International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 10 | Oct 2024 www.irjet.net p-ISSN: 2395-0072

Development of AI model for Reactivity estimation of NPPs

Rahul Raj[1], P. Parimalam[2]

[1]*Scientific officer C , Indira Gandhi Centre for Atomic Research, Kalpakkam, Tamil Nadu, India*
[2]*Scientific officer G , Head- DDCSS, Indira Gandhi Centre for Atomic Research, Kalpakkam, Tamil Nadu, India*

Abstract - *The reactivity of nuclear power plants (NPPs) is a critical parameter, essential for ensuring the safe and secure operation of these systems. In Pressurized Heavy Water Reactors (PHWRs), reactivity measurement typically relies on the Inverse Point Kinetics Equation (IPKE), a deterministic approach based on predefined mathematical formulations, which can be rigid in practice. This paper presents an alternative methodology using Machine Learning (ML), a subset of Artificial Intelligence (AI). ML models offer a data-driven approach for reactivity estimation, providing greater flexibility by learning from historical data. Several ML models are explored, including Linear Regression, Regression Trees, Random Forest, Support Vector Regression (SVR), and Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks. Simulated datasets of reactivity versus neutron concentration, generated using Point Kinetics Equations (PKEs), are employed to train the ML models. After training, the models are evaluated on testing datasets, and their performance is assessed using Mean Squared Error (MSE), which quantifies the squared deviation between predicted and actual values. A comparative performance analysis based on MSE values identifies the most optimal ML model for accurate reactivity estimation.*

Key Words: **Artificial intelligence (AI), Reactivity estimation, Machine Learning (ML), LSTM, Point Kinetics Equation (PKE), Nuclear Power Plants (NPPs)**.

1. INTRODUCTION

Fast Breeder Reactors (FBRs) are a class of nuclear reactors that produce more fissile material than they consume, making them a critical component of sustainable nuclear energy production. However, the inherently dynamic behavior of FBRs poses significant challenges for reactor control and safety management, necessitating precise and timely reactivity estimation to ensure safe and efficient operation.

Traditionally, reactivity in FBRs is estimated by monitoring power changes through the solution of Inverse Point Kinetics Equations (IPKE), which account for six groups of delayed neutrons. This deterministic approach relies on predefined mathematical models, limiting adaptability in dynamic conditions. In contrast, Machine Learning (ML) models provide a data-driven alternative, offering greater flexibility by learning from historical data trends.

Among ML techniques, Deep Neural Networks (DNNs), particularly Long Short-Term Memory (LSTM) networks, have demonstrated superior performance with lower prediction errors compared to traditional methods. To train these models, simulated datasets of neutron concentration versus reactivity are generated using input reactivity values ranging from +20 pcm to -20 pcm. The datasets incorporate three types of reactivity inputs: step, ramp, and sinusoidal reactivity.

The generated datasets are used to train multiple ML models, and their performance is evaluated through comparative analysis using random test datasets. This study aims to assess the feasibility of ML-based reactivity estimation as a complementary or alternative approach to the conventional IPKE method, providing insights into the potential of ML models for enhancing the safety and control of FBR operations.

AI/ML as an Alternative to Conventional Methods.

Conventional models like IPKE offer a deterministic, physics-based approach to reactivity estimation but are limited by their assumptions, rigidity, and sensitivity to input parameter inaccuracies. They are reliable within well-defined conditions but struggle with dynamic and complex scenarios. AI/ML models such as LSTMs provide a flexible, data-driven alternative that excels in handling time-dependent, non-linear, and noisy data. They learn directly from historical patterns, are more adaptable to changing conditions, and offer robust performance in scenarios where conventional models may fall short.

