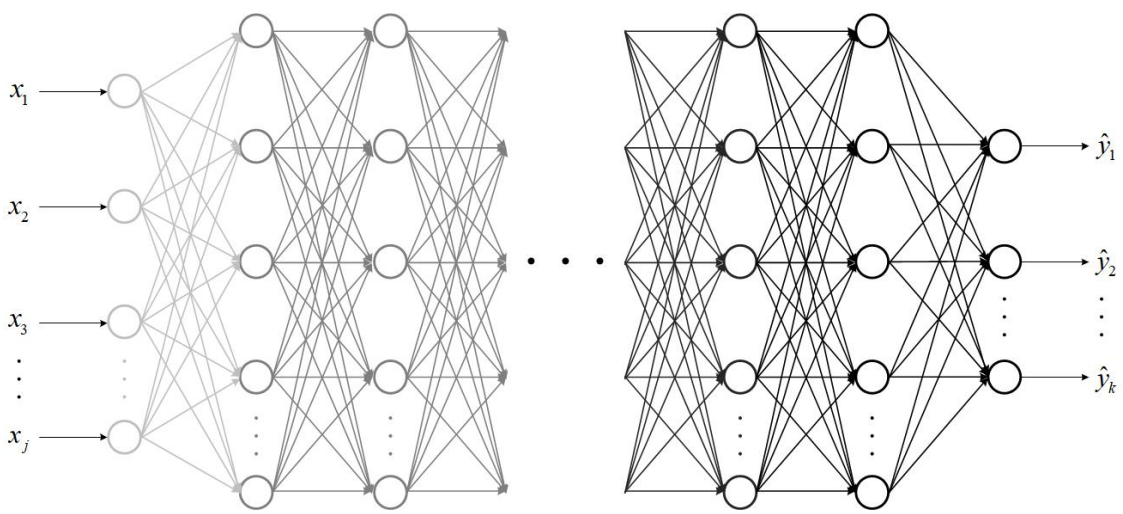


Fig. 8. Vanishing gradient problem in the DNNs

The vanishing gradient problem, indicated in Fig. 8, means that the error can not be well propagated backward from the output layer to the hidden layers. Basically, this phenomenon is caused since most of the ANNs are based on the multiplying operation. Concretely, Fig. 9 shows that sigmoid, one of the activation functions, is repeatedly multiplied. The output becomes considerably flattened when the sigmoid function is multiplied more than three or four times. Therefore, the DNNs can be hard to be trained by excessively reduced gradient which is nearly zero.

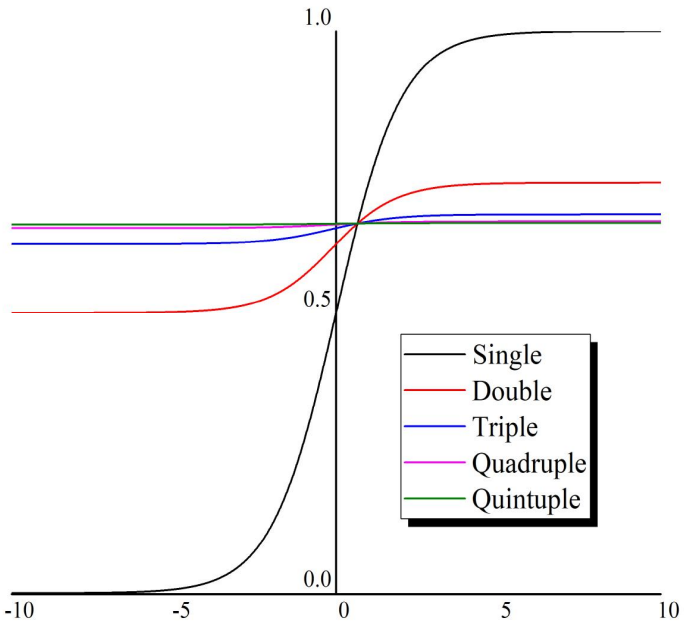


Fig. 9. Multiplication of the sigmoid function

Therefore, a variety of the activation functions such as rectified linear unit (ReLU), leaky ReLU, tanh, maxout, bipolar sigmoid, and so on are proposed to avoid the vanishing gradient problem (refer to Fig. 10). Each activation function with different specific output can not be vulnerable to the vanishing gradient or even can propagate the gradient through deeper neural networks.

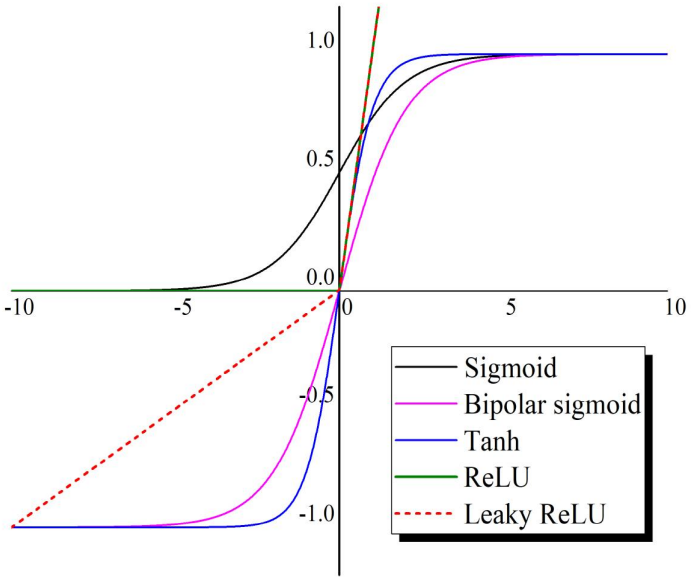


Fig. 10. Outputs of the activation functions

The overfitting problem, another consideration by complicated networks, denotes that the DNNs are over-trained using the learning data. With the introduction of the dropout method [26], recently, this problem related to generalization for the deep learning methods is able to be easily solved. The idea of the dropout method is a reduction of the complexity of the DNN by deactivating the arbitrary nodes selected according to the user-specified droprate in a single training procedure (refer to Fig. 11). In other words, the DNNs are trained by only activated nodes.

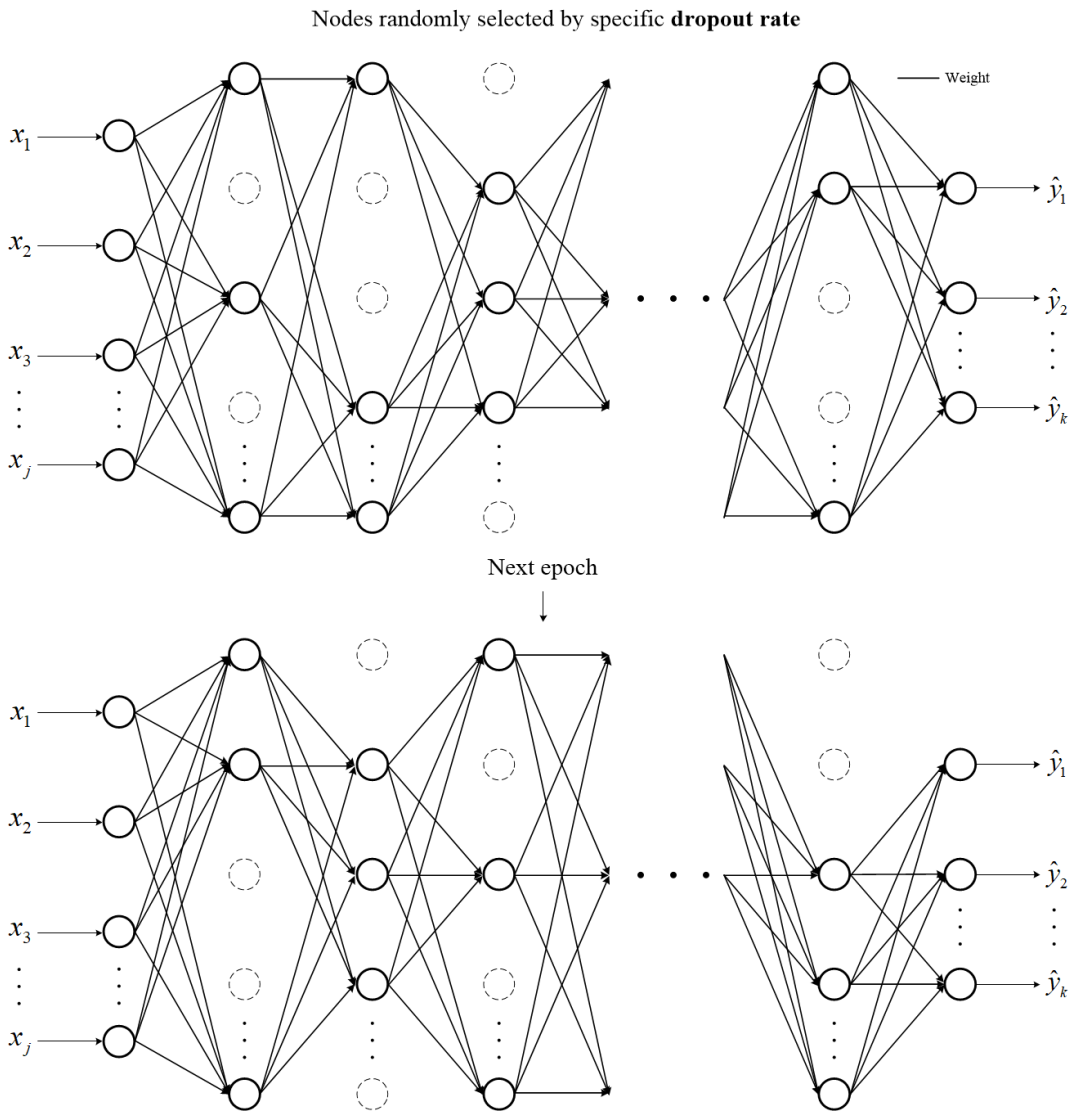


Fig. 11. The DNN model employing the dropout

Furthermore, cross validation is another method to prevent the DNN from the overfitting problem by usually dividing the data into the learning data set, the validation data set, and the test data set according to a specific proportion by users. The learning data set is used for the DNN training and the validation data set is used to check whether the overfitting occurs or not, which are directly related to the development of the DNN model. The test data set is literally used to independently test the extent to which the developed DNN model fits separate samples well. Fig. 12 indicates the development process of the model using the learning and validation data sets. In each epoch, the validation data are utilized to compute an error after the DNN model is trained using the learning data set. When the model is optimally trained, the error on the validation data is generally minimum. If the DNN model is over-trained (too much accurate for only the learning data) beyond this point, it can be regarded that the overfitting happens at the point that the error on the validation data starts to increase.

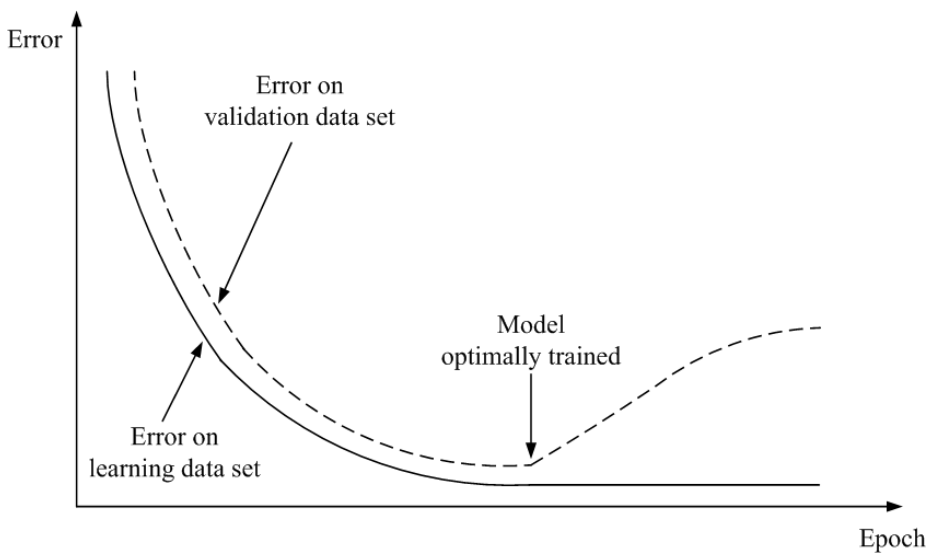


Fig. 12. Cross validation process for the model using the validation data set

The hyper-parameters correlative to the local minima which is a common problem for the optimizing using back-propagation are the cost function, learning rate α, weight initialization, and so on. As aforesaid, the convex function is generally used as the cost

function to easily approach to the global minima. Plus, the weight initialization technique is utilized not to be stuck in the false minima by changing a starting point of the gradient descent (refer to Fig. 13). Moreover, selecting a proper learning rate is crucial since it may fail to converge at the global minima on account of a local minima or divergence depending on the value of α in the convex function (refer to Fig. 14).

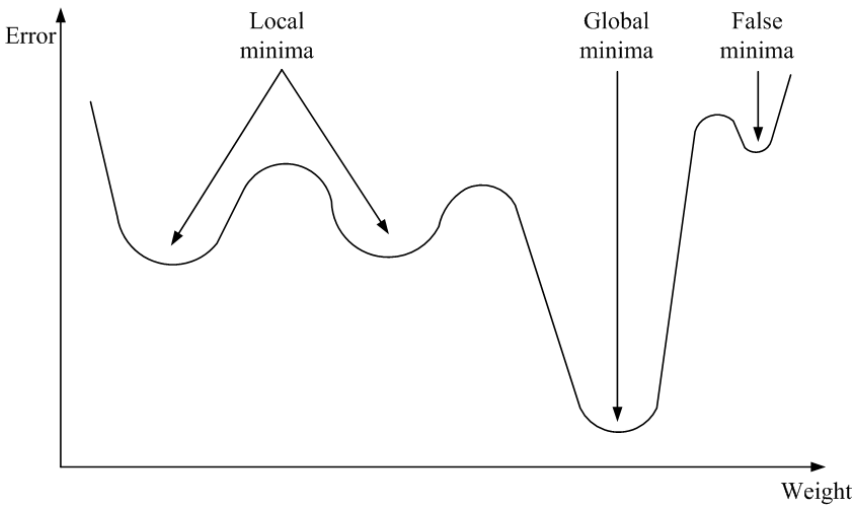


Fig. 13. Local minima and global minima in a non-convex function

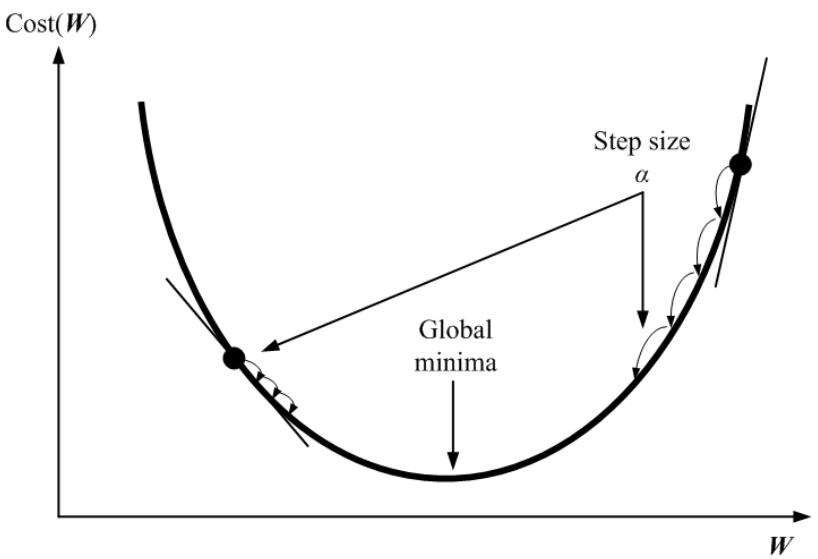


Fig. 14. Step size of gradient descent depending on an alpha value

In addition to the aforementioned hyper-parameters, there are several parameters and techniques to enhance the performance of the DNNs. However, the optimal setting for the hyper-parameters of the deep learning methods, called as a golden rule, has yet to be established. Although several searching methods for the hyper-parameters were proposed in the previous studies [27]-[29], the initial setting for these design parameters can be variable depending on the domains. Thus, it is obvious that a lot of efforts to guarantee powerful performance of the DNN model are required due to the fact that the deep learning methods have generally more design parameters than the traditional machine learning methods.

