A digital twin of the AGN-201 reactor to simulate nuclear proliferation

Ryan Stewart, Ashley Shields, Chad Pope, John Darrington, Katherine Wilsdon, Samuel Bays, Kolton Heaps, Nathan Woodruff, Jonathan Scott, Gustavo Reyes, Mark Schanfein, Eduardo Trevino, Jaden Palmer, Christopher Ritter

May 13, 2023

Abstract

Nonproliferation organizations must understand the potential proliferation pathways in a facility that could lead to weaponizable nuclear material. The International Atomic Energy Agency (IAEA) provides a source of confidence in the safe growth in global nuclear power through its assurance that states are continuing to abide by their international commitments while dampening the risk of undetected instances of proliferation. To succeed in this critical endeavor, the physics, design features, and proliferation indicators must be understood, and proliferation pathways mitigated.

This requires a fundamental understanding of all aspects of an operating facility at the design level to conduct a diversion pathway analysis. Modeling breakthroughs, along with emerging technologies, offer a unique opportunity to develop digital twins (DTs) to accurately perform a diversion pathway analysis. Indicators can then be analyzed using artificial intelligence and machine learning engines, and the same DT can be turned into a real-time monitoring system. In the end, a DT is an important approach for nonproliferation knowledge retention and remains a foundation capability that any organization can maintain and expand to other nuclear fuel cycles over time.

This paper focuses on the development of a DT of the Idaho State University AGN-201 reactor. The DT will be fed real-time data streams from an operating facility to detect early indicators in a timely manner so countermeasures can be taken. The AGN-201 virtual testbed will generate data representing activities at the physical testbed. This data will be used for algorithm development, virtual replay of campaigns, and prospective inspector training. It will also indicate which data feeds, both nuclear and non-nuclear, are most practical for successfully detecting errant reactor facility behavior. This will be a critical capability as the IAEA currently safeguards over 200 reactors around the world, a number expected to grow substantially in the near future. As it stands, 50 additional member states have expressed interest in pursuing nuclear power and roughly half of those states are in the pre-nuclear planning stage. If development of a digital twin is performed during the reactor design phase, the result can be the inclusion of timely safeguards by design features to ensure effective and efficient safeguards.

Demonstrating a DT of a real reactor is the natural technology maturation stepping stone to a high-power reactor DT. Basic nuclear behaviors are similar at 1 W to behaviors at 1,000 MW. The basic process, previously demonstrated with a virtual DT of a sodium-cooled fast reactor, and the AGN-201 physical twin is highly tractable to a much larger scale for real-world reactors.

1 Introduction

Digital engineering is defined by the United States Department of Defense as “an integrated digital approach that uses authoritative sources of system data and models as a continuum across disciplines to support lifecycle activities from concept through disposal” [1]. Essentially, this definition requires a series of dynamic and integrated models to describe a product, rather than static documents. These models are integrated across various platforms to help produce a design product that can be supported across its lifetime [2, 3]. Digital twins (DTs) utilize the concept of digital engineering to comprise a model representative of either a current physical product or the design of a future physical product [3]. This allows for both the creation of a virtual DT (i.e., a DT mimicking the behavior of a physical product by means of modeling and simulation) or a physical system DT (i.e., a DT containing both a physical asset and computational models representing that asset).

Idaho National Laboratory is currently developing a nuclear safeguards digital twin (SG-DT) [4–7]. Given the potential benefits of utilizing digital engineering (and DTs specifically), researchers developed a SG-DT to examine how a DT could be used to detect diversion and misuse. This virtual SG-DT was developed to model, predict, and analyse the behavior of a nuclear reactor. The SG-DT consists of five individual components: the graphical user interface, physics models, the optimization algorithm, the data warehouse, and the machine learning (ML) adapter. The graphical user interface provides direct access to the core design, where users (i.e., safeguards inspectors) can select where they expect target positions to be placed or diversion to occur. The determination of appropriate core designs allows domain experts to directly inputs their knowledge to help reduce the design space examined by the optimization algorithm and ML adapter. The physics models consist of high-fidelity Serpent simulations to explicitly model the behavior of each reactor type.

