Improving Visualization of Dynamic Probabilistic Risk Assessment Simulations for Risk Insight Extraction

Meredith R. Thibeault*, Joseph O'Leary, Yunfei Zhao, and Katrina M. Groth
Center for Risk and Reliability, University of Maryland, College Park, MD, USA
*thibeaum@umd.edu

Abstract: Dynamic probabilistic risk assessment (DPRA) is an advanced risk analysis method that more realistically captures the progression of events during accident scenarios that may affect the safety of a nuclear power plant (NPP). This method integrates the modeling of random events that may occur during an accident scenario with deterministic simulations of system behavior. This provides a detailed, accurate understanding of accident scenarios by improving the modeling of interdependencies between events and leveraging high-fidelity accident progression simulations. As a result, DPRA supports an improved understanding of the system-level risk and the impact on plant operation as a whole.

A DPRA analysis typically involves simulating thousands to millions of accident scenarios, each of which produces high-dimensional time-series data with numerous variables and temporal dependencies. The scale and complexity of this data introduce significant challenges when trying to efficiently and effectively analyze the data to extract actionable risk insights. These challenges create a critical barrier to the adoption of DPRA methods within the nuclear energy industry.

This work investigates the use of advanced machine learning techniques to enable efficient analysis of high-dimensional DPRA simulation data. Specifically, this work explores methods for dimensionality reduction of DPRA simulation results while preserving key information relevant to accident progression and risk. The proposed approach aims to facilitate improved visualization of key system behaviors and accident patterns within large simulation datasets. By improving the visualization of DPRA results, this work can promote a wider adoption of DPRA methods, ultimately contributing to more efficient, cost effective, and safer nuclear power plant operation.

1. INTRODUCTION

Nuclear power plants (NPPs) operate in complex, dynamic environments in which the safety and reliability of systems must be maintained under a wide range of both external and internal operating conditions. Probabilistic risk assessment (PRA) is a well-established methodology used to systematically evaluate the risks associated with NPP operation by identifying potential accident sequences, estimating their likelihood, and assessing their consequences [1]. Conventional PRA methods, however, rely on static modeling frameworks, which represent system behavior as discrete branching points [2]. Because of this, conventional PRA techniques are limited in their ability to efficiently capture the time-dependent interactions between systems, operators, and the physical processes that occur during the progression of accident scenarios [1, 2].

Dynamic probabilistic risk assessment (DPRA) was developed to address these limitations by explicitly incorporating the time dependence of accident progression into the risk assessment framework [1, 3]. This enables a more realistic and comprehensive characterization of system risk than conventional static methods can provide, particularly for complex scenarios where the timing and order of events has a significant impact on outcomes [2, 4, 5].

Despite these advantages, the widespread adoption of DPRA methods within the nuclear energy industry has been limited [6, 7]. A DPRA analysis typically involves simulating thousands to millions of accident scenarios, each producing high-dimensional time-series data, characterized by a large number of variables and many time steps per simulation [2, 8]. The scale and complexity of this data presents significant challenges when attempting to efficiently extract actionable risk insights. Current approaches to DPRA data analysis and visualization are not well-suited to handle data of this dimensionality [9, 10], creating a critical barrier to practical uses.

This work investigates the use of advanced machine learning (ML) techniques to address these challenges, with a focus on enabling efficient analysis of high-dimensional DPRA simulation data. Specifically, this paper introduces a case study involving a transient overpower event in a sodium fast reactor, simulated using the ADAPT dynamic event tree framework [11, 12] coupled with the SAS4A system analysis code [13, 14]. The main contribution of the paper is a preprocessing methodology that transforms the complex output of DPRA simulations, consisting of discrete events and continuous physical process variables, into a unified data format that can be used with recurrent neural network (RNN) [15, 16] and long short-term memory (LSTM) [17] based dimensionality reduction techniques. This preprocessing approach converts discrete events into physically meaningful continuous state variables by using knowledge of the underlying system behavior to define how each event changes the state of the system and how that state evolves over time, enabling the two data types to be represented in a single framework.

