Journal of KOSSE. (2023. 12) Vol. 19, No. 2 pp. 18-31
DOI: https://doi.org/10.14248/JKOSSE.2023.19.2.018
www.kosse.or.kr

ISSN (print) : 1738-480X
ISSN (online) : 2288-3592

A SE Approach for Machine Learning Prediction of the Response of an NPP Undergoing CEA Ejection Accident

Ditsietsi Malale[1)], Aya Diab[1),2)*]

1) Nuclear Engineering Department, KEPCO International Nuclear Graduate School, Ulsan, South Korea.

2) Mechanical Power Engineering Department, Faculty of Engineering, Ain Shams University, Cairo, Egypt.

Abstract: Exploring artificial intelligence and machine learning for nuclear safety has witnessed increased interest in recent years. To contribute to this area of research, a machine learning model capable of accurately predicting nuclear power plant response with minimal computational cost is proposed. To develop a robust machine learning model, the Best Estimate Plus Uncertainty (BEPU) approach was used to generate a database to train three models and select the best of the three. The BEPU analysis was performed by coupling Dakota platform with the best estimate thermal hydraulics code RELAP/SCDAPSIM/MOD 3.4. The Code Scaling Applicability and Uncertainty approach was adopted, along with Wilks' theorem to obtain a statistically representative sample that satisfies the USNRC 95/95 rule with 95% probability and 95% confidence level. The generated database was used to train three models based on Recurrent Neural Networks; specifically, Long Short−Term Memory, Gated Recurrent Unit, and a hybrid model with Long Short−Term Memory coupled to Convolutional Neural Network. In this paper, the System Engineering approach was utilized to identify requirements, stakeholders, and functional and physical architecture to develop this project and ensure success in verification and validation activities necessary to ensure the efficient development of ML meta−models capable of predicting of the nuclear power plant response.

Key Words: Control Element Assembly Ejection, Uncertainty Quantification, Best Estimate Plus Uncertainty, Machine Learning, Recurrent Neural Network, Long Short−Term Memory, Gated Recurrent Unit, Convolutional Neural Network

Received: June 22, 2023 / **Revised**: November 21, 2023 / **Accepted**: December 13, 2023

* Corresponding Author: Aya Diab / KEPCO International Nuclear Graduate School / aya.diab@kings.ac.kr

1. Introduction

The 2011 Fukushima Daiichi accident emphasized the need for more resilient nuclear power plants. An essential element in averting such incidents is effective decision-making. Given that humans may not always find themselves in optimal positions for sound decision-making, it becomes crucial to leverage all available resources to ensure operators are positioned to make the most informed choices and avert crises. Consequently, there is a pressing demand to create machine-learning tools that can support operators in making these critical decisions. This project has emerged as a response to this necessity and represents the initial phase in the development of a machine learning tool capable of predicting the Nuclear Power Plant (NPP) response under accident conditions.

The application of a Systems Engineering approach is crucial for the systematic development of a machine learning model designed to forecast the NPP response. The Systems Engineering approach was found to be holistic and useful in integrating different elements of the project (Sifat, et al., 2024)[16], (Buonocore, et al., 2023)[3], (Mahmoud & Diab, 2020)[11] and (van Erp, et al., 2023)[17] by providing guidelines to manage and execute the various tasks while ensuring that the overarching objectives are met with acceptable quality. This approach emphasizes a holistic view of the entire system, including requirements, interconnections, objectives, etc. Therefore, in the context of this work, the 'system' pertains to the development of a Machine Learning (ML) meta-model capable of predicting the response of the Korean APR-1400 during an accident scenario.

A system has interconnected elements that work together to realize a common objective (Sifat, et al., 2024).[16] For this work, these elements include processes, stakeholders, requirements, etc., all of which are arranged to meet the objective. Elements of this work consist of conducting conservative safety analysis, performing Best Estimate Plus Uncertainty (BEPU) analysis to create a statistically representative database for the development and training of the ML model, and lastly, conducting the training and prediction phases of the ML model.

2. Literature Review

ML is a subset of modern artificial intelligence that identifies hidden patterns by handling and learning from large datasets the inherent characteristics of the data and hence develops a statistical mapping between inputs and outputs without any prior pre-programming as is the case with physics-based modeling (Huang, et al., 2023).[7] This project explores the use of Recurrent Neural Networks (RNNs), specifically using long short-term memory (LSTM), convolutional neural networks (CNN), and Gated Recurrent Units (GRU)) given their good performance in the management of large training data, large input features, and long time series (Ramezani, et al., 2023).[14]

ML continues to increase in popularity for its ability to adequately predict the response of complex systems. Timely forecasting of the system response can be invaluable under accident conditions. For this work, ML is therefore used to predict the system response following a CEA ejection accident.