2. RELATED WORKS

Ma and Bao [1] document a project by Idaho National Laboratory for the Nuclear Regulatory Commission (NRC), investigating the use of advanced computational tools, including artificial intelligence (AI) and machine learning (ML), in nuclear plant operations. The report establishes connections between statistics and AI/ML, highlights commonly used supervised and unsupervised learning algorithms, and explores their applications in enhancing the safety and efficiency of nuclear plants and advanced reactors. Saurabh [2] presents research on applying LSTM-RNN models for stock price prediction, focusing on identifying the most effective optimizers for these models. Suman [3] discusses the complexities of nuclear reactors and the challenges of developing real-time AI models, primarily due to the lack of reliable datasets and the

absence of consensus on optimal AI techniques for specific applications. Mousavi Balgehshiri, Paydar, and Zohuri [4] emphasize that nuclear power plants involve multiple nonlinear systems and propose that integrating AI can improve operator performance, ensuring safer operations and enhancing resilience against natural and man-made disruptions.

3. METHODOLOGY

An efficient solution of the Point Kinetics Equations (PKE) was developed to generate neutron concentration profiles for various reactivity insertions, including step, ramp, and sinusoidal signals. The process involved pre-processing and feature engineering of the power series data to enhance the quality of inputs for training machine learning (ML) models. The generated neutron concentration versus reactivity data was used to train multiple ML models, which were subsequently tested with random samples from the datasets to assess their performance. A comparative analysis of the models' effectiveness was conducted to identify the optimal approach for accurate reactivity estimation.

3.1 System Design

The primary requirement for training ML systems is the availability of suitable datasets. In this study, simulated datasets of neutron concentration versus reactivity were generated, as real plant data was not used. These synthetic datasets served as labelled data, enabling the use of supervised ML models. The selection of appropriate supervised models plays a critical role in achieving reliable predictions. In this work, the following models were employed: Linear Regression, Regression Tree, Random Forest, and Support Vector Regression (SVR) [1]. Additionally, a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) layers [2] was implemented as part of the deep learning approach. The ADAM optimizer [2] was used to optimize the performance of the RNN-LSTM model.

3.2 Implementation of PKE

The Point Kinetics Equation (PKE) [6] is employed to generate neutron concentration versus reactivity datasets, which serve as the foundation for training machine learning models. These datasets are produced by implementing custom Python code. The PKE comprises a system of differential equations. To numerically solve these equations, the *odeint* function from the *scipy.integrate* library is utilized. The following differential equations, which constitute the PKE model, are applied to simulate neutron concentration as a function of reactivity:

$$\frac{dn}{dt} = k.(\rho-\beta).\frac{n}{l} + \sum_{i=1}^{6} \lambda_i.C_i$$

$$\frac{dCi}{dt} = k.\beta_i.\frac{n}{l} - \lambda_i.C_i$$

where:

- n(t) is the neutron population.
- ρ(t) is the reactivity.
- β is the delayed neutron fraction.
- k is neutron multiplication factor.
- l is prompt neutron lifetime
- λ_i is the decay constant for the i-th group of delayed neutron precursors.
- $C_i(t)$ is the population of the i-th group of delayed neutron precursors.
- βi is the delayed neutron fraction for the i-th group of precursors.

3.3 Generation of training data sets

The datasets are generated with neutron concentration versus reactivity values, sampled at 10 ms intervals. Due to the time-dependent nature of the Point Kinetics Equation (PKE), it is utilized for generating these training datasets. These datasets are subsequently used to train the system with various Machine Learning (ML) models, facilitating data-driven reactivity estimation.

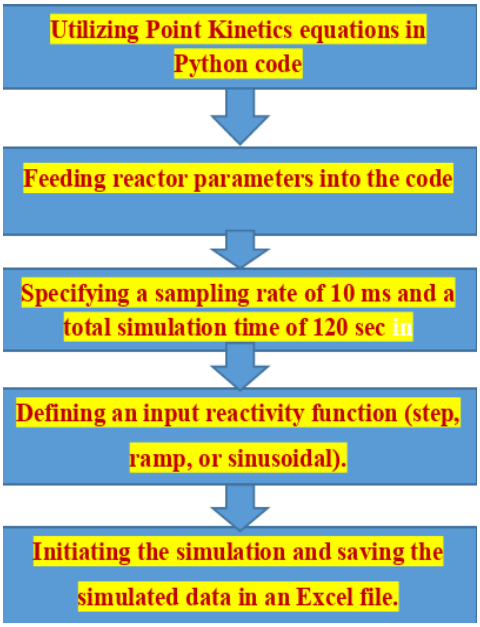

Figure 1. Flowchart represents data set generation process

Figure 1 illustrates the steps involved in the process of generating the training datasets. The types of synthetic reactivity inputs employed are ramp reactivity with varying slopes, step reactivity with different step values and sinusoidal reactivity with varying time periods.

