

## 냉각재 소량 누설 정량화를 위한 딥러닝의 적용

이상현\* · 조혜선\* · 나만균\*†

# Application of Deep Learning for Quantifying Small RCS Leakage

Sang Hyun Lee\*, Hye Seon Jo\*, Man Gyun Na\*†

*Key Words* : Deep learning(딥러닝), Relative humidity(상대습도), Small leakage(소량 누설), Leakage detection(누설 탐지), Nuclear power plant(원자력 발전소)

### ABSTRACT

In nuclear power plants, coolant leakage occurs for various reasons. Leak detection is important to ensure safety of nuclear power plants. Currently, a detection system for an unidentified reactor coolant system(RCS) leakage of less than 0.5gpm is being developed in Korea. The RCS leakage is detected through changes in radioactivity, humidity, and temperature in the containment air, and water level of sump. For small leaks, the change in humidity and temperature due to water vapor is very small, making the leak very difficult to detect until the leak accumulates in the instrument.

In order to solve these problems and increase the leak detection speed, it is necessary to develop a system capable of real-time detection using artificial intelligence. In this study, long short-term memory and bidirectional long short-term memory, which are types of recurrent neural networks among artificial intelligence methods, were applied to perform initial relative humidity prediction for leakage quantification. Also, an optimization technique that reduces learning time and improves prediction performance for the optimization of learning was applied. Finally, the prediction performance was evaluated using the developed model.

### 1. 서론

원자력 발전소 내의 기기는 많은 배관과 밸브로 구성되어 있다. 구성되어있는 밸브나 배관의 열화 또는 용접부에서의 열화 및 기기 진동이나 피로 파손에 의해 누설사고가 발생할 수 있다. 누설사고가 발생하게 되면 운전원은 적절한 절차를 기반으로 조치를 취함으로써 발전소를 안전한 상태로 회복시켜야 하지만, 상황을 인지하지 못했거나 잘못된 조치를 취한 경우 발전소는 불시정지상태로 이어질 수 있다. 누설사고는 방사선 누출과 직접적으로 연관되기 때문에 이를 방지하는 것이 안전성 측면에서 중요한 역할을 한다. 원전안전운영정보시스템(OPIS)에 의하면 2017년부터 2022년까지 총 12건의 기계결함에 의한 사고가 발생했으며, 그중 2017년도에 고리 4호기의 밸브 용접 부위에서 진동 피로로 인한 균열

발생으로 인한 냉각재 누설이 확인되어 원자로 불시정지 조치가 취해졌다<sup>(1,2)</sup>. 이러한 사고는 인명피해뿐만 아니라 경제적인 피해로도 이어질 수 있으므로 조기에 누설을 감지하는 것이 중요하다. 현재 누설 탐지방법은 공기 중 입자 방사능이나, 기체 방사능, 원자로 격납건물 대기 습도나 압력 및 온도, 냉각기로부터의 응축수 유량으로 탐지를 진행하거나 1차 계통에서 특정 기기나 위치에서 발생한 누설을 감시하는 음향누설감시계통, 불활성 방사성 기체를 이용하여 누설 탐지 시 핵종을 이용한 베타 방사선을 감시하여 누설을 탐지하는 방법으로 누설을 탐지하고 있다<sup>(3)</sup>. 그리고, 경수로형 원전을 대상으로 한 누설 탐지계통 성능 기준은 규제지침(KINS/RG-N06.01)에 명시된 바에 의하면, 모든 탐지계통에서 1시간 이내에 1 gpm의 누설증가를 탐지할 수 있어야 한다고 명시되어있다.

\* 조선대학교 원자력공학과(Department of Nuclear Engineering, Chosun University)

† 교신저자, E-mail : magyna@chosun.ac.kr

하지만 0.5 gpm 미만의 소량 누설일 경우에는 수증기에 의한 상대습도와 온도의 변화율이 매우 작기 때문에 계측기의 측정부에 누적되어 계측 시 변화량을 감지하기 어려워 미확인 누설을 탐지하기는 매우 어렵다. 이러한 단점을 극복하고자 한국원자력연구원에서는 원자로 냉각재 계통의 미확인 누설에서 실시간급 감시를 위한 누설감시기술을 개발하고 있다<sup>(4)</sup>. 구체적으로 소량 누설 시 누설 탐지를 위해 온도, 습도, 방사선 등 다중 복합형 계측기를 개발하고, 이와 더불어 누설량 정량화를 위한 알고리즘 개발을 수행하고 있다. 이러한 기술개발을 통해 소형 및 대형 누설 감지속도를 높임으로써 운전원 조치를 위한 시간 확보와 더불어 발전소 안전성을 향상시킬 수 있다.

