Enabling Dynamic Probabilistic Risk Assessment of Physical Security Using EMRALD and MAAP

Daniel Nevius[a] and Steven Prescott[a]
[a] Idaho National Laboratory, Idaho Falls, ID, USA, daniel.nevius@inl.gov, steven.prescott@inl.gov

Abstract: The optimization of physical security in nuclear power plants requires sophisticated methodologies that integrate operator actions and plant behavior through advanced simulation tools. To address this, Idaho National Laboratory has developed the Modeling and Analysis for Safety and Security using the Dynamic EMRALD Framework (MASS-DEF) methodology, an approach that integrates force-on-force simulations, dynamic probabilistic risk assessment, and thermal-hydraulics modeling to enhance security planning while reducing costs. We developed a tool that produces reduced order models using thermal hydraulic simulations from the Modular Accident Analysis Program (MAAP) [1]. These models can quickly evaluate reactor core behavior during attack simulations, and in so doing, address two barriers of traditional methods: (1) MAAP simulations are computationally intensive, and (2) attack scenarios must be run in a secure environment, which complicates analysis and validation. By precomputing scenario outcomes for a small number of modified parameters, the reduced order model significantly decreases the computational cost and enables offsite review of the results.

1. INTRODUCTION

Under the Department of Energy's Light Water Reactor Sustainability (LWRS) physical security pathway, Idaho National Laboratory has developed the Modeling and Analysis for Safety and Security using the Dynamic EMRALD Framework (MASS-DEF) methodology to address rising operations and maintenance (O&M) costs at U.S. nuclear power plants. A significant portion of these O&M costs are related to the armed response force. Traditional physical security analysis assumes there is core damage whenever an adversary sabotages a target set with little or no consideration of time; the MASS-DEF methodology uses coupled simulations and dynamic probabilistic risk assessment (PRA) modeling to account for the time it takes attackers to disable items in the target set, for operators to take preventative actions, and for other alternative equipment to be used in target sets to prevent damage to the core [2]. By including operator actions or other measures and a timing analysis of the plant's thermal hydraulic response, MASS-DEF shows statistically whether the proposed protection methods maintain a plant's safety margins. These results can then be used to ease the O&M financial burden on plants by justifying a reduction in on-site armed response while maintaining adequate protection [2].