The CEA ejection accident belongs to a group of accidents called reactivity-initiated accidents (RIAs). CEA ejection is assumed to be caused by mechanical failure resulting in rupture of the control element drive mechanism (CEDM), and therefore subsequent full withdrawal of the CEA and drive shaft caused by pressure of the RCS. For conservatism, loss of offsite power (LOOP) is assumed to coincide with the turbine trip that follows the accident (Korea Hydro and Nuclear Power Co., Ltd., 2018).[8]

Uncontrolled ejection of a CEA results in positive reactivity insertion into the reactor core. Reactivity is first lessened by Doppler feedback, followed by reactor trip. In reactivity-initiated accidents, the fuel pellet heats rapidly, with a corresponding temperature increase. This rapid heating results in pellet-cladding mechanical interaction (PCMI), which occurs when the fuel expands and presses on the cladding inner wall. PCMI causes cladding stress and deformation that could result in the formation of cracks, crack propagation, and fuel failure (Magnusson, et al., 2018).[12] Parameters of interest for this accident are peak fuel rod temperature and fuel rod enthalpy, for the evaluation of cladding temperature failure, pellet cladding mechanical interaction (PCMI) failure, and core coolability.

BEPU is an established accident analysis approach that can provide realistic predictions compared to the conservative approach; hence, resulting in larger safety margins and more operational flexibility. BEPU has been used in licensing activities of certain accidents; for example, at Angra-2 in Brazil, Kozloduy-3 VVER-440 in Bulgaria, Smolensk-3 RBMK in Russia, Balakovo-3 VVER-1000 in Russia, Atucha-2 in Argentina, and others (D'Auria, 2019).[4]

The BEPU approach is guided by The Code Scaling Applicability, and Uncertainty (CSAU) methodology, starting with the phenomena Identification and Ranking Table (PIRT) for the rod ejection accident in PWRs developed by a group of experts in a project led by the US NRC in 2001 (Boyak, et al., 2001).[2] Next, a list of uncertain parameters is derived from the PIRT with statistical details (range and probability density function) gathered from published literature for reactive-initiated accidents (Marchand, et al., 2018).[12] These uncertain parameters are propagated into a thermal-hydraulic (TH) model to assess the most probable system response.

According to the CSAU methodology, the Monte Carlo method should be used to arrive at a sample with a probability of 95% and a confidence level of 95%. However, for computational efficiency, a non-parametric method is used [8] following the Gesellschaft für Anlagen-und Reaktorsicherheit (GRS) based on Wilks' formula. The proposed method acquires the 95/95 tolerance limit by order statistics with a sample size that is independent of the number of uncertain parameters and much smaller than that of Monte Carlo which makes Wilks' method a preferred alternative for industrial applications (Han & Kim, 2019).[6] Wilks's 5^{th} order, with a minimum sample number of 181 samples was chosen for this study based on recommendations of previous studies (Han & Kim, 2019).[6]

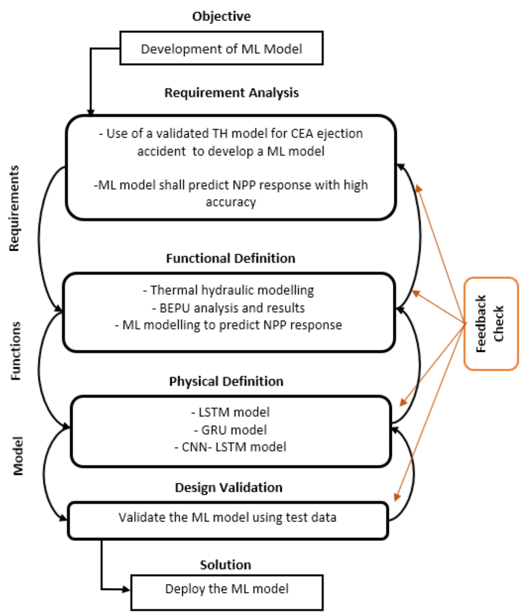
3. Systems Engineering Approach

3.1. Objective and Methodology

The main objective of this work is to develop an ML metamodel capable of predicting the response of the APR1400 undergoing a CEA ejection accident. The system referred to in this work is therefore the development of the ML meta-model.

The Systems Engineering (SE) approach is utilized to guide the development of the ML model by breaking down the tasks associated with model development into more manageable components. In this context, the Kossiahoff SE (Kossiakoff, et al., 2020)[9] method is applied, involving four distinct steps:

1. Requirement Analysis
2. Functional Definition
3. Physical Definition
4. Design Validation



[Figure 1] SE Method Objective Hierarchy

When applying the Kossiahoff SE method (Kossiakoff, et al., 2020)[9] for ML metamodel development, the objective hierarchy becomes valuable as shown in Figure 1. It serves the purpose of delineating objectives, tasks, solutions, and feedback relationships among tasks, with the ultimate aim of guaranteeing high-quality outcomes from the system.

3.2. Work Breakdown Structure

A work breakdown structure with the following activities was developed:

1. Develop a TH model of the APR-1400 undergoing CEA ejection accident scenario.
2. Validate the steady-state response of the TH model using conservative assumptions.
3. Validate the transient response of the TH model against the Design Control Document, DCD.
4. Develop an uncertainty quantification framework by coupling the validated TH model with DAKOTA.
5. Start with PIRT to identify key uncertain parameters.
6. Propagate uncertain parameters into the TH model.
7. Analyze the uncertainty and report the BEPU results.
8. Develop three ML meta-models.
9. Train, validate, and test the three ML models using results from uncertainty quantification.
10. Analyze the ML models results to select the best model.
11. Deploy the ML model.