본 논문에서는 소량 누설 시 센서를 기반으로 하는 누설 탐지와 더불어 측정부에서 취득할 수 있는 상대습도 정보를 기반으로 누설량 정량화를 위한 연구의 일환으로 인공지능 기술을 활용하여 초기(누설부) 상대습도 예측을 수행하였다. 즉, 원자로 냉각 시스템의 특정 주요 부위에 감지 센서를 배치하여 공기를 포집하고 이송 루프를 통해 포집부에서 떨어진 측정부에서 측정된 상대습도 변화를 기반으로 포집부 상대습도 예측을 통하여 누설량을 정량화하고자 한다. 인공지능 기술 중 Long short-term memory (LSTM)<sup>(5)</sup>와 Bidirectional LSTM (Bi-LSTM)<sup>(6)</sup>을 적용하였으며, 이는 시계열 데이터 예측에 특화되어있는 딥러닝 모델로 다양한 분야의 예측 연구에 활용되고 있다<sup>(7)</sup>. 그리고, 딥러닝 모델 학습에 있어서 하이퍼파라미터에 따라 예측성능이 달라지는 경향을 보이는데, 모델의 우수한 성능을 위해서는 하이퍼파라미터의 최적화가 매우 중요한 역할을 한다. 하지만, 하이퍼파라미터 최적화과정은 사용자의 손으로 일일이 확인하면서 하기에는 매우 많은 시간적 비용이 소모된다. 이를 극복하기 위해 모델을 위해 필요한 하이퍼파라미터를 선정한 후, 하이퍼파라미터 최적화 알고리즘을 연구에 활용하였다.

모델 개발을 위해 적용한 데이터는 한국원자력연구원의 고정밀 열수력 해석 코드인 Component Unstructured Program for Interfacial Dynamics (CUPID) 코드<sup>(8)</sup>를 활용하였다. CUPID 코드는 2상 유동을 해석하기 위해 과도, 2 유체, 3유동장 모델 (Transient, 2-fluid, 3-field model)을 채택하고 있다. 3개의 장은 연속 액체, 액적 및 증기를 나타낸다. CUPID 코드를 통해 센서 튜브를 모사한 후, 데이터를 취득하여 딥러닝 학습에 활용하였다. 초기상대습도 예측 성능평가를 통해 딥러닝 및 하이퍼파라미터 최적화 알고리즘 적용 가능성을 확인하고, 향후 소량 누설 시에 인공지능을 이용하여 신속한 누설 탐지 및 누설량 정량화를 위해 도움이 될 것으로 기대된다.

## 2. 인공지능 및 최적화 방법론

### 2.1 LSTM

LSTM은 순환신경망 중 하나이다. 순환신경망은 시계열 데이터를 예측하는데 이점을 가지고 있는 신경망으로, 기본적인 순환신경망은 입력단을 거쳐 나온 결과와 모델의 상태를 계산한 이후, 다음 시점에서 해당 결과를 입력단에 입력하여 결과를 얻어내는 방법이다. 기존의 순환신경망은 역전파를 할 때, 활성화 함수인 tanh 함수와 행렬 연산을 통해 학습이 진행되기 때문에 역전파를 반복해서 계산할수록 기울기가 1보다 작은 값이 계속적으로 곱해지는 구조이기 때문에 학습이 이루어지지 못하는 현상이 발생한다. 또, 이러한 순환신경망의 단점을 보완하기 위해 LSTM 신경망은 hidden state에서 이전 시점에서 전달된 상태를 조절하는 특성 게이트를 추가해 계산된다. 위의 특성 게이트는 셀 상태, 망각 게이트, 입력 게이트, 출력 게이트의 총 네 개의 게이트로 구성되어있다<sup>(5)</sup>. LSTM의 구조는 Fig. 1과 같다.

첫 번째 단계에서는 컨베이어벨트를 통하여 정보(셀 상태,  $C$ )가 전달되는 과정에서 이전 정보를 얼마나 간직하느냐를 결정하게 되며, 망각 게이트라 불린다. 망각 게이트는 식 (1)과 같이 계산된다.

$$F_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

식 (1), (2)와 (3)에서  $\sigma$ 는 시그모이드 활성화함수이고,  $x$ 와  $h$ 는 입력과 hidden state를 나타내며,  $W$ 는 가중치 매개변수이고, 그리고  $b$ 는 편향 매개변수이다.

두 번째 단계에서는 셀 상태에 새로운 정보를 저장하는 역할이다. 입력 게이트 안에 시그모이드 함수를 통해 업데이트 할 값을 정하고, tanh 함수를 이용해 셀 상태에 추가될 수 있는 새로운 후보 값을 생성한다. 입력 게이트는 식 (2)와 같이 표현된다.

$$I_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

마지막으로 다음 셀 상태에 새로운 정보를 저장하는 단계이다. 이전의 망각 게이트와 입력 게이트를 통해 잊고, 기억할 정보를 결정했으므로 이전의 셀 상태를 업데이트한다. 출력 게이트는 식 (3)과 같이 표현된다.