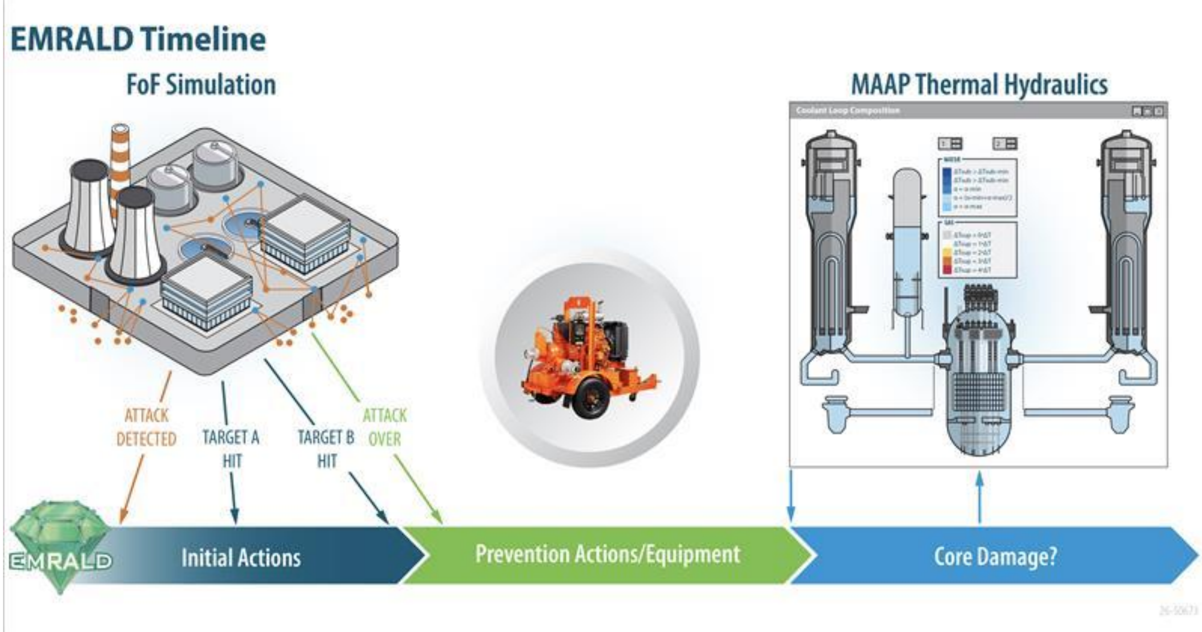
A key component of this analysis is the integration of thermal hydraulic simulation results into the Event Modeling Risk Analysis using Linked Diagrams (EMRALD) dynamic PRA model. The Modular Accident Analysis Program (MAAP) is a thermal hydraulic code for simulating core response in accident scenarios; it performs this simulation using time-dependent inputs supplied by EMRALD based on the current state of the model. The results produced by MAAP are read and integrated back into the model to provide a scenario-specific evaluation of whether core damage would occur. MAAP computations are, however, time-consuming and computationally expensive and need to be evaluated thousands of times in the running of the EMRALD model to be statistically reliable. A methodology and accompanying software tool have been developed for precomputing the MAAP results based on a limited number of parameter inputs. The tool produces a reduced order model (ROM) that enables thermal hydraulic results to be queried by and returned to EMRALD nearly instantly.

2. BACKGROUND

2.1 EMRALD and Dynamic PRA

EMRALD is a software tool developed by Idaho National Laboratory to enable the analysis of dynamic probabilistic risk assessment (PRA) models. Dynamic PRA differs from traditional PRA with the introduction of time-dependency, which allows models to evaluate the impact of time between events and time taken to complete actions. Models are built using a modified form of an event tree represented by states and diagrams. Time-dependent events trigger actions that move the model between states, and outcomes of events can conditionally alter the probabilities of subsequent events. In its application to physical security, this time dependency supports the methodology of MASS-DEF by enabling the analysis of operator actions and reactions in response to adversary actions, as shown in Figure 1. A key component of this analysis is the coupling of external simulations to the EMRALD model. A Run Application action type is available within EMRALD to supply time-dependent parameters from within the simulation to an external simulation code and use the results of that simulation in the model as it advances.

Figure 1: MASS-DEF Process Integrating Force-on-Force Simulations, EMRALD, and MAAP Thermal Hydraulics



MAAP is an application for simulating the thermal hydraulic response of a plant during accident scenarios. The coupling of thermal hydraulic simulations with EMRALD's dynamic PRA dramatically increases the value of such simulations because they consider the timing of operator actions that may directly influence the outcome of the scenario. EMRALD supports easy integration with MAAP through a graphical user interface (GUI) within the main EMRALD application, where the modeler can easily assign the time-dependent values of variables within the model to their corresponding parameters within the MAAP input deck [3]. Each MAAP run can take tens of minutes to complete, and previously a full run would have been necessary for each unique set of inputs. EMRALD runs through the model potentially millions of times, so waiting ten minutes for each MAAP run to complete is not feasible. Additionally, these simulations must be prepared by a thermal hydraulics expert and run on secure systems, both of which are barriers for coupling with EMRALD. A case study with industry collaboration [2] showed that MAAP was the most computationally limiting part of the analysis. The case study identified the key need to develop a simplified way to integrate MAAP with EMRALD with fast thermal hydraulics computation times to bring the MASS-DEF process to industry, and INL was awarded a Technology Commercialization Fund to do so [4].