3.3. Stakeholder Identification

Identification of interested and affected parties is paramount from the concept to the implementation stage of this project. Stakeholders for this work refer to individuals or groups of people who have a vested interest in the development of this ML model. These stakeholders have specific needs, expectations, concerns, or interests related to this project.

This project is undertaken as a research project at a university, therefore stakeholders interested in this work are predominately researchers, the university, and the scientific community. A list of stakeholders was identified and described in Table 1.

<Table 1> Stakeholders

Stakeholder	Justification
University Researchers	– Interests in simulations as this work can influence their projects. – Team members: shared knowledge of codes and analysis. – For review and advice relating to the development of this work
University	– To provide tools to perform the work – To sponsor the work – Publication of research
External stakeholders	Stakeholders who may be interested in the results of the project. These include: – Utility companies – Scientific journals – Nuclear regulator – Researchers and scientists

3.4. Requirement Analysis

Requirement analysis is essential in identifying both the functional and non-functional requirements of the system and establishing the scope of the project. This phase aids in ensuring that the system is designed and implemented to meet the expectations of stakeholders and users, ensuring it achieves the necessary performance and quality standards.

The requirements for development of an ML model will be divided into mission requirements and component requirements, i.e., ML Development requirements.

<Table 2> SE Requirements

Requirement	Description
Mission requirements	The ML model shall predict NPP response with good accuracy.
ML Development requirements	– Three RNN models shall be trained and evaluated for performance. – The best model shall be used to predict NPP response with an accuracy of more than 90%

3.4.1. Mission Requirements

In the systems engineering approach, the mission sets the foundation for the system development lifecycle. The project's overall mission is to develop an ML meta-model that can accurately predict NNP response undergoing a CEA ejection accident.

As stated in Table 1, university researchers may have an interest in the development of the ML model as this project could support research in the area of nuclear safety. For safety analysis determining the optimal time to deploy emergency operating procedures, or high level candidate actions within the severe accident management guidelines to prevent reactor pressure vessel breach, can be assisted via a robust ML tool that can assist the operators, and provide confidence that the correct actions are taken. Having an ML model that can predict the NPP response with good accuracy will enhance accident mitigation work by providing a tool that can expedite decision-making for operators during severe accidents or prevention thereof. Furthermore, researchers provide support for this work by documenting their findings and providing a body of knowledge on system codes and research methodologies available for this type of research.

The university has a vested interest as this work can play a fundamental role in fulfilling its mission and contributing to nuclear safety research. Universities help create knowledge and advancement in research in the nuclear industry and allow advancement of people's understanding of the nuclear field, stay current on new developments, attract talent, and solve nuclear safety problems. Having an ML model that can predict NPP response accurately and reliably can only improve the standing of the institution and advancement of the tools that are currently used in the nuclear industry.

Stakeholders that are external to the university, such as utility companies, and nuclear regulators are interested in the development of a ML model capable of predicting NPP response because this tool has the advantage of enhancing the safety, efficiency, and reliability of nuclear power plants. Some advantages of this tool include assisting nuclear operators to meet regulatory requirements.

Utility companies are interested in the development of this tool because it offers the potential to improve safety, increase efficiency, and optimize various aspects of plant operations. By leveraging the capabilities of ML, utility companies can enhance the overall performance and reliability of nuclear power plants.

Scientific journals, scientists, researchers, and others may be interested in this work because it has scientific value, it adds to the body of work done on this technology. This interest is similar to that of the university and researchers.

3.4.2. ML Development Requirements

The ML development requirements or component requirements refer to what is needed to develop the ML model. For this work to be a success, three key steps are necessary. First is developing a thermal hydraulic (TH) model, then performing BEPU analysis for the CEA Ejection accident on the APR-1400 NPP, and finally developing an ML meta-model to predict the transient response of the plant.

The CEA ejection accident is analyzed for the APR1400 using the best estimate code, RELAP5/SCDAPSIM/MOD3.4. The uncertainty quantification framework is developed by coupling RELAP5/SCDAPSIM/MOD3.4 with the statistical software, DAKOTA. The database collected from DAKOTA is subsequently used for training and validation of the machine learning models. Once trained, the machine learning (ML) model is used to predict the NPP response.

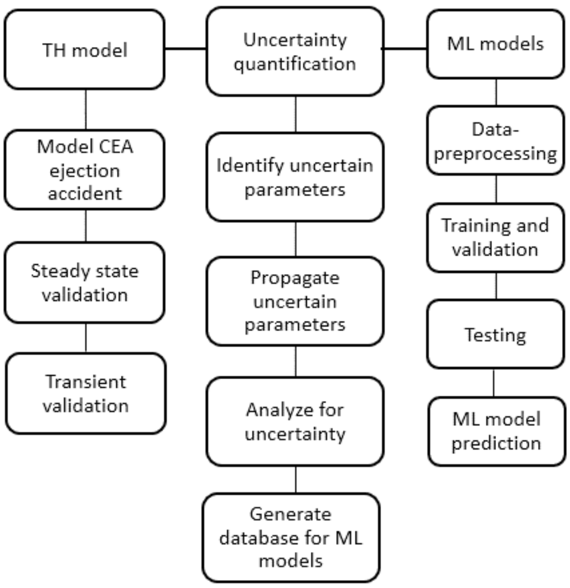
Three RNNs that are closely related must be tested and the best one selected for its ability to accurately predict NPP response at the lowest computational cost. For this reason, three RNNs were chosen for this work, LSTM, GRU, and a combination of CNN and LSTM.

