

Extracting Interpretable Accident Conditions from Large Simulation Datasets via Time-Series Clustering and Rule Induction

Junyong Bae and Hyeonmin Kim*

Korea Atomic Energy Research Institute, 111 Daedeok-daero 989beon-gil, Yuseong-gu, Daejeon, Republic of Korea
junyong8090@kaeri.re.kr, hyeonmin@kaeri.re.kr

Keywords : Dynamic PSA, Time-series clustering, Variational autoencoder, Rule induction, SGTR

1. Introduction

Advances in computational power, together with thermal-hydraulic (TH) system code analysis and its deep-learning surrogates, have enabled the analysis of large grids of accident scenarios for dynamic probabilistic safety assessment and passive safety system analysis [1-2]. A full-factorial scenario grid, however, quickly produces datasets of thousands of simulated trajectories that no analyst can inspect one by one.

One possible approach for this is to identify the decision boundary between plant critical states (e.g., core damage or not). However, such an approach is limited for high-dimensional (i.e., more than four dimensions) cases. To address this issue, Bae et al. suggested the concept of a success box and its conversion into a readable event tree form [3], and Jo et al. extended it by optimizing the locations of the boxes to maximize the coverage of success conditions [4]. Each box represents a single success path, and as such, this approach can induce clusters of success scenarios and their rules. Such rules, however, describe only whether a scenario succeeds, and not how the plant behaves along the way.

A complementary line of work groups scenarios by their behavior instead. For instance, Jin et al. clustered severe accident scenarios by considering the temporal evolution of source term release [5]. To address the high dimensionality of the time series, an autoencoder was used to reduce the dimension, and the clusters were formed in the resulting latent space. However, this study considered only the trends of the output source term release and did not relate the resulting clusters back to the input scenario conditions.

In this study, we take both perspectives at once and propose a three-step post-processing pipeline that turns a large set of simulated trajectories into a small number of human-readable accident-condition regimes. First, motivated by the use of variational autoencoders (VAEs) in image and video generation, we employ a VAE to encode the output time series of each scenario into the latents [6]. Second, we cluster them by latent similarity to obtain groups of scenarios with similar output-trajectory shape. Third, we employ a classification and regression tree (CART) to induce a rule for each cluster over the input scenario conditions. The rule interprets cluster membership as a handful of readable if-then conditions on the inputs. We apply the

pipeline to a steam generator tube rupture (SGTR) TH simulation dataset [7].

The remainder of this paper is organized as follows. Section 2 introduces the related concepts including VAEs, clustering algorithms, and CART. Section 3 describes the proposed pipeline. Section 4 reports the SGTR case study, and Section 5 concludes the paper.

2. Related concepts

2.1. Variational Autoencoder

The high dimensionality of images and video frames has limited the application of computationally expensive operations such as diffusion and attention. As such, models that reduce the dimension while keeping the critical information have been researched, and VAEs are widely used, as in latent diffusion models [8].

The basic structure of an autoencoder is an encoder and a decoder with a mirror-symmetric structure. Since the training objective is to reconstruct the original input at the output side, latents keeping the information are generated at the bottleneck between them. Unlike a plain autoencoder, a VAE encodes an input not into a single point but into a normal distribution, whose mean and standard deviation are the outputs of the encoder, and the latent is sampled from that distribution during training.

Recently, alternative schemes such as the vector-quantized VAE, which discretizes the latent space, and representations obtained with a task-relevant objective, such as the joint-embedding predictive architecture (JEPA), have also been researched. For nuclear time-series datasets, Bae et al. utilized a VAE to obtain the latents and built a model that predicts the latents for query conditions, rather than directly predicting the time series [6]. In this study, a VAE is trained on the simulated trajectories, and the resulting latents are used as the representation on which scenarios are clustered.

2.2. Clustering

Clustering is an unsupervised learning method that groups a dataset by the similarity between its points, without labels given in advance. Algorithms differ in whether the number of groups has to be specified. For instance, k -means partitions a set of points into k groups by minimizing within-cluster variance. The number of clusters is commonly chosen by maximizing the

silhouette coefficient, which scores each point by how much closer it is to its own cluster than to the nearest other cluster, averaged over all points.

Density-based spatial clustering of applications with noise (DBSCAN), on the other hand, does not take the number of clusters and uses density instead. For each point, it counts the points within a given radius, and a point with at least a minimum number of neighbors is regarded as a core point. Core points close to each other are chained into one cluster, so clusters of arbitrary shape can be found, and points near no core point are left as noise instead of being forced into the nearest group. The number of clusters is therefore determined by the data, though the result is sensitive to the radius, and a single setting of these parameters does not suit clusters of differing density. Since the number of scenario groups is not known in advance, both k-means and a density-based method are applied in this study and their results are compared.