2.2 Reduced Order Modeling Approach

As there are typically a small number of MAAP parameters modified by EMRALD in each model, a ROM can be utilized to avoid time-consuming direct calls to MAAP. This model is produced by pre-computing MAAP outputs across a representative sample of input parameter combinations, and it uses a regression model to interpolate using the pre-run values. The regression models can take sampled values from EMRALD and produce results that are accurate to a full MAAP call in microseconds rather than minutes, making the use of MAAP within the EMRALD simulation much more practical. Additionally, the ROM removes sensitive information that may be found in the MAAP input deck as well as the requirement of familiarity with thermal hydraulics and the MAAP code itself, easing barriers for MAAP's use by risk analysts. A new software tool, the MAAP Reduced Order Model Generator, has been developed to simplify the process of creating these ROMs using a GUI that powers the parameter sampling, MAAP execution, results collection, and model querying.

2.3 Industry Use Case

A utility is working with Idaho National Laboratory to evaluate physical security changes and use MASS-DEF to determine and validate post reductions. It provided a MAAP model for its site for the scenario and the MAAP input file containing the parameters to modify. For the attack scenarios and additional prevention features, three parameters in the MAAP model were defined: the time auxiliary feedwater (AFW) pumps are turned on, the time AFW pumps are turned off, and the time a make-up pump is activated. A linear distribution for the times of each was used that would be in line with the plant procedures and the attack scenarios. The MAAP model assumes conservative values of initial conditions for applicable scenarios such as loss of offsite power.

3. METHODOLOGY

3.1 Problem Formulation

The EMRALD model can perturbate $\boldsymbol{n}$ MAAP input parameters, each with a specified distribution and distribution parameters. The combined input parameters cover an $\boldsymbol{n}$-dimensional space, where each possible combination of inputs is represented by the vector $\boldsymbol{x}$. Let $\boldsymbol{f}(\mathbf{x})$ denote the scalar MAAP output produced by the combination of $\boldsymbol{n}$ input parameters contained in $\boldsymbol{x}$. The objective of this software is to approximate $\boldsymbol{f}$ using a set of training pairs $\{(\boldsymbol{x_i}, \boldsymbol{f}(\boldsymbol{x})_{\boldsymbol{i}})\}$ that are collected by running MAAP at selected points with the input space. To reduce the computational load of gathering this data, the program attempts to achieve this using the smallest possible training set based on a user-defined tolerance, denoted as $\boldsymbol{\tau}$.

3.2 Hierarchal Sampling Strategy

The input space is represented using normalized points where the dimensions of each input parameter are mapped within the interval $[\mathbf{0}, \mathbf{100}]$. This mapping is dependent on the distribution of each input parameter. Two distributions are currently supported, but this methodology is applicable to any desired distribution.

- Uniform Distribution: For inputs that are uniformly distributed, the lower bound of the interval corresponds to the lower bound of the distribution, and the upper bound of the interval corresponds to the upper bound of the distribution. The halfway point along the interval, represented by the coordinate 50, would therefore correspond to the midpoint of the distribution.
- Normal Distribution: For inputs that are normally distributed, the position $\boldsymbol{p}$ within the interval corresponds to the $\boldsymbol{p}$/100th quantile of the given normal distribution using the inverse cumulative distribution function. The interval for normal distributions is restricted to $[\boldsymbol{\tau}, \mathbf{1} - \boldsymbol{\tau}]$, discarding extreme tails.

This normalized approach ensures that the logic traversing the space operates uniformly regardless of the scale, distribution, or units of individual parameters, and the convergence criterion is applied consistently across dimensions.