3. Genetic Algorithm for the Hyper-parameters

In an attempt to find out the optimal hyper-parameters of the DNN model, the genetic algorithm (GA) [30,31] was used in this study. Briefly, the GA is a technique artificially modelling an evolutionary process of living organisms by natural evolution mechanisms such as selection, crossover, and mutation [32]. As the generation proceeds in the GA, the populations of chromosomes are arbitrarily generated, and then iteratively replaced with another population of chromosomes newly generated by natural evolution [19]. A chromosome means a candidate solution to a problem. Finally, the fittest chromosome in the populations is selected as an appropriate solution to the optimization for the hyper-parameters [32]. The GA employs the fitness function of Eq. (4) evaluating how well a chromosome in a population fits a desired objective by assigning a score to each chromosome in the corresponding population as follows:

$$F = \exp(\lambda_1 E_l + \lambda_1 E_v + \lambda_2 E_{\max \cdot l} + \lambda_2 E_{\max \cdot v}) \tag{4}$$

where λ_1 and λ_2 are the coefficients for the RMS errors (RMSE), E_l and E_v, and the maximum errors, $E_{\max \cdot l}$ and $E_{\max \cdot v}$ for the learning data and the validation data, respectively.

In this thesis, the number of the hidden layers and nodes was configured by the GA. As aforementioned above, altering the hidden layers is a primary way to determine the

performance of the DNN model, and moreover a variation of the performance is remarkably checked depending on this parameter option. Setting the proper number of the hidden layers is usually performed by a conventional method manually changing the number of the hidden layers after finding the point where the accuracy on the learning data set becomes too much increasing (overfitting). However, it seems to be a quite limited method to be applied to each different subject since the options for the hyper-parameters can be variable. For this reason, the GA was used to get closer to the global minima by automatically finding the optimized number of the hidden layers and nodes although the GA is likely to be computationally expensive [19]. Fig. 15 shows the flowchart of the DNN model employing the GA in this thesis.

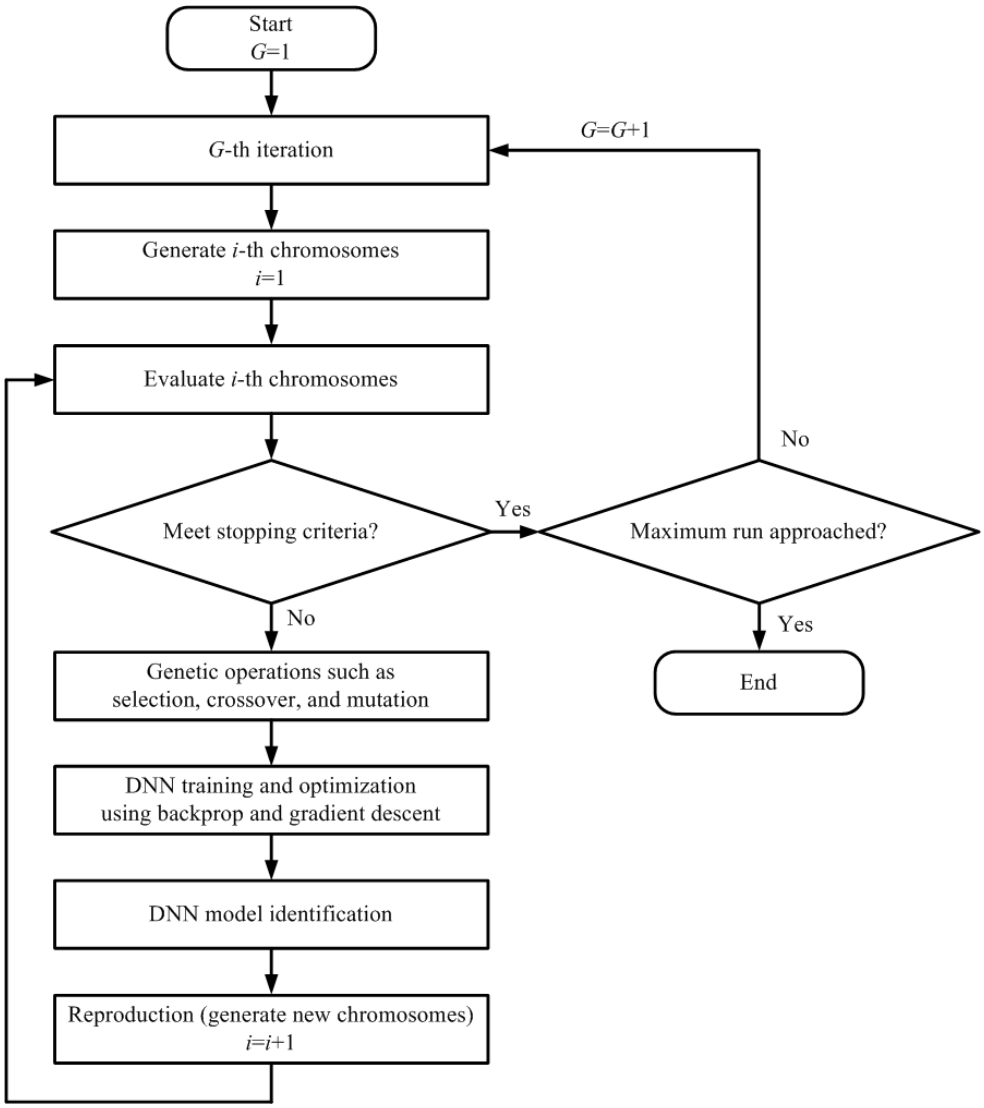


Fig. 15. Flowchart of the DNN model with the genetic algorithm

4. Employed DNN model

A part of the employed hyper-parameters of the DNN model utilized for the prediction of the major factors of the NPPs in this thesis are shown in Table 1. Notwithstanding there are several well-known activation functions and techniques such as the ReLU function and the dropout, the bipolar sigmoid function and the cross validation were employed for the DNN model in this thesis since the better performance of the DNN model was shown. Additionally, it is noted that the number of hidden layers and nodes, selected by the GA, is changeable depending on the range of the its maximum or minimum, and the GA options such as the probability of genetic operations.

Table 1. Initial settings applied to the DNN model

Hyper-parameter	Application setting
No. of hidden layers	Randomly selected by the genetic algorithm (variable depending on the option)
No. of hidden nodes	Randomly selected by the genetic algorithm (variable depending on the option)
Activation function	Bipolar sigmoid
Cost function	Root mean square
Learning rate (alpha)	0.1

B. FNN of Machine Learning Methods

Machine learning methods as a subset of the AI have been successfully applied to various fields and known for their good performances. In addition to the SVMs [9,10] which is one of the representative machine learning methods, there are several algorithms based on the neural networks such as the FNNs. It is noted that there is a main difference on the basic idea between the methods with a neural network structure. The FNN model is trained from the data through the specific processing layers.

1. Fuzzy Neural Networks

The conditional rule, described in a fuzzy *if-then* rule comprised of an antecedent and a consequent, is generally used in the FIS [5,6]. In the Mamdani-type FIS [33], the defuzzification is needed since its *if* and *then* parts are fuzzy linguistic while in the Takagi-Sugeno-type FIS [34], the inputs and outputs are usually real-valued variables, and its *if* part is fuzzy linguistic and its *then* part is crisp [21]. Therefore, the Takagi-Sugeno-type FIS was used for the FNN model in this thesis as follows:

$$\begin{aligned} &If\ x_1(k)\ is\ \mu_{i1}\ AND\ \cdots\ AND\ x_m(k)\ is\ \mu_{im} \\ &then\ \hat{y}^i(k)\ is\ f^i(x_1(k),\ \cdots, x_m(k)) \end{aligned} \tag{5}$$

where $x_j(k)$ is the input to the FNN model $(j = 1, 2, \cdots, m)$, μ_{ij} is the membership function of the *i*-th fuzzy rule $(i = 1, 2, \cdots, n)$ and the *j*-th input variable, and $f^i(x_1(k), x_2(k), \cdots, x_m(k))$ is the output of the *i*-th fuzzy rule and is expressed as a first-order polynomial of the input variables as shown in Eq. (6):

$$\hat{y}^i(k) = f^i(x_1(k), x_2(k), \cdots, x_m(k)) = f^i(\boldsymbol{x}(k)) = \sum_{j=1}^{m} q_{ij} x_j(k) + r_i \tag{6}$$

where q_{ij} is the weight at the *i*-th fuzzy rule and the *j*-th variable, and r_i is the bias at the *i*-th fuzzy rule.

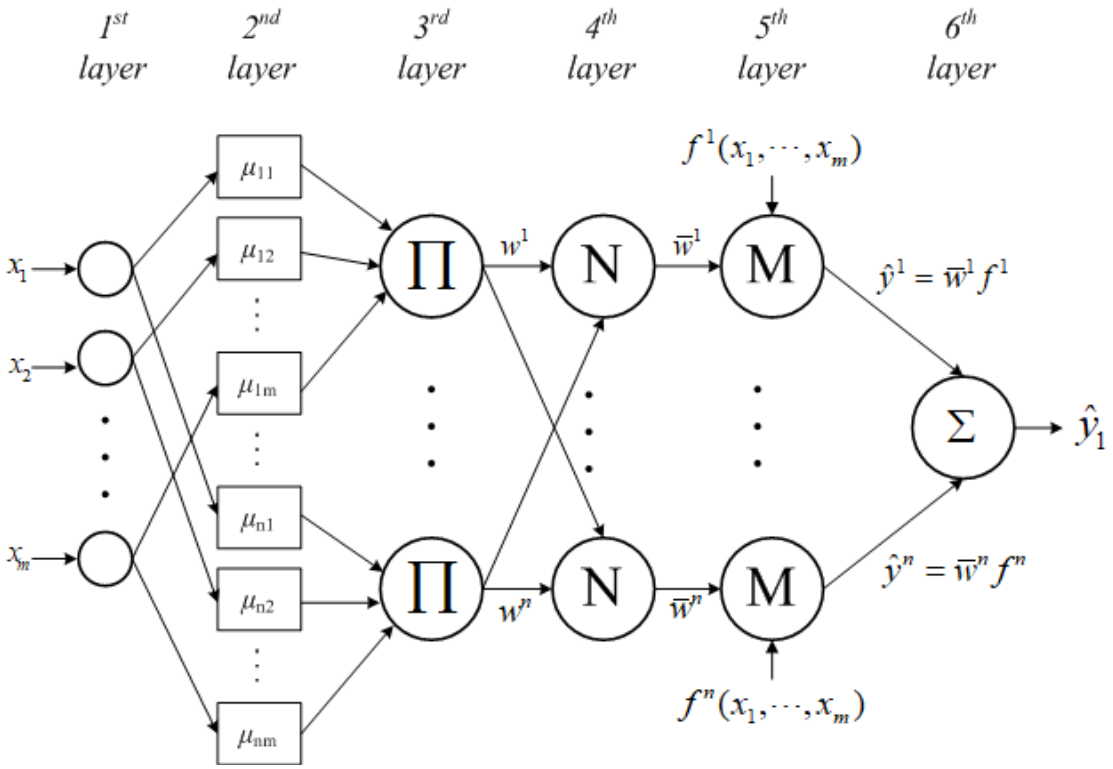


Fig. 16. Takagi-Sugeno type FIS in a neural network framework

Fig. 16 shows the FNN model using the Takagi-Sugeno-type FIS embodied in the neural network framework. The first layer consists of the nodes which transfer the input $x_j(k)$ to the next layer. The second layer is a fuzzification layer using a membership function. There are a variety of membership functions, which have no restriction of the shape. Among them, the symmetric bell-shaped Gaussian function was used as the membership function of each input at the *i*-th fuzzy rule for the FNN model expressed as Eq. (7):

$$\mu_{ij}(x_j(k)) = \exp\left(-\frac{(x_j(k)-c_{ij})^2}{2\sigma_{ij}^2}\right) \tag{7}$$

where c_{ij} and σ_{ij} are the center position of a peak and the sharpness of the Gaussian function at the *i*-th rule and the *j*-th input, respectively. These are the antecedent parameters for the FNN model as well as key parameters for the symmetric Gaussian membership function.

In the third layer, a product operator for the membership function is performed using Eq. (8) in order to calculate the membership values for each fuzzy rule. This membership value for rule *i*, $w^i(k)$, denotes a compatibility grade between the antecedent parts of a fuzzy *if-then* rule [21]. Normalization of the membership values for each fuzzy rule is carried out in the fourth layer using Eq. (9). In the fifth layer, the outputs for the *i*-th fuzzy rule are multiplied by the normalized values. In the sixth layer, all the calculated values from the fifth layer are summed up as expressed in Eq. (10). The output of a FIS with *n* fuzzy rules, $\hat{y}(k)$, is a weighted sum of the consequents of all the fuzzy rules [21]. Therefore, the predicted output using the FNN model is expressed by the vector product defined as Eq. (11).

$$w^i(k) = \prod_{j=1}^{m} \mu_{ij}(x_j(k)) \tag{8}$$

$$\overline{w}^i(k) = \frac{w^i(x(k))}{\sum_{i=1}^{n} w^i(x(k))} \tag{9}$$

$$\hat{y}(k) = \sum_{i=1}^{n} \overline{w}^i(k)\hat{y}^i(k) = \sum_{i=1}^{n} \overline{w}^i(k) f^i(\boldsymbol{x}(k)) \tag{10}$$

$$\hat{y}(k) = \boldsymbol{w}^T(k)\boldsymbol{q} \tag{11}$$

where $\boldsymbol{w}(k)$ and $\boldsymbol{q}$ are defined as

$$\boldsymbol{w}(k) = \left[\overline{w}^1 x_1(k) \ \cdots \ \overline{w}^n x_1(k) \ \cdots\cdots \ \overline{w}^1 x_m(k) \cdots \ \overline{w}^n x_m(k) \ \ \overline{w}^1(k) \ \cdots \ \overline{w}^n(k)\right]^T,$$

$$\boldsymbol{q} = \left[q_{11} \cdots \ q_{n1} \cdots\cdots \ q_{1m} \ \cdots \ q_{nm} \ \ r_1 \cdots \ r_n\right]^T.$$

2. Optimization of the FNN

The FNN model for predicting the major factors was optimized using the GA combined with the least square algorithm [32]. The GA was used to find the optimal antecedent parameters (c_{ij} and σ_{ij}) of the FNN model. Additionally, it is generally known that the GA is less vulnerable to convergence at the local minima than the traditional techniques and is a useful and effective method to solve the optimization problems with multiple objectives [32]. The least square algorithm was used to easily compute the consequent parameter $\boldsymbol{q}$ of Eq. (11). The squared error between the target value and the predicted value of Eq. (12) is used as the objective function.

$$J = \sum_{k=1}^{N_l} \left(y(k) - \hat{y}(k)\right)^2 = \sum_{k=1}^{N_l} \left(y(k) - \boldsymbol{w}^T(k)\boldsymbol{q}\right)^2 = \frac{1}{2}(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)^2 \tag{12}$$

where $\boldsymbol{y}_t = \left[y(1)\, y(2) \cdots y(N_l)\right]^T$ is a target value vector, $\hat{\boldsymbol{y}}_t = \left[\hat{y}(1)\, \hat{y}(2) \cdots \hat{y}(N_l)\right]$ is a predicted value vector from the FNN model, and N_l is the number of the learning data.

Eq. (13) is a solution to minimize the objective function of Eq. (12).

$$\boldsymbol{y}_t = \boldsymbol{W}_t \boldsymbol{q} \tag{13}$$

where the matrix $\boldsymbol{W}_t = \left[\boldsymbol{w}(1)\ \boldsymbol{w}(2)\ \cdots\ \boldsymbol{w}(N_l)\right]^T$ has $N_l \times (m+1)n$-dimensional and the vector $\boldsymbol{q}$ has $(m+1)n$-dimensional. The parameter $\boldsymbol{q}$ can be solved using the pseudo-inverse of the matrix $\boldsymbol{W}$ as follows:

$$\boldsymbol{q} = (\boldsymbol{W}_t^T \boldsymbol{W}_t)^{-1} \boldsymbol{W}_t^T \boldsymbol{y}_t \tag{14}$$

3. Fuzzy Neural Networks in a Cascade Structure

To improve the performance of the FNNs, the CFNNs [35] that consist of multi-connected FNN modules were proposed. The CFNN model used for the major factor prediction has a serially connected cascade structure and more than two FNN modules expressed as Fig. 17 [5]-[7]. The key aspect of the CFNNs is based on the syllogistic fuzzy inference where a calculated output from the previous FNN module is transferred to the next module, of which the calculation process is the same as all of the modules, as a fact. This is important to effectively establish a large-scale system with high-level intelligence [35]. The CFNNs perform the prediction process by iteratively adding the FNN modules. Eq. (15) is the random *i*-th fuzzy rule at the *l*-th FNN module of the CFNNs.

$$\begin{aligned} &If\ x_1(k)\ is\ \mu_{i1}^{l}\ AND\ \cdots\ AND\ x_1(k)\ is\ \mu_{im}^{l}, \\ &\quad AND\ \hat{y}_1(k)\ is\ \mu_{i(m+1)}^{l}(k)\ AND\ \cdots\ AND\ \hat{y}_{(l-1)}(k)\ is\ \mu_{i(m+l-1)}^{l}(k), \\ &then\ \hat{y}_l^i(k)\ is\ f_l^i(x_1(k),\ \cdots, x_m(k),\ \hat{y}_1(k),\ \cdots,\ \hat{y}_{(l-1)}(k)) \end{aligned} \tag{15}$$

Continuous training by adding the FNN modules can make the CFNNs powerful, but simultaneously it can be susceptive to the overfitting problem. Therefore, the cross validation was used to prevent this problem. The overfitting of the CFNN model was able to be checked by the fractional error for the validation data in Eq. (16).

$$E_f(G) = \frac{\sum_{k=1}^{N_v}\left(y_G(k) - \hat{y}_G(k)\right)^2}{\sum_{k=1}^{N_v}\left(y_G(k)\right)^2} \tag{16}$$

where N_v is the number of the validation data.