2.3. Classification and Regression Tree (CART)

A CART is a supervised learning method that predicts a label by asking a series of questions on single features. For a given dataset, it searches every feature and every threshold on it and selects the split that separates the labels most purely, commonly measured by the Gini index. The two subsets produced by the split are then split again in the same manner, and the recursion is stopped by a limit on the depth or on the number of samples in a node, so that the tree does not grow into one leaf per sample.

The benefit of CART is its readability. A path from the root to a leaf is a conjunction of threshold conditions on the features, so each leaf can be explained by a rule with a handful of if-then conditions. For this reason, CART is employed in this study to induce a rule for each cluster derived from the latent clustering.

3. Framework

Figure 1 outlines the proposed pipeline, which consists of three steps: VAE training, latent clustering, and rule induction.

VAE training. A VAE is trained on the output trajectories of the simulated scenarios, so that each trajectory is encoded into a low-dimensional latent vector. The encoder and the decoder are trained together to reconstruct the trajectory, while the latents are regularized toward a normal distribution. After training, only the encoder is used, and the mean of the encoded distribution is taken as the deterministic latent vector of the scenario, so that a single point in the latent space stands for one trajectory.

Latent clustering. The latent vectors are standardized per dimension and clustered by the algorithms described in Section 2.2. For k-means, the

number of clusters is selected by sweeping a candidate range and examining the silhouette coefficient.

For the density-based method, we used HDBSCAN [9]. Compared with DBSCAN, it alleviates the burden of hyperparameter choice by building a hierarchy of clusters over the whole range of radii and extracting the final partition from different levels at once. The hierarchy therefore allows dense and sparse groups to be obtained together. Afterward, clusters whose latent centroids lie closer than a threshold are merged to handle duplicated trajectories.

Rule induction. Each cluster is explained by a one-vs-rest CART, with cluster membership as the binary label. To feed the time-series input conditions to CART, we reduce each of them to two scalar features. The first is its initial value, and the second is the time of its first departure from that value. If the condition never departs, the time is set to the full horizon, i.e., the length of the input time series.

To obtain a readable rule for each cluster, we limit the tree depth and set a minimum number of samples per leaf. We also give more weight to the scenarios in the cluster, in inverse proportion to the class sizes, to alleviate the class imbalance of the one-vs-rest setting.

After training, we extract the conditions along the path to each leaf that predicts the cluster and prune them. Conditions on the same feature are merged into one interval, and a split whose branches both predict the cluster is removed, since it does not change the prediction and only raises the purity. Positive leaves whose purity falls below a threshold are then discarded. Finally, the retained rules are scored as a binary classifier against the cluster assignment, and the resulting precision and recall show how faithfully the rules describe the cluster.

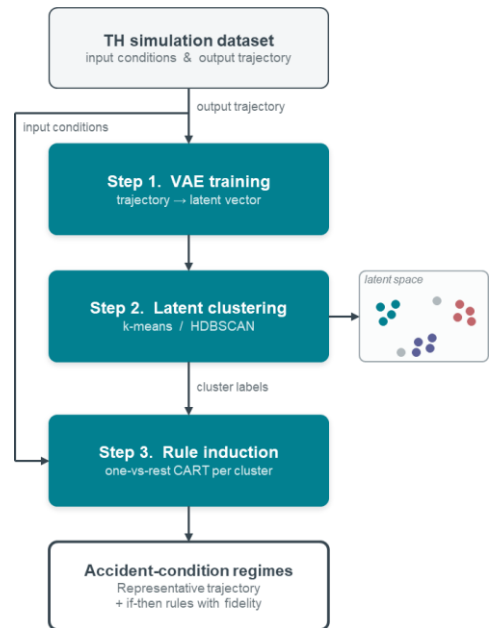


Fig. 1. The proposed pipeline.

4. Case study

The pipeline was applied to an SGTR TH simulation dataset of 6,555 scenarios, which collapse to 2,698 unique input trajectories once the exact duplicates of the full-factorial grid are removed. Each scenario has 171 input channels. Among them, 78 vary within the dataset and the rest stay constant, so the rule conditions can only come from the varying channels. The output target is the steam generator 2 pressure (SG2), recorded over 1,200 time steps (= 200 mins), and Figure 2 shows the trajectories of all scenarios.