A custom optimization algorithm (the multi-agent blackboard system) for handling large numbers of high-fidelity simulations was developed to analyze thousands of potential core designs. Along with this, the data produced by the optimization algorithm doubles as a training set for the ML adapter and provides a wide sampling of the design space, including an emphasis on optimal designs, which are able to successfully proliferate under the design constraints. The data warehouse utilizes a unique ontology for nuclear facilities, which allows for a common nomenclature to easily pass information between the physics engine, ML adapter, and future live data streams. The ML adapter currently utilizes data from the optimization algorithm to predict the potential for diversion and reactor misuse. It is not meant to replace a safeguards inspector but instead rapidly provide valuable information to help determine where sensors should be placed. The final aspect of the SG-DT is the ability to utilize both cloud and high-performance computing (HPCs). A majority of the SG-DT is housed on the cloud and communicates with Idaho National Laboratories HPC to perform the optimization algorithm and physics solves. This provides the SG-DT with the computing power of HPCs and the versatility and maneuverability of the cloud.

1.1 AGN-201

The AGN-201 is a small 5.0 W nuclear reactor at Idaho State University, developed for both research and operational training [8]. It consists of two major parts: the core region and the ex-core components. The core region is composed of nine fuel disks of polyethylene homogenously mixed with uranium dioxide fuel (enriched to 19.9% U-235). The rough dimensions for the core are about 24 cm tall and 25.6 cm in diameter [9]. The core is split into two parts and held together with a thermal fuse, which is polystyrene with a fuel loading twice that of the fuel disks. The thermal fuse acts as a safety mechanism that will melt and split the core into two pieces, thereby exterminating the chain reaction if the temperature in the thermal fuse reaches approximately 100°F.

To maintain criticality, four fueled control rods are inserted into the core. Two safety rods are always fully inserted, and a coarse control rod (CCR) and fine control rod (FCR) are used to increase, decrease, or maintain power. A central irradiation facility passes through the center of the core, allowing experiments to be inserted directly into the center of the core.

Surrounding the core is a graphite reflector contained within an aluminum core tank. Surrounding the core tank is an additional graphite reflector, followed by a lead shield, and finally a water shield. Above the core sits a graphite thermal column that can be removed for experiments; below the core sits the control rod drive mechanisms. The full AGN-201 configuration can be seen in Figure 1.

2 Methodology

2.1 AGN-201 Reactor Physics Model

For the AGN-201 Serpent model, material and geometric characteristics of the core were taken from the safety analysis report and from previous benchmarking performed for various AGN-201 reactors [9–11]. When possible, geometric information was obtained from the safety analysis report first, which is then supplemented with additional benchmarking data. All material compositions were obtained from [10]. Impurities in the

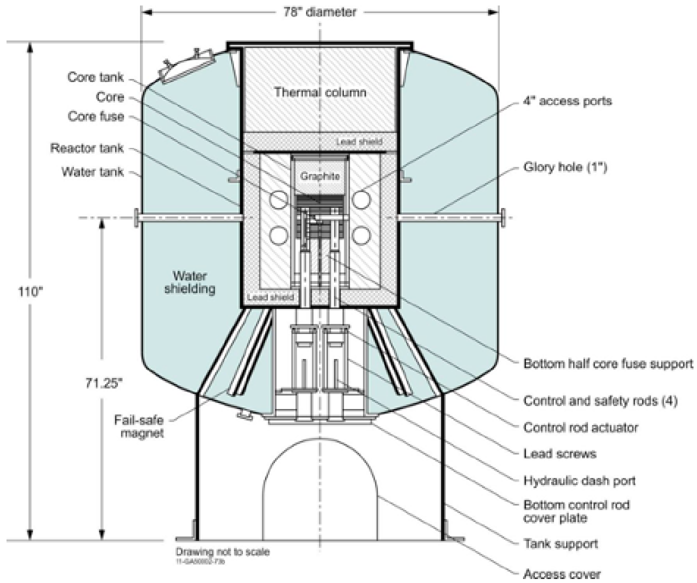


Figure 1: AGN-210 reactor, reflector, and shielding configuration [8].