When the condition $E_f(G+1))>E_f(G)$ is satisfied, it is regarded that the overfitting begins at the (G+1)-th module if an FNN module is continuously added. The process of adding an FNN module stops at this point. Additionally, if an error at L-th module $E_f(L)$ is less than the specific minimum fractional error, the training and validation of the CFNN

model will be stopped. The CFNN model with the minimum fractional error in L-th module consists of the number of L FNN modules [6].

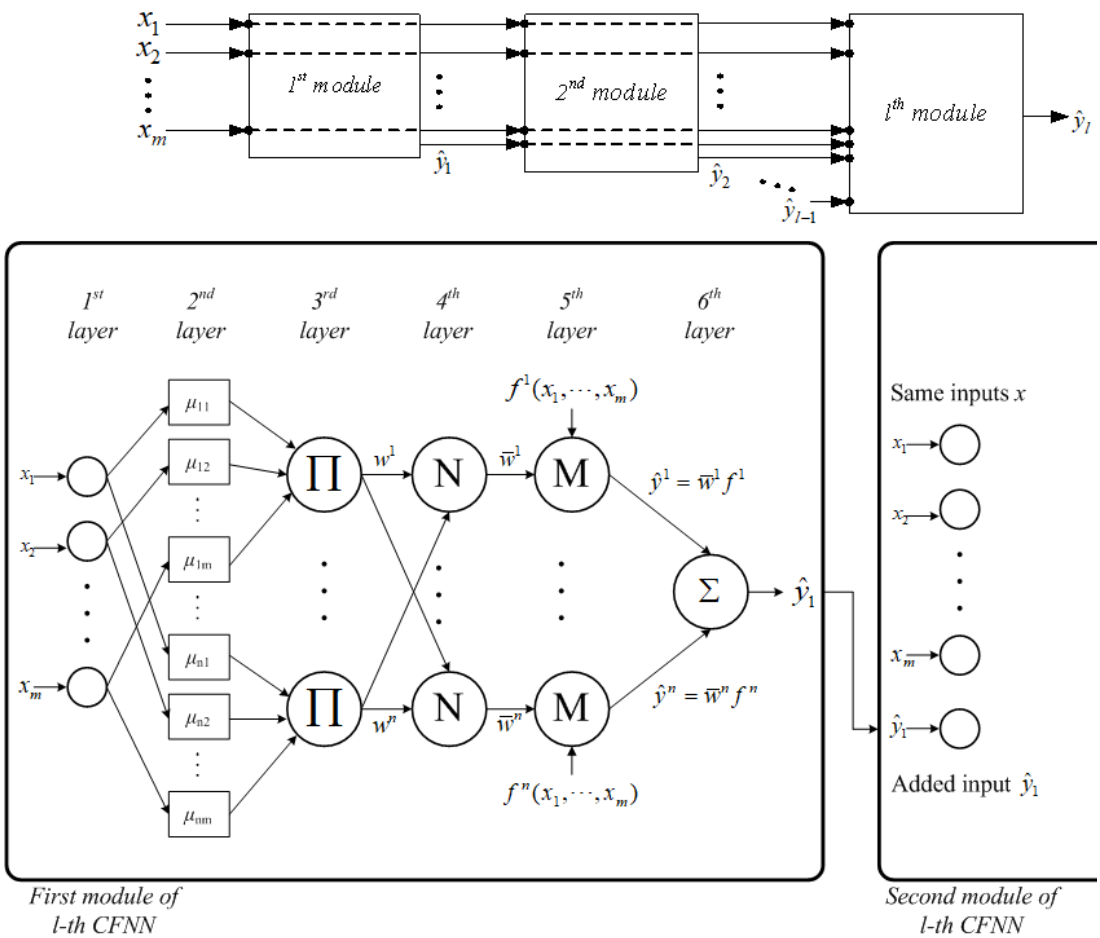


Fig. 17. Cascaded Fuzzy Neural Networks

III. Applied Data to Predict the Major Factors of NPPs

A. Simulation Data on the Postulated LOCAs

The data are required to train and verify the presented AI methods employing the supervised learning algorithms. However, since the data on the real NPP accident rarely exist, the MAAP [4] was used for acquisition of the accident simulation data for the NPPs. The MAAP code, as a severe accident analysis program for the PWR and BWR, is generally used to gain the data showing the behaviors of the major factors of the NPPs under the accident circumstances. In this thesis, the Hanul units 3 and 4, as a PWR type plant, were selected as the plant models. Some of the assumed loss of coolant accidents (LOCAs), which may occur in the NPPs, were simulated using the MAAP code in order to obtain the data applied to the mentioned AI models.

The postulated LOCAs were assumed to occur at three break positions such as hot-leg, cold-leg, and steam generator tube (SGT), and were classified into the small break LOCAs (SBLOCAs) and the large break LOCAs (LBLOCAs) depending on the break sizes. In addition, the accident scenario that several safety systems of the NPPs did not properly work under the LOCA circumstances was assumed.

To predict the major factors of the NPPs, a total number of 600 simulation data on these postulated LOCAs, obtained by simulating 200 different break sizes in each break position, respectively, were applied to the AI methods. In case of the hot-leg and cold-leg LOCAs, the 170 LOCA simulation data were for the larger break sizes and the remaining data were for the smaller break sizes, respectively. For the LOCAs at SGT (or steam generator tube rupture (SGTR)), the 100 LOCA simulation data were prepared for the smaller break sizes and the same amount of the LOCA data were prepared for the larger break sizes.

Moreover, these data were separated into three data sets for the cross validation. First, the test data were selected at fixed intervals and comprised of the number of 100 or 200 data points in each break position. After the test data points were removed, the validation

data points were selected in a similar way as the test data points. Among the rest of the data points, lastly, the learning data points were chosen at another fixed intervals.

B. Simulated Signals for Predicting the Major Factors

The accident simulation data from the MAAP code consist of various simulated instrumentation signals of the NPPs, which are numerically expressed. In this thesis, the postulated LOCA simulation data consist of the time-integrated values of 15 simulated instrumentation signals as follows:

$$x_j = \int_{t_s}^{t_s + \Delta t} g_j(t)dt, \;\; j = 1, 2, \cdots, 15 \tag{17}$$

where t_s is the elapsed time after the rector trip, Δt means a short integrating time interval after the reactor trip, and $g_j(t)$ denotes a specified signal.

Table 2 shows the signals of the postulated LOCA simulation data used in this thesis. Several simulated signals, which are considered as the monitoring parameters, such as containment pressure, RCS pressure, sump water level, and hydrogen concentration can be used to predict the major factor of the NPPs. It is noted that although the LOCA break size can not be measured in the actual accidents, it was considered as a parameter applicable to the prediction of the NPP factors in this thesis since it was quite accurately estimated in the previous studies using several AI methods [11]-[13], [36]. In this thesis, some of the NPP major factors were predicted applying a few signal data among the simulated signals to the proposed DNN model under the assumption that the NPP information was not ensured due to instrument unreliability by inability or malfunction in the severe accidents, which is the same as the previous studies [5]-[7].

Table 2. Simulated instrumentation signals from the MAAP

No.	Signal
1	Elapsed time after reactor trip
2	LOCA break size
3	Containment pressure
4	RCS pressure
5	Sump water level
6	Collapsed RV water level
7	Boiled-up RV water level
8	Hydrogen concentration in containment
9	Leak flow
10	Maximum leak flow
11	SG leak flow
12	Maximum SG leak flow
13	Maximum hydrogen concentration
14	Full RV water level
15	Maximum containment pressure

IV. Prediction of Major Factors Using Deep Learning

In this section, the prediction performance on the major factors using the proposed DNN model can be checked and are compared with the performance of the CFNN model of the previous studies [5]-[7]. In addition, a cause of the performance difference between the DNN and CFNN models is analyzed.

A. Prediction of RV Water Level

First, the RV water level, selected as a target factor in this thesis, was predicted using the DNN model under the assumed severe accident circumstances. The RV water level is a safety-critical factor directly related to determining the cooling capability for the nuclear fuels and the reactor core, and preventing the core from uncovery. Thus, it is obvious that the RV water level has to be accurately measured and monitored under the accident circumstances or normal operation condition. Three simulated signals (the elapsed time after the reactor trip, the estimated LOCA size, and the containment pressure) were applied to the DNN model to predict the RV water level. The containment pressure was used since the internal environment of the containment can be easier to be recognized than that of the RV or reactor coolant boundary [5].

Tables 3 shows the prediction performance of the DNN model and the number of hidden layers and nodes selected using the GA according to the LOCA cases. Table 4 shows the comparison of the prediction performance of the DNN and CFNN models [5] for the test data, respectively. According to Table 4, the proposed DNN model shows its more accurate prediction performance than the CFNN model in most of the LOCA cases except for the LBLOCAs at the hot-leg and the cold-leg.

Table 3. Prediction performance for the RV water level using the DNN model

LOCA break size	LOCA break position	Learning data		Test data		Hidden layers (nodes)
		RMSE (m)	Max. E (m)	RMSE (m)	Max. E (m)	
Small	Hot-leg	0.07	0.62	0.11	0.49	6 (19-10-20-20-6-20)
	Cold-leg	0.05	0.49	0.07	0.28	4 (17-19-17-12)
	SGT	0.04	0.38	0.03	0.10	6 (10-15-5-13-19-5)
Large	Hot-leg	0.03	0.41	0.08	0.61	9 (7-18-10-12-5-18-10-9-5)
	Cold-leg	0.07	1.16	0.37	1.71	9 (7-14-12-12-5-18-10-10-18)
	SGT	0.04	0.39	0.09	0.26	9 (7-14-12-12-5-18-10-10-18)

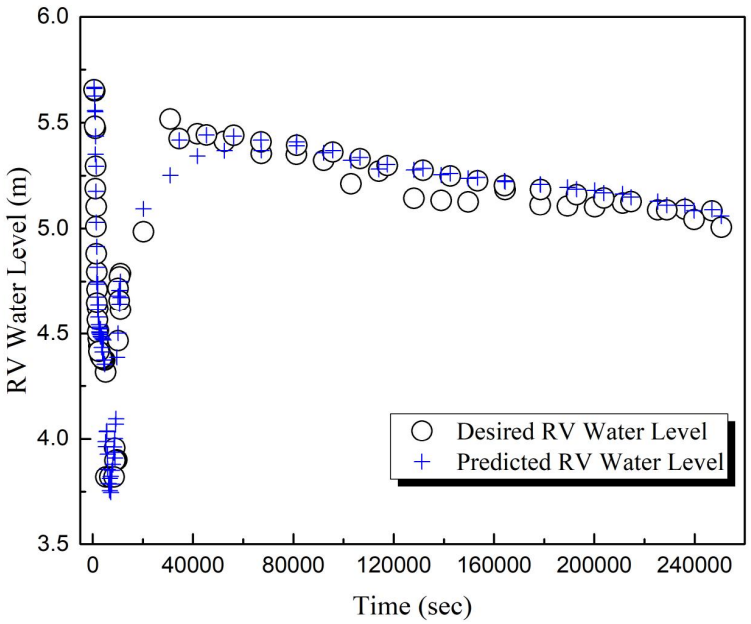
Table 4. Comparison of prediction performance of the DNN and CFNN models for the RV water level

LOCA break size	LOCA break position	Test data			
		DNN model		CFNN model	
		RMSE (m)	Max. E (m)	RMSE (m)	Max. E (m)
Small	Hot-leg	0.11	0.49	0.32	1.76
	Cold-leg	0.07	0.28	0.20	0.79
	SGT	0.03	0.10	0.22	0.82
Large	Hot-leg	0.08	0.61	0.07	0.64
	Cold-leg	0.37	1.71	0.15	0.60
	SGT	0.09	0.26	0.50	3.24

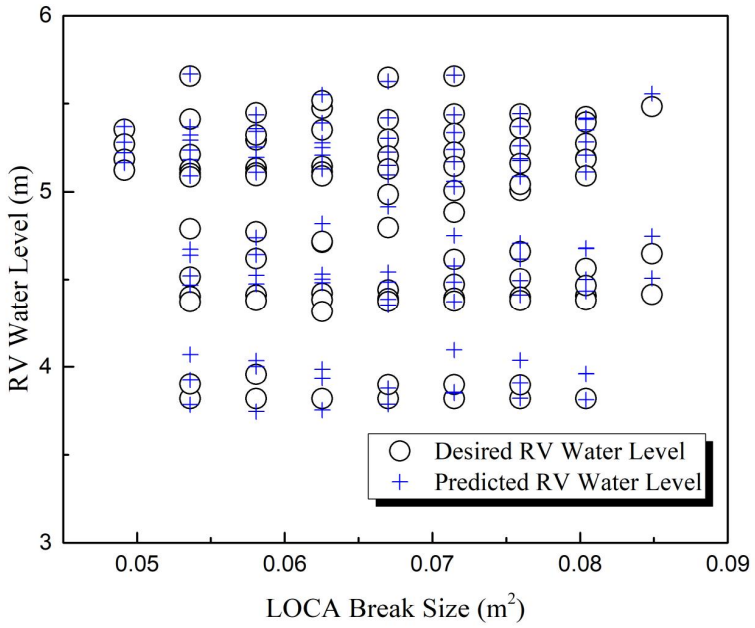
Figs. 18-20 indicate the prediction results of the RV water level according to the elapsed time after the reactor trip, the estimated LOCA break size, and the containment pressure for the test data using the DNN model in case of the SBLOCAs at each break position. Figs. 21-23 indicate the prediction results of the RV water level according to the elapsed time, estimated LOCA break size, and containment pressure for the test data using the DNN model under the LBLOCAs at each break position.

According to Figs. 20-25, the results with better prediction performance for the test data are made for the SBLOCAs and LBLOCAs at SGT. In these cases, the predicted values from the DNN model ('+' symbol) catch up with the target values ('○' symbol) relatively well. Some undesirable prediction results are caused by a rapid change of the RV water level in a short time in all the LOCAs.

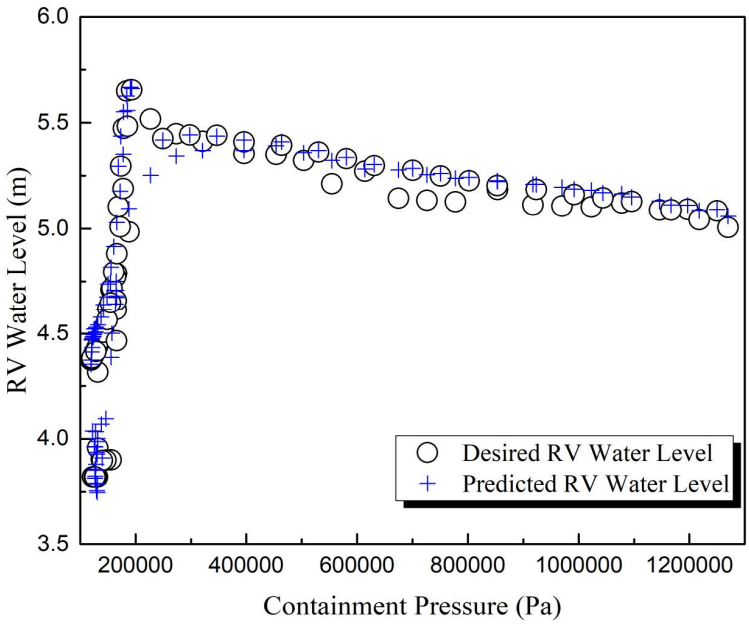
In result, although the CFNN model in the previous study [5] has also good accuracy for the RV water level prediction, the DNN model is regarded as a better method since its RMSE and maximum error for the test data are lower than the CFNN model in most cases.