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3)$$

이와 같은 특성 게이트의 구조를 이용하여 LSTM에서 오차 역전파는 셀 상태를 통해 이루어지기 때문에, 인공지능의

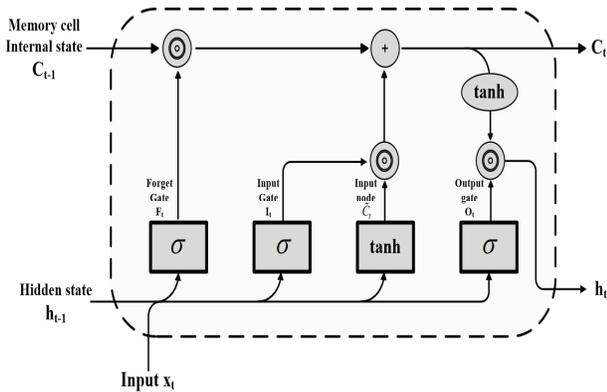


Fig. 1 Structure of LSTM

학습을 어렵게 하는 기울기 소실에 큰 영향을 받지 않는다. 이를 통해 오래된 정보를 저장하는 것에 대한 장기적 의존성이 없다는 장점이 있다.

### 2.2 Bi-LSTM

LSTM은 입력이 순차적으로 입력되기 때문에 출력값이 직전의 입력값에 가깝게 편향되는 단점을 가지고 있다. 이는 정보가 많아질수록 출력에 가까운 정보에 따라 출력값에 영향을 미칠 수 있다. 이러한 단점을 해결하기 위하여 Bi-LSTM이 고안되었다<sup>(6)</sup>. Bi-LSTM은 LSTM과는 달리 손실을 최소화하는 학습 과정에서 정방향으로 학습하는 Hidden state뿐만 아니라 역방향으로도 학습을 진행하는 Hidden state를 추가하여 모든 시점에서 미래와 과거의 정보를 보존하여 학습을 진행한다. Bi-LSTM의 구조는 Fig. 2와 같다. Fig. 2에서 각 시점에서 학습이 진행되어 출력된 결과값은  $h$ 와  $h'$ 로 출력이 되는데, 이때 출력값은 벡터 계산을 통해 결합된 값으로 출력된다. 이 출력값 계산은 결합 이외에도 평균을 내거나 단순히 더하는 방법 등으로 다양하게 적용할 수 있다.

### 2.3 Grid search

하이퍼파라미터는 딥러닝 모델 내부에서 결정되는 파라미터와는 달리 딥러닝의 학습에 있어 제어하는데 사용되는 파라미터를 의미한다. 이는 딥러닝에서 모델을 만들면서 사용자가 직접 설정해주는 값을 의미한다. 딥러닝 모델을 적용하는 데 고려해야 할 하이퍼파라미터의 종류는 매우 많다. 이는 사용자가 고려해야 할 점이 많다는 것을 의미하며, 성능이나 학습 시간에 영향을 미칠 수 있다. 하이퍼파라미터 최적화는 데이터와 딥러닝 모델에 따른 최적의 하이퍼파라미터 조합을 찾는 것을 의미한다.