2. CURRENT STATE OF ART

2.1. Dynamic Probabilistic Risk Assessment

Probabilistic risk assessment is a systematic methodology for evaluating the risks associated with complex engineering systems by identifying potential accident sequences, estimating their likelihood, and assessing their consequences [1]. Conventional, static PRA methods represent accident progressions through event trees and fault trees, which model system behavior as a series of discrete branching points. While these methods have been widely applied in the nuclear industry and form the basis of risk-informed regulatory decision making, they are limited in their ability to capture the time-dependent interactions between systems, operator actions, and physical processes that occur during complex accident scenarios [1, 2].

DPRA was developed to overcome these limitations by explicitly modeling the time dependence of accident progressions [1, 3]. Rather than representing accident sequences as static branching logic, DPRA combines the stochastic modeling of random events with deterministic simulation of system behavior, allowing the full time history of an accident scenario to be captured with high fidelity [2]. This integration enables DPRA to model interdependencies between events, system responses, and operator actions that are not easily representable in conventional static frameworks, producing a more realistic and comprehensive characterization of risk [2, 11].

A central element in DPRA is the dynamic event tree (DET), which extends the conventional static event tree by integrating time-dependencies into events [11, 18]. In a DET, each branch point represents a discrete event or action, such an automatic system response or an operator action. In addition to having an associated probability, each event has a specific time point where it occurs. The full set of scenarios generated by traversing the DET provides a probabilistic map of possible accident progressions that captures the likelihood and physical consequences of each path [2, 11, 14, 19]. The generation of dynamic event trees requires a computational framework capable of managing the branching logic of the DET and coordinating the execution of the deterministic simulation code. To address this need, the Analysis of Dynamic Accident Progress Trees (ADAPT) framework was developed to provide a code-agnostic platform for the dynamic generation of accident progression event trees [11, 12]. ADAPT manages the DET branching, assigns probabilities to each branch point, and interfaces with the simulation code to execute the deterministic system simulations.

To fully explore the space of accident sequences, DPRA relies on a large number of simulations, generating a substantial number of high-dimensional simulation data that presents a significant challenge for analysis and interpretation [2, 8]. Several approaches have been explored to address these challenges. Mandelli et al. developed methods for scenario clustering to identify groups of similar accident progressions within large DPRA datasets, enabling a more structured exploration of the scenario space [9]. Subsequent work extended these methods to mining techniques capable of identifying risk-important variables and behaviors within DPRA frameworks [10, 20]. While these methods show meaningful progress, they are generally limited in their ability to handle the full dimensionality of DPRA simulation output and do not leverage the representational capacity of modern machine learning architectures. The development of more capable analysis methods that can operate effectively on high-dimensional DPRA time-series data remains a challenge.

2.2. Machine Learning for High-Dimensional Time-Series Analysis

Machine learning (ML) methods have demonstrated strong capability for identifying structure in high-dimensional time-series data across scientific and engineering domains [21, 22]. Particularly relevant to DPRA data analysis are deep learning architectures designed for sequential and time-series data, which are well-suited to capture the temporal dependencies that characterize accident progression simulations [21].

RNNs are a class of deep learning architectures specifically designed to process sequential data by maintaining an internal state, or memory, that evolves as the network processes each element of the input sequence [15]. This makes RNNs well suited for time-series data, where the value of a variable at any given time step is dependent on previous time steps. LSTM networks are a specialized form of RNNs that address the vanishing gradient problem (where gradient signals decay during backpropagation, preventing the network from learning long-range dependencies) that occurs in standard RNNs, enabling the network to learn across longer time sequences [16, 17]. LSTMs achieve this through a gating mechanism that controls what information is retained, updated, or discarded as the sequence is processed, allowing relevant features to be preserved [17]. These properties make LSTMs particularly well-suited for the analysis of DPRA simulation data, which consists of long time sequences with complex temporal dependencies.