Reason for using the synthetic input signals:
Step Reactivity: Sudden changes in reactivity, simulating abrupt disturbances or control rod movements.
Ramp Reactivity: Gradual changes in reactivity over time, simulating slow control rod adjustments or changes in moderator temperature.
Sinusoidal Reactivity: Periodic changes in reactivity, simulating oscillations in reactor conditions or coolant flow variations.

Figure 2 provides a glimpse of the generated neutron concentration vs. reactivity data sets. It contains data corresponding to sinusoidal input reactivity only, with similar datasets generated for step and ramp reactivity as well. The reactivity values are fixed for the step reactivity case but vary for the ramp and sinusoidal reactivity cases. In this figure the reactivity is the target variable predicted in the study and neutron population, also called neutron concentration acts as the primary feature in the dataset that models the time-dependent response of the reactor to changes in reactivity.

(T= 6 sec)	20 pcm range	(T= 20 sec)	20 pcm range
Reactivity	N-population	Reactivity	N-population
0	1	0	1
0.209453136	0.995808866	0.209453136	0.995808866
0.4188833	0.99166069	0.4188833	0.99166069
0.62826752	0.98755477	0.62826752	0.98755477
0.837582832	0.98349042	0.837582832	0.98349042
1.046806279	0.979466972	1.046806279	0.979466972
1.255914912	0.975483773	1.255914912	0.975483773
1.464885798	0.971540186	1.464885798	0.971540186
1.673696015	0.967635588	1.673696015	0.967635588
1.882322662	0.963769371	1.882322662	0.963769371
2.090742857	0.959940942	2.090742857	0.959940942
2.29893374	0.95614972	2.29893374	0.95614972
2.506872476	0.952395138	2.506872476	0.952395138
2.714536261	0.948676642	2.714536261	0.948676642
2.921902316	0.944993691	2.921902316	0.944993691
3.128947898	0.941345753	3.128947898	0.941345753
3.335650299	0.937732312	3.335650299	0.937732312
3.541986847	0.93415286	3.541986847	0.93415286
3.747934912	0.930606903	3.747934912	0.930606903
3.953471904	0.927093955	3.953471904	0.927093955
4.158575282	0.923613541	4.158575282	0.923613541
4.363222549	0.920165197	4.363222549	0.920165197
4.56739126	0.916748468	4.56739126	0.916748468

Figure 2. Glimpse of generated datasets

Several fixed neutronics parameter values are used for generating the training data sets, as listed below:
Delayed Neutron Fraction (β): This value is used within the Point Kinetics Equation to model how the delayed neutrons contribute to the overall neutron population.
Prompt Neutron Lifetime (l): This parameter is part of the differential equations used to generate the synthetic neutron concentration
Decay Constants of Delayed Neutron Precursors (λi): The decay constants are utilized in the Point Kinetics Equation to model the time-dependent behaviour of delayed neutron precursors, affecting the neutron population.
Population of Delayed Neutron Precursors (Ci(t)): This is also part of the Point Kinetics Equation that helps generate the synthetic dataset.

4. EXPERIMENTAL SETUP

The experiments were conducted on a CPU with an Intel® Xeon® processor operating at a clock frequency of 2.5 GHz and 4 GB of RAM. The system ran on a 64-bit Windows 10 Pro operating system. The project was implemented in Spyder (version 5) using Python 3.11. Both the dataset generation and the training of ML models were performed within the Spyder environment using Python.

The generated datasets were sampled at 10 ms intervals for a total duration of two minutes, resulting in 12,000 data points for each type of reactivity input. The experiment included step reactivity with 40 different values (ranging from -20 pcm to +20 pcm, excluding 0 pcm), ramp reactivity with three slopes (0.5, 1.0, and 2 pcm/sec), and sinusoidal reactivity with two time periods (20 seconds and 60 seconds). In total, the datasets contained 300,000 data points.

