

- Scalability - how well can this algorithm perform with

large data sets?

- Arbitrary-Shaped Clusters - can this algorithm find clusters that are non-spherical?
- Parameters - how many user-defined parameters are required? How sensitive are these parameters?
- Noisy Data - how does this algorithm handle deviations in the data?
- High Dimensionality - will preprocessing of high-dimensional data be necessary?

Several clustering methods have been previously used with event trees. Fuzzy c-means is an algorithm that has been used to cluster the data from a DDET for a steam generator tube rupture in a light water reactor [13]. This algorithm chooses initial centroids and iteratively creates clusters based on points surrounding the centroids. It then assigns points a membership weight between 0 and 1 depending on how much they belong to the cluster. While this method is useful when the boundaries between clusters are not well-defined, it is highly susceptible to noisy data and it is unable to identify clusters that are non-spherical.

The mean-shift (MS) algorithm has been used in DDETs for an airplane crash in a sodium fast reactor, as well as a station blackout in a pressurized water reactor [14][15]. MS works by choosing an arbitrary data point, defining a circular region around that point of with a radius equal to a user-defined bandwidth, and calculating the center of mass of the points in the region. This center of mass is then used as the new center point, and the process is repeated until the distance between the new center of mass and the old one is below a fixed threshold. While MS is able to identify non-spherical clusters, unlike fuzzy c-means, it is sensitive to both the choice of bandwidth and to noise in the data.

CURRENT APPROACH

Density Peaks Clustering with Dynamic Time Warping

Our proposed clustering algorithm is density peaks (DP), which was recently proposed in [16]. This algorithm has several advantages over other available clustering algorithms, including the following:

- DP is able to find clusters of arbitrary shapes.
- It only requires 2 user-defined parameters, neither of which are exceptionally sensitive.
- The algorithm has the ability to deal with noisy data, something that is particularly important with the data that is generated from the severe accident simulators.
- As we will discuss below, the distance measure used can be easily extended into multi-dimensions.

The DP algorithm works to cluster data points based on their local density. To determine cluster centers, two

values, ρ and δ , must be found for each point. The steps for these calculations are shown in Pseudo-Algorithm 1. The ρ calculation requires an input of a distance matrix, D , as well as a user-determined threshold, d_c . Lines 2-4 show that for each point in the matrix, the number of data points within d_c is determined and denoted as ρ_i . The values of all ρ_i are then sorted in descending order and this sorted list is used as an input for the δ calculation. Lines 6-9 show that for each point, the distance of the closest data point of higher density, δ_i , is found. Cluster centers are then defined as those points that have the highest values of $\rho_i * \delta_i$, as seen in line 11. After the cluster centers have been decided, points are assigned to the nearest neighbor cluster center from the list of points with higher density [17].

Algorithm 1 Density Peaks

```

1: procedure  $\rho\_CALCULATION(D, d_c)$ 
2:   for  $i = 1:n$  do
3:      $\rho(i) = \text{count}(D(i, \text{otherObjects}) < d_c)$ 
4:    $\text{ord\_}\rho = \text{sort}(\rho, 'descend')$ 
5: procedure  $\delta\_CALCULATION(\text{ord\_}\rho)$ 
6:   for  $i = 1:n$  do
7:      $\delta(\text{ord\_}\rho(i)) = \max(D(\text{ord\_}\rho(i)))$ 
8:     for  $j = 1:(i-1)$  do
9:        $\delta(\text{ord\_}\rho(i)) = \min(\text{dist}(\text{ord\_}\rho(i), \text{ord\_}\rho(j)))$ 
10: procedure  $\text{CLUSTERCENTERS}(\rho, \delta)$ 
11:    $\text{clusterCenters} = \text{top}(\text{sort}(\rho * \delta))$ 

```

Multi-Dimensional Dynamic Time Warping

For time series data, there are numerous distance measures that can be used to create the distance matrix that is used in the density peaks algorithm. Many in the research community have determined that Dynamic Time Warping (DTW) is a superior choice as a time series distance measure, and it has been found to regularly outperform the Euclidean Distance (ED) [17]. DTW is more robust than the ED due to the fact that it can identify similarities between two time series, even if there is a lag between the two series. Figures 2 and 3 show how clustering using DTW is able to find similar time series paths even though they are temporally offset.

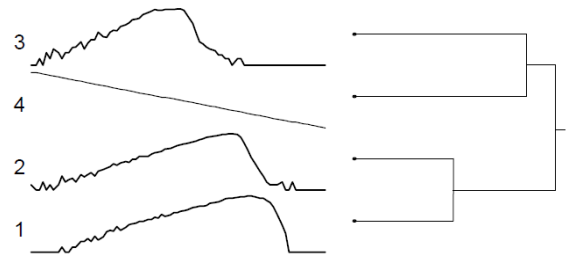


Fig. 2. Clustering using ED. Though series 3 is visually more similar to series 1 and 2, the ED clustering is unable to account for the time lag between them [18].