LSTMs have been successfully applied to a range of time-series analysis task in the nuclear engineering field. Lee at al. demonstrated the use of dynamic event trees combined with deep learning for real-time emergency planning in nuclear power plant operation, highlighting the usability of these methods in nuclear safety [23]. Yang et al. applied LSTMs to accident diagnosis in nuclear plants, demonstrating the ability to classify plant states from time-series sensor data [24]. Bae et al. further demonstrated the use of LSTMs for plant parameter trend prediction, showing that these networks can accurately model the change in physical process variables across a variety of operating conditions [25]. These studies show LSTM models are well-suited for NPP time-series data and motivate their application to DPRA simulation data.

A particularly relevant application of LSTM models is dimensionality reduction of high-dimensional time-series data. It is possible to use an LSTM-based encoder to create a compressed representation of high-dimensional, sequential data by identifying patterns within the large data set [22]. This can be directly applied to DPRA simulation data, where each accident scenario is a high-dimensional time series and the goal is to identify and preserve key features relevant to risk assessment. Reducing the dimensionality of this data would substantially improve the interpretability of DPRA results and enable more efficient risk assessment.

3. DPRA DATASET OVERVIEW

3.1. DPRA Simulation Data

A DPRA analysis generates two fundamentally different types of data that together describe the full progression of an accident scenario: discrete event data and continuous time-series data. Understanding the nature of each data type and the relationship between them is essential to understanding the challenges associated with DPRA analysis.

Discrete event data describes the occurrence of a specific event or action at a distinct point during an accident scenario. These events represent branching points in the dynamic event tree and may include automatic system responses, such as a reactor protection system response, or operator interventions, such as a manual pump trip. Each event occurs once within a given scenario at a specific time point, and the set of events that occur defines the unique pathway through the dynamic event tree [2, 11]. In addition to the event itself, each branching point has an associated conditional probability, reflecting the likelihood that a particular event occurs, given the previous events in the scenario. The discrete event data determines the branching structure of the DET and provides the sequences of events through the possible accident progressions.

In contrast, continuous time-series data describes the physical process variables throughout the duration of an accident scenario. These variables are produced by the deterministic system simulation code coupled to the DET framework and capture the thermal-hydraulic and neutronic state of the plant at each time step over the course of the simulation [11, 12]. The continuous data is produced for every time step of the simulation, making it much denser than the discrete event data, where each event occurs at a single time point. The number of variables and time steps in a DPRA analysis means that a high-dimensional set of data is generated for each simulated scenario, and the full set of thousands to millions of scenarios further increases the dataset size and complexity [8, 19].

Together, these two data types provide a comprehensive picture of accident progressions: the discrete events define what happens and when, while the continuous variables describe how the physical system responds. However, the fundamental difference in structures, one sparse and categorical, the other dense and numerical, present a significant challenge for any analysis method that intends to use them in combination. This incompatibility is one obstacle to applying modern machine learning techniques directly to raw DPRA simulation output, and motivates the preprocessing methodology described in Section 4.

3.2. Case Study: Transient Overpower Event in a Sodium-Cooled Fast Reactor

The dataset used in this study was generated for a DPRA analysis of a transient overpower (TOP) event in a sodium-cooled fast reactor (SFR) **[5, 14, 26]**. A TOP event represents an uncontrolled increase in reactor power and is a design-basis accident scenario of particular importance for SFR safety analysis **[27]**. The dynamic event tree for this scenario was constructed using the ADAPT software [11, 12], and deterministic system simulations were performed using SAS4A/SASSYS-1 [13], a well-established code for transient analysis of sodium-cooled fast reactors.

The dataset consists of 2,052 unique accident scenarios. Each scenario represents a possible pathway through the dynamic event tree, defined by a unique combination of system responses and operator actions following the initiating event. The discrete event data describes the branching points of the DET, capturing component and subsystem state changes and operator interventions at specific points during each scenario. The continuous simulation data consists of 106 physical process variables recorded at each time step, containing thermal-hydraulic and neutronic values across five reactor channels and a whole-core model. Individual scenarios vary in duration, terminating either upon cladding melt, the cold pool temperature reaching a critically high threshold, or successful completion of the full 24-hour simulation window. Of the 2,052 scenarios, 1,088 reach the full 24-hour duration, the remainder terminate early due to one of the two failure conditions.