3.5. Functional Architecture

For the successful execution of the project, it is indispensable to develop the system architecture reflecting both physical and functional details.

The system architecture for this work follows the three elements contained in the development of the ML model. Functional architecture is necessary in developing criteria for evaluation for each of the steps of the system. As shown in Figure 2, three main functions are proposed.

The first function is the TH modelling, which must have sufficient credibility. This entails modelling the accident to replicate the design of the APR1400 as documented in the DCD, this includes steady state and transient validation.



[Figure 2] Functional Architecture

The second function is the uncertainty quantification. For this function, the evaluation criteria are meeting the USNRC 95/95 rule using Wilks' 5^{th} order formulae.

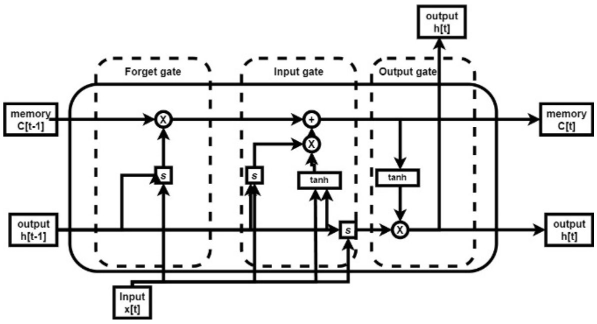
Lastly, the ML development function of this work entails training three ML models. These models must all be trained on the same dataset, and their performance evaluated and compared. The result of this step must be prediction accuracy of above 90% with the lowest computational cost.

3.6. Physical Architecture

The physical structure of the proposed three RNNs is presented below for each of the three ML meta-models.

3.6.1. Long Short-Term Memory (LSTM)

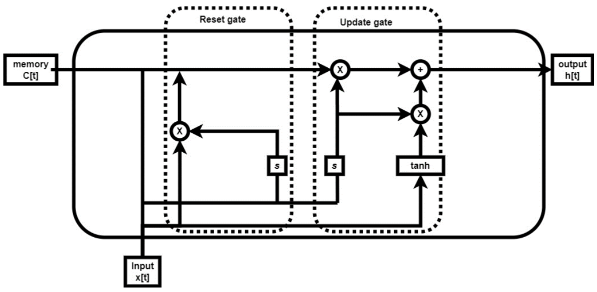
As stated earlier in this paper, the LSTM model belongs to a group of RNNs, and as is commonly known, RNNs have input, hidden, and output layers, where the output of a layer is saved and fed back into the input for it to predict the output layer. RNNs were created to solve issues of feed-forward neural networks such as not being able to manage sequential data (Qian & Liu, 2023)[13], i.e. inability to memorize input and only being able to process one input. LSTM is distinguished from the generic RNN by its use of gates (forget, update, and output) to manage the data. The functions of these gates comprise: discarding undesired information from the previous state and output of the upper hidden layer, updating the current state by controlling the information to be added to the next state from the previous hidden state, and filtering out desired and undesired information that will be reproduced by the output gate, respectively (Qian & Liu, 2023)[13] (Agga, et al., 2022)[1] (GitHub, 2023).[5] LSTM has shown reasonable capability in multivariate times series forecasting (Qian & Liu, 2023)[13], however, there are issues of vanishing and exploding gradients even though the model does allow easy modeling of dependencies (GitHub, 2023).[5]



[Figure 3] LSTM Cell

3.6.2. Gated Recurrent Unit

GRUs are like LSTMs but have two gates instead of three: reset and update gates (Qian & Liu, 2023).[13] The update gate determines the amount of information passed on from the previous time step; whereas the reset gate decides which information can be eliminated. This RNN cell also uses sigmoid and tanh activation functions. GRUs, as shown in the figure below, have simpler architecture and fewer parameters. The basic cell of GRU is shown in Figure 4.

