Transactions of the Korean Nuclear Society Virtual Autumn Meeting
October 21-22, 2021

Comparison of Consequence Estimation Model Using Machine Learning Technology

Sunghyun Park and Moosung Jae*
Nuclear Engineering Department, Hanyang University, 222, Wangsimni-ro, Seongdong-gu, Seoul, Republic of Korea, 04763
Corresponding author: jae@hanyang.ac.kr

1. Introduction

Nowadays, various industrial fields are introducing and utilizing machine learning technologies to their fields. The nuclear industry in the world, including Korea, is about to introduce machine learning technologies. However, the problem is that nuclear technologies, including Probabilistic Safety Assessment (PSA), have many uncertainties. Recently, in Korea, Kim et al. proposed a fast-running model using deep learning techniques to obtain plausible accident scenarios [1]. However, this work covers only thermo-hydraulic results, not consequences. To fill that void, we reviewed various machine learning technologies and applied them to the estimation consequences with different release fractions. This paper introduces various machine-learning technologies and their simple application.

2. Methods and Results

2.1 Machine Learning

Machine learning is the study of computer algorithms that improve automatically through experience and by the use of data [2]. In other words, machine learning is to find a relationship between input data and answers, called supervised learning. The simple description for supervised learning is presented in Fig. 1.

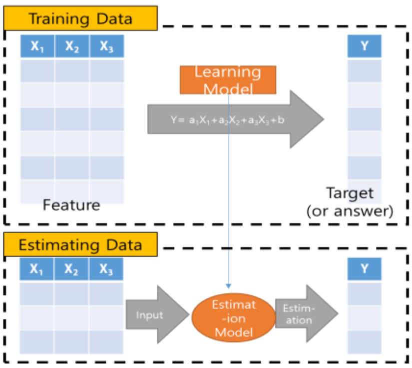

Fig. 1. Supervised learning process

In Fig.1, the training data consists of features (X) and targets (Y). Machine learning could be expressed as 'to find a relationship between X and Y, such as Y=aX+b. Based on it, the answer can be estimated on new data.

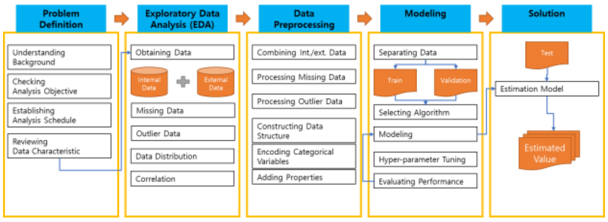

Fig. 2. Machine learning process [3]

The general machine learning process is presented in Fig. 2. The process consists of five steps; problem definition, Exploratory Data Analysis (EDA), data preprocessing, modeling, and solution. This study also followed this process. This paper describes some parts considered as a relatively important factor.

2.2 Problem Definition

Level 3 PSA calculates the consequence of radioactive material released from a nuclear power plant (NPP). The calculation time of Level 3 PSA is relatively short compared to source term analysis, but plume segmentation and weather data might increase the calculation time. However, in recent years, a new Atmospheric Transport and Diffusion (ATD) model, particle-based model, not a Gaussian plume model, is provided in MACCS [4]. The new model increases the calculation time by tens or hundreds of times. In summary, the more optimal calculation of Level 3 PSA is required, the more increased calculation time. In this regard, this study was conducted to determine whether predicting the consequence with input values would be possible.

The release fractions of nine radioactive nuclide groups and latent cancer fatality in 16 km from the NPP (Shin-kori 1) are considered features and answers, respectively. The features and answers are arbitrary. The whole data was obtained by Latin Hypercube Sampling (LHS) with 10,000 realizations as Fig. 3.