The sampling algorithm follows an iterative, hierarchical grid refinement strategy. The algorithm begins at the coarsest possible level of search and increases the fidelity of the search with each successive iteration. This approach is similar to the limit surface search algorithm implemented in RAVEN, which has previously been validated for producing reduced order models of plant behavior for the case of a PWR station blackout using RELAP-7 [5]. RAVEN was not suitable for this project as the development of this tool required deeper customization, including an interface for querying the model from EMRALD. The algorithm is as follows:

1. *Initialization* — A uniform and coarse grid is constructed over the $\boldsymbol{n}$-dimensional hypercube consisting of the domains of the $\boldsymbol{n}$ input parameters, $[\mathbf{0}, \mathbf{100}]^{\boldsymbol{n}}$. At the first, coarsest level, only the endpoints $\{\mathbf{0}, \mathbf{100}\}$ are considered for a total of $\mathbf{2}^{\boldsymbol{n}}$ points. The coarseness is internally represented as the "fidelity level," beginning with a value of 100. The algorithm iterates through decreasing fidelity levels, dividing by two each iteration. This value directly represents the spacing between points in the sample space. On the second iteration, with a value of 50, points at $\{\mathbf{0}, \mathbf{50}, \mathbf{100}\}$ are considered, for a total of $\mathbf{3}^{\boldsymbol{n}}$ points, and so on. This functionally means that in the next iteration, one point between each point in the current level will be considered.
2. *Iteration and Refinement* — MAAP is run for each point under consideration, with values for the input parameters corresponding to the coordinates of the point and the normalization described above. The maximum number of levels is limited by a user-defined parameter $\boldsymbol{d}$, which corresponds to a maximum fidelity of $\mathbf{100}/\mathbf{2}^{\boldsymbol{d}}$. The purpose of this parameter is to guarantee that the search will end in a finite amount of time. At each fidelity level $\boldsymbol{h}$, the grid contains $\left(\frac{\mathbf{100}}{\boldsymbol{h}} + \mathbf{1}\right)^{\boldsymbol{n}}$ total points. However, the algorithm limits the number of points that need to invoke MAAP by not evaluating points that have previously been evaluated, and by eliminating space within points, described next.
3. *Convergence Detection* — After evaluating all points under consideration for a given fidelity level, each pair of axis-aligned neighbors within the grid is examined. If $\left|\boldsymbol{f}(\boldsymbol{x_i}) - \boldsymbol{f}(\boldsymbol{x_j})\right| \leq \boldsymbol{\tau}$, the box defined by $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ is marked as eliminated space. In other words, if two neighboring pairs of inputs produce MAAP results that fall within the user-defined tolerance, the algorithm assumes the space between those points is flat and it does not need to run MAAP for any more points that fall within that space.
4. *Termination* — The main loop terminates when either all points at the current fidelity level fall within already evaluated or eliminated subspaces, or the fidelity level falls below the user-defined depth limit.

As the algorithm finishes evaluating each level, a model is output using the data collected so far. This allows the analyst to test the model before a termination condition is met and determine whether the algorithm needs to continue running to produce results with an acceptable margin of error. When a MAAP run successfully completes, extra artifacts of the run are deleted to prevent excessive use of space, and a single file named "summary" is produced that contains the values of the input parameters and the ultimate MAAP result. The result is currently read from the .D69 output file of the MAAP dose module, with the result being the calculated time to core damage (if any) stored in that file. Allowing the user to define exactly where results are read from is a planned future enhancement to the software.

Depending on the shape of the MAAP outputs, this adaptive approach can greatly reduce the time required to gather data to construct the model. This is, however, a potential weakness of the approach when considering general applications, as sharp peaks within the MAAP output shape will not be captured. The existence of such peaks is theoretically dependent on what MAAP parameters are being perturbated, but for the application to physical security modeling, the response to our parameters of interest has not exhibited this behavior.

3.3 Reduced Order Model

The result of the algorithm is a ROM that takes only the $\boldsymbol{n}$ input parameters and produces an interpolation of MAAP's output that corresponds to those parameters. This model is represented as a JSON file that contains two arrays:

- *X values* — An $\boldsymbol{m} \times \boldsymbol{n}$ matrix of sampled parameter vectors of length $\boldsymbol{n}$ for all $\boldsymbol{m}$-evaluated points.
- *Y values* — A vector of length $\boldsymbol{m}$ containing the corresponding MAAP outputs.