Table 1. Input output relation of training data sets

S.no	Neutron Concentration (Ni)	Reactivity (Ri)
1	N1	R1
2	N2	R2
.....	.....	.....
.....	.....	.....
99	N99	R99
100	N100	R100

The datasets were structured such that every 100 neutron concentration values corresponded to a single reactivity value, with the reactivity value matching the 100th neutron concentration point. This arrangement facilitated efficient mapping between input and output for model training. Table 1 illustrates the input-output relationship between the neutron concentration and reactivity values in the dataset.

5. TRAINING USING DIFFERENT ML MODELS

Machine Learning (ML), a branch of artificial intelligence [5], leverages datasets to develop algorithms that identify underlying patterns. These algorithms can make predictions on new, similar data without the need for explicit programming for each task. In this study, various ML models are employed for training, including Linear Regression, Regression Tree, Support Vector Regressor (SVR), and Random Forest. Additionally, the Long Short-Term Memory (LSTM) model, a type of deep neural network, is used to enhance performance. The selection of these models is informed by a literature review of prior research studies.

The ML models are implemented using Python programming. Upon completing the training, the Mean Squared Error (MSE) metric is used to evaluate the system's performance. The MSE measures the deviation between predicted and actual values, with higher MSE values indicating lower prediction accuracy, thereby deeming the model unsuitable for practical use. An 80:20 split is applied to the dataset, where 80% is allocated for training and 20% for testing.

After training, the system undergoes testing with random datasets of normalized neutron flux. These datasets serve as input to the trained models to predict reactivity values. The predicted reactivity values are then compared with the actual reactivity values (the corresponding values of the input neutron flux) to assess the model's accuracy.

6. TESTING OF TRAINED ML MODELS

Once training is complete, the trained system undergoes testing using random data sets of normalized neutron flux. These data sets are inputted into the system to predict the output, known as predicted reactivity values. The reactivity values corresponding to the input normalized neutron flux, used for testing, are referred to as actual reactivity values. For testing purposes, random data sets are selected from the sinusoidal column of the generated data sets, where the time period is 20 seconds. The sinusoidal signal is chosen due to its curved path, which differs from the fixed path of step signals or the fixed slope path of ramp signals.

7. RESULTS AND DISCUSSION

The performance of the trained ML models is analyzed based on the results from the test datasets and the MSE values. Figure 3 presents a graph illustrating the deviations between predicted and actual reactivity values for the different ML models.In Figure 3 blue line is for actual value of reactivity with respect to time and rest other line are of predicted values trained at different ML models. Only LSTM (purple line) and Regression Tree (Yellow line) traces the actual value line properly.

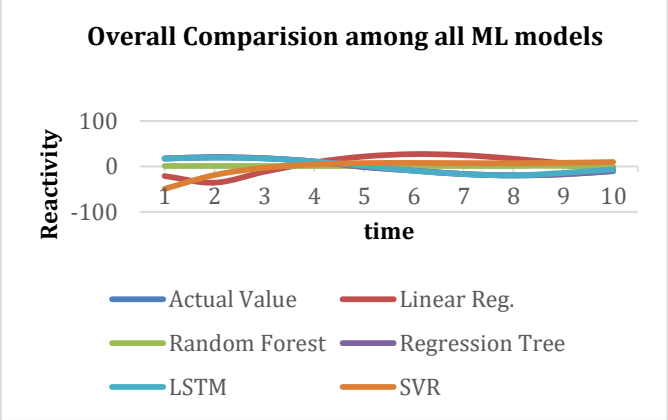

Figure 3. Test output of different ML models under no noise conditions

Table 2 summarizes the overall MSE values for each ML model, with a corresponding graphical representation provided in Figure 4 . In all the five models, only LSTM has the minimum MSE value which is 0.665.