A	B	C	D	E		O
	Xe	Cs	Ba	I		LF-16
Model1.out	0.935813	0.271075	0.180771	0.296865		10200
Model2.out	0.944064	0.475762	0.312053	0.182343		10200
Model3.out	0.058822	0.996954	0.735196	0.620115	…	10200
Model4.out	0.496441	0.770491	0.208338	0.98046		10300
Model5.out	0.265422	0.352622	0.191012	0.061863		10300
Model6.out	0.700237	0.81749	0.941062	0.021987		10300
	⋮					
Model9997.out	0.356604	0.981764	0.516944	0.979931		9140
Model9998.out	0.033572	0.351718	0.526632	0.668728		9990
Model9999.out	0.333517	0.013866	0.871355	0.579371		10300
Model10000.out	0.141152	0.611811	0.931245	0.740499		10600

Fig. 3. LHS with 10,000 realizations

2.3 Modeling

The various tools for constructing machine learning models are summarized in Table I. Python used in this study is preferred for programming due to its vast features, applicability, and simplicity. The python programming language best fits machine learning due to its independent platform and its popularity in the programming community.

Table I: Major tools for machine learning

Tool	Descriptions
Data acquisition	MACCS ver. 4.0
Code editor	visual studio code ver. 1.57.1
Language	python ver. 3.9.6
Library	TensorFlow, sklearn, NumPy, pandas, seaborn

The obtained learning data from the MACCS was divided into two parts, training data, and test data, as Fig.4. Using 20-30% of the total data is appropriate for the test, and this study randomly spilt the total data.

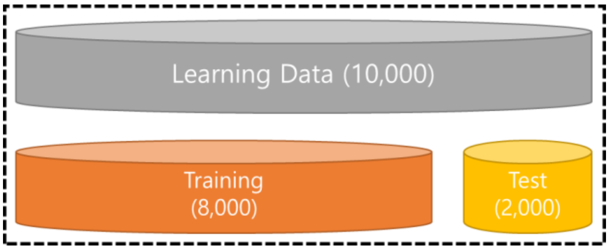

Fig. 4. Division of learning data

This study considered various machine learning models: linear/polynomial regression model, non-linear regression models, and multi-layer perceptron.

It is hard to say that the estimated value is accurately consistent with the actual value. The difference between the actual value (answer) and the estimated value is called the residuals. To evaluate the residuals of machine learning models, the Mean Squared Error (MSE) as Equation (1) was used.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y_i})^2 \tag{1}$$

MSE : Mean Squared Error
n : Total number of data
Y_i : The actual value of i^{th} data
$\widehat{Y_i}$: The estimated value of i^{th} data

By comparing the training MSE with the estimating MSE, the degree of the fit (overfitting or underfitting) can be determined. Overfitting refers to a model that models the training data too well. The overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the model's performance on new data. On the other hand, underfitting refers to a model that neither models the training data nor generalizes to new data. An underfit machine learning model is not a suitable and obvious model as it will have poor performance on the training data [5].

2.3.1 Linear and Polynomial Regression Model

The linear and polynomial regression are machine learning algorithm that performs a regression task. In general, the polynomial regression model rather than the linear model can lead to complex structures that increase the predictivity of the model, as shown in Table II.

Table II: MSE of the linear and polynomial regression model

Order	Training MSE	Test MSE	Fit
1st (Linear)	0.260	0.267	Over
2nd	0.012	0.011	Under
3rd	0.005	0.004	Under
4th	0.002	0.002	Over
5th	0.001	0.003	Over
6th	0.000	0.007	Over
7th	0.000	0.057	Over
8th	0.000	0.053	Over
9th	0.000	0.057	Over
10th	0.000	0.065	Over
11th	0.000	0.080	Over
12th	0.000	0.102	Over
13th	0.000	0.134	Over
14th	0.000	0.178	Over
15th	0.000	0.240	Over

2.3.2 Non-linear Regression Model

This study considered the decision tree model (a single tree), random forest model (multiple trees), and XGBoost model (sequential trees). The decision tree is simply a series of sequential decisions made to reach a specific result. The random forest model is also a tree-based algorithm that leverages the power of multiple decision trees. The XGBoost is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework. The result of the non-linear regression model is presented in Table III.

Table III: MSE of the non-linear regression model

Type	Training MSE	Test MSE	Fit
Decision Tree	0.131	0.295	Over

Random Forest	0.063	0.129	Over
XGBoost	0.000	0.059	Over

2.3.3 Multi-layer Perceptron Model

A multi-layer perceptron (MLP) is a class of feedforward artificial neural networks. The MLP consist of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Except for the input nodes, each node is a neuron that uses a non-linear activation function [6]. The architecture of MLP in this study is described in Fig. 5.