This model is used on demand to query results. Because the model itself is a simple representation of sampled data, it is agnostic toward the regression model used to interpolate results. Currently, k-nearest neighbors (KNN) and random forest models are available, and any other model desired by the user can be added. A utility is included within the software to compute the root mean square error (RMSE) of the data model in combination with the selected regression model, to provide users with a sense of the fit of the regression model to their data.

4. IMPLEMENTATION

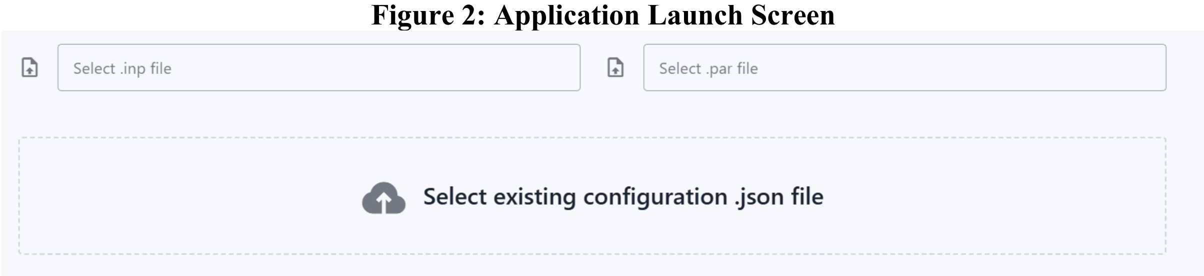
4.1 Model Preparation

A MAAP input file with all variables defined and used in applicable conditions, such as turning AFW pumps on and off, needs to be developed for the scenario. The time required to collect samples for the model is exponentially related to the number of variables selected to sample upon. The user can simplify the MAAP model to reduce the number of variables. For example, AFW pumps can be modeled as actuating simultaneously to reduce the number of variables needed to sample their behavior to just two, as having any one pump on almost maximizes the flow. Simplifications like this are not required but will reduce the computational cost of creating the model without sacrificing significant accuracy.

4.2 User Interface

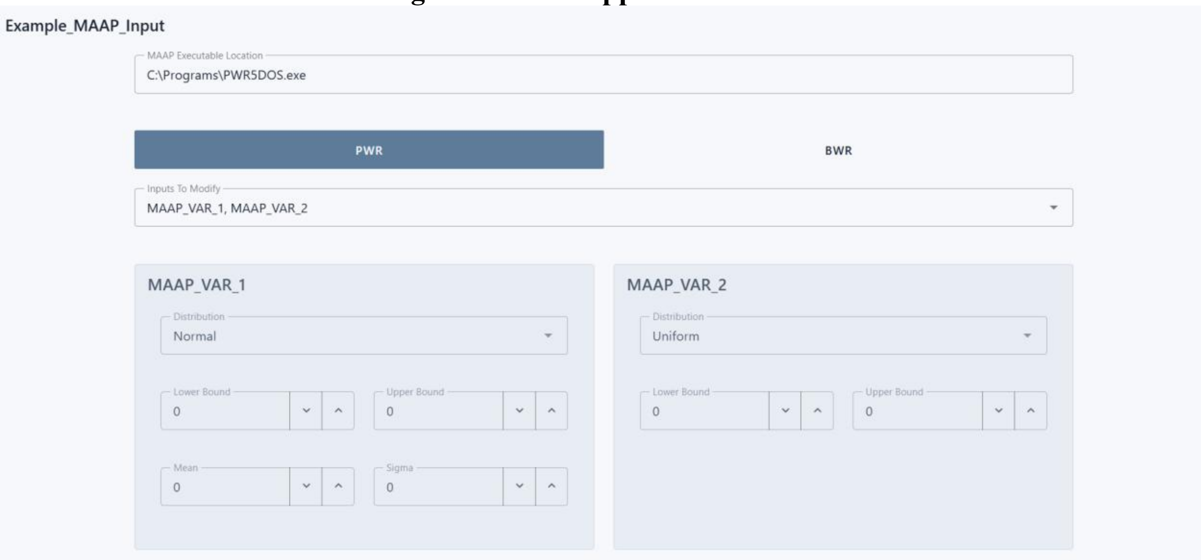
The entry point of the software is a desktop application that includes both a user interface for running the sampling and a command line interface (CLI) for querying the model, as well as advanced options not available through the GUI. Upon launching the application, the user has two options, shown in Figure 2: begin a new configuration from scratch by providing the MAAP input (.inp) and parameter (.par) files, or open an existing configuration previously created through the GUI.