(a) Prediction of RV water level according to elapsed time

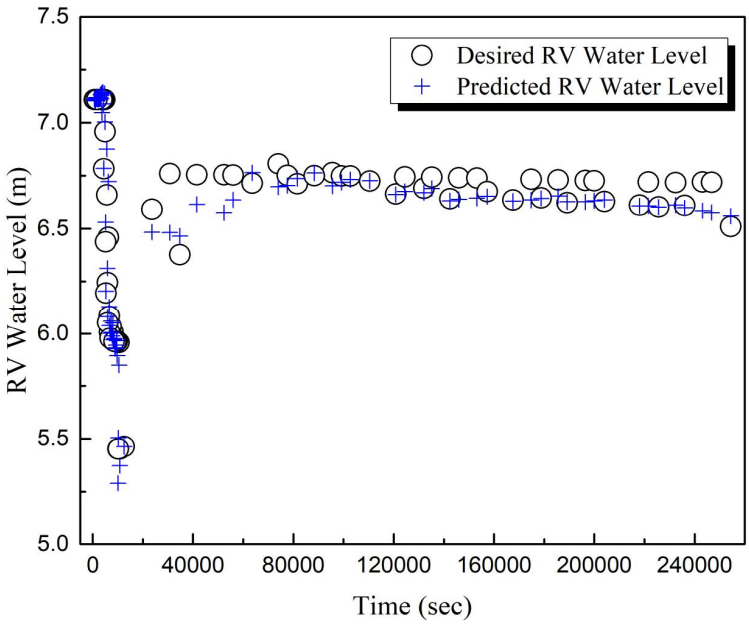


(b) Prediction of RV water level according to estimated LOCA break size

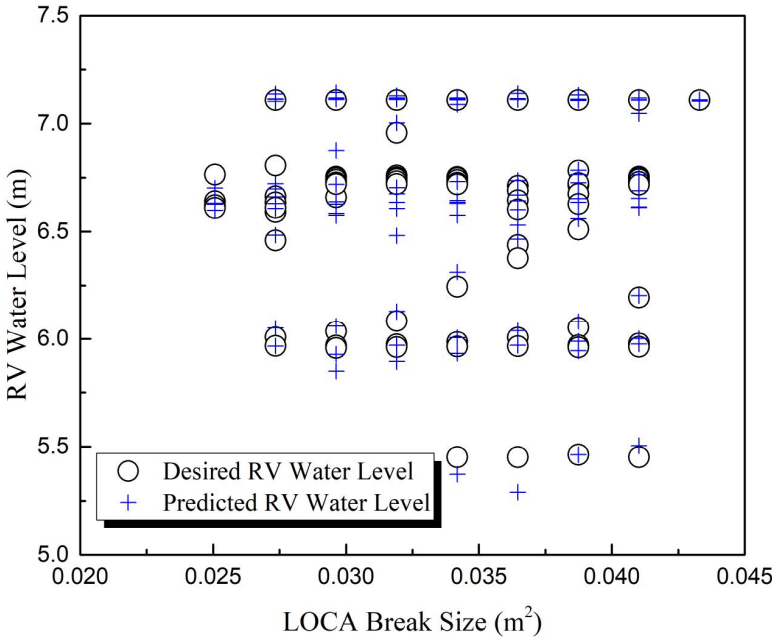


(c) Prediction of RV water level according to containment pressure

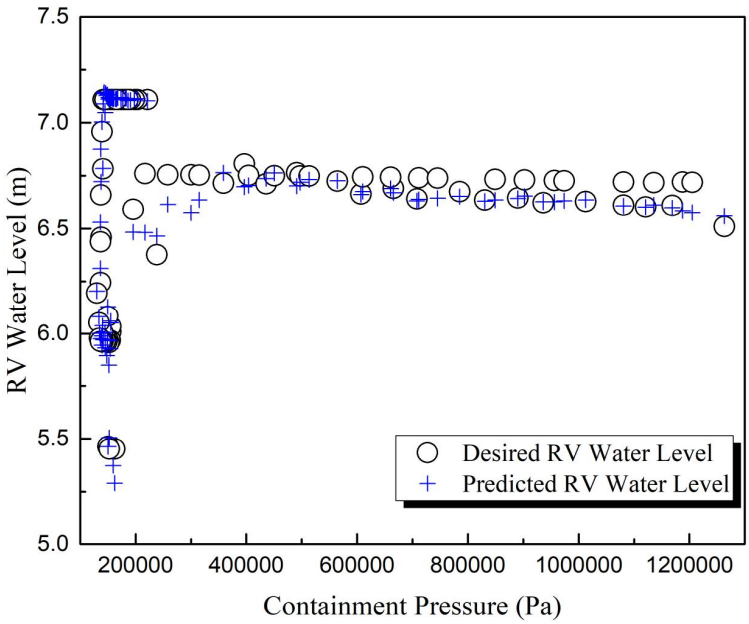
Fig. 18. Performance of the DNN model for the test data under SBLOCAs at hot-leg (RV water level)



(a) Prediction of RV water level according to elapsed time

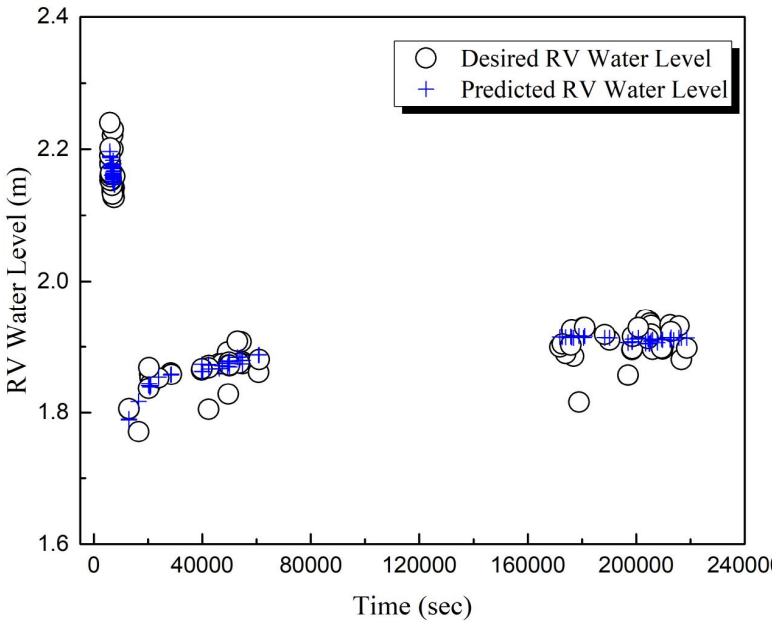


(b) Prediction of RV water level according to estimated LOCA break size

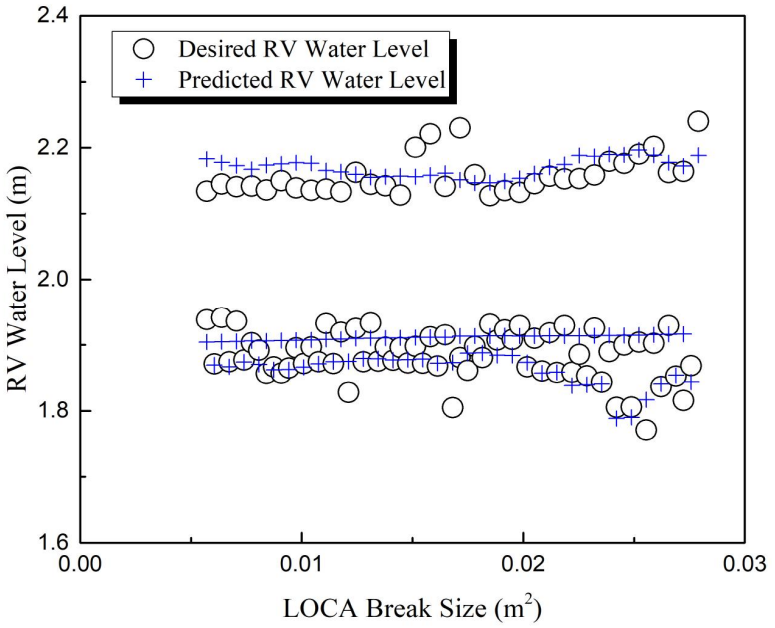


(c) Prediction of RV water level according to containment pressure

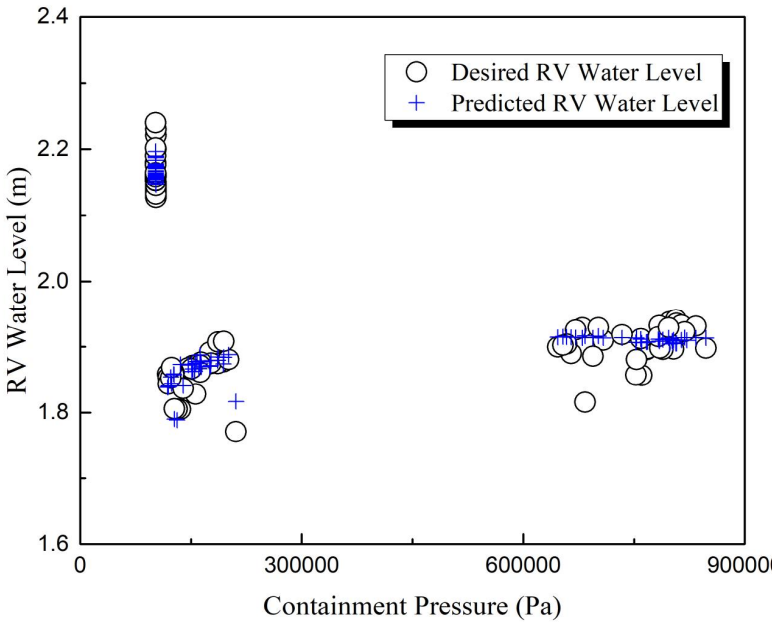
Fig. 19. Performance of the DNN model for the test data under SBLOCAs at cold-leg (RV water level)



(a) Prediction of RV water level according to elapsed time

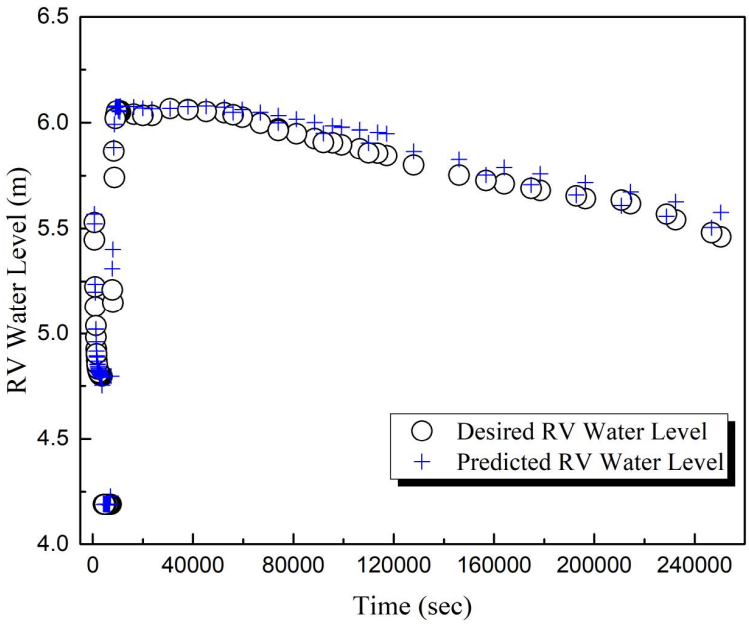


(b) Prediction of RV water level according to estimated LOCA break size

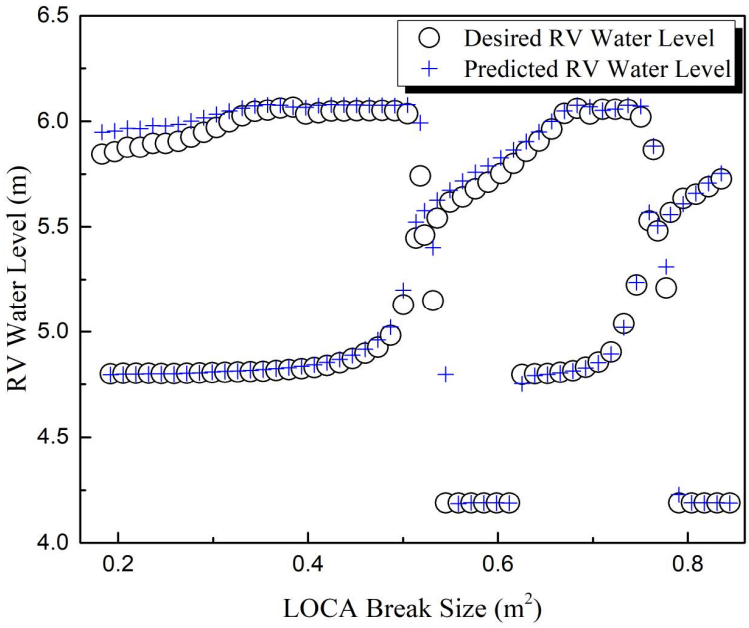


(c) Prediction of RV water level according to containment pressure

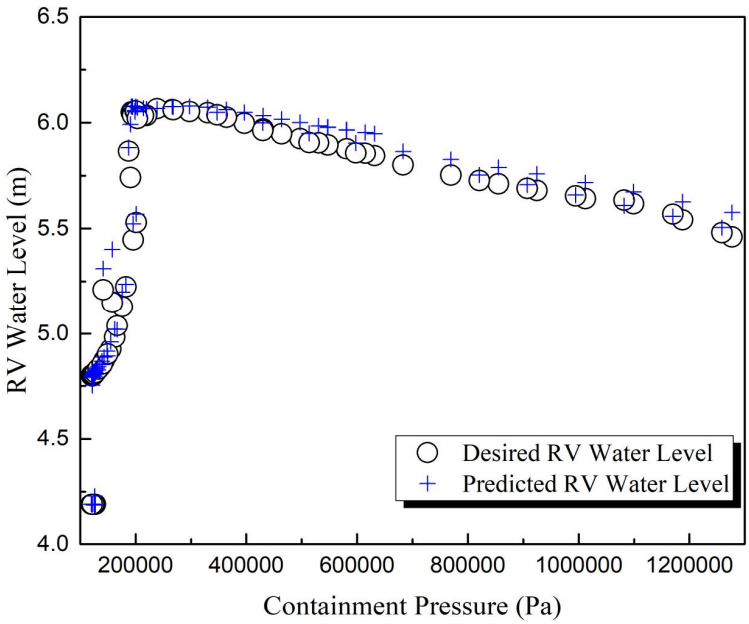
Fig. 20. Performance of the DNN model for the test data under SBLOCAs at SGT (RV water level)



(a) Prediction of RV water level according to elapsed time

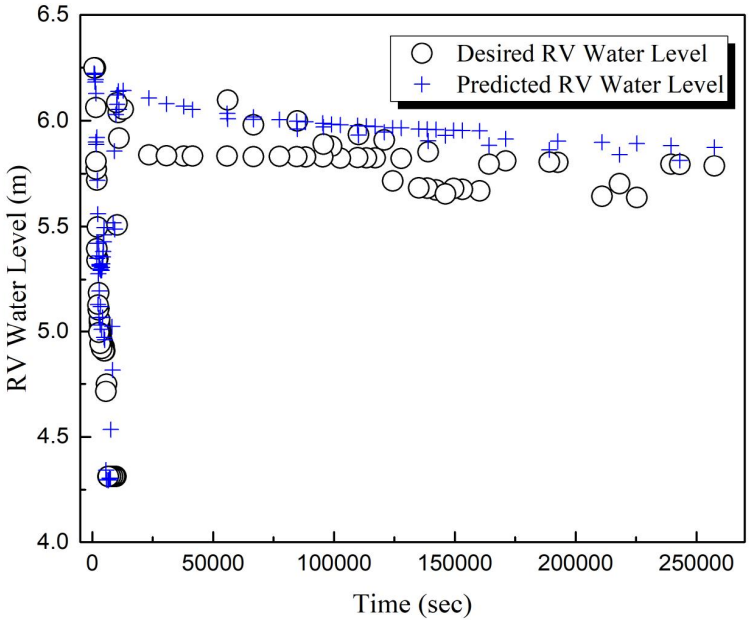


(b) Prediction of RV water level according to estimated LOCA break size

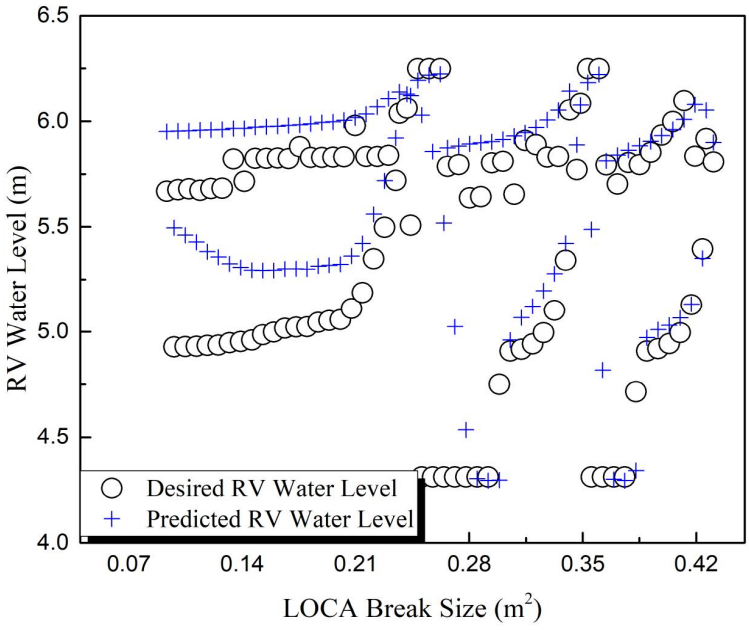


(c) Prediction of RV water level according to containment pressure

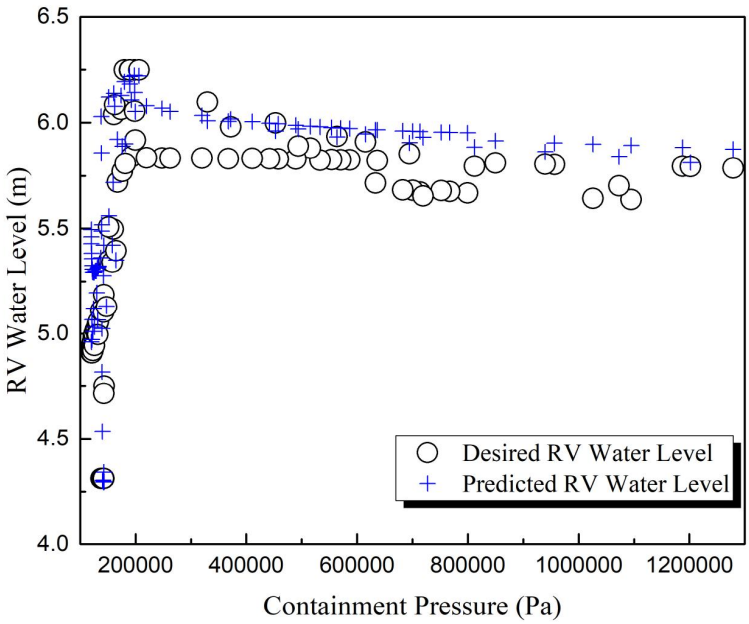
Fig. 21. Performance of the DNN model for the test data under LBLOCAs at hot-leg (RV water level)