The scale and diversity of the dataset make it well-suited as a case study for the development and demonstration of machine learning-based analysis methods for DPRA data. At the same time, the physical complexity of the SFR transient scenario ensures that the dataset captures the nonlinear interactions and time-dependent behaviors that motivate the use of DPRA in the nuclear industry and present the greatest challenge for conventional approaches.

4. DATASET PREPARATION FOR MACHINE LEARNING APPLICATIONS

4.1. Preprocessing Requirements for Machine Learning Compatibility

The raw output of a DPRA analysis presents a fundamental structural incompatibility that must be resolved before machine learning methods can be applied. As described in Section 3, DPRA simulation data consists of two distinct data types: discrete event data, which describes the occurrence of specific system responses and operator actions at individual time steps, and continuous time-series data, which describes the physical process variables across the full duration of each accident scenario. While both data types are necessary to fully record the accident progressions, their structural differences make them incompatible for direct use with sequence-based machine learning architectures such as RNNs and LSTM networks [15, 17].

RNNs and LSTMs operate on sequential data in which every input feature is represented at every time step, forming a regularly structured tensor of samples, timesteps, and features [15, 17]. The continuous SAS4A output satisfies this requirement, with each of the physical process variables being recorded at every time step throughout the simulation. The discrete event data, however, does not. Each discrete event occurs at most once within a given scenario, at a single point in time, and is not represented at any other time step. This sparse, one-dimensional encoding is structurally incompatible with the dense, multi-timestep format required by RNN and LSTM architectures. Simply appending the discrete events to the continuous data as additional features is not possible without first transforming them into a time-series representation.

An additional challenge arises from the many-to-many relationships between discrete events and system states. A single discrete event may affect multiple system state variables simultaneously, and a single system state variable may be modified by multiple different events at different points during a scenario. Any preprocessing approach must therefore account for these dependencies, ensuring that the state of each system variable is correctly represented at every time step in a manner that accurately reflects the cumulative effect of all events that have occurred up to that point.

4.2. Preprocessing Methodology

To resolve the structural incompatibility described in section 4.1, we developed a preprocessing methodology that converts the discrete event data into continuous time-series representations using physical system knowledge. For each discrete event type in the DET, we defined a new time-series state variable to represent the relevant system or operational condition throughout the duration of the scenario. We assigned each state variable a default nominal value reflecting the initial condition of the system prior to any event occurrence. As discrete events occur during the simulation, we use them to update the value of the corresponding state variable at that time step, with the new value persisting for all subsequent time steps until another event modifies it. This approach transforms each discrete event into a physically meaningful state variable that evolves over time in a manner consistent with the actual behavior of the system.

The control rod position and primary pump variables provide an example of this approach. In the nominal state, we initialize the control rods as withdrawn and the primary pump status to be running. When a “reactor protection system scram and trip success” event occurs, we transition the control rods to the inserted state and the primary pump status to stopped at that time step, reflecting the physical change in the reactor protection system. These inserted and running states then persist for the remainder of the scenario. The result is a time-series variable that accurately represents the control rod position at every time step, directly derived from the occurrence and timing of the discrete scram event.

The preprocessing is challenged by the many-to-many relationships between discrete events and state variables. Some events affect multiple state variables simultaneously. For example, an RPS Scram and Trip Success event requires us to update both the control rod position and the primary pump status in a single branching action. Conversely, some state variables are updated by multiple different events. The primary pump status, for example, is modified by scram and trip outcomes, operator trip and recovery actions, and operator responses at varying levels of thermal damage. We handle these relationships by evaluating all events that affect each state variable and applying their effects in the order in which they occur during the scenario.