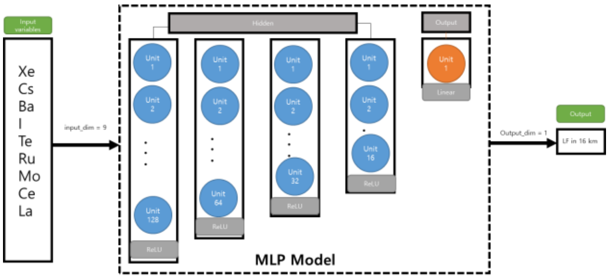

Fig. 5. The architecture of MLP model

The above MLP was constructed by the Keras module of TensorFlow. As a result, the number of parameters is 12,161 (=128(9+1)+64(128+1)+32(64+1)+16(32+1)+1(16+1)).

In the training process, there is a problem that training takes a long time to update the weights by inputting data one by one. To fix this problem, mini-batch training that divides the training data into small batches and inputting them into the model was used, as shown in Fig. 6.

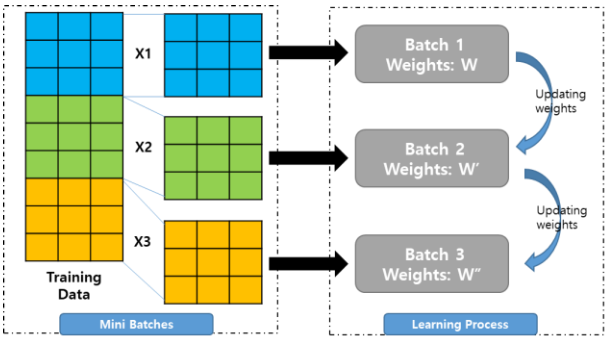

Fig. 6. Mini batches and learning process

In this study, 313 mini-batches are made by dividing total training data by 32, and the number of training (epoch) is 100. Therefore, the total number of updating weights is 31,300(=313*100). The MLP model was learned fast before the 3rd epoch, and then the learning rate over training epochs becomes flat, as shown in Fig. 7. Finally, the MSE is presented in Table IV.

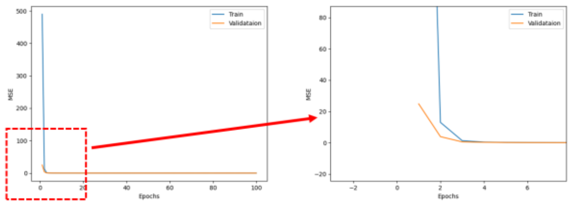
Fig. 7. Learning rate over training epochs

Table IV: MSE of the MLP model

Type	Training MSE	Test MSE	Fit
MLP	0.008	0.003	Under

3. Conclusion

This study was performed to introduce various machine learning technologies and identify applicability to Level 3 PSA. We considered linear/polynomial regression, non-linear, and multi-layer perceptron models and calculated Mean Squared Error (MSE) as a comparison measurement. It was found that various machine learning models could estimate consequences sufficiently. In this study, simple static data (only release fraction) was used. However, the learning process would be more complicated if time series data is included and many features are considered. The consideration of other features remains future work.

Acknowledgment

This work was supported by the Nuclear Safety Research Program through the Korea Foundation Of Nuclear Safety (KOFONS), granted financial resource from the Multi-Unit Risk Research Group (MURRG), Republic of Korea (No.1705001).

REFERENCES

[1] H. Kim, J. Cho, and J. Park, Application of a Deep Learning Technique to the Development of a Fast Accident Scenario Identifier, IEEE Access, Vol. 8, p. 177363-177373, 2020.
[2] en.Wikipedia.org [Internet]. [cited 2021 Aug 2]. Available from:https://en.wikipedia.org/wiki/Machine_learning#cite_note-1
[3] Seunghwan Oh. Python Deep Learning Machine Learning. Information Publishing Group; 2021.
[4] Nathan Bixler, et al., MACCS User's Guide and Reference Manual-Draft Report. 2020.
[5] Machine Learning Mastery [Internet]. [cited 2021 Aug 5]. Available from: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/
[6] en.Wikipedia.org [Internet]. [cited 2021 Aug 5]. Available from: https://en.wikipedia.org/wiki/Multilayer_perceptron