UO_2-poly material were obtained from mass spectroscopy of two fuel shaving samples; however, the mass of uranium was based on material inventory paperwork [9, 10]. Other materials (lead, graphite, etc.) were obtained via best estimates based on historical data.

2.2 Physics-Based Surrogate Models

The surrogate model (SM) generation was broken into two parts: the mathematical SM of the reactor physics and the point kinetic equations surrogate model (PKE-SM). The overall flow of data from the AGN-201 through the SM is in 2. The SM is informed of the reactor state through hundreds of Serpent calculations performed during normal and off-normal operations. Serpent utilizes the water temperature, FCR height, and CCR height to determine the k_{eff} (i.e., the reactor state—critical, subcritical, supercritical). These data are passed to a model (polynomial regression, support vector machine, etc.) to create the mathematical SM relating the input and output variables [12]. During live operations, the SM will take the water temperature, FCR height, and CCR height at each time step to determine the k_{eff}. This information can then determine if the reactor state matches the predicted state. For example, during normal operations, if the FCR and CCR heights are within an acceptable band of uncertainty (based on the temperature), the reactor state would show a critical reactor. During an incident where an experiment was inserted into the core, the FCR and CCR heights would have to be adjusted to compensate for the gain or loss of reactivity. Without a priori knowledge of the insertion event, the reactor would likely indicate a super- or subcritical system based on the given FCR and CCR heights. The SM provides an understanding of the reactor state in snapshots, so each time step is independent of the previous time steps.

Along with the mathematical SM, we are generating a secondary SM. This SM utilizes data obtained from Serpent and is combined with the PKEs to track reactor power as a function of time. Within the SM, there are eight key pieces of information passed from the physical asset (AGN-201) to the SM to assess the change of power in the physical system. Live data from AGN-201 will be passed to both the mathematical SM and PKE. The water temperature, FCR position, and CCR position are passed to the mathematical SM to determine the k_{eff} at each step. Each state consists of roughly 0.1 second of live operation time. The reactivity change from the mathematical SM is passed to the PKEs along with the initial reactor power from AGN-201. Given the reactivity change and previous (or initial) power, the PKE-SM solves the point kinetics equations to determine the power based on the reactivity. Aspects that might affect the reactor, including the declared experiment insertion or control rod movement, would manifest themselves in the SM outputs, where undeclared events would cause a deviation in the expected outputs.

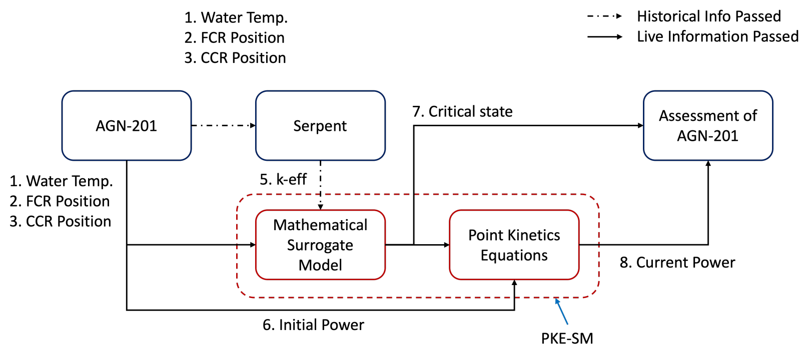


Figure 2: Flow of data from the AGN-201 through the mathematical SM and PKE-SM.