(a) RV water level prediction according to elapsed time

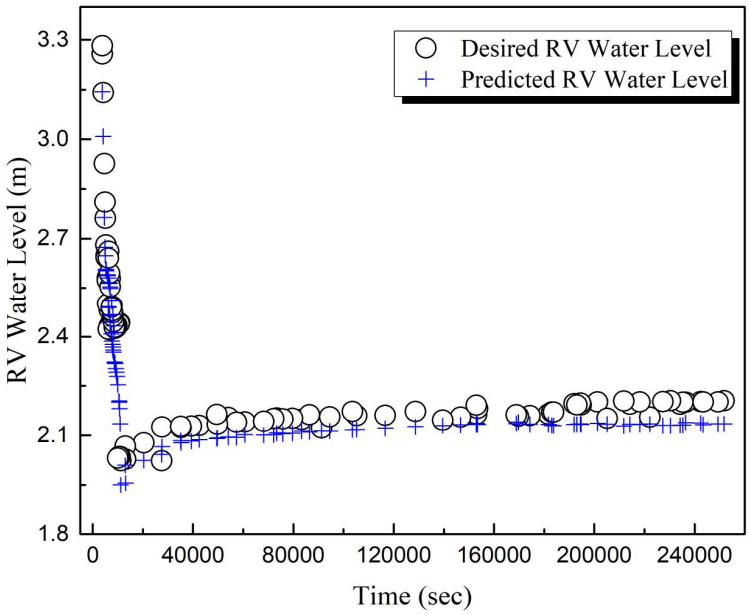


(b) Prediction of RV water level according to estimated LOCA break size

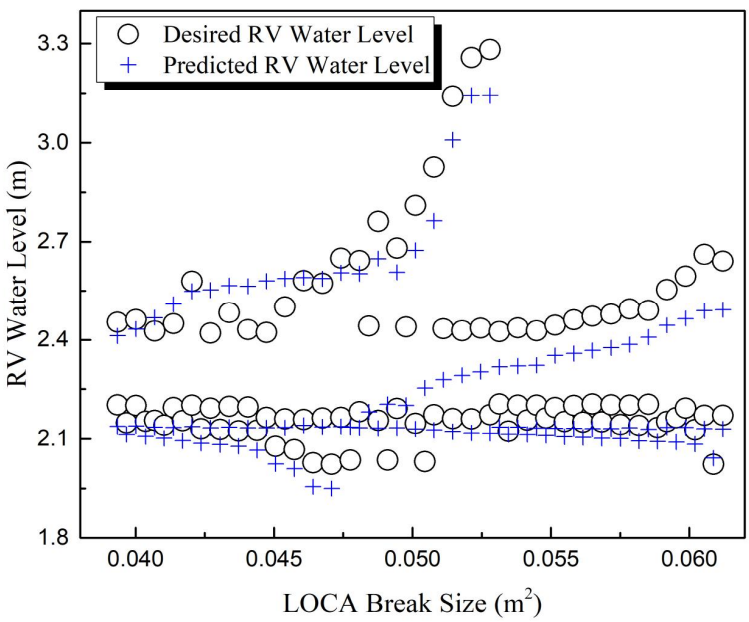


(c) Prediction of RV water level according to containment pressure

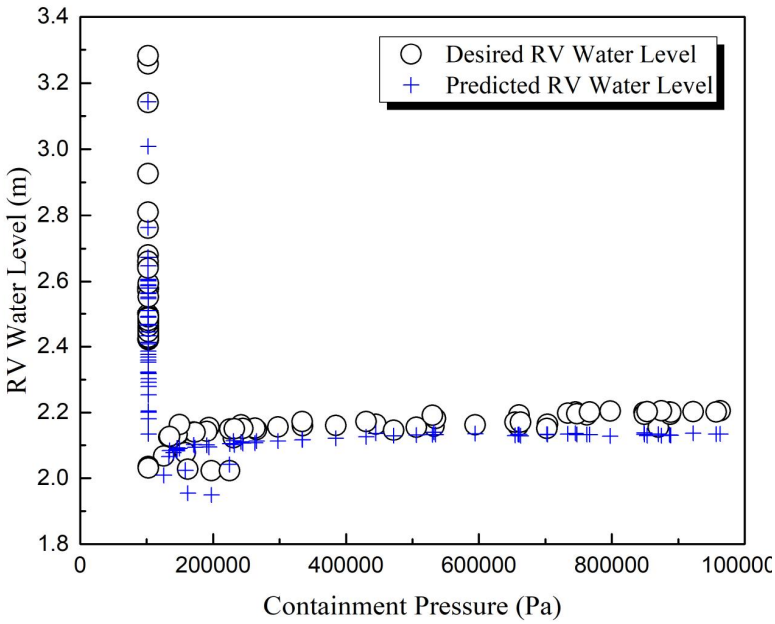
Fig. 22. Performance of the DNN model for the test data under LBLOCAs at cold-leg (RV water level)



(a) Prediction of RV water level according to elapsed time



(b) Prediction of RV water level according to estimated LOCA break size



(c) Prediction of RV water level according to containment pressure

Fig. 23. Performance of the DNN model for the test data under LBLOCAs at SGT (RV water level)

B. Prediction of Hydrogen Concentration

As another target factor in this thesis, an increase in the hydrogen concentration in the containment is mainly caused by metallic material oxidation during the progress of the severe accident phenomena [6], [37,38]. Specifically, the hydrogen gas is generated from metal oxidation such as zirconium (Zr) cladding by the water (or steam) during the in-vessel phenomena. Especially, the Zr cladding rapidly reacts with the water (or steam) at high temperature environment which is originated from the LOCAs, and accordingly the hydrogen generation is accelerated and the hydrogen concentration is considerably increased. Ex-vessel oxidation of the metallic material during direct containment heating (DCH) and molten core-concrete interaction (MCCI) makes also the hydrogen concentration in the containment higher. For these reasons, the containment failure can occur by explosion of the mixture of a increasingly higher hydrogen with air oxygen. Thus, the hydrogen concentration in the containment must be maintained below 4% by accurately measuring the hydrogen gas to prevent detonation and progression to the major accident circumstance. To predict the hydrogen concentration in the containment, two simulated signals (the elapsed time after the reactor trip and the estimated LOCA break size) were applied to the DNN model.

Table 5 indicates the number of each hidden layer and node and prediction performance of the DNN model for the hydrogen concentration in the containment. As listed in Table 6, the DNN model shows around 1% RMSE and considerably reduced maximum error for the test data than the CFNN model. For these reasons, therefore, the DNN model is a more reliable method for the hydrogen concentration prediction.

Table 5. Prediction performance for the hydrogen concentration in the containment using the DNN model

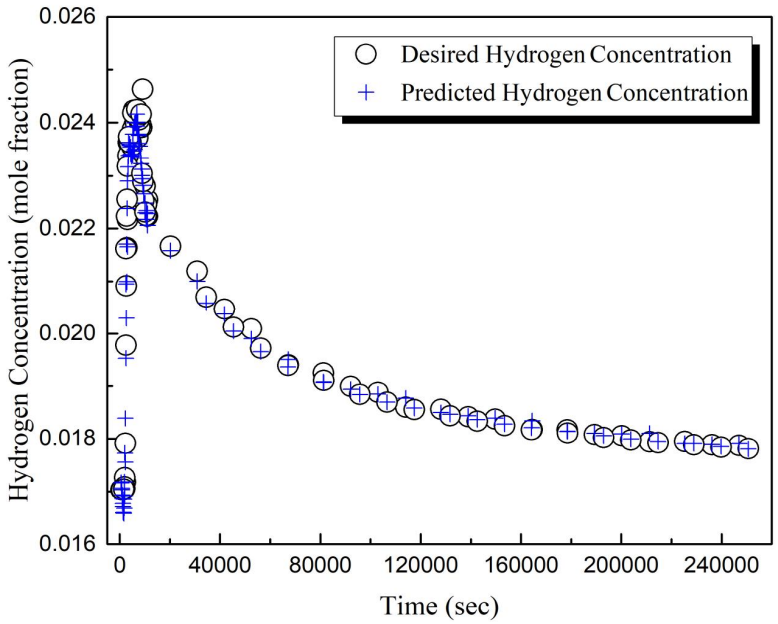
LOCA break size	LOCA break position	Learning data		Test data		Hidden layers (nodes)
		RMSE (%)	Max. E (%)	RMSE (%)	Max. E (%)	
Small	Hot-leg	0.58	4.29	0.86	4.54	4 (17-19-17-12)
	Cold-leg	0.21	5.13	0.43	1.91	6 (10-16-6-13-19-11)
	SGT	0.71	7.06	1.33	6.56	3 (19-20-8)
Large	Hot-leg	0.36	4.90	0.87	6.72	4 (5-12-8-17)
	Cold-leg	0.39	4.78	0.77	2.61	4 (5-12-8-17)
	SGT	0.71	10.41	1.64	6.24	9 (20-5-17-20-6-15-10-10-5)

Table 6. Comparison of prediction performance of the DNN and CFNN models for the hydrogen concentration in the containment

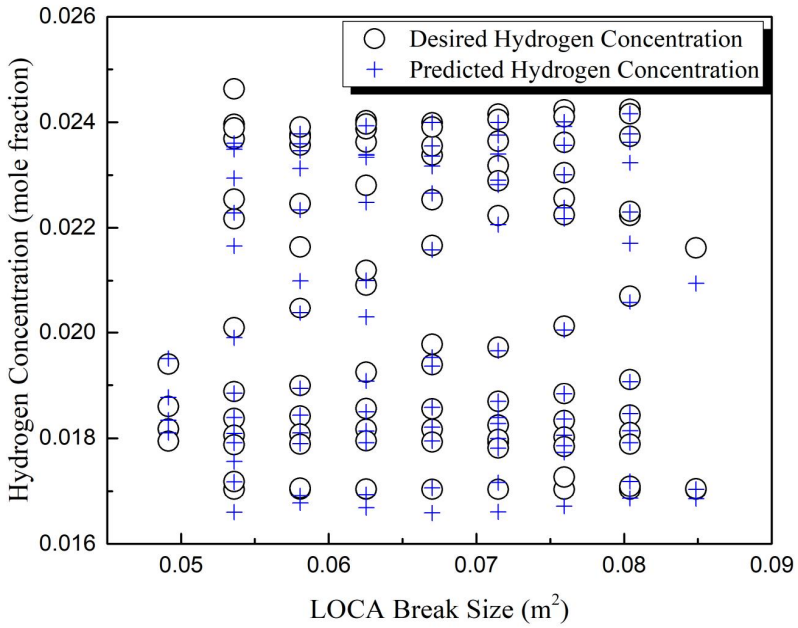
LOCA break size	LOCA break position	Test data			
		DNN model		CFNN model	
		RMSE (%)	Max. E (%)	RMSE (%)	Max. E (%)
Small	Hot-leg	0.86	4.54	2.10	13.09
	Cold-leg	0.43	1.91	5.38	48.84
	SGT	1.33	6.56	6.70	44.01
Large	Hot-leg	0.87	6.72	0.54	2.17
	Cold-leg	0.77	2.61	1.69	10.15
	SGT	1.64	6.24	7.21	44.66

Figs. 24-26 indicate the graphs showing the prediction results of the hydrogen concentration in the containment according to the elapsed time after the reactor trip and the estimated LOCA break size for the test data using the DNN model under the SBLOCAs at each break position. Figs. 27-29 show the graphs showing the prediction results of the hydrogen concentration in the containment according to the elapsed time after the reactor trip and the estimated LOCA break size for the test data using the DNN model under the LBLOCAs at each break position.

In case of the small and large SGTRs, there are predicted values relatively fitted away from the target values. These phenomena arise from the hydrogen concentration which is not affected in the initial severe accident phases, especially in the SGTRs [5,6]. Notwithstanding, the predicted values using the DNN model track a downward trend after the specific time. Hence, the proposed DNN model with an advanced performance is considered as a appropriate model to predict the hydrogen concentration in the containment under the severe accident circumstances.

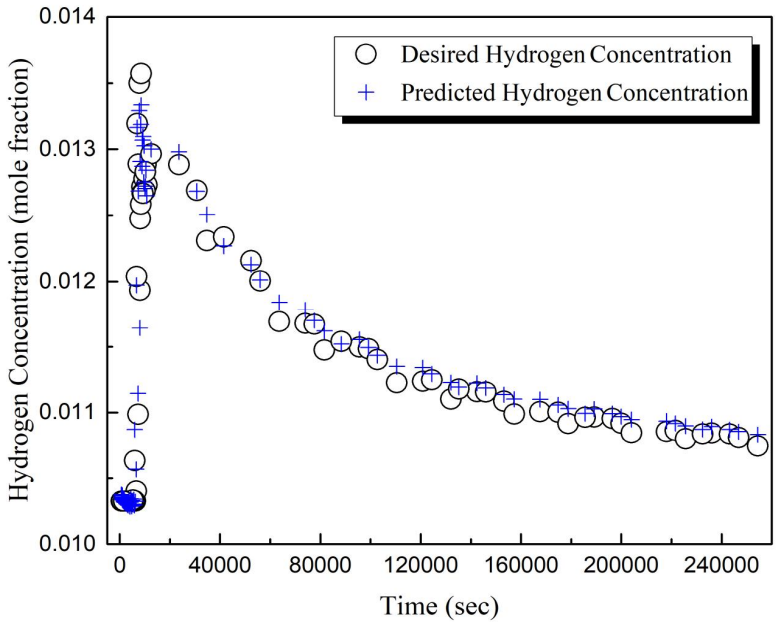


(a) Prediction of hydrogen concentration according to elapsed time

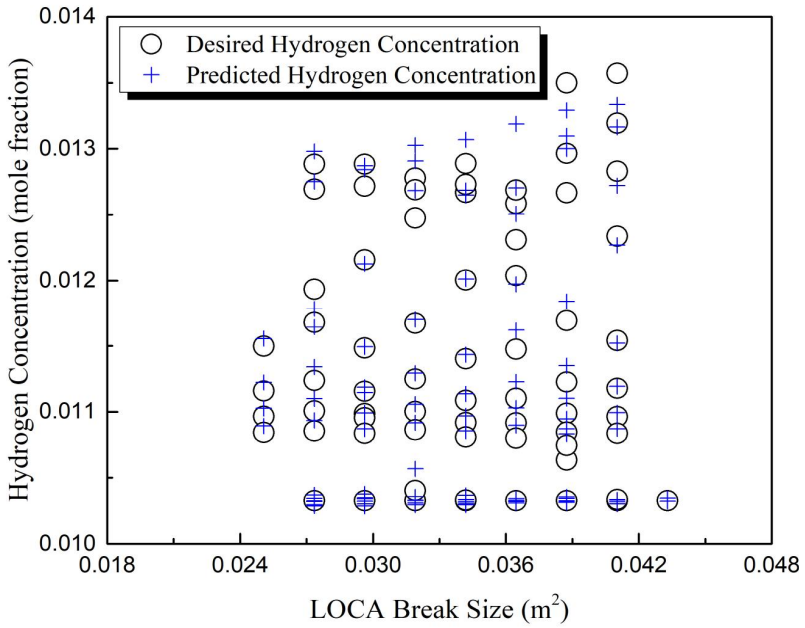


(b) Prediction of hydrogen concentration according to estimated LOCA break size

Fig. 24. Performance of the DNN model for the test data under SBLOCAs at hot-leg (hydrogen concentration in the containment)