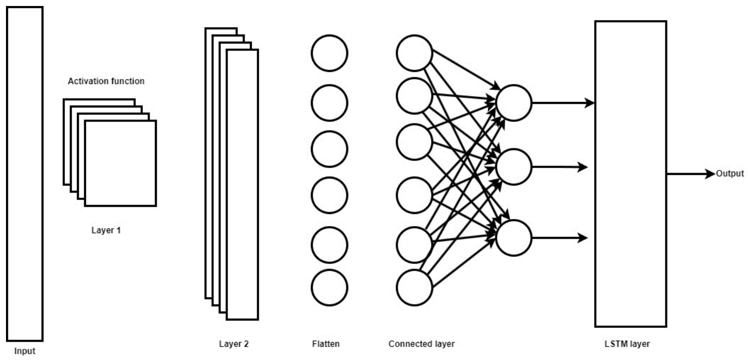


[Figure 4] GRU Cell

3.6.3. CNN + LSTM

CNN is a deep learning model that has historically been used to find patterns and to classify images as it has a grid-like two-dimensional topology (Agga, et al., 2022).[1] Although this neural network is predominantly used to classify images, it can be used for the extraction of features such as time series data and can apply 1D, 2D, and 3D convolution (Nguyen & Diab, 2023).[18] In CNNs, the input is filtered through one convolutional layer, and as the input passes through, a weighted summation is conducted before passing onto the next convolutional layer. The algorithm takes the input, assigns weights and biases to unique features of the input, learns from these, and can differentiate the objects from the input. The last fully connected layer extracts characteristics of the input, and the input is finally labeled by the last dense layer (Shin, et al., 2023).[15] CNNs usually use convolution and RELU, followed by pooling, and flatten, a fully connected layer, SoftMax in the output layer.

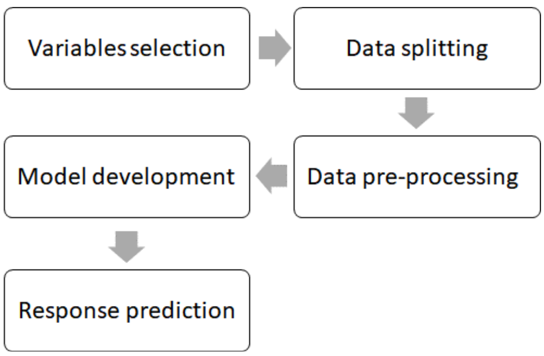
The combination of CNN and LSTM in this work was chosen because of the robustness of hybrid models in managing time series data (Agga, et al., 2022)[1], and the increased computational efficiency as the CNN layer removes noisy data before passing the input onto the LSTM model (Nguyen & Diab, 2023).[18] In this work, the dataset is first preprocessed (which includes normalization, shaping, and inverse transforming), then passed onto the CNN model, followed by LSTM before being transformed back into the desired values as is illustrated in Figure 5.



[Figure 5] CNN + LSTM Cell

3.7. ML Model Development

The database generated from the uncertainty quantification step of this work is used for the training, validation, and testing of three ML Meta models. This section outlines the process used to perform the ML prediction and follows the process in Figure 6.

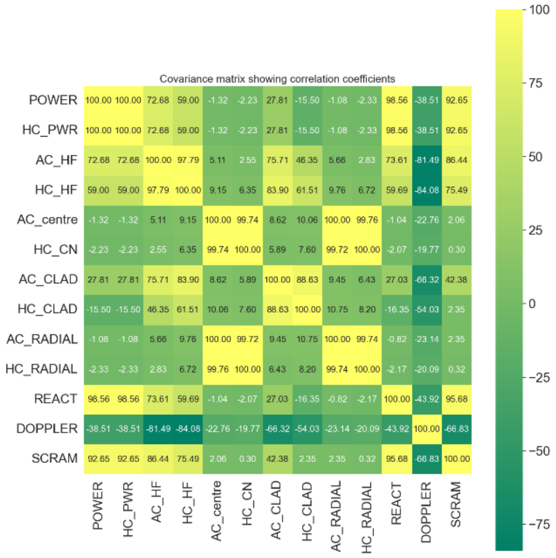


[Figure 6] ML Model Workflow

The first step is to select key input variables for the ML models. For this work, 14 variables from the transient response were selected, i.e., power, hot channel power (HC_PWR), average channel power, average channel heat flux (AC_HF), hot channel heat flux (HC_HF), average channel fuel centerline temperature (AC_centre), hot channel fuel centerline temperature, average channel fuel clad temperature (AC_CLAD), hot channel fuel clad temperature (HC_CLAD), average channel fuel radial temperature (AC_RADIAL), hot channel fuel radial temperature (HC_RADIAL), total reactivity (REACT), Doppler reactivity and scram.

The covariance matrix below was used to inform the relationship commonalities between the variables. As shown in Figure 7, not all the variables are related in the same manner. The coefficient matrix below can be read from 0 to 100. Positive values indicate that both are directly proportional, so when one increases the other increases too e.g., there is a positive relationship between power and scram; while negative values indicate that when one value is increasing, the other is decreasing, e.g., there is an inverse relationship between the Doppler reactivity and average channel heat flux. Values close to zero indicate that there is no relationship between the variables; whereas high values indicate a strong relationship.

The next step involves the selection of training and testing data. As is common practice in supervised learning, the training dataset is used to fit the model, which simply means adjusting weights and biases applied to the data as it passes through the network so the ML model can learn from the dataset. The validation dataset is used to evaluate how well the machine learns, then finally the testing dataset is used to predict the NPP response. For this work, the data split was 80% for training data of which 20% is used for validation and 20% for testing.



[Figure 7] Covariance Matrix for all Variables

The following step is data pre-processing. The standard scaler function was used to transform the variables in the dataset to the same scale. Initially, the dataset was not scaled, and it was clear that large values from temperature variables were dominating the model, making it inaccurate. Scaling the dataset to a consistent scale made the model more accurate and effective in predictions. This step was followed by creating a dataset with dependent and independent variables, i.e., here a dataset of y and x was created. The last step of preprocessing was shaping the dataset, as the model expects a specific shape (i.e., samples, time steps, and features) for it to run.