2.3 Machine Learning Model

Our ML approach is effectively an automated QA-QC along with methods exploration. This strategy involves model development for the CCRs, FCRs, Channel 2 power (Watts), and Channel 3 power (Watts) as a response in the data using all variables except for the target response as predictors, shown in Table 1 and including: Channel 1 (Ch1, CPS), Channel 2 (Ch2, W), Channel 3 (Ch3, W), Fine Control Rod Heights (FCR, cm), Coarse Control Rod heights (CCR, cm), Temperature (Temp, °C), and Inverse Period (Inv-P, sec). Three model types have been implemented in this work, specifically Multilayer Perceptron (MLP) Regression [13], GrowNet [14], and TabNet [15]. Each of these methods implements a different approach to tabular data analysis. While MLP Regression is a classic neural network approach applied here as a baseline technique, GrowNet and TabNet represent leading edge techniques in the domain of tabular data analysis. GrowNet [14] is a gradient boosting technique that incrementally builds complex deep neural networks, this allows shallow neural networks as "weak learners" to sequentially construct a powerful, higher-order model for general loss functions. Along with this, GrowNet incorporates a fully corrective step, which updates the parameters of all weak learners rather than just the current one. TabNet [15] is a sequential attention based technique for deep learning that identifies connections between individual variables and global characteristics to focus attention on variables with the greatest importance.

2.4 Digital Twin Integration

The AGN-201 DT has many moving parts, as discussed in previous sections. A goal of all DT construction is "gluing" those parts together into a single, cohesive whole. We also expect the whole AGN-201 DT can be deployed and used quickly. To accomplish this task, we utilized an existing data warehouse software, DeepLynx, and wrote custom software to integrate AGN-201 with DeepLynx and the software driving the ML/AI portion of this project. We will briefly touch on each integration piece and how they make up the whole. A general overview of DeepLynx is show in Figure 3.

DeepLynx

DeepLynx is an ontological data warehouse specializing in storing data for DTs [16]. Not only can we store a graph representing the actual reactor, its requirements, and other data—we can also store the raw data flowing in from various sensors and collectors in the same space. This combination of metadata and tabular sensor data is stored securely in the cloud using a DeepLynx instance running in Azure Government. The data is available near real time from the reactor and can be returned via simple queries or more complex requests for data, such as the ML/AI software requires.

Jester

To store data in DeepLynx we must first access and preprocess this data. We use a publicly available software called "Jester" to retrieve this data from the data acquisition system (DAS) and send it to the DeepLynx

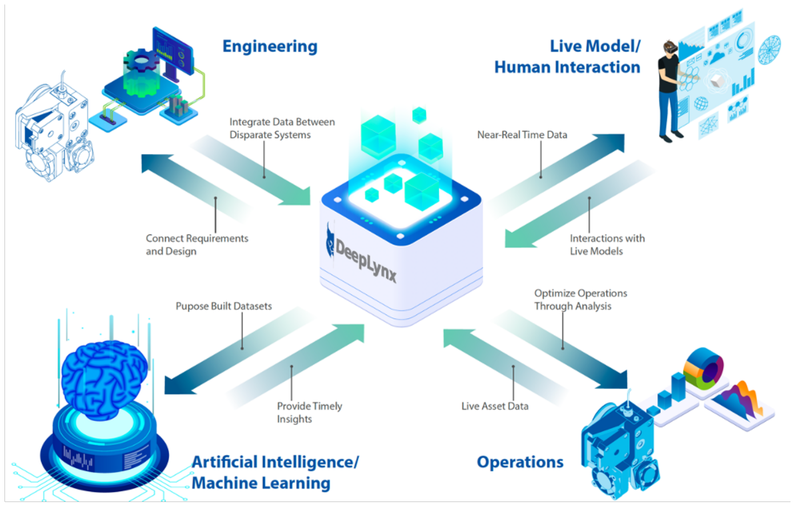


Figure 3: General DeeyLynx capabilities for the generation of DTs.

cloud instance. This software runs on the same machine the data is recorded to and is constantly relaying that data to DeepLynx in near real time. This software was written in Rust and was made publicly available on the Idaho National Laboratory GitHub.

DeepLynx-loader

Once data has been retrieved and stored in the cloud it must become readily available to the ML/AI software that requires it—in near real time. A custom Rust-Python module allows users to quickly retrieve their tabular or timeseries data from DeepLynx and use it as they see fit. With a direct integration to popular data manipulation libraries like pandas, we were able to develop ML integrations that operate in near real time with data from AGN-201. This software is currently undergoing an internal software disclosure process and will be released under an open-source license as soon as possible.

