Assessing Vulnerability of Nuclear AI using Layer-wise Relevance Propagation and Bayesian Neural Networks for Adversarial Attack Mitigation

Young Ho Chae [a], Seoryong Koo [a*]
[a]*Korea Atomic Energy Research Institute, 111 Daedeok-daero 989 beon-gil, Daejeon, 34057*

[*]*Corresponding author: srkoo@kaeri.re.kr*

1. Introduction

The emergence of various data-collecting devices has spurred proposals for AI-based research across multiple fields. Within the nuclear industry, several AI-based methods have been suggested for various tasks [1-3].

AI technology utilized in the nuclear field, being a safety-critical infrastructure, necessitates a level of reliability surpassing that required by AI in general systems. The installation of various software in digitalized nuclear power plants, in contrast to the previous analog power plants, necessitated the introduction of the V-model to ensure the software's integrity. However, when applied directly to deep learning-based AI technology, the V-model presents a challenge. In the case of non-deep learning-based software, V&V focuses on the software's integrity and malfunction. Conversely, AI-based software requires a comprehensive verification of both the software's reliability and the data utilized for its training, as the training process and the data are closely intertwined. Therefore, deep learning-based software necessitates a more inclusive verification process to ensure its trustworthiness.

Therefore, this paper suggests a framework that augments AI trustworthiness through the promotion of robustness and explainability while concurrently identifying data-induced vulnerabilities. This comprehensive framework comprises a data feature extraction module that relies on Layerwise-relevance propagation (LRP) and a neural network confidence estimator that is based on Bayesian neural network (BNN) methodology. Implementation of suggested framework enables the verification of data robustness, which is a pivotal facet of AI trustworthiness and allows for the exploration of viable alternatives, including data augmentation and perturbation robustness, via analysis of the data analysis results. Additionally, the framework facilitates the identification of the most salient variables for prioritizing data integrity checks.

2. Data trustworthiness assessing framework

Fig. 1 illustrates the all-inclusive framework that comprises two modules performing distinct functions. The framework operates by initially utilizing the LRP module to extract variables that significantly impact DL software. The subsequent step entails the application of perturbations to the extracted critical data types and quantitatively assessing the changes in software response confidence via the BNN module.

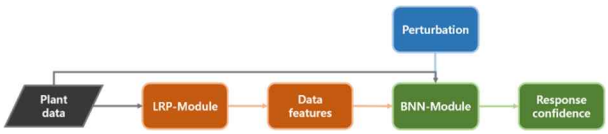

Fig. 1 Data trustworthiness assessing framework

2.1 Layerwise relevance propagation-based data feature extraction module

Relevance propagation is a quantitative technique for determining the contribution of each layer to the output after decomposition. The calculation of relevance across all hidden layers can enable the determination of the relevance of an element x within a dataset vector **X**. The overall relevance propagation can be calculated using Eq. 1, whereas the propagation in a specific layer can be calculated using Eq. 2. Typically, a multivariate Taylor series expansion is used for each propagation computation. Relevance propagation can be computed for both forward and backward propagation.

$$f(x) \approx \sum_{d=1}^{k} R_d \qquad \text{Eq. 1}$$

Where,
x is a data element from dataset **X**
k is a dimension of dataset
R is a relavance score

$$R_i^l = \sum R_{i\to k}^{l,l+1} \qquad \text{Eq. 2}$$

i is input for neuron k
l is layer

By determining the relevance of element x within the dataset, we can assess its importance and contribution to the output at a particular time, especially after an incident. The extracted data is crucial for the proper functioning of DL-based software compared to other data.

2.2 Bayesian neural network-based response confidence estimate module

We have developed a Bayesian Neural Network (BNN) based confidence estimator that discerns critical variables via the Layer-wise Relevance Propagation (LRP) module and quantitatively assesses the influence

of Deep Learning (DL)-based software when perturbations or noise influences these critical variables. Quantifying confidence is important for ensuring the trustworthiness of Artificial Intelligence (AI) systems.

For example, envision a scenario wherein the objective is to design an agent capable of binary classification (*a, 1-a*). Assuming both models are constructed upon a neural network-based architecture and employ a softmax layer for probability estimation by transforming classification outcomes into probabilities. Given a specific data x with label a, Agent1 ascertains a 90% probability of it being *a*, while Agent2 deduces a 60% probability. Despite the performance-based metric yielding identical results, one could argue that Agent1 exhibits superior design. Consequently, to consider the aforementioned issue, we utilize the BNN-grounded confidence estimation module to quantitatively ascertain the impact of data on the confidence level of the DL-based software's response.

A salient distinction between the BNN-based model and other artificial neural networks lies in the perception of weights and biases within the neural network as distributions rather than discrete values. The neural network computation is executed utilizing Eq 3.

$$y = \sigma(w^T x + b)$$

Where,
w is a weight
b is a bias
σ is an activation function
x is a input data

Eq. 3

In Bayesian Neural Networks (BNNs), weights (w) and biases (b) in Equation 3 are considered as distributions. Bayesian inference entails defining a prior distribution and updating the posterior based on observations. Several approaches exist to define a prior in a deep neural network, such as weight-space priors, function-space priors, BNN ensembles, and more. In this study, we employed a weight-space prior to constructing a data-dependent prior. The fundamental architecture of the weight-space prior is adapted from Atanov et al. [4]. As per the same paper, the assignment of a prior is facilitated by the algorithm illustrated in Figure 2.

