ANS PSA 2013 International Topical Meeting on Probabilistic Safety Assessment and Analysis
Columbia, SC, September 22-26, 2013, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2013)

MINING NUCLEAR TRANSIENT DATA THROUGH SYMBOLIC CONVERSION

D. Mandelli and C. Smith
Idaho National Laboratory (INL)
2525 Freemont Ave., Idaho Falls (ID) 83415
diego.mandelli@inl.gov; curtis.smith@inl.gov

A. Yilmaz and T. Aldemir
The Ohio State University
281 W. Lane Ave, Columbus (OH) 43210
aldemir.1@osu.edu; yilmaz.15@osu.edu

ABSTRACT

Dynamic Probabilistic Risk Assessment (DPRA) methodologies generate enormous amounts of data for a very large number of simulations. The data contain temporal information of both the state variables of the simulator and the temporal status of specific systems/components. In order to measure system performances, limitations and resilience, such data need to be carefully analyzed with the objective of discovering the correlations between sequence/timing of events and system dynamics. A first approach toward discovering these correlations from data generated by DPRA methodologies has been performed by organizing scenarios into groups using classification or clustering based algorithms. The identification of the correlations between system dynamics and timing/sequencing of events is performed by observing the temporal distribution of these events in each group of scenarios. Instead of performing "a posteriori" analysis of these correlations, this paper shows how it is possible to identify the correlations implicitly by performing a symbolic conversion of both continuous (temporal profiles of simulator state variables) and discrete (status of systems and components) data. Symbolic conversion is performed for each simulation by properly quantizing both continuous and discrete data and then converting them as a series of symbols. After merging both series together, a temporal phrase is obtained. This phrase preserves duration, coincidence and sequence of both continuous and discrete data in a uniform and consistent manner. In this paper it is also shown that by using specific distance measures, it is still possible to post-process such symbolic data using clustering and classification techniques but in considerably less time since the memory needed to store the data is greatly reduced by the symbolic conversion.

Key Words: Data Mining, Time Series, Symbolic Conversion, Dynamic PRA

1 INTRODUCTION

State of the art PRA methods, i.e. dynamic PRA (DPRA) methodologies [1], largely employ system simulators codes to accurately model system dynamics. Typically, these system simulator codes (e.g., RELAP [2], MELCOR [3], MAACS [4]) are coupled with other codes (e.g., ADAPT [5], ADS [6], RAVEN [7,8], MCDT [22]) that monitor and control the simulation. The latter codes, in particular, introduce both deterministic (e.g., system control logic, operating procedures) and stochastic (e.g., component failures, variable uncertainties) elements into the simulation.

The overall DPRA analysis is performed by 1) sampling values of a set of parameters from the uncertainty space of interest, and 2) simulating the system behavior for that specific set of parameter values. For complex systems, such analyses typically require a very large number of simulations and, thus, a large amount of data are generated [9].

The data are typically very heterogeneous (see Fig. 1), i.e., it consists of two datasets:

- *Continuous data*: which contains temporal profiles of state variables $\theta(t)$ (e.g., temperature, pressure of specific nodes of the simulator)
- *Discrete data*: which contains timing of events. Note that a generic event A_i can occur:
    - at a time instant t_{A_i}, i.e., the event can be defined as: (A_i, t_{A_i}), or,
    - over a time interval $[t_{A_i,\alpha}, t_{A_i,\omega}]$, i.e., the event can be defined as: $(A_i, [t_{A_i,\alpha}, t_{A_i,\omega}])$

In traditional static PRA (fault-tree and event-tree based) [10] timings such as t_{A_i} or $[t_{A_i,\alpha}, t_{A_i,\omega}]$ are not explicitly considered[1] and data analysis is performed by considering the Cut Sets or the Prime Implicants (see Fig. 1). However, in a DPRA framework, the temporal scale is implicitly modeled and, thus, events are defined over a time instant or an interval of occurrence.

The ability to analyze and identify correlations among timing of events through system dynamics/software/human action interactions is essential for nuclear power plant safety analysis and post-processing of the data generated by DPRA methodologies towards such a purpose is still a research topic.