Integration with AGN-201

Our goal with the software and integrations developed to support AGN-201 was to create a readily available framework that could be quickly adapted for other DT efforts. At time of writing, we have made great progress in developing a set of tools, procedures, and integrations that could be adapted to suit various DT needs. With all software being developed and used in this framework being published under open-source licensing, we expect to need only to provide sufficient documentation and examples for other projects to duplicate our integration results. We have successfully built an automated monitoring loop with a live reactor and have laid the groundwork for more complex tools and visualization to be added as the project progresses.

3 Results

3.1 Reactor Physics Model and Physics-Informed Surrogate Model

The first set of data is the historical critical control rod heights. From the historical data, 30 critical configurations that had no additional changes to the reactor configuration were selected to describe the estimated k_{eff} value for a critical core. Historical critical control rod heights provide a baseline for the critical core and allow us to examine the variance in our data to capture any bias between the physical core and our reactor physics model. We expect variance in the historical start up data will be larger than any variance in the DAS data due to the addition of a human factor for determining the temperature and rod heights. Figure 4 shows the k_{eff} values for each of the available startups. Each dot represents a Serpent calculation, where the statistical uncertainty for each k_{eff} value was less than 2E-5. The mean k_{eff} is roughly 1.00772 with a standard deviation of 1.81E-4 (i.e., 36 cents).

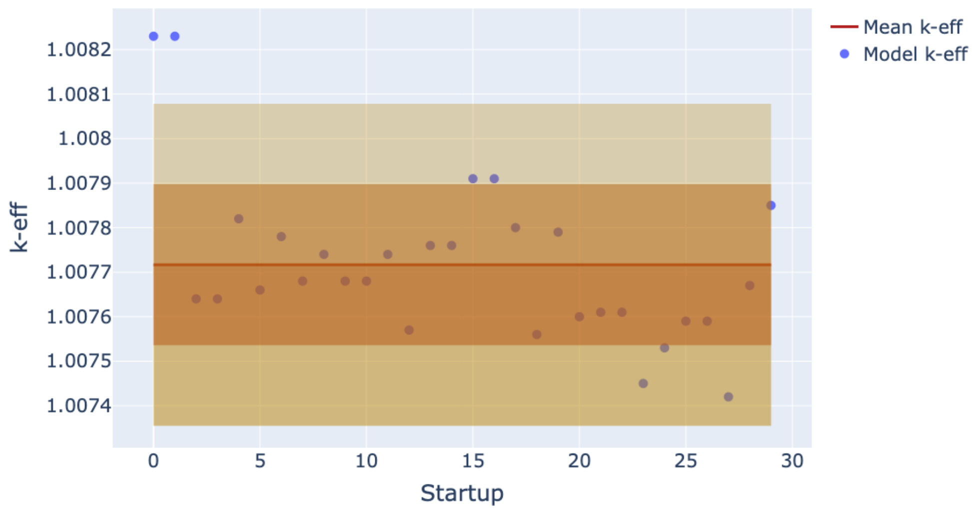


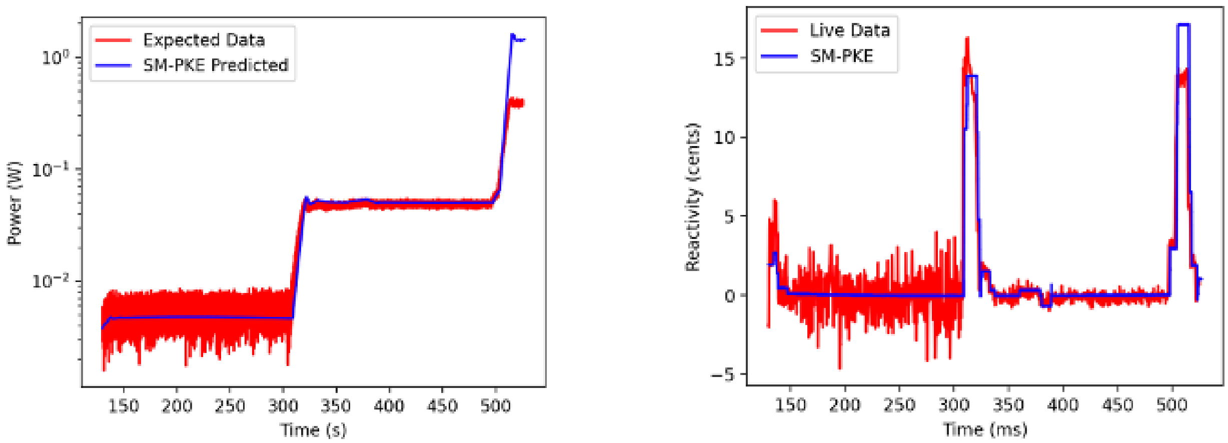
Figure 4: Historical k_{eff} for critical cores during reactor start up.