Algorithm 1 Stochastic Variational Inference With Implicit Prior Distribution

Require: dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
Require: variational approximations $q(w \mid \theta_{ij}^l)$ and reverse models $r(z \mid w; \psi_l)$
Require: reconstruction models $p(w \mid z; \phi_l)$, priors for auxiliary variables $p_l(z)$
while not converged **do**
  $M \leftarrow$ mini-batch of objects form dataset $\mathcal{D}$
  $\hat{w}_{ij}^l \leftarrow$ sample weights from $q(w|\theta_{ij}^l)$ with reparametrization
  $\hat{z}_{ij}^l \leftarrow$ sample auxiliary variables from $r(z \mid \hat{w}_{ij}^l; \psi_l)$ with reparametrization
  $\hat{\mathcal{L}}^{aux} \leftarrow L_{\hat{M}} + \sum_{l,i,j} - \log q(\hat{w}_{ij}^l \mid \theta_{ij}^l) - \log r(\hat{z}_{ij}^l \mid \hat{w}_{ij}^l; \psi_l) + \log p_l(\hat{z}_{ij}^l) + \log p(\hat{w}_{ij}^l \mid \hat{z}_{ij}^l; \phi_l)$
  Obtain unbiased estimate $\hat{g}$ with $\mathbb{E}[\hat{g}] = \nabla \mathcal{L}^{aux}$ by differentiating $\hat{\mathcal{L}}^{aux}$
  Update parameters θ and ψ using gradient $\hat{g}$ and a stochastic optimization algorithm
end while
return Parameters θ, ψ

Fig. 2 Weight-space prior algorithm (Atanov et al. [4])

The potency of BNNs is evident through the results of applying a BNN to a simple function for regression. Assuming we have a data distribution as depicted in Figure 3 ($0.3 * \sin(3\pi(x + noise)) + 0.3 * \cos(4\pi(x + noise))$), executing a regression analysis using a 3-layer fully connected neural network results in the line showcased in Figure 3. As BNNs compute weights as a distribution, they can concurrently express the confidence bounds of the output. The blue-shaded region represents the 95% confidence interval of the neural network output. The confidence interval is notably narrower within the interpolation region, signifying elevated confidence in the current output. In contrast, the confidence interval broadens within the extrapolation region, indicating diminished confidence in the output. Fluctuations in the confidence interval enable the quantification of perturbation effects on the output of DL-based software.

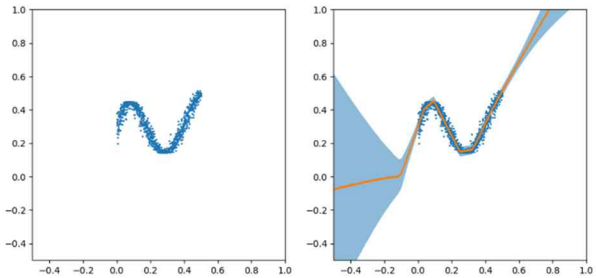

Fig. 3 Sample dataset (Left), Regression results (Right)

3. Conclusions

In conventional software Verification and Validation (V&V), the emphasis lies in examining the reliability of software operations. However, the algorithm and data warrant verification when employing Deep Learning (DL)-based software. Therefore, this study proposes a data verification framework that enhances data robustness and transparency, constituting an aspect of AI trustworthiness. The methodology readily applies to various previously developed DL-based algorithms, as it can be analyzed by modifying the trainable parameter, transforming the value to be computed as a distribution through the assignment of a prior distribution. Utilizing the suggested framework, crucial data can be selected, and algorithmic robustness against data can be assessed. Additionally, it is anticipated to facilitate the identification of critical data assets requiring protection to ensure the safety of DL-based software from cyberattacks.

Future research will be conducted as follows: To evaluate the validity of the proposed methodology, experiments on diverse data types are currently underway. The experiments may necessitate alterations to the methodology for assigning prior distributions—moreover, further investigation of the results under various conditions to establish the threshold level. At present, the confidence interval can be calculated, yet no

criteria exist for assessing a confident response up to a specific confidence interval level.

The study's limitations are as follows: The principal variables estimated by the Layer-wise Relevance Propagation (LRP) module may encompass variables with minimal physical meaning. BNNs compute all weights as a distribution, signifying that variables treated as single values must be calculated as one by n matrix, a calculation considerably slower than traditional algorithms. Nonetheless, this is not a significant concern as the tool is intended for V&V rather than real-time evaluation.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) (No. RS-2022-00144150)

REFERENCES

[1] Chae, Y. H., Kim, S. G., Kim, H., Kim, J. T., & Seong, P. H. (2020). A methodology for diagnosing FAC induced pipe thinning using accelerometers and deep learning models. Annals of Nuclear Energy, 143, https://doi.org/10.1016/j.anucene.2020.107501
[2] Lee, G., Lee, S. J., & Lee, C. (2021). A convolutional neural network model for abnormality diagnosis in a nuclear power plant. Applied Soft Computing, 99, https://doi.org/10.1016/j.asoc.2020.106874
[3] Chae, Y. H., Lee, C., Han, S. M., Seong, P. H.(2022) Graph neural network based multiple accident diagnosis in nuclear power plants: Data optimization to represent the system configuration. Nuclear Engineering and Technology, 54, https://doi.org/10.1016/j.net.2022.02.024.
[4] Atanov, A., Ashukha, A., Struminsky, K., Vetrov, D., & Welling, M. (2019). The Deep Weight Prior (arXiv:1810.06943). arXiv. http://arxiv.org/abs/1810.06943