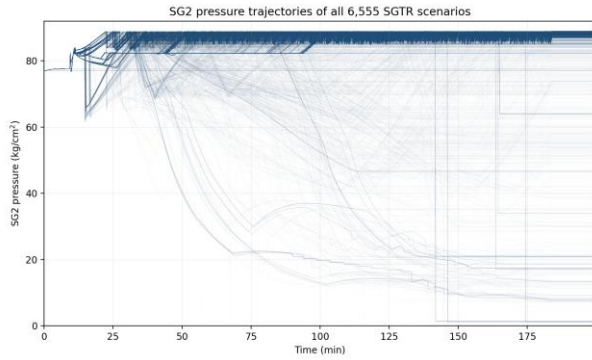


Fig. 2. Overall trajectories of SG2 pressure under SGTR scenarios

The VAE was constructed as a one-dimensional convolutional network with a latent dimension of 32 and trained on the SG2 trajectories with a mean-absolute-error reconstruction loss and a regularization loss. The reconstruction error on the held-out scenarios was $2.6e-3$ in the scaled unit, and each trajectory was then encoded into its latent vector by the trained encoder. Figure 3 illustrates the resulting latent space.

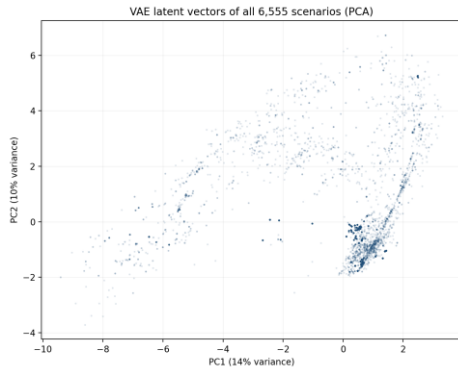


Fig. 3. PCA projection of the VAE latent vectors of all scenarios. Note that these two components explain only 24 percent of the variance.

The latent vectors were clustered by both methods. HDBSCAN with a minimum cluster size of 30 and a minimum sample count of 3, followed by the merge, found 65 clusters and left 19.6 percent of the scenarios as noise, with a silhouette coefficient of 0.68 over the clustered scenarios. Figure 4 shows the representative trends of the six largest clusters.

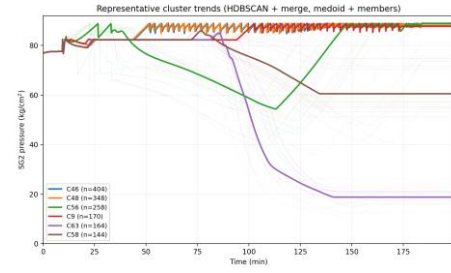


Fig. 4. Trends of the six largest clusters found by HDBSCAN.

The silhouette coefficient of k-means was swept over k up to 200, as shown in Figure 5. The coefficient increased steeply until around $k = 80$ and saturated near 0.60.



Fig. 5. Silhouette coefficient of k-means over the number of clusters.

Figure 6 shows the representative trends of the six largest clusters found by k-means with $k = 88$. Unlike HDBSCAN, none of them is a depressurization case, and they all stay in the band of the cyclic opening of the main steam safety valve (MSSV), which corresponds to the critical state defined in [7]. Further investigation shows that HDBSCAN gathers the depressurization scenarios into fewer and larger groups than k-means (18 versus 51 groups). Note that k-means assigns every scenario to a cluster and leaves no noise.

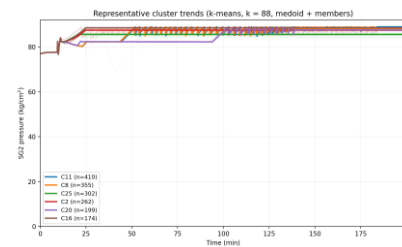


Fig. 6. Trends of the six largest clusters found by k-means.

The one-vs-rest CART was fitted for the six largest HDBSCAN clusters with a tree depth of six, a minimum of ten samples per leaf, and a purity threshold of 0.5. Table I shows the details of the six clusters displayed in Fig. 4. The recall ranged from 0.92 to 1.00 for all six clusters, and the precision ranged from 0.83 to 0.99 except for C56 at 0.58. Its rule recovers almost all members of the cluster (recall of 0.97) but also admits about 180 other scenarios as false positives, half of which are noise scenarios adjacent to the cluster in the latent space.

Table I. Induced rules for the six largest clusters

Cluster	N	Rules	Conditions	Precision	Recall
C46	404	1	4	0.98	1.00
C48	348	1	3	0.97	1.00
C56	258	1	4	0.58	0.97
C9	170	1	4	0.99	1.00
C63	164	2	3	0.91	1.00
C58	144	3	3	0.83	0.92

We further converted these cluster rules into natural language with an LLM (Claude), and the descriptions were verified against the signal definitions as follows.