Once preprocessed, the dataset was fed into the ML model. The hyper-parameters in Table 3 were used for each of the models. The hyper-parameters were selected guided by the open literature and confirmed by trial and error following the common practice.

<Table 3> ML Model Hyper−Parameters

Parameter	GRU	LSTM	CNN+LSTM
Optimizer	Adam	Adam	Adam
Activation function	Relu	Relu	Relu
Kernel regularizer	L1(1´10−6)	L1(1´10−6)	L1(1´10−6)
Epoch	40	40	40
Batch size	100	100	100
Hidden layers	1	1	2
Learning rate	1	1	1
Input shape	(81935,10,13)	(81935,10, 13)	(81935,10,13)

The training process was followed by model evaluation. Several model performance metrics were used in this work; specifically, the Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), coefficient of determination (R^2), and prediction accuracy. It is important to observe the error during training using more than one metric as some metrics may be more sensitive to outliers than others. It is worth noting that all three error metrics, MAE, MSE, and RMSE were reasonably low for this work (which is desirable).

MAE monitors the averaged absolute error between the actual values of the dataset and the predicted value, it is demonstrated using Eq.1. MSE is the square of the error between the actual and predicted value as illustrated using Eq.2. This error penalizes larger errors compared to smaller ones because of squaring. The RMSE returns the square root of MSE while still penalizing higher errors, as is denoted by Eq.3.

$$MAE = \frac{1}{n}\sum_{i}^{n} y_i^{actual} - y_i^{predicted} \quad (1)$$

$$MSE = \frac{1}{n}\sum_{i}^{n} (y_i^{actual} - y_i^{predicted})^2 \quad (2)$$

$$RMSE = \sqrt{MSE} \quad (3)$$

The R^2 monitors the variance in the dependent variable, it shows how well the predicted values fit the actual values. The prediction accuracy was calculated by dividing the average of the predicted values by the actual values and multiplying by 100 to get a percentage. Ideally, values for MAE, MSE, and RMSE should approach 0, while R^2 and accuracy should approach 1 and 100%, respectively.

3.8. Verification and Validation

Verification and validation are required to ensure that requirements are met. This work had two requirements, mission (the ML model shall predict NPP response with good accuracy), and ML development requirements (three RNN models shall be trained and evaluated for performance and the best model shall be used to predict NPP response with the accuracy of more than 90%).

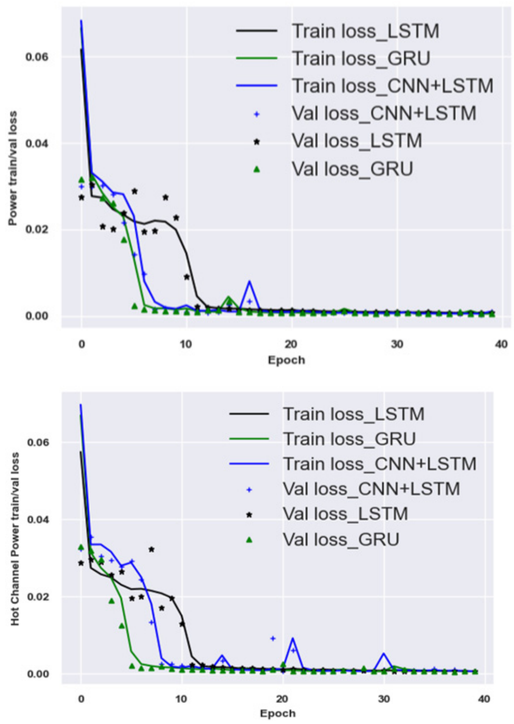
Both the mission requirements as well as the ML development requirements were met: all three ML that were developed in this work are capable of predicting the NPP response with reasonable accuracy.

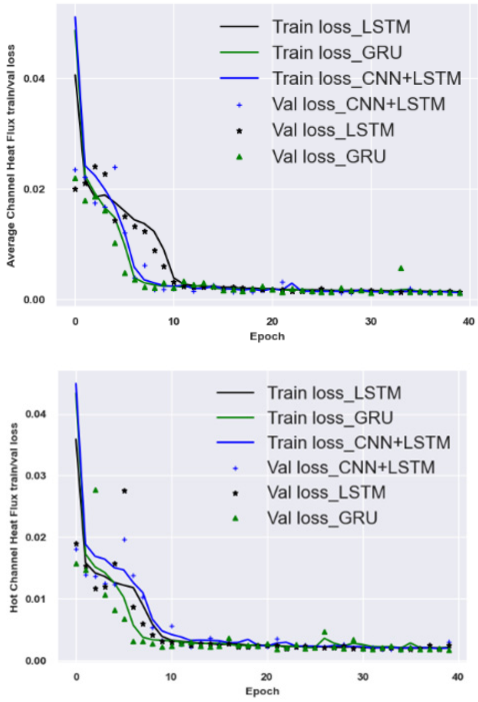
3.8.1. ML Development Result

As explained in section 3.7, the model performance was monitored to ensure that the model learns as expected. Training and Validation loss functions were monitored for the Core Power, Hot Channel Power, Average Core Heat Flux, and Hot Channel Heat Flux, the results are shown in Figure 8.