Figure 2: Application Launch Screen



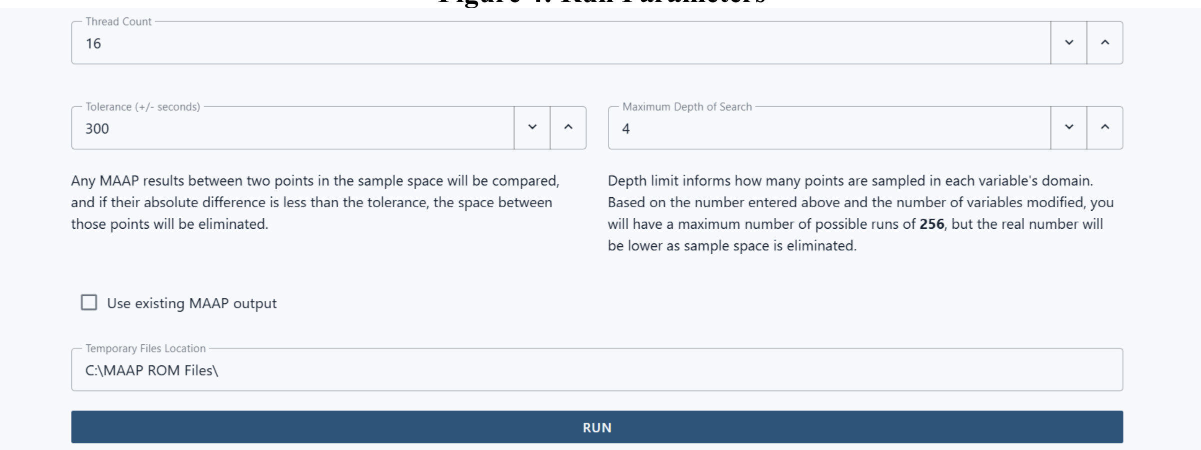
The MAAP input file is selected here, and variables assigned in the PARAMETER CHANGE section are presented as options to sample upon. When both an input and parameter file or existing configuration have been opened, the application proceeds to the main screen, shown in Figure 3. As MAAP is licensed software, the first input required of the user is the path to a local copy of the executable on their machine. As there are slight differences in the outputs from MAAP between the PWR and BWR versions of the code, the user can select which format is applicable to their model. Next is a list of variables the user may select to sample, any of which may be selected. A card that corresponds to each selected variable will display below it, where the user can input distribution type and distribution parameters. The distributions and parameters that are input here will define the points used in the sample space for the algorithm. For each MAAP run, a unique copy of the input file will be generated that is identical to the original, except the sampled values of the selected variables will be replaced in their assignment in the PARAMETER CHANGE section.

Figure 3: Main Application Screen



After the input parameters are configured, the next section contains options for constructing the model, shown in Figure 4. The software supports parallel runs, the maximum number of which can be specified in the thread count input. The run will utilize up to this many threads concurrently but may use fewer at times depending on the number of points under consideration. Each parallel run operates independently in its own directory, with separate copies of the MAAP executable, input, and parameter files. Next, the algorithm parameters of tolerance $\boldsymbol{\tau}$ and maximum depth $\boldsymbol{d}$ are entered.