C46 (Cyclic) One main steam isolation valve (MSIV) closes within 122 min while the other closes only after 36 min, low-pressure safety injection starts within 10 min, and the charging line isolates only after 157 min.

C48 (Cyclic) One main feedwater isolation valve (MFIV) closes within 63 min, the downcomer feedwater control valve actuates only after 105 min, and the charging line isolates within 10 min.

C56 (Depressurization and cyclic) Auxiliary feedwater starts within 56 min and the reactor coolant pump trip occurs only after 158 min. The rule region also covers many noise scenarios, which lowers the precision to 0.58.

C9 (Cyclic) One MFIV closes within 147 min, while the letdown line changes only after 152 min and the MSIV signal appears only after 94 min.

C63 (Depressurization) An MSSV signal appears within 88 min, auxiliary feedwater starts only after 92 min, and the MSIV signal appears only after 62 min.

C58 (Depressurization) Scenarios similar to C63 but with auxiliary feedwater within 94 min, split into three sub-rules by the exact timing combination.

5. Conclusion

In this research, we proposed a data-driven post-processing method that combines a VAE, clustering, and CART. The case study on an SGTR simulation database showed that the method can organize thousands of simulated trajectories without manual inspection and induce readable rules for the resulting scenario groups.

Two applications look promising. First, the pipeline can support operators when combined with real-time prediction of the plant status. The rise of data-driven models has enabled the anticipation of various future scenarios, but the practical bottleneck is deriving the guidance finally delivered to the operator, and the induced rules can fill this gap as a real-time advisory for emergency responses. Second, the pipeline can support safety analysis of the plant in terms of dynamic probabilistic safety assessment, where the induced rules group the accident scenarios and yield the success criteria.

However, this research is a proof of concept with substantial limitations. Technically, the latents are trained for reconstruction, which does not guarantee their relevance to the grouping task. Adversarial losses from image generation, prototypical learning, or JEPA can be adopted to obtain more task-relevant latents. Another drawback is the impurity of the classification. As shown in Table I, the induced rules allow substantial misclassification. Considering the safety-critical nature of nuclear power plants, such impurity can be unacceptable. One possible remedy is conservative clustering, which allows false positives only toward the critical cases such as core damage. A further limitation concerns the treatment of response scenarios, which take the form of time-series records of control signals. Based on our investigation of the data, this research considers only the initial value of each signal and the time at which it is activated. A more general representation of such scenarios, in a form that the rule induction algorithm can handle, should therefore be devised.

ACKNOWLEDGEMENTS

This research was supported by the National Research Council of Science & Technology (NST) grant by the Korea government (MSIT) (No. GTL24033-400)

REFERENCES

- [1] J. Bae, J. W. Park, and S. J. Lee, Limit surface/states searching algorithm with a deep neural network and Monte Carlo dropout for nuclear power plant safety assessment, *Applied Soft Computing*, Vol. 124, pp. 109007, 2022.
- [2] T. Glingler et al., Dynamic event tree analysis of a severe accident sequence in a boiling water reactor experiencing a cyberattack scenario, *Annals of Nuclear Energy*, Vol. 192, pp. 109994, 2023.
- [3] J. Bae et al., Postprocessing of Multidimensional Limit Surfaces Searched by a Deep Learning-based Algorithm, *Transactions of the Korean Nuclear Society Autumn Meeting*, Gyeongju, Korea, 2023.
- [4] W. Jo et al., Automatic accident sequence generation using optimized simulations for dynamic probabilistic safety assessment, *Reliability Engineering & System Safety*, Vol. 266, pp. 111643, 2026.
- [5] K. Jin, J. Cho, and S. Kim, Machine learning-based categorization of source terms for risk assessment of nuclear power plants, *Nuclear Engineering and Technology*, Vol. 54, No. 9, pp. 3336-3346, 2022.
- [6] J. Bae, K. S. Kim, and S. J. Lee, Two-stage deep learning model for nuclear power plant parameter trend prediction with epistemic uncertainty quantification, *Applied Soft Computing*, Vol. 186, pp. 114233, 2026.
- [7] J. Park and H. Kim, A case study to address the limitation of accident scenario identifications with respect to diverse manual responses, *Reliability Engineering & System Safety*, Vol. 251, pp. 110406, 2024.
- [8] R. Rombach et al., High-Resolution Image Synthesis with Latent Diffusion Models, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684-10695, 2022.
- [9] R. J. G. B. Campello, D. Moulavi, and J. Sander, Density-Based Clustering Based on Hierarchical Density Estimates, *Advances in Knowledge Discovery and Data Mining (PAKDD)*, *Lecture Notes in Computer Science*, Vol. 7819, pp. 160-172, 2013.