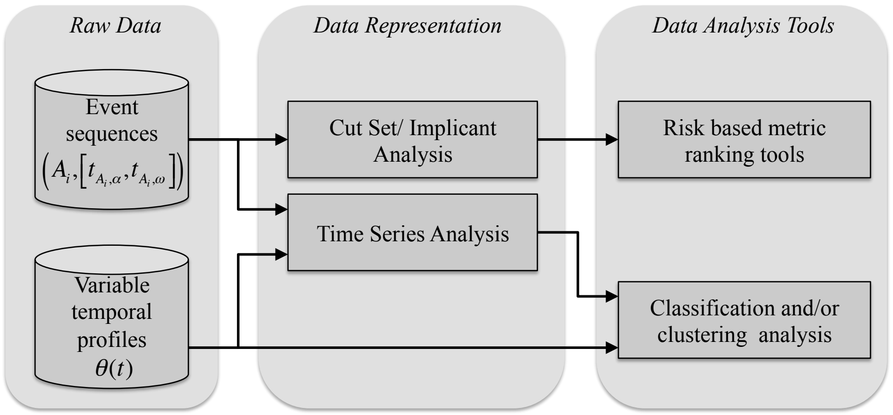

Raw Data
Data Representation
Data Analysis Tools
Event sequences $(A_i,[t_{A_i,\alpha},t_{A_i,\omega}])$
Variable temporal profiles $\theta(t)$
Cut Set/ Implicant Analysis
Time Series Analysis
Risk based metric ranking tools
Classification and/or clustering analysis

Figure 1. Overview of data analysis for PRA applications

A first approach toward discovering these correlations from data generated by DPRA methodologies has been developed using Fuzzy classification [12] and clustering [9] algorithms (see Fig. 1). In particular, clustering algorithms have allowed users to fully analyze these correlations by considering the complete system dynamics (i.e., continuous data) and not only

[1] Sequence of events is indeed considered (e.g., sequencing is embedded in the construction of event-trees) but no timing information (i.e., time of activation or duration of an event) is included

the final outcome[2] [9]. Clustering based algorithms can be used to identify groups (i.e., clusters) of scenarios having similar temporal behavior of the state variables. In [9] only continuous data are used to represent each scenario while discrete data are considered after the clustering process to identify the set of events that caused a similar temporal behavior.

This paper presents an alternative method to represent each scenario that can help the user to extract knowledge from large volume of data for DPRA applications by considering both continuous and discrete data simultaneously and in a coherent fashion. This alternative still considers the raw data as time series (see Fig. 1) as shown in [9] but it performs a symbolic conversion on both continuous and discrete data. This is accomplished by quantizing both datasets and by associating to each quantized element a symbol (see Sections 2 and 3). Section 4 will show how it is still possible to perform classical data mining applications[3] (i.e., clustering and classification) by considering specific metrics. Section 4 will also show how such symbolic conversion drastically improves computational performances for classical data mining applications both in terms of memory requirements and computational time.

2 TIME SERIES ANALYSIS

The scope of our time series algorithm is to perform the analysis of scenarios by considering both continuous and discrete types of data (see Section 2.1) in a coherent fashion. This is performed by symbolically representing both continuous and discrete datasets. Symbolic representation means that the data are transformed into a series of symbols.

Two algorithms are being used (see Fig. 2):

1. A modified version of SAX [13] which symbolically converts continuous types of data (see Section 2.1)
2. Time Series Knowledge Representation (TSKR) [14] which symbolically converts discrete types of data (see Section 2.2)

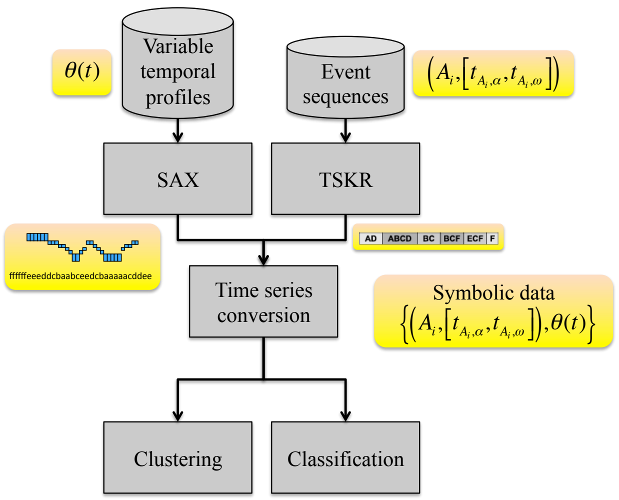

$\theta(t)$
Variable temporal profiles
Event sequences
$\left(A_i,\left[t_{A_i,\alpha},t_{A_i,\omega}\right]\right)$
SAX
TSKR
ffffffeeeddcbaabceedcbaaaaacddee
AD ABCD BC BCF ECF F
Time series conversion
Symbolic data
$\left\{\left(A_i,\left[t_{A_i,\alpha},t_{A_i,\omega}\right]\right),\theta(t)\right\}$
Clustering
Classification

Figure 2. Overview of the methodology proposed

[2] For example by considering only system failure and system success

[3] Which are typically defined for real-value data

For each scenario, both SAX and TSKR produce in output a sequence of symbols (i.e., characters): a phrase. When both phrases are obtained, they are merged using the Time series conversion algorithm (see Fig. 2) in such way that duration, order and coincidence of events is preserved (see Section 3). When this process is repeated for all scenarios, then data-mining algorithms, such as classification and clustering, can be used to analyze the symbolically converted data (e.g., identify failure patterns).