Grid search는 사용자가 하이퍼파라미터의 후보 값을 선정한 후, 각 후보 값의 조합에 따라 최적의 출력값을 찾아 주

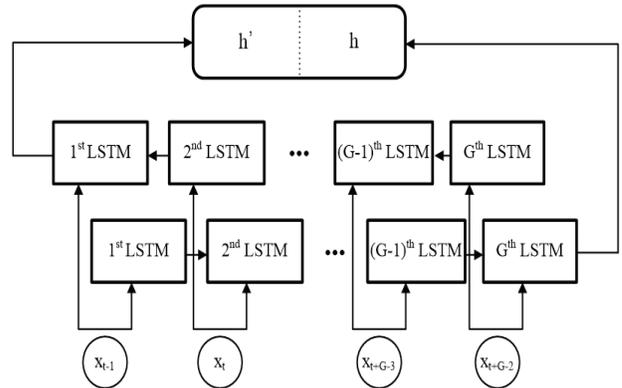


Fig. 2 Structure of Bi-LSTM

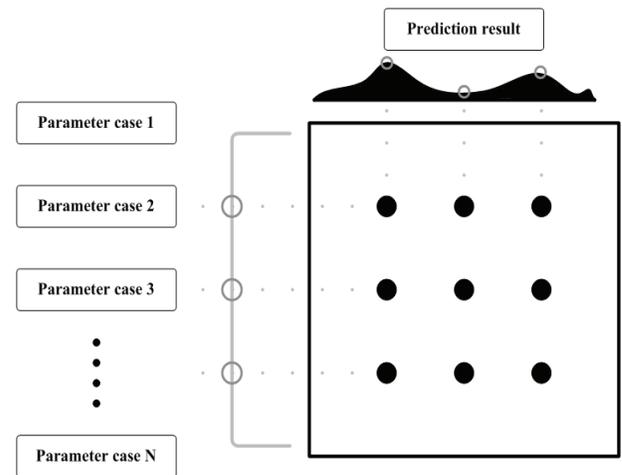


Fig. 3 Structure of grid search

는 탐색 방법이다<sup>(9)</sup>. 즉, 하이퍼파라미터 조합을 순차적으로 적용하여 학습하고 결과를 출력한다. Grid search의 대략적인 구조는 Fig. 3과 같다. Grid search는 모든 하이퍼파라미터 조합을 비교할 수 있다는 장점이 있지만, 모든 조합을 순차적으로 비교하기 때문에 결과값을 출력하기에 오랜 시간이 걸리는 단점이 존재한다.

## 3. 데이터 수집 및 처리

### 3.1 데이터 수집

열수력 코드는 원자력 발전소의 다양한 사고를 분석할 수 있는 능력을 갖추고 있다. 하지만, 물리적 모델이나 수치적 방법에서 다차원 유동 현상을 1차원으로 모사한다는 한계점을 보인다<sup>(8)</sup>. 이러한 단점을 극복한 고정밀 열수력 해석 코드인 CUPID 코드<sup>(8)</sup>를 활용하여 센서 튜브를 모사하였고, 이송 거리에 따른 상대습도 변화 데이터를 취득하였다. CUPID 코드는 원자력 발전소 기기 중 증기발생기, 원자로 압력용기, 격납용기 등에서 발생하는 다차원 2상 유동 현상을 해석하기

위해 개발된 코드이다. 이는 단상뿐만 아니라 2상, 상변화, 공기-물 유동 등을 수치적으로 정확하고 안정되게 풀 수 있다는 장점이 있다.

본 논문에서는 소량 누설감지를 위한 감시시스템 개발 연구를 위해 원자로 냉각 시스템의 특정 주요 부위에 감지 센서를 배치하여 공기를 포집하고 이송 루프를 통해 상대습도 변화를 기반으로 초기상대습도 예측하는 연구에 활용했다. 이를 위해 모사된 튜브는 누설부의 공기를 포집부에서 포집하여 측정부까지 이송시키는 센서 튜브이다. 센서 튜브의 총 길이는 100 m이며, 직경은 6 mm인 배관이며 100 m 중 초기 1 m는 공기를 포집하는 포집부로 설정하고, 유속은 현재 개발 중인 누설감지 시스템의 유속인 11 m/s로 설정하였다. Fig. 4는 모사된 센서 튜브를 보여준다.

CUPID 코드 시뮬레이션을 통한 데이터를 취득하기 위해 CUPID 코드 입력으로 온도, 압력, 건도, 이송 속도를 입력 변수로 선정하였고, 이를 딥러닝 학습에 필요한 변수로 변환하려면 일련의 과정이 필요하다. 일반적으로 상대습도는 수증기압과 수증기량을 바탕으로 계산되며, 본 연구에서는 수증기압을 바탕으로 식 (4)와 같이 상대습도 값을 계산하였다.

$$Relative\ Humidity\ (\%) = \frac{P_w}{P_{ws}} \times 100\% \quad (4)$$

$$P_{ws} = P_c \cdot e^{\frac{T_c}{T}(C_1\theta + C_2\theta^{1.5} + C_3\theta^3 + C_4\theta^{3.5} + C_5\theta^4 + C_6\theta^{7.5})} \quad (5)$$

$$\theta = 1 - \frac{T}{T_c} \quad (6)$$

식 (4)에서  $P_w$ 는 수증기압,  $P_{ws}$ 는 포화수증기압이다. 식 (5)는 포화수증기압을 계산하는 식이다. 식 (5)에서  $P_c$ 는 임계압력으로 220,640 hPa이고,  $T$ 는 절대온도를,  $T_c$ 는 임계 온도인 647.096 K이다. 식 (5)에서 각  $C$  값은 계수들이다. ( $C_1 = -7.85951783$ ,  $C_2 = 1.84408259$ ,  $C_3 = -11.7866497$ ,  $C_4 = 22.6807411$ ,  $C_5 = -15.9618719$ ,  $C_6 = 1.80122502$ )<sup>(10)</sup>.