As shown in Figure 8, the model performed adequately. The loss starts high and drops significantly, then maintains a low loss for training and validation. The model also does not show underfitting or overfitting. As explained in Section 3.7, it is important to monitor the error using more than one metric for model performance as listed in Table 4.

Test data that was kept aside from the BEPU dataset, described in Section 3.7, and deployed to predict the NPP response, as can be seen in Figure 9, the proposed models predicted the NPP response reasonably well.



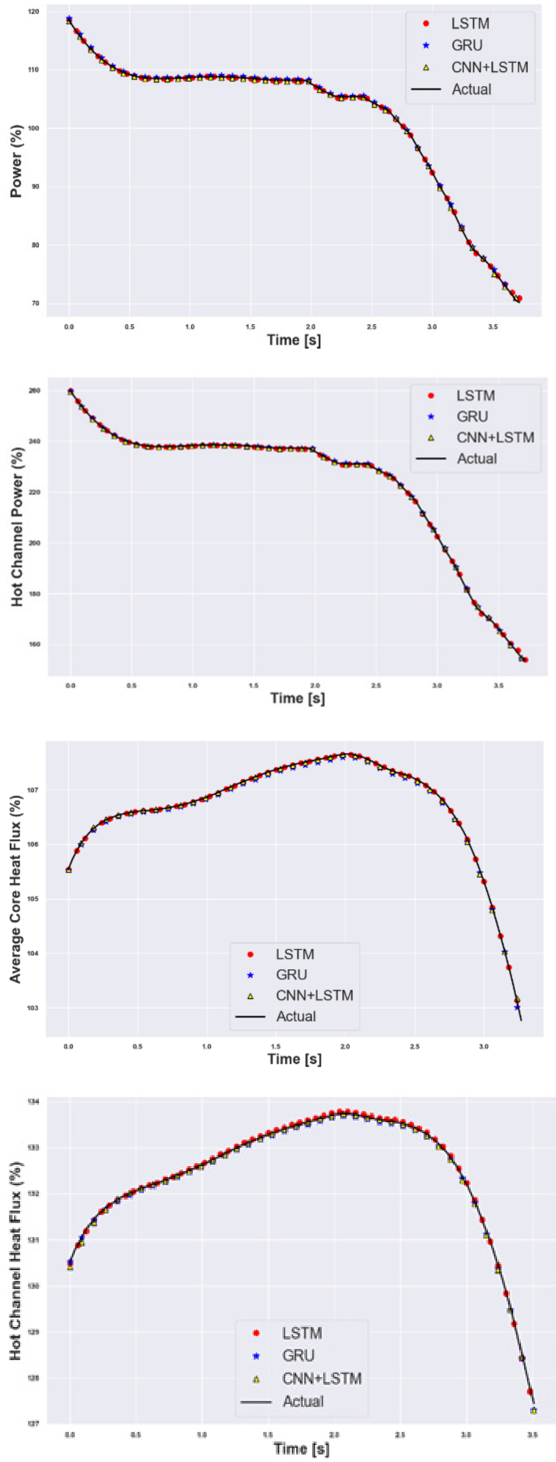


[Figure 8] ML Loss

<Table 4> ML Performance

Parameter	ML Model	MSE	MAE	RMSE	R^2	Accuracy (%)
Average Power	LSTM	0.001	0.008	0.035	0.999	94.50
	GRU	0.003	0.013	0.006	0.997	93.32
	CNN+ LSTM	0.001	0.017	0.032	0.999	91.72
Max. Power	LSTM	0.003	0.010	0.055	0.997	94.08
	GRU	0.003	0.020	0.057	0.997	94.28
	CNN+ LSTM	0.003	0.007	0.053	0.999	94.99
Average Heat Flux	LSTM	0.004	0.006	0.063	0.996	97.38
	GRU	0.004	0.010	0.063	0.996	95.03
	CNN+ LSTM	0.004	0.009	0.062	0.996	93.10
Max. Heat Flux	LSTM	0.003	0.020	0.050	0.997	91.13
	GRU	0.001	0.010	0.033	0.999	94.34
	CNN+ LSTM	0.001	0.010	0.032	0.999	94.17

The LSTM outperformed GRU and the hybrid model of CNN+LSTM for 3 of the 4 parameters by a small margin. The downside of LSTM is slow training while GRU showed the highest error. The computational efficiency of LSTM and GRU were similar. The hybrid CNN + LSTM is deemed the most competent, albeit at a slightly lower accuracy.



[Figure 9] ML Prediction Results

Conclusion

This project used the Systems Engineering approach to guide the development of an ML meta-model capable of predicting the response of APR-1400 NPP under the CEA ejection accident using three RNNs (LSTM, GRU, and CNN+LSTM). The SE method was used from concept to implementation stage.

First a TH model for the CEA ejection accident on APR-1400 was developed, followed by a BEPU analysis, and finally the ML was developed and trained to accurately predict the NPP response.