We defined seven new continuous state variables in total, corresponding to the discrete event categories present in the DET. A complete summary of the state variables, their default nominal values, and the system changes caused by each branch event is provided in Table 1.

Table 1: Summary of new system state variables and associated branch events

New State Variable	Default Value	Branch Event	State Change
Reactivity coefficient	0	Reac. 1 – Reac. 10	Reactivity coefficient: 1-10
Transient overpower (TOP)	0.0	No TOP	TOP: 0.0
		0.06 TOP	TOP: 0.06
		0.3 TOP	TOP: 0.3
		0.5 TOP	TOP: 0.5
Primary pump	Running	RPS Scram and Trip Success	Stopped
		RPS Scram Success and Trip Failure	Running
		RPS Scram Failure and Trip Success	Stopped
		RPS Scram and Trip Failure	Running
		Operator trips RPS	Stopped
		Operator recovers RPS	Running
		Operator Trips Primary Pumps at High Temp (0%, 50%, 100% damage)	Stopped
		Operator Does Not Trip Primary Pumps at High Temp (0%, 50%, 100% damage)	Running
Control rods	Withdrawn	RPS Scram and Trip Success	Inserted
		RPS Scram Success and Trip Failure	Inserted
		RPS Scram Failure and Trip Success	Withdrawn
		RPS Scram and Trip Failure	Withdrawn
Thermal damage	0%	Operator Trips/Does Not Trip at 0% damage	0%
		Operator Trips/Does Not Trip at 50% damage	50%
		Operator Trips/Does Not trip at 100% damage 100%	100%
Secondary pump	No action	Secondary pump enhancement	Enhanced
		No secondary pump enhancement	No action
		Secondary Pump Degradation	Degraded
DRACS	No attempt	Successful DRACS heat exchange attempt	Success
		No DRACS heat exchange	No attempt
		Failed DRACS heat exchange attempt	Fail

Following this conversion, each of the seven new state variables are represented as a continuous time series of the same length as the SAS4A simulation output for that scenario. We then concatenate these new variables with the 106 physical process variables from SAS4A, producing a single unified data matrix for each accident scenario with 113 columns, one per variable, and a row for each time step. This structure satisfies the input requirements of RNN and LSTM architectures and enables the full information content of the DPRA simulation output to be used in the machine learning model.

5. CONCLUSION

In this work, we developed a preprocessing methodology to address a fundamental challenge in the application of machine learning to dynamic probabilistic risk assessment data. DPRA simulations produce two structurally incompatible data types, discrete event data describing actions that occur at specific points in time and continuous time-series data that describes the physical process variables during an accident scenario. These different structures prevent the direct application of sequence-based machine learning architectures, which require all input features to be represented at every time step. We developed a methodology that resolves this incompatibility by leveraging physical system knowledge to convert discrete events into time series system state variables, enabling both data types to be represented in a format usable with RNN and LSTM based machine learning models.

We demonstrated this methodology on a dataset of 2,052 accident scenarios generated from a DPRA analysis of a transient overpower event in a sodium-cooled fast reactor, simulated using the ADAPT dynamic event tree framework coupled with SAS4A/SASSYS-1. By defining seven physical system variables affected by the DET branching conditions we produced a data matrix of 113 variables for each accident scenario. This preserves the data output of the original simulation while creating a format compatible with future machine learning efforts.

This preprocessing methodology is a necessary step towards enabling machine learning analysis of DPRA simulation data. Future work will build on this foundation by applying RNN and LSTM based dimensionality reduction techniques to the preprocessed dataset to extract actionable risk insights from the high-dimensional simulation output.

The ability to efficiently analyze and interpret DPRA simulation data has a large impact on the adoption of DPRA methods within the nuclear industry. The complexity and size of DPRA output has been a major barrier to its practical application, limiting the risk assessment benefits that DPRA methods offer over conventional static PRA techniques. By demonstrating a path toward machine learning-enabled analysis of this data, this work contributes to reducing that barrier and supports the broader goal of making DPRA a practical tool for nuclear power plant risk analysis. This has the potential to improve the quality of risk-informed decision making in the nuclear industry, contributing to safer, more reliable, and more cost-effective nuclear power plant operation.