Fig. 5는 CUPID 코드를 활용하여 특정 지점에 따른 온도, 건도, 수증기압을 보여준다. 데이터의 초기조건은 저습도 영역의 온도와 상대습도를 50 °C, 10 %로 설정하였고, 고습도 영역의 온도와 상대습도를 각 2 °C와 2 % 단위로 변경하였고, 건도는 조건에 따른 최적값을 위해 임의의 건도값을 넣은 다음 도출된 수증기압으로 습도를 계산하고 원하는 습도에 도달 할 때까지 구해 데이터를 취득하였다. 취득된 데이터의 case 수는 총 336개이다.

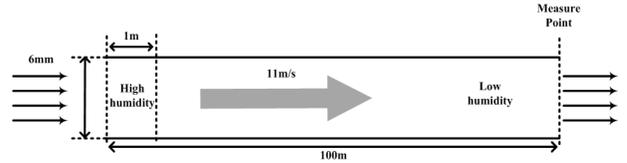


Fig. 4 Simulated sensor tube using CUPID code

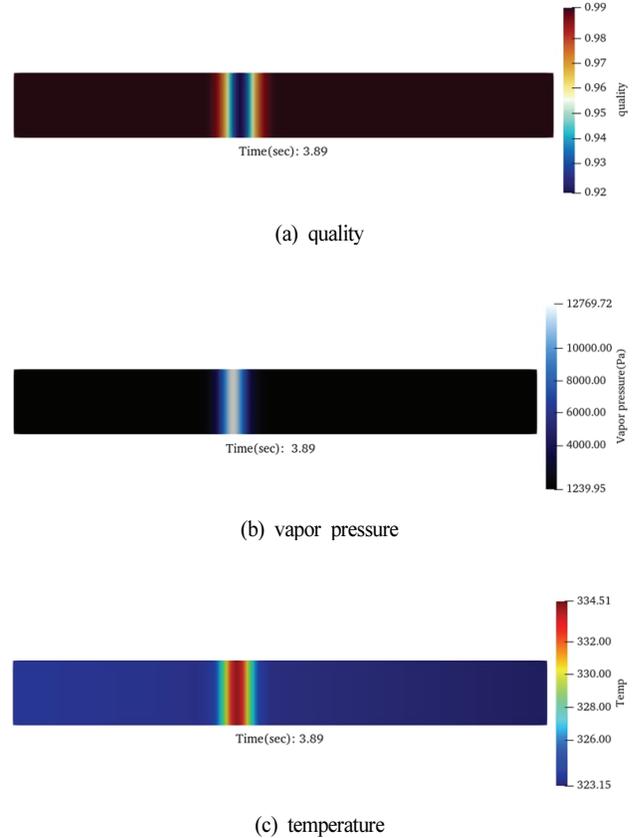


Fig. 5 Analysis result using CUPID code

### 3.2 데이터 전처리

CUPID 코드를 통해 취득한 데이터는 딥러닝 모델에 적용을 위해 전처리하는 과정을 거쳤다. 데이터 전처리는 크게 3가지로 구분되며 1) 변수선정, 2) 정규화, 3) 시계열 처리를 수행하였다.

먼저 초기 상대습도 예측을 수행하는 딥러닝 모델 개발에 필요한 입력변수를 선정하였다. CUPID 코드를 활용하여 추출된 데이터는 압력, 상대습도 등 여러 변수에 대한 정보를 포함하고 있다. 이 중에서 딥러닝 학습에 불필요한 변수이거나 실시간 탐지를 함에 있어 현실적으로 사용하기 어려운 변수를 제외하고, 실질적으로 사용 가능한 주요 변수로써 온도, 거리, 상대습도 3가지의 변수를 입력변수로 선정하였다. 또한, 여기서 상대습도는 측정부의 상대습도를 의미하며 측정부의 상대습도를 입력값으로 사용하여 누설부의 초기 상대습도를 예측함을 의미한다.

Table 1 Input data combination

Temperature (°C)	Input humidity (%)	Distance from measuring point (m)	Total data length
60, 62, 64, ⋮ 96, 98, 100	60, 62, 64, ⋮ 86, 88, 90	1.1, 2.2, 3.3, 4.4, ⋮ 96.6, 97.7, 98.8, 99.9	26,880

두 번째로는 정규화를 통해 변수별 데이터 값을 조정하였다. 정규화는 데이터의 높은 변화 값의 범위를 줄이고 특정 데이터가 딥러닝 모델 예측에 부정적인 영향을 줄 수 있는 확률을 줄인다. 그리고, 특정 범위에서 인공지능 학습 속도에 좋은 영향을 미친다. 본 연구에서는 데이터의 범위를 0과 1 사이의 값으로 조정하는 min-max 정규화를 수행했다.

