SUNI Review Complete
Template=ADM-013
E-RIDS=ADM-03

ADD: Matthew Dennis, Tray Hathaway, Mary Neely
Comment (1)
Publication Date: 7/5/2022
Citation: 87 FR 39874

PUBLIC SUBMISSION

As of: 7/27/22 6:10 AM
Received: July 19, 2022
Status: Pending_Post
Tracking No. l5s-nj85-nd8n
Comments Due: August 19, 2022
Submission Type: API

Docket: NRC-2022-0095
NRC's Fiscal Years 2023-2027 Artificial Intelligence Strategic Plan

Comment On: NRC-2022-0095-0001
NRC's Fiscal Years 2023–2027 Artificial Intelligence Strategic Plan

Document: NRC-2022-0095-DRAFT-0002
Comment on FR Doc # 2022-14239

Submitter Information

Name: Edward Chen
Address:
Raleigh, NC, 27606
Email: echen2@ncsu.edu
Phone: 7347306673

General Comment

See attached pdf document.

Attachments

NRC_RFC

Docket ID: NRC-2022-0095

Comment on the "NRC's Fiscal Years 2023-2027 Artificial Intelligence Strategic Plan":

There are 2 major issues with the NRC's strategic plan that it does adequately addresses:

1. The maintenance requirements for models incorporated over the lifetime of the plant also known as technical debt.
2. The overarching premise that baseline data collected in the development of said predictive models are representative of the problem scope for the entirety of operation.

These two problems are even more significant than any technical development issues of predictive models. In NUREG/CR-7294, the NRC has explored existing state of technology of AI/ML models. Repeated throughout the report is the issue of data quality, quantity, applicability, and uncertainty. These are significant non-trivial problems that currently plague all industries that utilize AI/ML. However, the difference is that in nuclear energy, the consequence of model failure is significant compared to other industries. The design of the model is less relevant as nearly all models suffer from the two problems above. Take the following statement from NUREG/CR-7294 as an example:

> *However, due to AI/ML uncertainty, the insufficiency of data quality and quantity, and lack of cognition about how to efficiently incorporate knowledge and data, challenges of adapting AI/ML techniques still exist. New perspectives and advanced frameworks should be proposed for different purposes in nuclear engineering. Particularly, the "black box" nature of ML/AI brings challenges with respect to the trustworthiness and transparency of the results in nuclear industry. This challenge makes the deployment of ML/AI-guided applications difficult to satisfy the regulatory requirements of NRC.*

This leads to the first problem that the strategic plan has not yet addressed. Suppose for the sake of argument that sufficient quality data exists during the software development process of the ML/AI model to develop highly accurate predictive models for an arbitrary safety critical variable (e.g., fuel centerline temperature). The model is dispatched to a plant to assist operator decisions. At first glance, this seems to be the optimal 'goal' of the ML/AI project, a dispatchable model that can improve safety; however, significant issues will arise through the lifetime of the plant. The primary problem is that the assumption that sufficient quality data is available is fundamentally flawed. The operating physics of nuclear power plants are constantly changing from beginning of life (BOL) to end of life (EOL). The data collected can only represent a subset snapshot of the reactor physics and will never represent all possible states. Therefore, it is highly probable that the model will slowly (or suddenly) become irrelevant at some changing point in the reactor lifecycle. Models developed based on BOL or snapshot reactor physics data will lose predictive accuracy overtime and be detrimental to operators rather than useful. On the other hand, sudden changes in the reactor state (known as context shifts) can disrupt temporal predictive models (e.g., RNN). Anticipated sudden changes such as replacing fuel, or a single sensor degrading are examples of context shifts. More serious context shifts could be power transients could cause serious predictive issues. This data problem cannot be easily resolved by including all possible states of concern in the data set. Even limiting the data set to the subset including anticipated operational occurrences (AOO) or design basis accidents (DBA), would rely on strong assumptions about how systems fail that may be physically infeasible. In any regard, the dataset for anticipated possible states would be intractable, expensive, and would most likely rely on multiple models developed for different context scenarios. The last point is due

to the over generalization problem of neural network-based ML/AI models. In essence, as the problem scope increases to include more scenarios, the accuracy of the model steadily decreases. In general, there is no acceptable solution to the latter problem as there are too many highly speculated (and unsupported) scenarios to consider to develop a comprehensive model. Therefore, if the former approach is adopted, it may be possible to update the ML/AI model routinely to reflect changing states of the NPP.