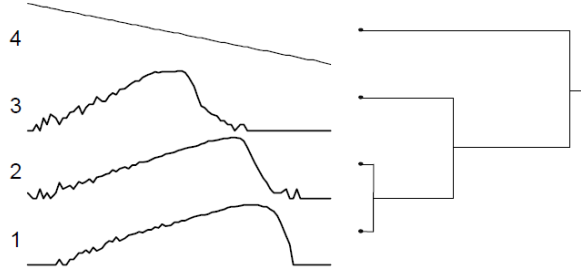


Fig. 3. Clustering using DTW. This is more intuitive in that it is able to identify that series 1,2 and 3 are similar [18].

DTW is a relatively simple algorithm that first calculates the pairwise Euclidean distance between each time step in two time series. These pairwise distances form a matrix, and the algorithm works to find an optimal path through the matrix that minimizes the Euclidean distance. The optimal path is formed as follows: let Q and C be two time series. Define $d(q_i, c_j)$ to be the squared Euclidean distance between points q_i and c_j . Then, we can create a distance matrix D as seen in Equation 1. Figure 4 shows an example of the optimal path that is formed through the distance matrix.

$$D(i, j) = d(q_i, c_j) + \min[D(i-1, j-1), D(i-1, j), D(i, j-1)] \quad (1)$$

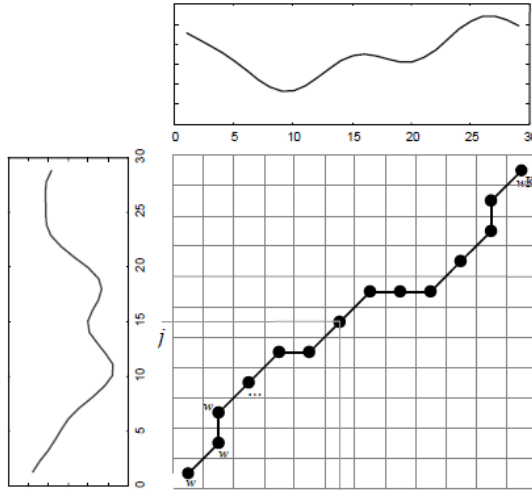


Fig. 4. An example of the optimal path through the pairwise distance matrix of two time series [18].

The extension to the multi-dimensional case is straightforward. In Multi-Dimensional Dynamic Time Warping (MD-DTW), it is assumed that each dimension is independent. The one-dimensional DTW distance is calculated for each dimension separately, and then those values are summed to get a total distance measure. If $D(Q_m, C_m)$ is the DTW distance of the m^{th} dimension of Q and C , then we can calculate the MD-DTW distance matrix as:

$$MD(Q, C) = \sum_{m=1}^M D(Q_m, C_m) \quad (2)$$

Pruning

Pruning will occur after cluster centers have been determined using the DP algorithm. During initial testing, pruning will be done on fully grown DDETs so that the full tree can be used as an evaluation tool against the pruned tree. Branches that are on the outer edge of clusters will be pruned. The number of branches pruned will be related to a user-defined acceptable amount of fidelity reduction in the final tree. Following the initial testing, the pruning method will be implemented during the growth of the tree.

PRELIMINARY RESULTS

A preliminary test was performed on a relatively small DDET with 219 total branches and 112 end state branches that was created using ADAPT and SAS4A. Each branch is comprised of 18 time-varying output variables. A MD-DTW distance matrix was formed at the final branch point of the tree, and the DP clustering algorithm was performed. Figure 5 shows the output values of four variables of interest - outlet temperature, inlet temperature, cool average temperature, and saturation temperature. The DP algorithm produced 7 clusters, shown in different colors in the figure.

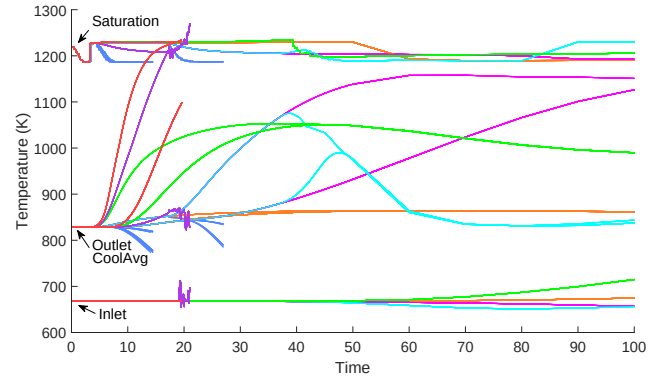


Fig. 5. Outlet temperature, inlet temperature, cool average temperature, and saturation temperature for branches clustered by color.

Figure 6 shows the first 30 seconds of the outlet and cool average temperatures, and it demonstrates the benefit of using DTW in the DP algorithm. The dark blue time series show similar paths; they would intuitively be clustered together even though they are separated by approximately 10 seconds. Since DTW is able to account for temporal disparities, it is able to group the two series into the same cluster. A Euclidean-based distance matrix would not be able to adjust for any time lag between the two series.