마지막으로 데이터를 시계열 구조로 바꾸기 위해 sliding window 기법을 이용한 시계열 처리를 하였다<sup>(11)</sup>. CUPID 코드를 통해 취득한 이송 거리에 따른 상대습도 데이터를 sliding window 기법을 활용하여 0.1 초 단위 시계열 데이터로 변환하였다. 전처리를 모두 거친 데이터의 구성은 Table 1과 같다. 인공지능 모델의 입력변수는 거리(time step= 0.1초), 온도, 상대습도이며, 이용된 데이터는 이들 입력변수 3개의 조합으로 26,880개 데이터로 이루어져 있다.

## 4. 딥러닝 적용 및 결과

### 4.1 하이퍼파라미터 선정

인공지능 예측 모델의 하이퍼파라미터는 사용자가 어떤 값을 결정하느냐에 따라 예측성능에 큰 영향을 미치기 때문에 우수한 성능 조합을 찾기 위하여 하이퍼파라미터 최적화 알고리즘을 적용하였다. 본 연구에서는 하이퍼파라미터로 batch size, 딥러닝 모델의 layer 수, 그리고 최적화 함수를 선정하였다. 그리고 개발한 딥러닝 모델이 훈련 데이터에 대해서만 우수한 성능을 보이고 테스트 데이터에 대한 성능이 낮게 나오는 현상인 과적합을 방지하기 위해 조기 종료 기능을 추가하였다.

먼저 batch size는 딥러닝 모델 학습 중 파라미터를 최신화할 때 사용되는 데이터의 개수를 의미한다. 그 크기에 따라 학습 시간이나 성능이 달라지기 때문에 본 연구에서는 16, 32, 64, 128을 채택하였다.

두 번째로는 딥러닝 모델의 layer 계층이다. layer는 딥러닝 모델 안에서 역전파 방법을 통해 가중치를 최신화하며 계산하는 역할을 하는 계층이다. layer 계층이 깊어질수록 연결 가중치가 늘어나고 학습에 필요한 계산량이 늘어나 학습 시간이 증가한다. 본 연구에서는 layer 계층을 3, 4, 5개로 조절하였다.

Table 2 Hyperparameter values

Hyperparameter name	Values
Batch size	16
	32
	64
	128
Number of layer	3
	4
	5
Optimizer function	Adam
	RMSProp

마지막으로 최적화 함수는 가중치와 편향을 최적화하기 위해 사용되는 함수인데, 다양한 방법 중 root mean square propagation (RMSProp)과 adaptive moment estimation (Adam)을 사용하였다. 먼저, RMSProp은 기울기에 따라 학습률을 조절하여 최적의 결과값에서 멀 때는 학습률을 높게, 가까울 때는 학습률이 낮아지도록 하는 최적화 함수이다<sup>(12)</sup>. Adam 최적화 함수는 현재 널리 사용되는 최적화 함수로 RMSProp과 관성의 개념을 결합한 함수로 학습률 조절을 RMSProp처럼 하되 지역적인 최솟값에 편향되지 않게 관성을 적용하여 계산되는 방법론이다<sup>(13)</sup>. 하지만 Adam은 초기 가중치 최신화가 느리다는 단점이 존재한다. Table 2는 Grid search를 통해 선정된 모델의 하이퍼파라미터 값을 보여준다.

### 4.2 초기(누설부) 상대습도 예측 결과

학습과 테스트를 위한 데이터 비율은 9:1로 설정하였다. 초기 상대습도 예측 결과는 오차를 통해 계산되었으며, CUPID 코드를 통해 취득한 데이터와 딥러닝 모델을 통해 출력된 결과값을 평균 제공된 오차와 평균 절대오차를 활용하여 계산하였고, 각각은 식 (7)과 (8)로 표현된다:

$$Mean\ absolute\ error = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i| \quad (7)$$

$$Root\ mean\ square\ error = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}} \quad (8)$$

식에서  $Y_i$ 는 실제값을 나타내며,  $\hat{Y}_i$ 는 인공지능을 통해 예측된 값을 의미한다.  $n$ 은 데이터의 개수이다.

두 성능평가 지표 모두 실제값과 예측값을 통해 계산되기 때문에, 오차의 값이 낮을수록 성능이 높다는 것을 의미한다.

하이퍼파라미터 최적화를 위해 Grid search를 적용한 결과를 통해 몇 가지의 특징을 발견할 수 있었다. 모든 학습은 조기 종료 기능을 설정하여 학습에 과적합이 일어나지 않도록 하였으며, patience를 설정하여 10번 동안 손실함수의 진전이 없다면 종료하는 알고리즘을 적용하였다.

초기 상대습도 예측 결과 일반적으로 batch size가 작아질수록 학습 시간이 오래 걸린다는 단점이 있지만, 예측성능에는 긍정적인 영향을 미치는 것으로 확인되었다.