Figure 4: Run Parameters



The interface includes a checkbox titled "Use existing MAAP output." When checked, the algorithm runner will look for existing run directories and summary files, and if they exist it will not rerun MAAP to evaluate that point. This option naturally has no effect on the first run but allows a run to begin again after being interrupted without needing to regenerate results. The final text box allows the user to define the path where the individual run directories will be stored and executed, which defaults to the location of the interface executable file.

At the end of the screen is the Run button, which will begin the model construction using the previously entered configuration. The configuration is saved in its entirety in JSON format to a directory named run_config, alongside the templated input and parameter files. For future runs or for resuming a run in the event it is interrupted, the corresponding JSON file can be selected on the first screen to restore all values entered for this run's configuration. Additionally, selecting an existing JSON file will automatically enable the use of an existing output option for convenience. Beginning the run will open a window within the interface that pipes output from the runs, both messages from the sampler and the individual MAAP runs, allowing the user to monitor progress and see potential warnings or errors.

4.3 Industry Pilot Model

The surrogate generation tool was used with the MAAP .par and .inp files provided by the industry collaborator. The three parameters were assigned with linear distributions. The tool ran for approximately 5 days on a 32 core 2.1 GHz machine to develop a model with a target convergence of 300 sec and a depth limit of 4.

4.4 Command Line Interface

A CLI is included in the same executable file as the GUI. The CLI provides additional functionality; most importantly it queries the generated model. Model querying is integrated into EMRALD through a Run Application action that invokes the CLI with the following arguments:

- --model: The path to the model file output by the interface / sampler tool.
- --values: A list of sampled values for each variable that was selected to generate the model.
- --output: A path to a file to write the result to.

Invoking the application with these CLI options will read the collected data into the application from the given model file and train the desired regression model on that data. The values provided will then be passed to the regressor for interpolation. The interpolated result is printed to the console and the specified local file, which can then be read back into EMRALD through the use of a Document Link variable.

Additional functionality provided through command line options includes:

- --config: Run the model generation using the configuration in the given file. This is the same functionality as running the tool through the GUI.
- --clean: Clean up residual run artifacts in all run directories.
- --validate: When used in combination with "model," this option will calculate the RMSE of the model. The calculation withholds a random 10% of values from the model data and trains the regression model on the remaining 90%. The RMSE is calculated using the withheld 10% as known values.

5. RESULTS

The adaptive hierarchical sampling approach provides two efficiency gains over traditional sampling strategies:

1. Reduced MAAP evaluations: By eliminating subspaces that have converged according to the user's specified tolerance before refining the grid, the algorithm avoids evaluating MAAP in regions where the shape of the output is flat and therefore easily interpolated within by a regression model. In scenarios where the output is dominated heavily by one or two parameters and is generally smooth, the eliminated space greatly reduces the number of MAAP runs required to build a model without losing notable accuracy in the predicted results. This allows the majority of computation time to focus on turbulent areas in the domains of the inputs. The

algorithm is also capable of handling rough output shapes as well, although with diminished time savings, and as previously discussed it may develop blind spots to sudden peaks within an otherwise smooth area of output. In the application of thermal hydraulic simulation for physical security, MAAP's response to a low number of variable inputs is smooth enough to benefit greatly from this strategy.

2. Reuse of prior results: Other available sampling implementations do not account for long runs being interrupted and can jeopardize days' worth of results if the machine shuts off in the middle of a run. The option to use existing output provided by this software ensures that the sampling can pick up immediately where it left off, resulting in no wasted computation time. Additionally, this enables the user to rerun the same configuration with higher fidelity to get more accurate results later without having to redo all the work on prior levels.