The mathematical SM captures snapshots in time, where the reactivity difference between an expected k_{eff} value and a given k_{eff} value, can be used to determine the reactor state. This provides the first of two SMs to determine if off-normal conditions are occurring within the core. The second method combines the mathematical SM with the PKE to capture the time dependence of the core. This method utilizes the mathematical SM to determine the k_{eff} at each time step during operations (i.e., every tenth of a second). This information is used to determine the change in reactivity between the current and previous time steps. The change in reactivity is then used to calculate the power at the current time step, where we can track power as a function of operational time.

To demonstrate this, we examined a segmented portion of the AGN-201 operations from February 10, 2023. Data from the AGN-201 operations are stored in a csv file, and the CCR height, FCR height, and temperature are extracted at each time step and fed into the mathematical SM. The mathematical SM provides the k_{eff} for each time step, which is used to calculate the reactivity change and resulting reactor power; these can then be plotted to show the comparison with live data.

Figure 5 shows the AGN-201 power and reactivity insertion as a function of operations on February 10th, where both reactor data (in red) and PKE-SM generated data (in blue) are presented. For the entirety of the operation, the PKE-SM provides an accurate trend of the power; however, the power magnitude tends to degrade after approximately 500 seconds during the last power ramp.

Figure 5b shows that the PKE-SM predicts a larger reactivity insertion, which results in a subsequently larger power. The overprediction is due to two major effects: changes in the reactor kinetics parameters and sensor drift and bias. For reactor kinetics parameters, minor changes can have drastic effects on reactivity insertion or removal, resulting in an accurate description of the power trend but a failure to capture the true power. The DAS provides the CCR and FCR heights, which are used in the reactor physics models to provide changes in reactivity. During operation the DAS shows sensor drift as a function of time for both the CCR and FCR, where the CCR and FCR heights tend to decrease over time. The resulting decrease in CCR and FCR heights is handled by adding a simple bias; however, a more robust model is needed to provide an accurate SM. An uncertainty quantification and reduction is currently being performed by the authors to capture these effects and provide an appropriate band for predicting reactor power.



(a) Power during operation

(b) Reactivity during operation

Figure 5: PKE-SM results for the sample operation.

3.2 Machine Learning Models

Our goal in this work is to identify which methods are most appropriate for final implementation by assessing the pros and cons of each method in terms of accuracy, access, development, efficiency, and transfer learning approaches. In the future we will also investigate explainability and interpretability, assess model performance on live data streams, and identify strategies for model verification.

The MLP Regression model development is one of the most straightforward neural networks to implement. It is a very well documented tool, open source, and easily accessed via scikit-learn. Multiple MLP Regression models resulted in the following R^2 values: 0.995 for Channel 2, 0.993 for Channel 3, 0.960 for Fine Control Rod Heights, and 0.870 for Coarse Control Rod Heights. These characteristics, along with the model requiring few lines of code and very little (if any) parameter tuning to develop and implement a high-quality model, make this method a great point of comparison as we explore leading-edge techniques. This model type is also very simple to transfer for efficient implementation via saving and loading the model as a pickle file. All of these characteristics make this tool a perfect first pass to understand whether or not a data set is appropriate for solving a particular problem. The disadvantage of this method is that it allows for very little parameter tuning and so the model, and so there isn't much that can be done to tailor the model beyond preprocessing and minor parameter adjustments. Overall, despite its simplicity, MLP Regression is a preeminent tool for preliminary analysis and variable exploration because of its relatively fast performance and straightforward implementation. It's also an excellent tool to implement during overall code framework development (e.g., data input, preprocessing, variable management, outputs), because its small input can easily be exchanged for more complex methods, such as GrowNet or TabNet.