Table 2. MSE values of different ML models

S. No	Ml Model	MSE
1	Linear Regression	24.974
2	Regression Tree	17.215
3	Random forest	40.379
4	Long Short Term Memory (LSTM)	0.665
5	Support Vector Regressor (SVR)	59.393

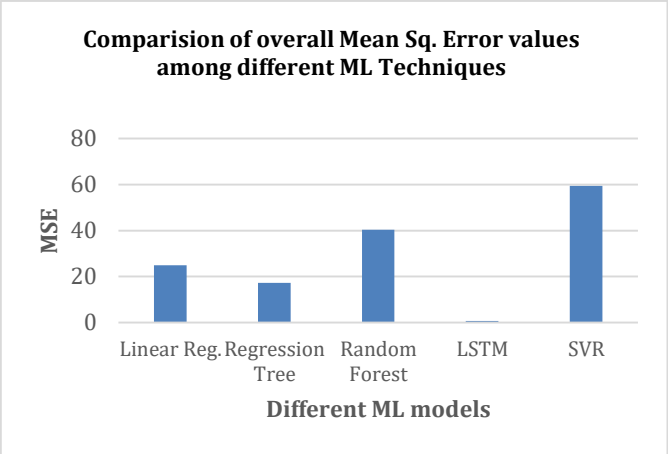

Figure 4. Bar graph of MSE values of different ML models

8. CONCLUSIONS

Both the LSTM and Regression Tree models closely follow the actual values (as shown in Figure 3). However, when considering MSE values, the LSTM model demonstrates superior accuracy, achieving an MSE of 0.665 (Table 2). The Regression Tree model comes in second, but with a

significantly higher MSE of 17.215 (Table 2). Since Figure 3 is based on a few data points, whereas the MSE is calculated from 20% of the training dataset, the results from Table 2 (or Figure 4) are deemed more reliable. Therefore, the LSTM model emerges as the more dependable choice for predicting reactivity.

Suitability of LSTM for Reactivity estimation:

LSTMs are capable of learning complex temporal patterns due to their architecture, which includes mechanisms to retain relevant information and forget irrelevant details over time. This capability is essential for accurately estimating reactivity, which can be influenced by various factors like sudden changes in neutron flux.

Linear Regression(LR) assumes a linear relationship between the input features and the target variable (reactivity).

Regression Trees(RT), make predictions by dividing the data into branches based on feature values. However, they are prone to over fitting on training data and fail to effectively model time-dependent or sequential data like changes in reactivity but more flexible than LR.

Random Forest(RF) aggregates the predictions of multiple regression trees, improving accuracy and reducing over fitting. However RF treats each input instance independently and does not consider sequential relationships, which results in poorer performance for time-dependent problems like reactivity estimation.

SVR is primarily designed to find a hyper plane that best fits the data points. However, it lacks mechanisms to understand or remember sequential patterns and dependencies that are crucial in time-series problems and it is very sensitivity to noise and is effective for many regression problems. Reactivity estimation involves subtle, time-dependent variations that SVR finds hard to model effectively.

Although traditional RNNs can model sequences, they have limitations in remembering long-term dependencies, which is crucial in modeling reactivity changes over time. LSTM networks, on the other hand, overcome this with specialized units (gates) designed to retain relevant information over extended periods and LSTMs have a well-suited architecture for time-series data, enabling them to capture long-term dependencies and effectively model non-linear patterns in dynamic systems like nuclear reactors.

REFERENCES

[1] Z. Ma, H. Bao "Exploring Advanced Computational Tools and Techniques with Artificial Intelligence and Machine Learning in Operating Nuclear Plants". Idaho National Lab. (INL), Idaho Falls, ID (United States), 2022.

[2] Nikitha Saurabh "LSTM -RNN Model to Predict Future Stock Prices using an Efficient Optimizer." International Research Journal of Engineering and Technology (IRJET) 2020, pno. 672-677 .

[3] Siddharth Suman "Artificial intelligence in nuclear industry: Chimera or solution?". Journal of Cleaner Production, 2021.

[4] Seyed Kamal Mousavi Balgehshiri, Ali Zamani Paydar and Bahman Zohuri "Artificial Intelligence Driven Nuclear Power Reactors ", Journal of Energy and Power Engineering 16 (2022).

[5] Textbook, "Introduction to Machine learning with Python" - 1st Edition (Author- Andreas C. Müller & Sarah Guido)

[6] Textbook, "Nuclear Reactor Engineering" - 4th Edition (Author- Samuel Glasstone & Alexander Sesonske)