The industry pilot was able to run 100 simulations in less than 1 minute, whereas it took more than 3 days in the previous case study for MASS-DEF to run the final model, not counting the days to run each minor change when the physical security model coupled to it was tested and calibrated. The target convergence τ was set to 5 minutes, and the depth limit was set to 5. With these parameters, using random forest with 200 trees as a regressor produced a model with a margin of error of $5{:}30 \pm 0{:}18$ at a 95% confidence level. This indicates that for a particular call to the ROM, the value predicted is within 5:48 seconds of the value MAAP would produce if it had been run with the same parameters.

The output of collected data as a stand-alone JSON file additionally offers several advantages. The user has the freedom to select the regression model that suits their data and modify the parameters of the regression model as needed. The performance of model querying is entirely dependent on the performance of the regression model. For KNN, the results are nearly instant, but other models may introduce lag time when queried repeatedly by EMRALD. In this scenario, the JSON data file enables the user to save a pretrained model using other technologies such as pickle or ONNX files. Including pretraining as an option within the CLI is planned future work. Storing the abstracted data in the model file eases restrictions around analysis in two ways. First, a MAAP expert does not need to be present for the results to be used in dynamic PRA analysis. Second, the data can be analyzed outside of a secure environment.

6. CONCLUSION

The MAAP Reduced Order Model Generator software automates the construction of ROMs for MAAP thermal hydraulic simulations with an adaptive hierarchical sampling strategy and interpolation. Using a convergence-based space elimination strategy with parallelized subprocess execution, and incremental model outputs, the tool minimizes the number of computationally expensive MAAP runs required to characterize the parameter space of interest. The resulting ROM can be queried in real time via a CLI designed to integrate directly with EMRALD, enabling dynamic PRA analysis of physical security that would otherwise be practically impossible. Future work will include configuring the MAAP output location and pretraining regression models for increased performance and enabling the tool to automatically and intelligently select regression models.

The pilot project successfully developed a surrogate model with real plant data and physical security scenarios connected to run through EMRALD. Post-optimization work is ongoing, and the site plans to use the results to develop a change package that will be included in physical security plan changes.

Acknowledgements
The researchers thank RhinoCorps and the Palo Verde Generating Station for their collaboration.

represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

References

[1] Electric Power Research Institute. "*Modular Accident Analysis Program 5 (MAAP5) Applications Guidance: Desktop Reference for Using MAAP5 Software—Phase 3 Report*," Electric Power Research Institute, November 2017, Palo Alto, CA. Available at: https://www.epri.com/research/products/000000003002010658 (last accessed May 26, 2026).

[2] S. Prescott, R. Christian, S. W. St. Germain, V. Yadav, and C. P. Chwasz. "*An Evaluation of The Dynamic Physical Security Risk Assessment Methodology for Fleet-Wide Applications*," INL/RPT-24-80303, Idaho National Laboratory, September 2024, Idaho Falls, ID.

[3] S. Prescott, D. Nevius, Z. Ma, and S. Lawrence. "*Using EMRALD to Simplify and Perform Dynamic Analysis with MAAP*," Probabilistic Safety Assessment and Management (PSAM) 16, June 2022, Honolulu, Hawaii, USA (2022). Available at: https://www.iapsam.org/PSAM16/papers/ST93-PSAM16.pdf (last accessed May 20, 2026).

[4] Idaho National Laboratory. "*Technology Transfer Annual Report 2025*," Idaho National Laboratory, September 2025, Idaho Falls, ID. Available at: https://inl.gov/content/uploads/2023/07/25-50922_2025_TD-Report_R4.pdf (last accessed May 26, 2026).

[5] A. Alfonsi, C. Rabiti, D. Mandelli, J. Cogliati, S. Sen, and C. Smith. "*Improving Limit Surface Search Algorithms in RAVEN Using Acceleration Schemes*," INL/EXT-15-36100, Idaho National Laboratory, July 2015, Idaho Falls, ID. Available at: https://www.osti.gov/servlets/purl/1244620.