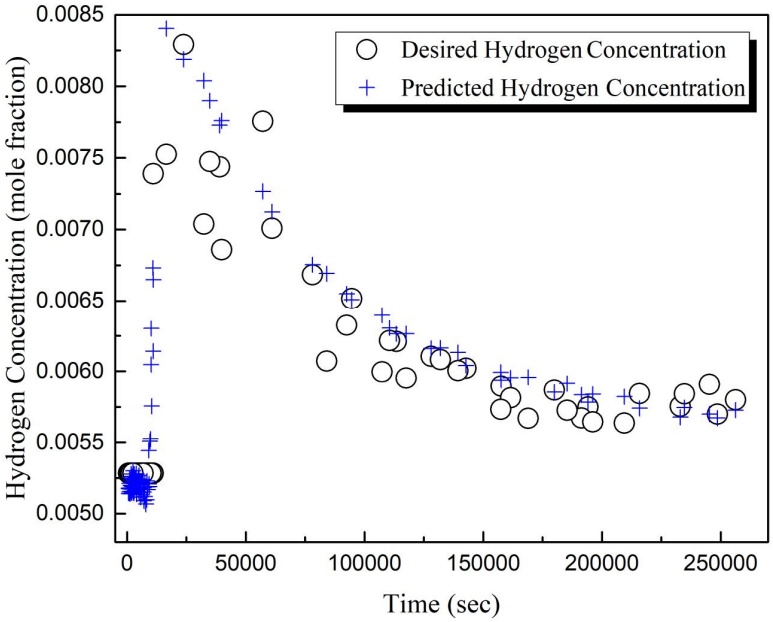


(a) Prediction of hydrogen concentration according to elapsed time

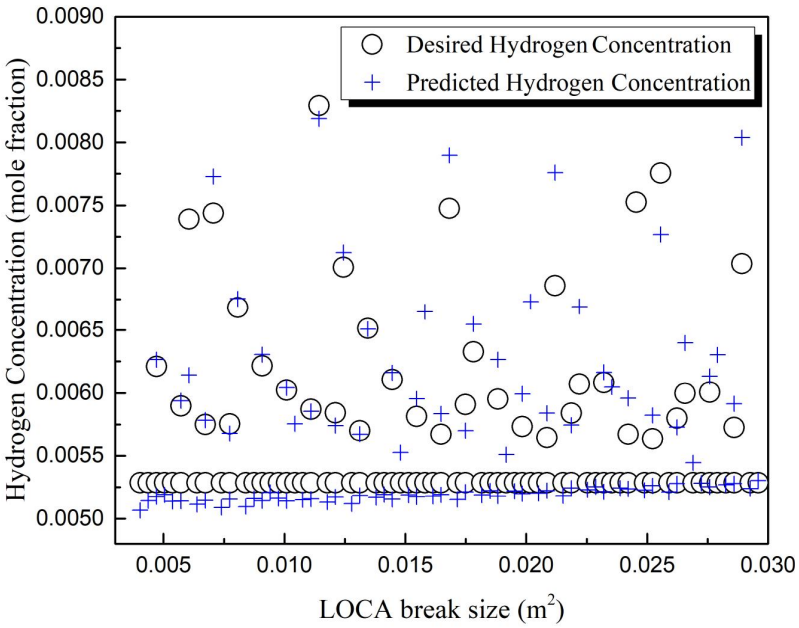


(b) Prediction of hydrogen concentration according to estimated LOCA break size

Fig. 25. Performance of the DNN model for the test data under SBLOCAs at cold-leg (hydrogen concentration in the containment)

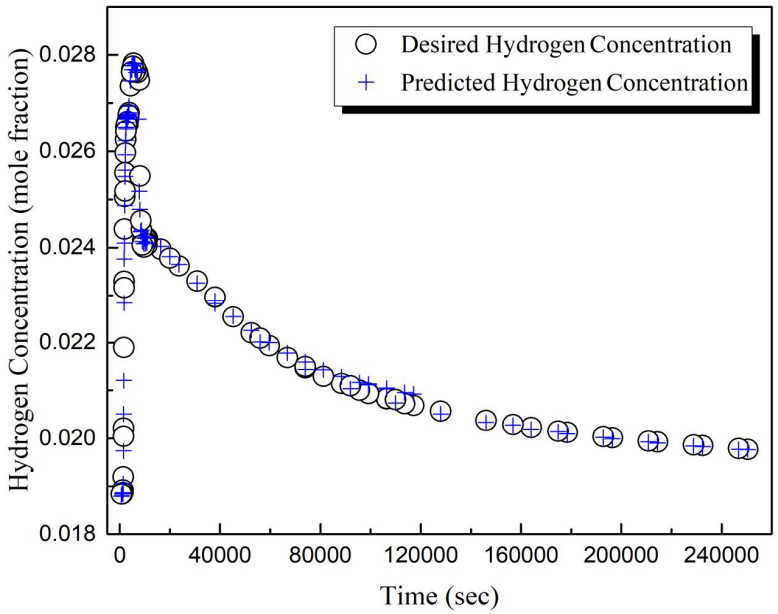


(a) Prediction of hydrogen concentration according to elapsed time

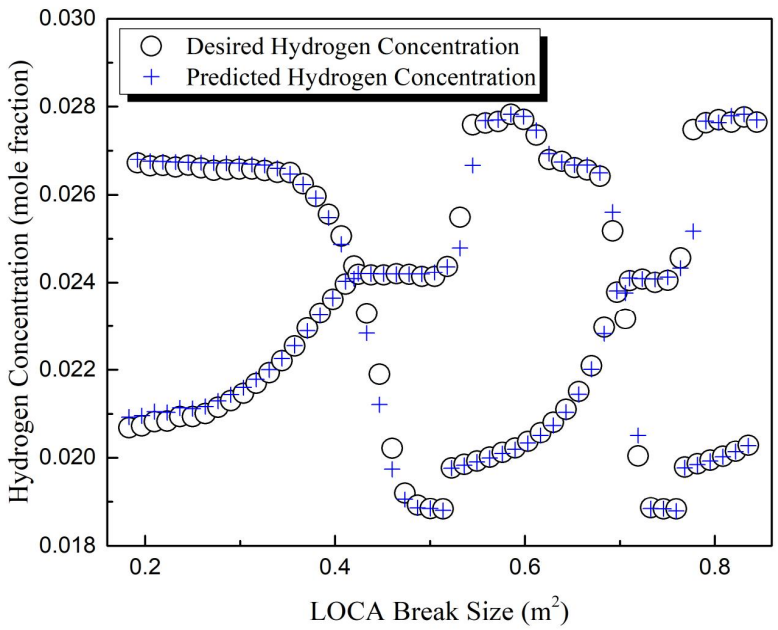


(b) Prediction of hydrogen concentration according to estimated LOCA break size

Fig. 26. Performance of the DNN model for the test data under SBLOCAs at SGT (hydrogen concentration in the containment)

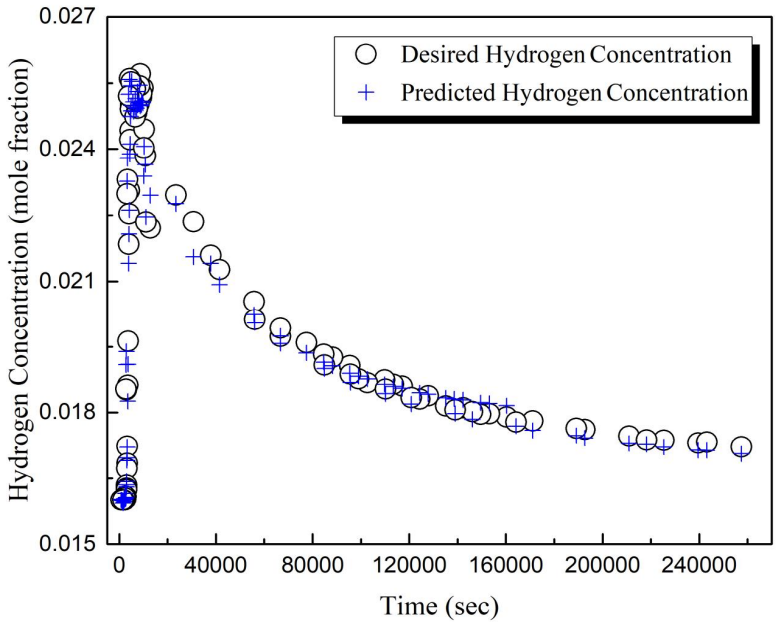


(a) Prediction of hydrogen concentration according to elapsed time

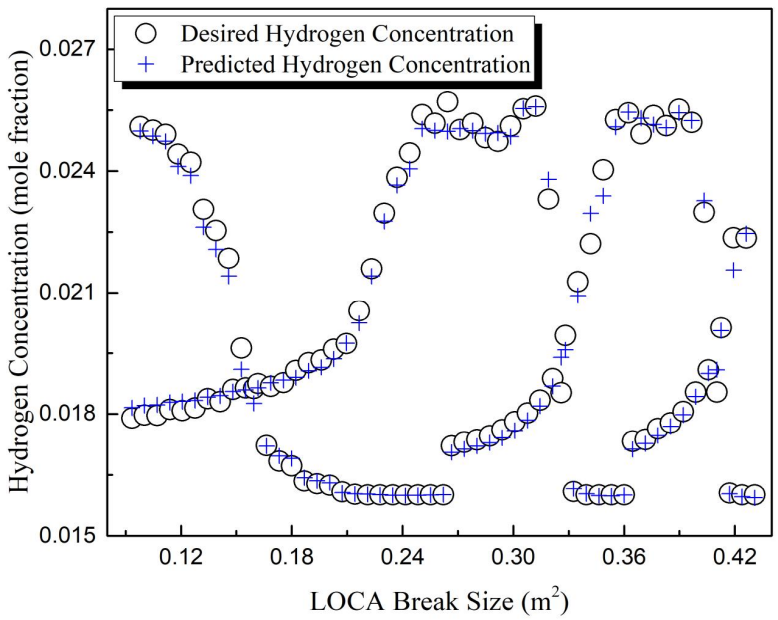


(b) Prediction of hydrogen concentration according to estimated LOCA break size

Fig. 27. Performance of the DNN model for the test data under LBLOCAs at hot-leg (hydrogen concentration in the containment)

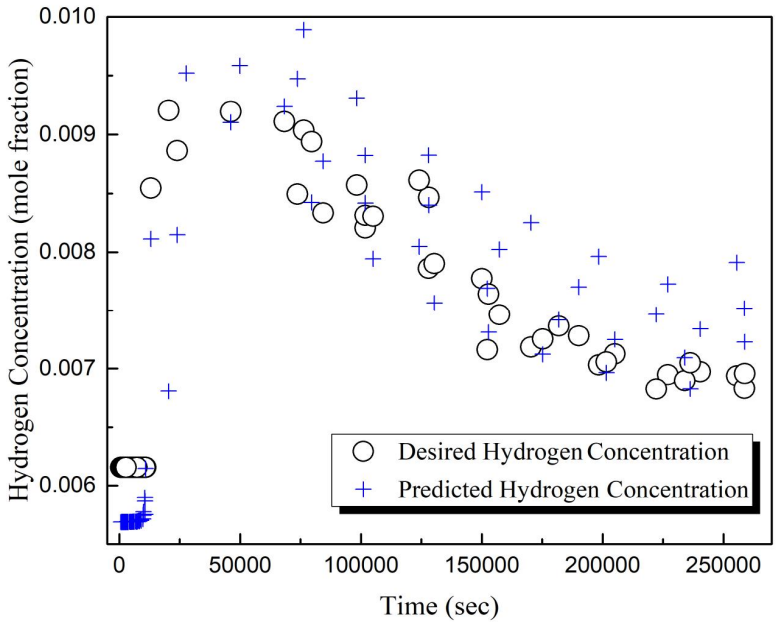


(a) Prediction of hydrogen concentration according to elapsed time

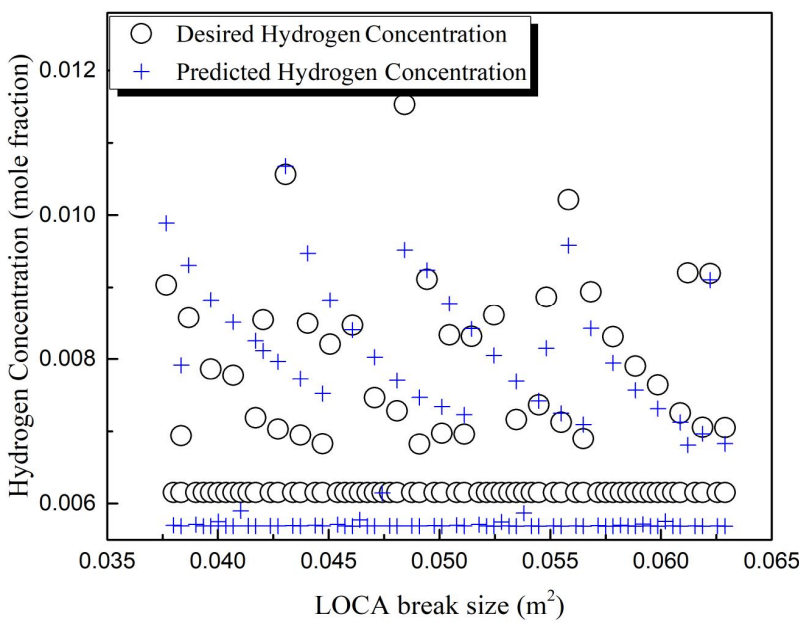


(b) Prediction of hydrogen concentration according to estimated LOCA break size

Fig. 28. Performance of the DNN model for the test data under LBLOCAs at cold-leg (hydrogen concentration in the containment)



(a) Prediction of hydrogen concentration according to elapsed time



(b) Prediction of hydrogen concentration according to estimated LOCA break size

Fig. 29. Performance of the DNN model for the test data under LBLOCAs at SGT (hydrogen concentration in the containment)

C. Prediction of Containment Pressure

The containment pressure can rapidly increase due to several reasons such as coolant leak, hydrogen gas generation, and so on under the severe accident phenomena [37,38] in which the integrity of the reactor is not maintained. As expressed in Fig. 1, since the containment surrounding the RCS and safety systems is a final barrier of the NPPs to prevent the release of radioactive materials outside and protect inner structures against external factors, accurately measuring the containment pressure is critical to avoid the containment failure by overpressure. As a parameter determining the containment environment, therefore, the containment pressure was selected as the third target factor of the NPPs in this thesis and was predicted applying the elapsed time after the reactor trip and the estimated LOCA break size to the DNN model under the severe accident circumstances.

Table 7 shows the prediction performance of the DNN model for the containment pressure according to the number of hidden layers and nodes, selected by the GA. As shown in Table 8, the performance of the DNN model is much better than that of the CFNN model [7] for the test data. Especially, the maximum error of the DNN model for the SGTR cases is notably reduced around 1%.