While beneficial throughout the entire project lifecycle, it is crucial to initiate the process of defining the project's needs, stakeholders, requirements set by stakeholders, components, and expected quality as early as possible. Establishing these aspects in advance ensures the availability of a guiding framework, particularly in moments of confusion. This project demonstrated a systematic process to ensure project quality, and served as a reliable guide, by outlining expectations at each stage of the project.

The SE approach was found to be an invaluable tool for organization, management, and planning and hence ensuring the efficient execution of high quality work within the available resources and imposed timelines.

Acknowledgments

This research was supported by the 2023 Research Fund of KEPCO International Nuclear Graduate School (KINGS), Republic of Korea.

References

1. Ali Agga, Abbou Ahmed, Labbadi Moussa, El Houm Yassine, CNN-LSTM: An Efficient Hybrid Deep Learning Architecture for Predicting Short-Term Photovoltaic Power Production. Electric Power Systems Research, Volume 208, 2022, p. 107908
2. B. E Boyak, A Motta, K Peddicord, Phenomenon Identification and Ranking Tables (PIRTs) for Rod Ejection Accidents in pressurized Water Reactors Containing High Burnup Fuel. NUREG/CR-6742, 2001
3. Sara Buonocore, Di Gironimo Giuseppe, Favaretto Mauro, Grasso Tiberio, Zanon Fabio, and Grazioso Stanislao, Systems engineering approach for the iterative concept design and virtual simulation of the DTT hyper redundant manipulator. Fusion Engineering and Design, Volume 190, 2023, p 113534.
4. F D'Auria, Best Estimate Plus Uncertainty (BEPU): Status and Perspectives, Nuclear Engineering and Design, Issue 352, 2019. p. 110-190.
5. GitHub, Introduction to RNN, 2023. [Available Online and Accessed 01 June 2023 at: https://github.com/cs231n/cs231n.github.io/blob/master/rnn.md.
6. Seola Han and Kim Taewan, Numerical experiments on order statistics method based on Wilks' formula for best estimate plus uncertainty methodology, Journal of environmental management, Volume 235, 2019, p 28-33.
7. Qingyu Huang, Peng Shinian, Deng Jian, Zeng Hui, Zhang Zhuo, Liu Yu, and Yuan

Peng, A review of the application of artificial intelligence to nuclear reactors: Where we are and what's next, 2023, Heliyon.

8. Korea Hydro and Nuclear Power Co., Ltd., APR1400 Design and Control Document Tier 2: Chapter 15 Transient and Accident Analyses, 2018, Revision 3.
9. Alexander Kossiakoff, Steven M. Biemer, Samuel J. Seymour, and David A. Flanigan. Systems engineering principles and practice. John Wiley & Sons, 2020.
10. Magnusson Per Anna-Maria Alvarez-Holston, Katja Ammon, Guido Ledergerber, Marcus Nilsson, David Schrire, Klaus Nissen, and Jonathan Wright, Effects of Zr-hydride distribution of irradiated Zircaloy-2 cladding in RIA-simulating pellet-clad mechanical interaction testing, Nuclear Engineering and Technology, Volume 50(2), 2018, p246-252.
11. Mahmoud, Abd El Rahman, and Aya Diab, A Systems Engineering Approach to Multi-Physics Load Follow Simulation of the Korean APR1400 Nuclear Power Plant, Journal of the Korea Society of Systems Engineering, Volume 16(2), 2020, p 1-15.
12. Marchand, Olivier, Jinzhao Zhang, and Marco Cherubini, Uncertainty and sensitivity analysis in reactivity-initiated accident fuel modeling: synthesis of organization for economic co-operation and development (OECD)/nuclear energy agency (NEA) benchmark on reactivity-initiated accident codes phase-II, Nuclear Engineering and Technology, Volume 50(2), 2018, p280-291.
13. Qian, Gensheng, and Jingquan Liu, Fault diagnosis based on gated recurrent unit network with attention mechanism and transfer learning under few samples in nuclear power plants, Progress in Nuclear Energy, Volume 155, 2023.
14. Ramezani Iman, Naser Vosoughi, Khalil Moshkbar-Bakhshayesh, and Mohammad B. Ghofrani, Evaluation of the performance of different feature selection techniques for identification of NPPs transients using deep learning, Annals of Nuclear Energy, Volume 183, 2023.
15. Shin, Ji Hyeon, Junyong Bae, Jae Min Kim, and Seung Jun Lee, An interpretable convolutional neural network for nuclear power plant abnormal events, Applied Soft Computing, Volume 132, 2023, p109792.
16. Sifat Md Mahmud Hussen, Sajal K. Das, and Safwat Mukarrama Choudhury, Design, development, and optimization of a conceptual framework of digital twin electric grid using systems engineering approach, Electric Power Systems Research, Volume 226, 2024.
17. van Erp Tim, Cecilia Haskins, Wayne Visser, Holger Kohl, and Niels Gorm Maly Rytter, Designing sustainable innovations in manufacturing: A systems engineering approach, Sustainable Production and Consumption, Volume 37, 2023, p 96-111.
18. Nguyen Tran Canh Hai, and Aya Diab, Assessing the response of a typical PWR undergoing a steam generator tube rupture accident using physics-based and data based approaches, Nuclear Engineering and Technology, 2023.