Acknowledgements

This work is supported by the U.S. Nuclear Regulatory Commission, grant number 31310025M0004. Any opinions, conclusions, or recommendations expressed in this paper belong to the authors alone and do not necessarily represent the views or opinions of the sponsoring agency.

References

[1] N. Siu, “Risk assessment for dynamic systems: An overview,” *Reliab. Eng. Syst. Saf.*, vol. 43, no. 1, pp. 43–73, Jan. 1994, doi: 10.1016/0951-8320(94)90095-7.

[2] T. Aldemir, “A survey of dynamic methodologies for probabilistic safety assessment of nuclear power plants,” *Ann. Nucl. Energy*, vol. 52, pp. 113–124, Feb. 2013, doi: 10.1016/j.anucene.2012.08.001.

[3] G. Cojazzi, “The DYLAM approach for the dynamic reliability analysis of systems,” *Reliab. Eng. Syst. Saf.*, vol. 52, no. 3, pp. 279–296, Jun. 1996, doi: 10.1016/0951-8320(95)00139-5.

[4] K. M. Groth, M. R. Denman, J. N. Cardoni, and T. A. Wheeler, “Proof of Principle Framework for Developing Dynamic Risk-Informed Severe Accident Management Guidelines,” Sandia National Laboratories, Albuquerque, NM, SAND2013-8324, Sep. 2013.

[5] A. D. Lewis and K. M. Groth, “A Dynamic Bayesian Network Structure for Joint Diagnostics and Prognostics of Complex Engineering Systems,” *Algorithms*, vol. 13, no. 3, p. 64+, Mar. 2020, doi: 10.3390/a13030064.

[6] N. Siu, “Dynamic PRA for Nuclear Power Plants: Not If but When?,” Technical Opinion Paper, Draft for Comment ML19066A390, US Nuclear Regulatory Commission. Available: https://www.nrc.gov/docs/ML1906/ML19066A390.pdf

[7] M. R. Denman, M. C. Darling, K. M. Groth, J. N. Cardoni, and Z. K. Jankovsky, “Formulation of Platform Enhancement Specifications and Demonstration Problem for Discrete Dynamic Event Tree Analysis,” Sandia National Laboratories, Albuquerque, NM, Nov. 2015.

[8] J. O’Leary, N. Wasserman, and Y. Zhao, “Dynamic Probabilistic Risk Assessment: A Review of Techniques to Reduce Computational Costs,” *Nucl. Sci. Eng.*, pp. 1–21, Feb. 2026, doi: 10.1080/00295639.2025.2594887.

[9] D. Mandelli, A. Yilmaz, T. Aldemir, K. Metzroth, and R. Denning, “Scenario clustering and dynamic probabilistic risk assessment,” *Reliab. Eng. Syst. Saf.*, vol. 115, pp. 146–160, Jul. 2013, doi: 10.1016/j.ress.2013.02.013.

[10] D. Mandelli *et al.*, “Mining data in a dynamic PRA framework,” *Prog. Nucl. Energy*, vol. 108, pp. 99–110, Sep. 2018, doi: 10.1016/j.pnucene.2018.05.004.

[11] A. Hakobyan *et al.*, “Dynamic generation of accident progression event trees,” *Nucl. Eng. Des.*, vol. 238, no. 12, pp. 3457–3467, Dec. 2008, doi: 10.1016/j.nucengdes.2008.08.005.

[12] U. Catalyurek *et al.*, “Development of a code-agnostic computational infrastructure for the dynamic generation of accident progression event trees,” *Reliab. Eng. Syst. Saf.*, vol. 95, no. 3, pp. 278–294, Mar. 2010, doi: 10.1016/j.ress.2009.10.008.