Table 7. Prediction performance for the containment pressure using the DNN model

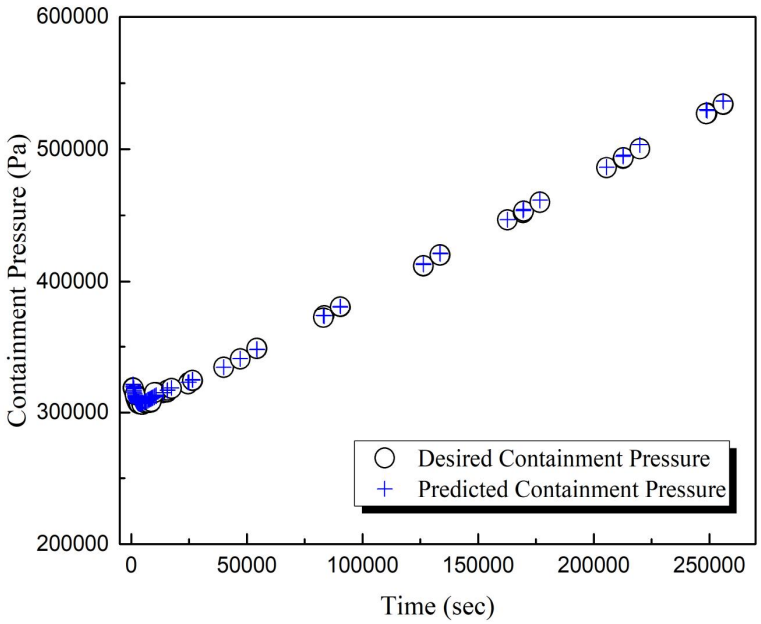
LOCA break size	LOCA break position	Learning data		Test data		Hidden layers (nodes)
		RMSE (%)	Max. E (%)	RMSE (%)	Max. E (%)	
Small	Hot-leg	0.05	0.62	0.10	0.23	6 (12-20-7-20-6-14)
	Cold-leg	0.06	0.63	0.10	0.33	5 (19-19-17-19-19)
	SGT	0.13	1.19	0.19	0.91	6 (8-19-18-15-17-11)
Large	Hot-leg	0.04	0.61	0.08	0.26	9 (20-14-19-11-5-18-5-6-8)
	Cold-leg	0.06	0.38	0.12	0.51	10 (16-13-13-10-11-11-15-10-16-8)
	SGT	0.13	1.32	0.26	1.14	9 (20-10-20-18-17-14-18-13-20)

Table 8. Comparison of prediction performance of the DNN and CFNN models for the containment pressure

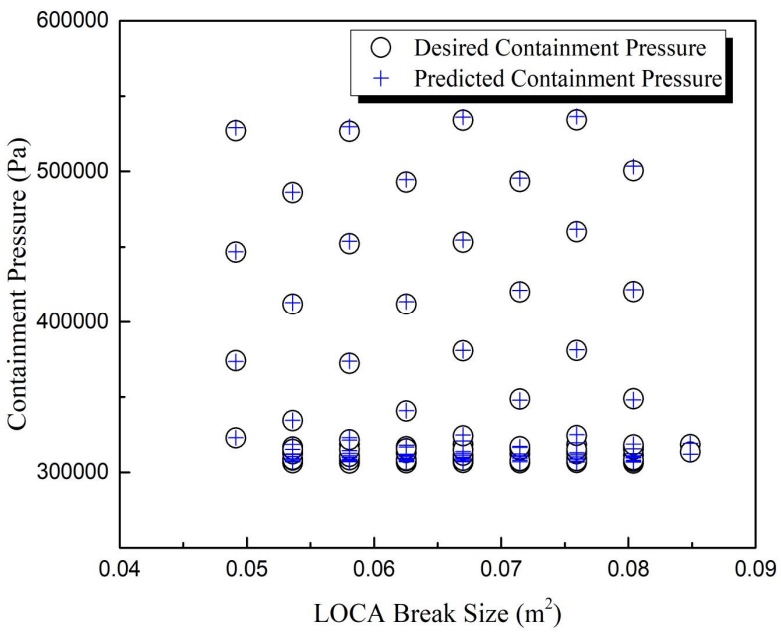
LOCA break size	LOCA break position	Test data			
		DNN model		CFNN model	
		RMSE (%)	Max. E (%)	RMSE (%)	Max. E (%)
Small	Hot-leg	0.10	0.23	0.13	0.73
	Cold-leg	0.10	0.33	0.24	0.77
	SGT	0.19	0.91	0.94	4.64
Large	Hot-leg	0.08	0.26	0.06	0.52
	Cold-leg	0.12	0.51	0.26	0.83
	SGT	0.26	1.14	1.41	7.22

Figs. 30-32 show the prediction results of the containment pressure according to the elapsed time and the estimated LOCA break size for the test data using the DNN model under the SBLOCAs at each break position. Figs. 33-35 show the prediction results of the containment pressure according to the elapsed time and the estimated LOCA break size for the test data using the DNN model under the LBLOCAs at each break position.

Among the Figs. 30-35, the containment pressure is precisely predicted using the DNN model especially in case of the hot-leg and cold-leg LOCAs. The reason why some of the inaccurate prediction values are shown in the SGTRs is that the containment pressure is not almost affected in the initial phases of these accidents. However, since the maximum error for the SGTR cases are approximately 1% (refer to Table 6), it is regarded that the proposed DNN model has more reliable prediction performance. Moreover, if the containment pressure signal, which was used for other factor prediction in this thesis, is not ensured in the severe accidents, its accurately predicted values from the DNN model can be used to predict other major factors.

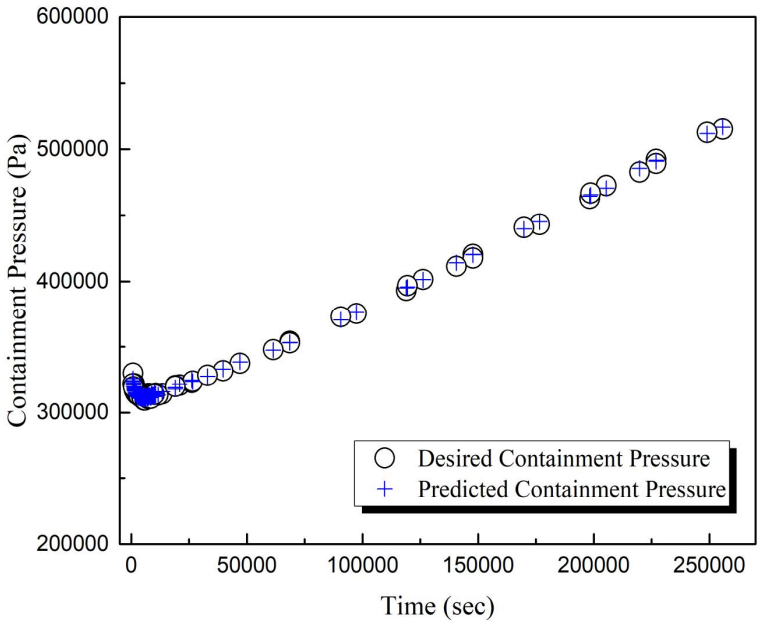


(a) Prediction of containment pressure according to elapsed time

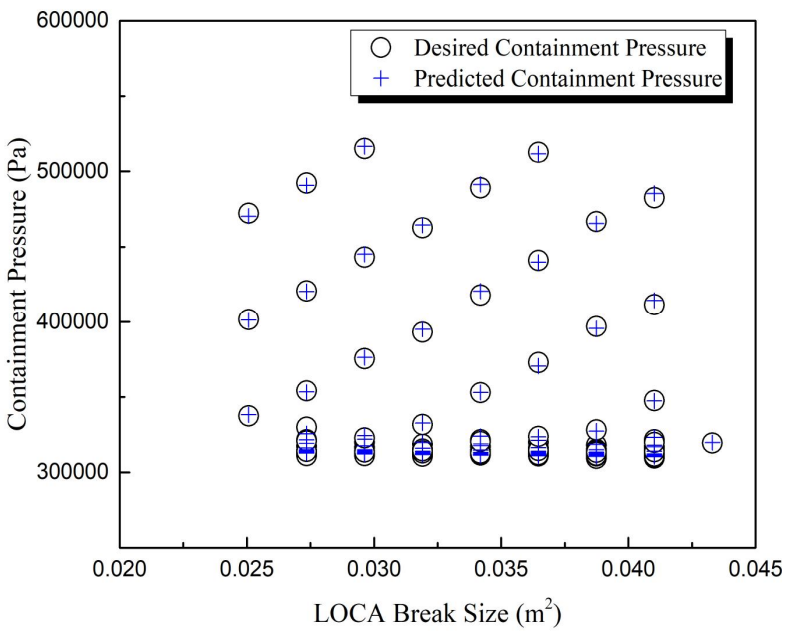


(b) Prediction of containment pressure according to estimated LOCA break size

Fig. 30. Performance of the DNN model for the test data under SBLOCAs at hot-leg (containment pressure)

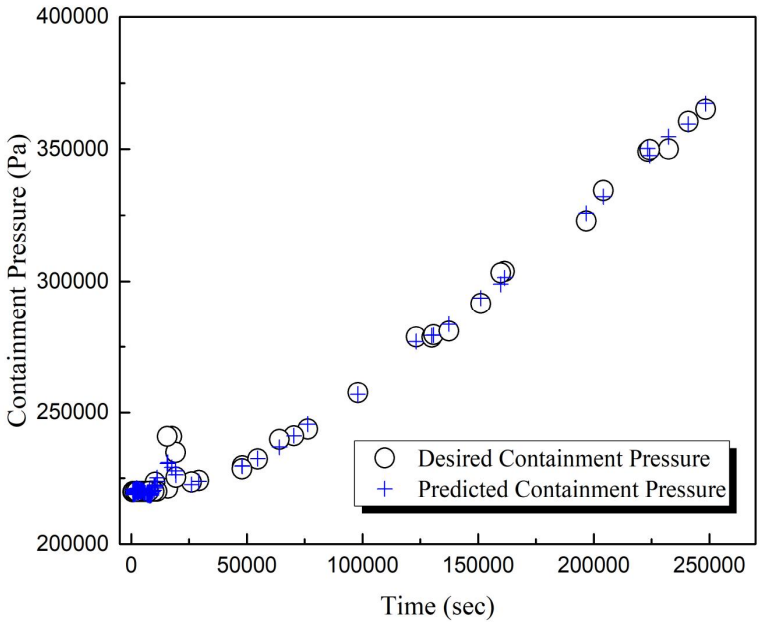


(a) Prediction of containment pressure according to elapsed time

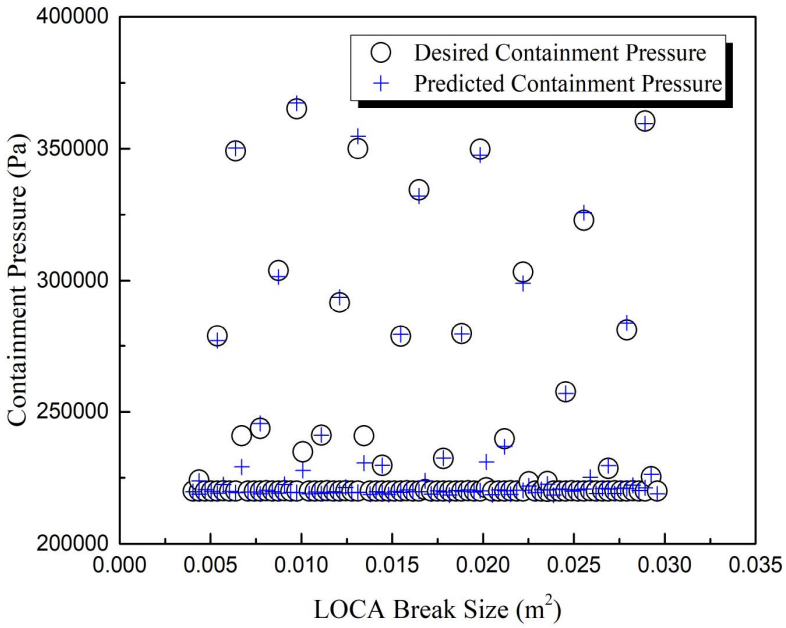


(b) Prediction of containment pressure according to estimated LOCA break size

Fig. 31. Performance of the DNN model for the test data under SBLOCAs at cold-leg (containment pressure)

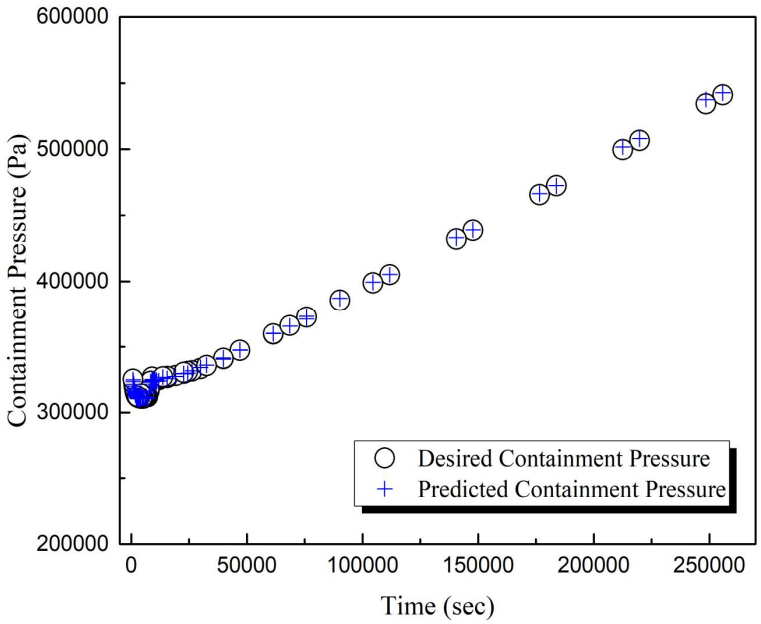


(a) Prediction of containment pressure according to elapsed time

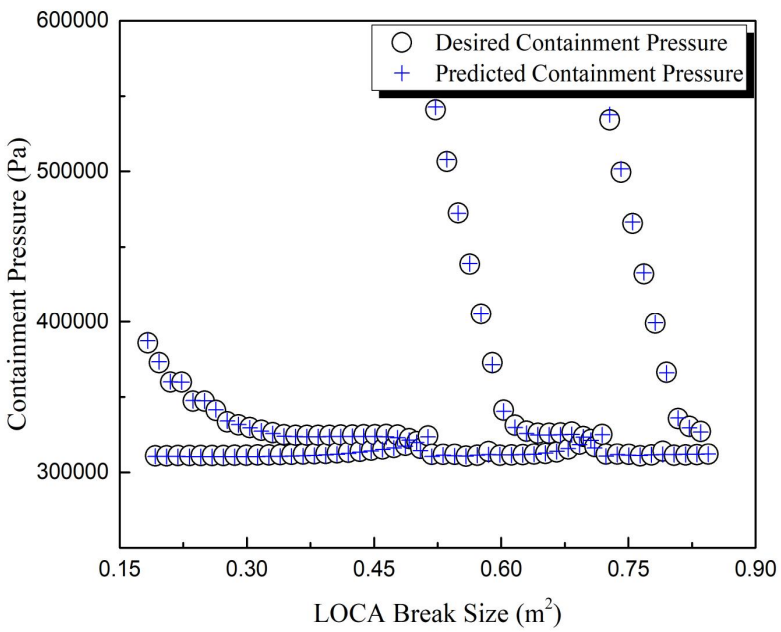


(b) Prediction of containment pressure according to estimated LOCA break size

Fig. 32. Performance of the DNN model for the test data under SBLOCAs at SGT (containment pressure)

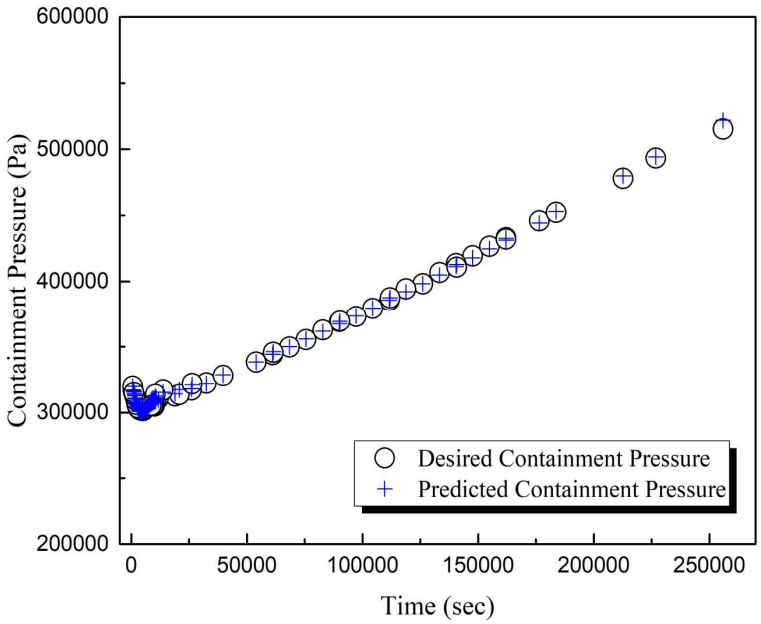


(a) Prediction of containment pressure according to elapsed time

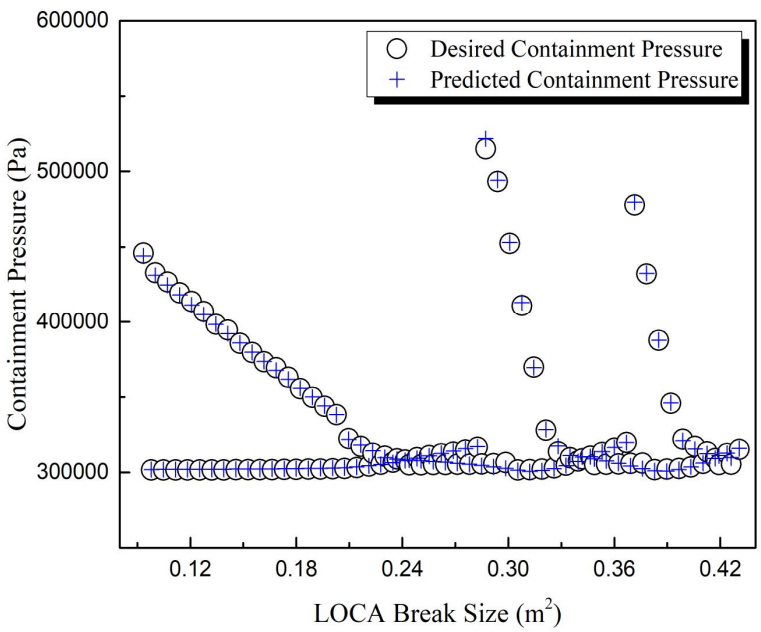


(b) Prediction of containment pressure according to estimated LOCA break size

Fig. 33. Performance of the DNN model for the test data under LBLOCAs at hot-leg (containment pressure)