그리고, layer의 깊이는 딥러닝 데이터와 모델 특성에 따라 성능이 달라지겠지만 본 연구에서는 layer가 깊어질수록 연산량이 많아져 예측 결과가 개선되는 결과를 보이지만, 학습 시간이 오래 걸리는 단점을 보였다.

마지막으로, 손실함수의 값을 낮춰주는 역할을 하는 최적화 함수는 모든 경우에서 Adam이 RMSProp보다 우수한 결과를 나타내었다.

Table 3, 4는 각 딥러닝 모델에 Grid search를 적용하여 하이퍼파라미터 조합의 성능 척도를 보인다. 총 48개의 하이퍼파라미터 조합에서 낮은 성능을 보이는 조합, 중간 정도의 성능을 보이는 조합, 높은 성능을 보이는 조합을 나타내었다. 이는 각 조합에 따라 성능이나 학습시간에서 차이를 보인다. 두 딥러닝 모델에서 낮은 성능을 보이는 하이퍼파라미터 조합은 batch size는 128, layer의 깊이는 5, 최적화 함수는 RMSProp이었고, 높은 성능을 보이는 조합은 size는 16, 깊이는 5, 최적화 함수는 Adam이었다. Table 5, 6에는 각 하이퍼파라미터 조합의 예측성능을 성능 지표를 활용하여 나타낸 표이다. LSTM과 Bi-LSTM을 비교하여 볼 때, Bi-LSTM이 오차 측면에서 우수한 결과를 보였다.

Fig. 6은 각 딥러닝 모델의 초기 상대습도 예측 결과 성능이 높은 결과를 나타낸 그래프이며, 그래프에서 빨간 점은 실제 데이터의 초기 상대습도를, 파란 점은 딥러닝 모델을 이용한 초기 상대습도 예측 결과를 나타낸 것이다. 두 그림을 통해 보았을 때, LSTM 모델보다 Bi-LSTM 모델이 더욱 정밀한 예측 정확도를 보인다.

Table 3 Hyperparameters combination by performance class of LSTM

	Batch size	Layer depth	Optimizer function	Learning time
Low performance	128	5	RMSProp	5 min 55 sec
Middle performance	32	4	Adam	4 min 47 sec
High performance	16	5	Adam	9 min 42sec

Table 4 Hyperparameters combination by performance class of Bi-LSTM

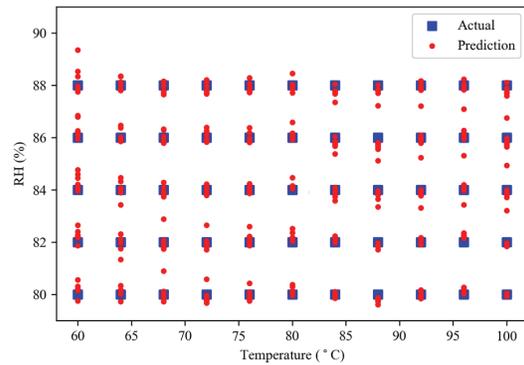
	Batch size	Layer depth	Optimizer function	Learning time
Low performance	128	5	RMSProp	2 min 20 sec
Middle performance	128	3	Adam	1 min 16 sec
High performance	16	5	Adam	1 min 52 sec

Table 5 Prediction result through performance metrics of LSTM

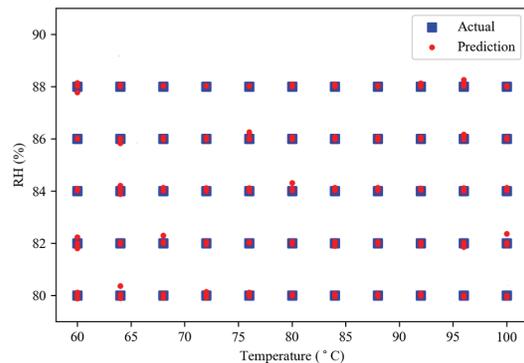
	RMS Train	RMS Test	MAE Train	MAE Test	Train Max	Test Max
128_5_RMSProp	0.8679	0.6431	0.6757	0.7242	5.2168	4.7490
32_4_Adam	0.3615	0.4220	0.2828	0.3177	2.3838	2.6599
16_5_Adam	0.2235	0.2531	0.1651	0.1735	2.4861	2.4589

Table 6 Prediction result through performance metrics of Bi-LSTM

	RMS Train	RMS Test	MAE Train	MAE Test	Train Max	Test Max
128_5_RMSProp	0.8743	0.9219	0.6971	0.7496	1.3032	1.3878
128_3_Adam	0.3330	0.4476	0.2611	0.3094	0.8805	0.8251
16_5_Adam	0.0798	0.2275	0.0429	0.0791	0.4195	0.4206