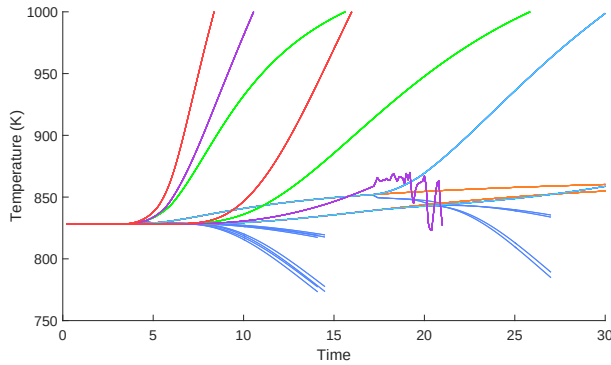


Fig. 6. The cluster in dark blue shows how dynamic time warping is able to find similar clusters at different times.

CONCLUSIONS

DDET is a valuable tool for understanding the risks present in a nuclear power plant severe accident. However, the rapid growth of a DDET poses significant computational requirements, as well as difficulties in interpreting the results. Therefore, a method to reduce the number of branches during the tree growth will help reduce the resources needed to perform the analysis. Further work in this study will determine the best method of choosing the distance threshold in the density peaks algorithm, d_c , as well as test the sensitivity of that choice. In addition, a metric will be defined that will help determine the effect of pruning on the overall predictive capability of the event tree.

ACKNOWLEDGEMENTS

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the US Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

NOMENCLATURE

ADAPT	Analysis of Dynamic Accident Progression Trees
DDET	Discrete Dynamic Event Tree
DP	Density Peaks
DTW	Dynamic Time Warping
ED	Euclidean Distance
MD-DTW	Multi-Dimensional Dynamic Time Warping
MELCOR	Not an acronym, Reactor Accident Code
MS	Mean-Shift
PRA	Probabilistic Risk Assessment
SCEPTRE	System for Circuit Evaluation and Prediction of Transient Radiation Effects
SNL	Sandia National Laboratories

REFERENCES

1. A. HAKOBYAN ET AL., "Dynamic Generation of Accident Progression Event Trees," *Nuclear Engineering and Design* (2007).
2. L. HUMPHRIES ET AL., "MELCOR Computer Code Manuals Vol. 1: Primer and Users' Guide," Tech. rep., Sandia National Laboratories, Albuquerque, NM (2015).
3. J. MAYO ET AL., "Approaches for Scalable Modeling and Emulation of Cyber Systems: LDRD Final Report," Tech. rep., Sandia National Laboratories, Livermore, CA (2009).
4. J. CARDONI ET AL., "Generic Severe Accident Model for Cyber Attack Scenarios," Submitted to American Nuclear Society, Las Vegas, NV (2016).
5. M. DENMAN ET AL., "Preliminary Cyber-Informed Dynamic Branch Conditions for Analysis with the Dynamic Simplified Cyber MELCOR Model," (2016).
6. Z. JANKOVSKY ET AL., "Extension of the ADAPT Framework for Multiple Simulators," (2016).
7. Z. JANKOVSKY ET AL., "Dynamic Importance Measures in the ADAPT Framework," (2016).
8. Z. JANKOVSKY ET AL., "Visual Tree Reduction in the ADAPT Framework," (2016).
9. S. GALUSHIN and P. KUDINOV, "Scenario Grouping and Classification Methodology for Postprocessing of Data Generated by Integrated Deterministic-Probabilistic Safety Analysis," (2015).
10. M. KEARNS and Y. MANSOUR, "A Fast, Bottom-Up Decision Tree Pruning Algorithm with Near-Optimal Generalization," (1998).
11. F. ESPOSITO, D. MALERBA, and G. SEMERARO, "A Comparative Analysis of Methods for Pruning Decision Trees," *IEEE* (1994).
12. A. ALFONSI ET AL., "Adaptive Dynamic Event Tree in RAVEN Code," Idaho National Laboratory, Idaho Falls, ID.
13. D. MERCURIO, L. PODOFILLINI, E. ZIO, and V. DANG, "Identification and Classification of Dynamic Event Tree Scenarios via Possibilistic Clustering: Application to a Steam Generator Tube Rupture Event," *Accident Modelling and Prevention*, **41** (2009).
14. D. OSBORN, T. ALDEMIR, R. DENNING, and D. MANDELLI, "Seamless Level 2/ Level 3 Dynamic PRA Clustering," Tech. rep., Sandia National Laboratories (2010).
15. D. MANDELLI and A. YILMAX, "Scenario Analysis and PRA: Overview and Lessons Learned," Tech. rep., The Ohio State University (2007).
16. A. RODRIGUEZ and A. LAIO, "Clustering by Fast Search and Find of Density Peaks," *Science*, **344** (2014).
17. N. BEGUM, L. ULANOVA, J. WANG, and E. KEOGH, "Accelerating Dynamic Time Warping Clustering with a Novel Admissible Pruning Strategy," *ACM* (2015).
18. E. J. KEOGH and M. J. PAZZANI, *Principles of Data Mining and Knowledge Discovery: Third European Conference*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–11 (1999).