[13] Argonne National Laboratory, *The SAS4A/SASSYS-1 Safety Analysis Code System*. Argonne, IL, USA: Argonne National Laboratory, 2025. Available: https://sas-doc.nse.anl.gov/latest/

[14] Z. K. Jankovsky, M. R. Denman, and T. Aldemir, “Dynamic event tree analysis with the SAS4A/SASSYS-1 safety analysis code,” *Ann. Nucl. Energy*, vol. 115, pp. 55–72, May 2018, doi: 10.1016/j.anucene.2018.01.001.

[15] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” Sep. 03, 2014, *arXiv*: arXiv:1406.1078. doi: 10.48550/arXiv.1406.1078.

[16] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proceedings of the 30th International Conference on Machine Learning*, S. Dasgupta and D. McAllester, Eds., in Proceedings of Machine Learning Research, vol. 28. Atlanta, Georgia, USA: PMLR, Jun. 2013, pp. 1310–1318. Available: https://proceedings.mlr.press/v28/pascanu13.html

[17] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[18] C. Acosta and N. Siu, “Dynamic event trees in accident sequence analysis: application to steam generator tube rupture,” *Reliab. Eng. Syst. Saf.*, vol. 41, no. 2, pp. 135–154, Jan. 1993, doi: 10.1016/0951-8320(93)90027-V.

[19] X. Zheng, H. Tamaki, T. Sugiyama, and Y. Maruyama, “Dynamic probabilistic risk assessment of nuclear power plants using multi-fidelity simulations,” *Reliab. Eng. Syst. Saf.*, vol. 223, p. 108503, Jul. 2022, doi: 10.1016/j.ress.2022.108503.

[20] D. Mandelli, Z. Ma, C. Parisi, A. Alfonsi, and C. Smith, “Measuring risk-importance in a Dynamic PRA framework,” *Ann. Nucl. Energy*, vol. 128, pp. 160–170, Jun. 2019, doi: 10.1016/j.anucene.2018.12.035.

[21] D. W. Otter, J. R. Medina, and J. K. Kalita, “A Survey of the Usages of Deep Learning for Natural Language Processing,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 604–624, Feb. 2021, doi: 10.1109/TNNLS.2020.2979670.

[22] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised Learning of Video Representations using LSTMs," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., in Proceedings of Machine Learning Research, vol. 37. Lille, France: PMLR, Jul. 2015, pp. 843–852. [Online]. Available: https://proceedings.mlr.press/v37/srivastava15.html
[23] J. H. Lee, A. Yilmaz, R. Denning, and T. Aldemir, "Use of Dynamic Event Trees and Deep Learning for Real-Time Emergency Planning in Power Plant Operation," *Nucl. Technol.*, vol. 205, pp. 1–8, Nov. 2018, doi: 10.1080/00295450.2018.1541394.
[24] J. Yang and J. Kim, "An accident diagnosis algorithm using long short-term memory," *Nucl. Eng. Technol.*, vol. 50, no. 4, pp. 582–588, May 2018, doi: 10.1016/j.net.2018.03.010.
[25] J. Bae, J. Ahn, and S. J. Lee, "Comparison of Multilayer Perceptron and Long Short-Term Memory for Plant Parameter Trend Prediction," *Nucl. Technol.*, vol. 206, no. 7, pp. 951–961, Jul. 2020, doi: 10.1080/00295450.2019.1693215.
[26] A. D. Lewis, "Dynamic Bayesian network updating approaches for enabling causal prognostics and health management of complex engineering systems," PhD dissertation, University of Maryland, College Park, MD, 2022. doi: 10.13016/1nbz-0yde.
[27] M. R. Denman, K. M. Groth, J. Cardoni, and T. Wheeler, "Advance Liquid Metal Reactor Discrete Dynamic Event Tree/Bayesian Network Analysis and Incident Management Guidelines (Risk Management for Sodium Fast Reactors)," Sandia National Laboratories, Albuquerque, NM, SAND2015-2484, Apr. 2015. doi: 10.2172/1177842.