GrowNet was selected for this work to assess how its ability to incrementally develop complex neural networks impacts its ability to draw conclusions from this relatively simple real world data set. The Grownet models resulted in the following preliminary R^2 values: 0.935 for Channel 2, 0.931 for Channel 3, 0.937 for Fine Control Rod Heights, and 0.915 for Coarse Control Rod Heights. Developing GrowNet involves selecting the optimal number of hidden layers and units for weak learners, as adding more layers may lead to stronger predictors but may also cause overfitting. Effectively incorporating the corrective step is crucial for addressing potential local minima and correlation among weak learners, as well as dynamically adjusting the boosting rate for better performance. The original paper provides comprehensive documentation and resources, which allow GrowNet to be adapted for a diverse range of ML tasks, including classification, regression, and learning to rank. With numerous hyperparameters available for tuning, GrowNet can be customized to specific data sets and applications. However, it may not be the optimal approach for tasks outside its primary domains. GrowNet's versatile and flexible framework, combined with its incorporation of second-order statistics, corrective steps, and dynamic boost rates, offers a robust solution for a variety of ML problems under a unified structure, addressing the shortcomings of traditional gradient boosting decision trees.

TabNet is an advanced neural network model creating a PyTorch neural network architecture for tabular data. There are many ways to develop a TabNet model, and the implementation can get quite complicated depending on the complexity and architecture of the desired model. There are no access restrictions that this model poses as the library containing it can be easily installed with the same process one would use for scikit-learn or PyTorch. There is plentiful documentation of the scikit-learn implementation of TabNet, including pretraining examples, defaults parameters and their purposes, and explanations for each TabNet model. One downfall of this model is the implementation of other PyTorch packages, such as PyTorch Optimizers other that the default “Adam” optimizer. This is due to the fact that the TabNet model does not require the data to be in tensor form for implementation and therefore will not operate with optimizers that require the data to be in this form. However, overall the TabNet regression model proves to be a very useful tool in terms of solving this ML problem. By using decision-based tree learning in its neural network architecture, TabNet boasts a high accuracy for this data set that is yet to be further improved through hyperparameter tuning and feature engineering.

Table 1: Model Descriptions

Predictor (y)	Response (x)	MLP R^2	GrowNet R^2	TabNet R^2
Channel 2 (Watts)	Ch1 (cps), Ch3 (W), FCR (cm), CCR (cm), Temp (°C), Inv-P (sec)	0.995	0.935	0.995
Channel 3 (Watts)	Ch1 (cps), Ch2 (W), FCR (cm), CCR (cm), Temp (°C), Inv-P (sec)	0.993	0.931	0.992
Fine Control Rod Heights (cm)	Ch1 (cps), Ch2 (W), Ch3 (W), CCR (cm), Temp (°C), Inv-P (sec)	0.960	0.937	0.938
Coarse Control Rod Heights (cm)	Ch1 (cps), Ch2 (W), Ch3 (W), FCR (cm), Temp (°C), Inv-P (sec)	0.870	0.915	0.986

4 Conclusion

The AGN-201 has been used in this study as a testbed to explore developing a DT to examine and detect proliferation-like activities in a real operating nuclear reactor. While the AGN-201 is a small 5 W reactor, the lessons learned and techniques applied can easily be translated to larger scale reactors to anticipate pitfalls or areas of interest. For example, due to the small reactor size, the inherent noise made it difficult to create a physics-informed SM. Through applying proper bias-adjusting and data-cleaning techniques, a more robust physics-informed SM can be developed. Similar trends are expected in larger reactors where aspects such as feedback play a more prominent role. Utilizing the lessons learned from AGN-201, a robust virtual DT can be developed to explore aspects, such as safeguard by design, before a reactor is fully constructed.