(a) LSTM



(b) Bi-LSTM

Fig. 6 Prediction result of LSTM and Bi-LSTM

## 5. 결 론

본 연구에서 RCS 소량 누설 시 누설량을 정량화하기 위하여 딥러닝을 이용해 초기 상대습도를 예측하는 연구를 수행하였다. 딥러닝의 학습에 활용된 모델은 LSTM과 Bi-LSTM이며, 두 딥러닝 모델은 시계열 예측에 특화된 딥러닝 모델이다. 이를 위해 데이터를 시계열 형태로 재처리하는 방법을 활용하였다. 데이터는 고정밀 열수력 해석 코드인 CUPID 코드를 활용하여 센서 튜브를 모사해 데이터를 취득하였고, 총 26,880개의 데이터를 취득할 수 있었다. 취득한 데이터는 학습 데이터와 테스트 데이터로 각각 나누어 진행하였다. 딥러닝 모델 학습을 할 때 사용자가 조정해야 하는 하이퍼파라미터를 알고리즘을 통해 조정함으로써 학습성능이 우수한 하이퍼파라미터 조합을 탐색하였다.

초기 상대습도 예측 결과 LSTM보다 Bi-LSTM이 파라미터 간의 양방향 종속성을 학습하여 우수한 결과를 보였다. 모든 학습은 과적합 방지를 위해 조기 종료를 적용하였고, 그 결과 LSTM의 오차와 학습시간은 0.2531과 9분 45초이고, Bi-LSTM의 오차와 학습시간은 0.0791, 1분 52초이다. 이를 통해 양방향으로 학습을 하는 Bi-LSTM의 성능이 우수함을 확인할 수 있었다. 연구를 통해 딥러닝을 적용하여 초기 상대습도 예측을 활용하여 누설량을 정량화하고, 향후 원자력 발전소 내의 기기나 배관 및 밸브의 열화 또는, 진동이나 파손에 의한 누설사고에 대처함으로써 원자력 발전소의 안전성 향상에 기여할 수 있을 것으로 기대된다.

## 후 기

본 연구는 산업자원부 기술혁신사업의 일환으로 연구(과제 번호 : 20211510100050, 과제명 : 0.5gpm 미만의 원자로 냉각재계통 미확인 누설 실시간 감시시스템 개발)가 수행되었습니다. KETEP과 연구 지원 관계자 여러분께 감사드립니다.

## References

(1) Korea Institute of Nuclear Safety, 2018, "Reports of Accident and Breakdown in the Nuclear Power Plant," Korea Institute of Nuclear Safety, Daejeon, 170327K4.

(2) "Operational Performance Information System for Nuclear Power Plant," Nuclear accident and failure status, last modified Jan 05, 2023, accessed Jan 16, 2023, <http://opis.kins.re.kr/opis?act=KROBA3100R>.

(3) Kim, Y. S., Euh, D. J., Kim, W. S. and Kwon, T. S., 2019, "Investigation of Leakage Characteristics on Major Equipment/Component in Reactor System," The KSFJ Journal of Fluid Machinery, Vol. 22, No. 6, pp. 30~35.

(4) Kwon, T. S., Kim, J. R., Choi, C. K., Park, J. S. and Choi, C. R., 2021, "Development of a Unidentified RCS Leakage Detection Sensor System Less than 0.5 gpm," The KSFJ Journal of Fluid Machinery, Vol. 24, No. 2, pp. 13~19.

(5) Hochreiter, S. and Schmidhuber, J., 1997, "Long Short-Term Memory," Neural Computation, Vol. 9, No. 8, pp. 1735~1780.

(6) Schuster, M. and Paliwal, K. K., 1997, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, Vol. 45, No. 11, pp. 2673~2681.

(7) Siami-Namini, S., Tavakoli, N. and Namin, A. S., 2019, "The Performance of LSTM and BiLSTM in Forecasting Time Series," 2019, IEEE International Conference on Big Data (Big Data), pp. 3285~3292.

(8) Jung, J. J., Yun, H. Y., Park, I. K. and Jo, H. G., 2010, "The CUPID Code Development and Assessment Strategy," Vol. 42, No. 6, pp. 636~655.

(9) Chicco, D., 2017, "Ten Quick Tips for Machine Learning in Computational Biology," BioData Mining, Vol. 10, article No. 35.

(10) Vaisala Oyj, 2013, Calculation Formulas for Humidity—Humidity Conversion Formulas: Vaisala: Helsinki, Finland.

(11) Koc, C. K., 1995, "Analysis of Sliding Window Techniques for Exponentiation," Computers & Mathematics with Applications, Vol. 30, No. 10, pp. 17~24.

(12) Ruder, S., 2016, "An Overview of Gradient Descent Optimization Algorithms," arXiv preprint arXiv:1609.04747.

(13) Kingma, D. P. and Ba, J. L., 2014, "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980.