Generally, there are three approaches for implementing ML/AI models in continuously evolving environments: (1) locked models, (2) updating locked models, and (3) continuously updating models. In the most basic approach, a locked model is a model with weights and parameters that cannot change regardless of the state of the plant. The benefits to this approach are that the model can be reviewed and controlled for quality without concern of any deviation from function. The outputs from the model can be anticipated based on the training data and determined if they are valid or not. However, the drawback is that the model is guaranteed to become irrelevant over time in any continuously evolving environment. It also removes the primary benefit of ML/AI models which is the ability to adapt and continuously 'learn' the operational environment. The second approach is the updating locked model approach. In this approach, locked models (with updated weights) are periodically dispatched as software updates whenever a sufficient change in the operational environment is detected. Each of the locked models are verified against a benchmark to ensure that they meet performance and safety requirements. In this approach, the ML/AI models can still update and 'learn' however under the strict oversight of developers. The drawback to this approach, however, is that given multiple versions of the same model will be developed, which model should be trusted and used. If an earlier version of the model contradicts later versions of the same model, should operators not trust the later version of the model? Version contradiction and management is the chief problem with the updating locked model approach. The last approach is the conventional approach to ML/AI models, that is to develop a continuous learning model with build-in restrictions and limitations and to hope that they ML/AI model will always perform accordingly. The model will adapt to any minute changes in the operational environment from BOL to EOL and to theoretically provide always accurate results. However, this approach is incredibly dangerous. Routinely we have seen optimal models that have been developed in the ideal laboratory conditions only to be corrupted, cause unanticipated side effects, or abuse constraints (in the reward function) and result in loss to stakeholders. It is also incredibly difficult to design comprehensive constraints such that the model will always perform as expected. It is difficult in conventional systems (which is why software failure routinely occurs) and it is nearly impossible for a continuously learning system.

This leads to the second major problem not covered by the strategic plan, that is the data control for risk applications. A new emerging topic in ML/AI is out-of-distribution (OOD) detection. The premise being that ML/AI models are only highly accurate whenever the input data is within the training data subset (i.e., interpolation). However, ML/AI models routinely fail at extrapolation task. OOD detection therefore is 'sensing' or calculating how 'far' away the input data is from known training data. The Mahalanobis distance is one example metric used to gauge distance of an input to training data distributions. Training data will always be a subset of the operational environment (for the reasons discussed above). Furthermore, training data is typically developed in a highly augmented environment (i.e., without data noise, normalized, trimmed, synthesized, etc.). This means that while training performance can be incredible high (>90%), when applied to real world applications, the models typically experience a 30-40% decrease in predictive accuracy. Therefore, a way to measure OOD is absolutely required to trust the

predictions made by an ML/AI model. The significance of OOD detection is not mentioned in the strategic plan nor in NUREG/CR-7294.

In summary, the current strategic plan should incorporate:

1. The NRC's anticipated plan on post ML/AI deployment and continued maintenance. Specifically, how will developers maintain their models to keep them relevant.
    a. Will ML/AI models be locked models or continuously updating models and what type of framework is sufficient to ensure models remain relevant over time?
2. Out-of-distribution is a key area of research when it comes to the trustworthiness and reliability of ML/AI models. Regardless of the type of model developed, a framework to ensure model relevancy (via OOD detection methods) must be in place to ensure models are relevant.