This work culminates in developing a DT framework utilizing DeepLynx, which ingests real-time data from the AGN-201 DAS, compares this with data from a physics-informed SM and a data-driven ML model to determine if the reactor is operating under normal conditions. Developing the physics-informed SM allows for the culmination of real-world data and modeling and simulation to determine when the reactor is critical and how the reactor changes power over time. Once the DT is completed, we will run equivalent diversion and misuse scenarios in the reactor with direct data streams to the DT to determine the detection probability.

References

[1] "Digital engineering strategy," U.S Department of Defense, Tech. Rep. [Online]. Available: https://man.fas.org/eprint/digeng-2018.pdf

[2] V. Yadav et al, "The state of technology of application of digital twins," U.S. Nuclear Regulatory Commission, Tech. Rep. TLR/RES-DE-REB-2021-01, 2021. [Online]. Available: https://www.nrc.gov/docs/ML2116/ML21160A074.pdf

[3] V. Yadav et al, "Technical challenges and gaps in digital-twin-enabling technologies for nuclear reactor applications," U.S. Nuclear Regulatory Commission, Tech. Rep. TRL/RES-DE-REB-2021-17, 2021. [Online]. Available: https://www.nrc.gov/docs/ML2136/ML21361A261.pdf

[4] C. Ritter, R. Hays, J. Browning, R. Stewart, S. Bays, G. Reyes, M. Schanfein, A. Pluth, P. Sabharwall, R. Kunz, A. Shields, J. Koudelka, and P. Zohner, "Digital Twin to Detect Nuclear Proliferation: A Case Study," *Journal of Energy Resources Technology*, vol. 144, no. 10, 03 2022, 102108. [Online]. Available: https://doi.org/10.1115/1.4053979

[5] R. Stewart, A. Shields, F. Gleicher, S. Bays, M. Schanfein, G. Reyes, and C. Ritter, "Utilizing a Digital Twin to Analyze and Monitor Proliferation Scenarios in a Sodium-Cooled Fast Reactor," *Journal of Science and Global Security (submitted)*, 2023.

[6] S. Wen, R. Stewart, F. Shields, Ashley Gleicher, S. Bays, G. Reyes, M. Schanfein, and C. Ritter, "Utilizing Advanced Statistics to Determine Anomalistic Conditions in Pebble-Bed Reactors," *Journal of Nuclear Material Management (submitted)*, 2023.

[7] R. Stewart, A. Shields, C. Pope, J. Darrington, K. Wilsdon, S. Bays, K. Heaps, J. Scott, G. Reyes, M. Schanfein, E. Trevino, J. Palmer, and C. Ritter, "A digital twin of the AGN-201 reactor to simulate nuclear proliferation," *Proceedings of the INMM/ESARDA 2023 Joint Annual Meeting*, 2023.

[8] C. Pope and W. Pheonix, *Idaho State University AGN-201 Low Power Teaching Reactor - An Overlooked Gem*, ser. Nuclear Reactors Spacecraft Propulsion, Research Reactors, and Reactor Analysis Topics, 2021.

[9] "Safety analysis report of the idaho state university agn-201m research reactor," Tech. Rep. License No. R-110, Docket No. 20-284, 2003.

[10] M. Gorham, "Experimental parameterization of the idaho state university agn-201 research and training reactor," Master's thesis, Idaho State University, 2012.

[11] L. Wetzel, R. Busch, K. Carpenter, and D. Bowen, "Nondestructive and supplemental measurements of the university of new mexico," Tech. Rep. ORNL-TM-2019/1410, 2109.

[12] F. Pedregosa et al, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5-6, pp. 183–197, 1991.

[14] S. Badirli, X. Liu, Z. Xing, A. Bhowmik, K. Doan, and S. S. Keerthi, "Gradient boosting neural networks: Grownet," *arXiv preprint arXiv:2002.07971*, 2020.

[15] S. Ö. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6679–6687.

[16] J. W. Darrington, J. M. Browning, and C. S. Ritter, "Deep lynx: Digital engineering integration hub," 7 2020. [Online]. Available: https://www.osti.gov//servlets/purl/1668295