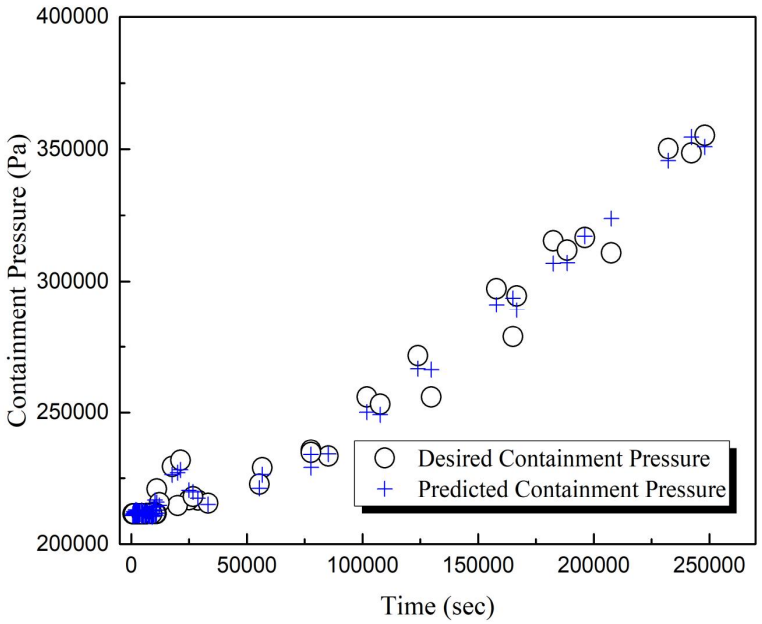


(a) Prediction of containment pressure according to elapsed time

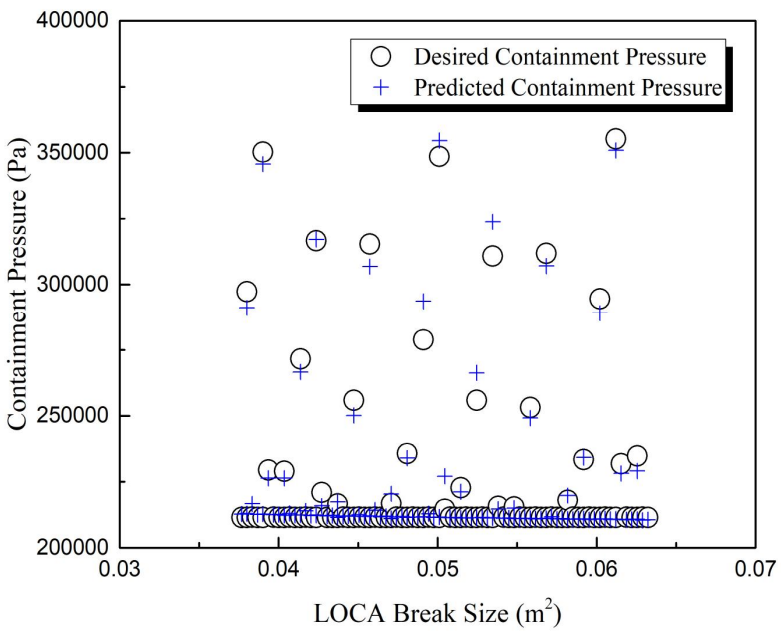


(b) Prediction of containment pressure according to estimated LOCA break size

Fig. 34. Performance of the DNN model for the test data under LBLOCAs at cold-leg (containment pressure)



(a) Prediction of containment pressure according to elapsed time



(b) Prediction of containment pressure according to estimated LOCA break size

Fig. 35. Performance of the DNN model for the test data under LBLOCAs at SGT (containment pressure)

D. Comparison of Performance between the AI Methods

According to Andrew Ng [39], the deep learning methods can become more powerful than the traditional machine learning algorithms by applying the larger amount of data. Therefore, the traditional machine learning methods are generally superior to the deep learning methods in the smaller amount of data.

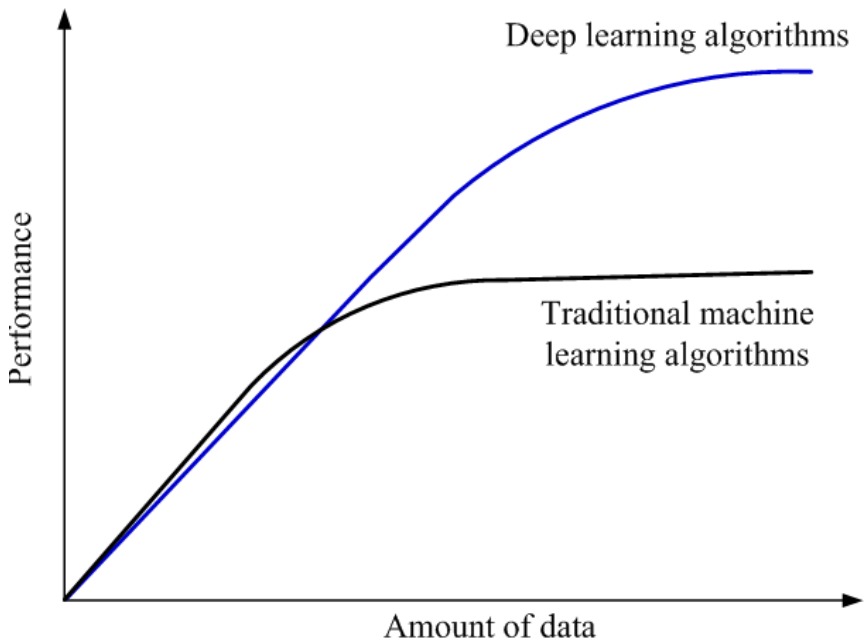


Fig. 36. Comparison of performance between deep learning and machine learning [39]

Nevertheless, the performance of the proposed DNN model, as a deep learning method used for predicting the major factors of the NPPs, is better than the CFNN model. It is supposed that one of the main reasons why these phenomena occur is due to the GA technique for hyper-parameter optimization. That is, the optimized performance of the DNN model is obtained by automatically finding the optimal number of hidden layers and nodes, using the GA in this thesis. Although the amount of applied accident simulation data can not be precisely pointed at Fig. 36, a performance gap between the DNN model and the

CFNN model was able to be reduced using the hyper-parameter optimization technique even in the small amount or the limited data in this thesis. Therefore, it is considered that the proposed DNN model becomes more competitive in NPP factor prediction.

V. Summary and Conclusions

In an effort to provide the supporting information to the operators, some of the major factors of the nuclear power plants (NPPs) were predicted applying other NPP information to the deep learning in the severe accidents. Specifically, the NPP factors such as the reactor vessel (RV) water level, the containment pressure, and the hydrogen concentration in the containment were predicted employing a few signals data under the severe accident circumstances originated in the postulated loss of coolant accidents (LOCAs) which may occur in NPPs.

Since the proposed deep neural network (DNN) model is generally with the supervised learning algorithms such as back-propagation and gradient descent, the data for training and verification are needed. Therefore, the data for predicting the major factors were obtained by simulating some of the various postulated LOCAs using the modular accident analysis program (MAAP) code. These LOCA simulation data consist of the behaviors of the various simulated signals for NPP parameters, which are numerically expressed. In this thesis, the NPP critical factors were predicted employing two or three simulated signals such as the elapsed time after the reactor trip, the estimated LOCA break size, and the containment pressure. Although the LOCA break size is not easily recognizable in the actual accidents, since the high estimation accuracy for the LOCA break size of the AI methods was shown in the previous studies, the LOCA break size was used as an accurately estimated signal data applicable to the major factor prediction.

To check the applicability of the DNN model used to predict the NPP factors using a few simulated signals, it was compared with the performance of the cascaded fuzzy neural network (CFNN) model of the previous studies. The DNN model is superior to the CFNN model in most cases, whereas its root mean square error (RMSE) and maximum error are relatively higher than those of the CFNN model in a few cases. Even though it is generally known that the performance of the traditional machine learning method is better than the deep learning in the smaller amount of the data set, that of the proposed DNN model was able to be more accurate than the CFNN model in this thesis. This is because the hyper-parameters of the DNN were optimized using the genetic algorithm (GA). It was

possible to obtain the better performance of the proposed DNN model in most cases even in the small data set by using the GA automatically optimizing the magnitude of hidden layers and nodes, which can be variable depending on the target factors and the LOCA cases.

Consequently, it is known that the DNN model with the GA is a more proper method to predict the major factors of the NPPs under the severe accident circumstances than the CFNN method. Furthermore, if a study to check the prediction performance of the DNN model by using other signals or considering the instrumentation error is carried out based on this thesis, the applicability of the DNN model to the NPP fields will be more improved in several aspects, and eventually it is expected that the proposed DNN model can be used for monitoring, diagnosis, and prediction of the NPP states or NPP equipment in the future.

REFERENCES

[1] KEPCO E&C, https://www.kepco-enc.com/eng/contents.do?key=1532

[2] Y. LeCun, Y. Bengio, and G. Hinton, Deep Learning, Nature, Vol. 521, pp. 436-444, 2015.

[3] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, Massachusetts, 2016.

[4] R. Henry et al., MAAP4: Modular Accident Analysis Program for the LWR Power Plants, User's Manual, Fauske and Associates Inc., 1994-2005.

[5] D. Y. Kim, K. H. Yoo, G. P. Choi, J. H. Back, and M. G. Na, Reactor Vessel Water Level Estimation During Severe Accidents Using Cascaded Fuzzy Neural Networks, Nuclear Engineering and Technology, Vol. 48, No. 3, pp. 702-710, 2016.

[6] G. P. Choi, D. Y. Kim, K. H. Yoo, and M. G. Na, Prediction of Hydrogen Concentration in Nuclear Power Plant Containment under Severe Accidents Using Cascaded Fuzzy Neural Networks, Nuclear Engineering and Design, Vol. 300, pp. 393-402, 2016.

[7] Y. D. Koo, G. P. Choi, and M. G. Na, Prediction of the Containment Pressure under Severe Accidents Using CFNN, Transactions of the Korean Nuclear Society Autumn Meeting, Gyeongju, Korea, Oct. 27-28, 2016.

[8] Michael Copeland, What's the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning?, https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

[9] C. Cortes and V. Vapnik, Support-Vector Networks, Machine Learning, Vol. 20, No. 3, pp. 273-297, 1995.

[10] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[11] M. G. Na, W. S. Park, and D. H. Lim, Detection and Diagnostics of Loss of Coolant Accidents Using Support Vector Machines, IEEE Transactions on Nuclear Science, Vol. 55, No. 1, pp. 628-636, 2008.

[12] S. H. Lee, Y. G. No, M. G, Na, K. I. Ahn, and S. Y. Park, Diagnostics of Loss of Coolant Accidents Using SVC and GMDH Models, IEEE Transactions on Nuclear Science, Vol. 58, No. 1, pp. 267-276, 2011.

[13] K. H. Yoo, Y. D. Koo, and M. G. Na, Identification of LOCA and Estimation of Its Break Size by Multiconnected Support Vector Machines, IEEE Transactions on Nuclear Science, Vol. 64, No. 10, pp. 2610-2617, 2017.

[14] J. Coble, P. Ramulhalli, L. Bond, J. Hines, and B. Upadhyaya, Prognostics and Health Management in Nuclear Power Plants: A Review of Technologies and Applications, Pacific Northwest National Laboratory (PNNL), Richland, Washington, 2012.

[15] J. W. Hines, D. J. Wrest, and R. E. Uhrig, Signal Validation Using an Adaptive Neural Fuzzy Inference System, Nuclear Technology, Vol. 119, No. 2 pp. 181-193, 1997.

[16] H. Y. Yang, S. H. Lee, and M. G. Na, Monitoring and Uncertainty Analysis of Feedwater Flow Rate Using Data-Based Modeling Methods, IEEE Transactions on Nuclear Science, Vol. 56, No. 4, pp. 2426-2433, 2009.

[17] S. L. Chiu, Fuzzy Model Identification Based on Cluster Estimation, Journal of Intelligent and Fuzzy Systems, Vol. 2, No. 3, pp. 267-278, 1994.

[18] M. G. Na, On-line Estimation of DNB Protection Limit via a Fuzzy Neural Network, Nuclear Engineering and Technology, Vol. 30, No. 3, pp. 222-234, 1998.

[19] M. G. Na, DNB Limit Estimation Using an Adaptive Fuzzy Inference System, Conference Record on Nuclear Science Symposium, Seattle, Washington, Oct. 24-30, Vol. 3, 1999.

[20] J. W. Hines, J. Coble, and B. Upadhyaya, Application of Monitoring and Prognostics to Small Modular Reactors, Publishings on Future of Instrumentation International Workshop (FIIW), Oak Ridge, Tennessee, Nov. 7-8, 2011.

[21] M. G. Na, S. H. Shin, S. M. Lee, D. W. Jung, K. B. Lee, and Y. J. Lee, Estimation of Axial DNBR Distribution at the Hot Pin Position of a Reactor Core Using Fuzzy Neural Networks, Journal of Nuclear Science and Technology, Vol. 41, No. 8, pp. 817-826, 2004.

[22] P. Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph. D. thesis, Harvard University, Cambridge, Massachusetts, 1974.

[23] D. Rumelhart, G. Hinton, and R. Williams, Learning Representations by Back-propagating Errors, Nature, Vol. 323, pp. 533-536, 1986.

[24] P. Kim, MATLAB Deep Learning: with Machine Learning Neural Networks and Artificial Intelligence, Apress Inc., New York, 2017.

[25] A. Ng, K. Katanforoosh, and, Y. B. Mourri, Neural Networks and Deep Learning, Coursera.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Stuskever, and R. Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, Vol. 15, No. 1, pp. 1929-1958, 2014.

[27] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kegl, Algorithms for Hyper-parameter Optimization, Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS), Granada, Spain, Dec. 12-15, pp. 2546-2554, 2011.

[28] J. Bergstra and Y. Bengio, Random Search for Hyper-Parameter Optimization, Journal of Machine Learning Research, Vol. 13, pp. 281-305, 2012.

[29] A. Ng, K. Katanforoosh, and Y. B. Mourri, Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization, Coursera.

[30] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Massachusetts, 1989.

[31] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Massachusetts, 1996.

[32] M. G. Na, Y. J. Lee, and I. J. Hwang, A Smart Software Sensor for Feedwater Flow Measurement Monitoring, IEEE Transactions on Nuclear Science, Vol. 52, No. 6, pp. 3026-3034, 2005.

[33] E. H. Mamdani and S. Assilian, An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.

[34] T. Takagi and M. Sugeno, Fuzzy Identification of Systems and Its Applications to Modelling and Control, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-15, No. 1, pp. 116-132, 1985.

[35] J. C. Duan and F. L. Chung, Cascaded Fuzzy Neural Network Model Based on Syllogistic Fuzzy Reasoning, IEEE Transactions on Fuzzy Systems, Vol. 9, No. 2, pp. 293-306, 2001.

[36] M. G. Na, S. H. Shin, D. W. Jung, S. P. Kim, J. H. Jeong, and B. C. Lee, Estimation of Break Location and Size for Loss of Coolant Accidents Using Neural Networks, Nuclear Engineering and Design, Vol. 232, No. 3, pp. 289-300, 2004.

[37] T. Haste, Severe Accident Phenomena part 1: In-vessel, IAEA Workshop on Severe Accident Management Guidelines, Vienna, Austria, Oct. 19-23, 2015.

[38] T. Haste, Severe Accident Phenomena part 2: Ex-vessel, IAEA Workshop on Severe Accident Management Guidelines, Vienna, Austria Oct. 19-23, 2015.

[39] A. Ng, What Data Scientists Should Know about Deep Learning, Baidu Research.