2.1 Continuous Data Conversion

SAX [13] is an algorithm that allows the user to represent continuous time-varying data S as a series of n symbols $\bar{\bar{S}} = \bar{\bar{s}}_1, \bar{\bar{s}}_1, \dots, \bar{\bar{s}}_n$ where $\bar{\bar{s}}_i$ is a symbol.

This is performed by discretizing:

- Time into n intervals
- Range of $\theta(t)$ into α intervals

While temporal discretization is performed by partitioning the time axis into n intervals having the same length, the discretization of $\theta(t)$ is typically performed by dividing the range of θ into α equi-probable regions. Each region has a character $\bar{\bar{s}}$ associated to it and the alphabet size has cardinality[4] α. The resulting conversion generates a time series of length n and an alphabet size equal to α. The SAX algorithm (Algorithm 1) consists of the following steps (also see Fig. 3):

Algorithm 1: SAX Algorithm

```
1: Input: $n$ and $\alpha$
2: Normalize the data (mean equal to 0 and standard deviation equal to 1)
3: Partition the temporal interval into $n$ equal sized intervals
4: Divide the distribution of $\theta(t)$ into a equi-probable regions and assign
   a symbol to each region (alphabet has cardinality equals to $\alpha$)
5: Consider the average value $\bar{s}$ of $\theta(t)$ in each interval
6: For each $\bar{s}$ assign its own $\bar{\bar{s}}$ according to the discretization performed in
   Step 4
7: Generate a phrase $\bar{\bar{S}}$ as a timely ordered sequence of symbols
```

The end result is a *phrase* $\bar{\bar{S}}$: a timely ordered sequence of symbols. An example of discretization for the temporal profile of a scenario taken from [9] is shown in Fig.4.

Typically, data generated by simulations contain the temporal profile of multiple variables; moreover, as also shown in Fig. 4, a fixed number (i.e., n) of time intervals having equal length is not optimal to capture rapid changes of S.

The issue of dealing with multiple variables can be solved by:

- performing Steps 2 and 4 in Algorithm 1 independently for each variable, and,
- maintaining the order of symbols for every variable in each time interval

[4] Cardinality of the alphabet refers to the number of characters used.

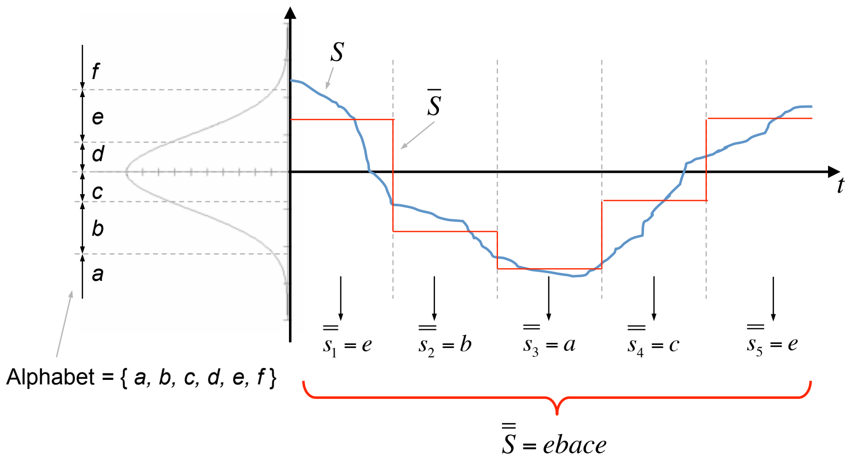

f
e
d
c
b
a
Alphabet = { a, b, c, d, e, f }
S
$\overline{S}$
t
$\overline{\overline{s}}_1 = e$
$\overline{\overline{s}}_2 = b$
$\overline{\overline{s}}_3 = a$
$\overline{\overline{s}}_4 = c$
$\overline{\overline{s}}_5 = e$
$\overline{\overline{S}} = ebace$

Figure 3. Example of symbolic conversion with a = 6 and n = 5

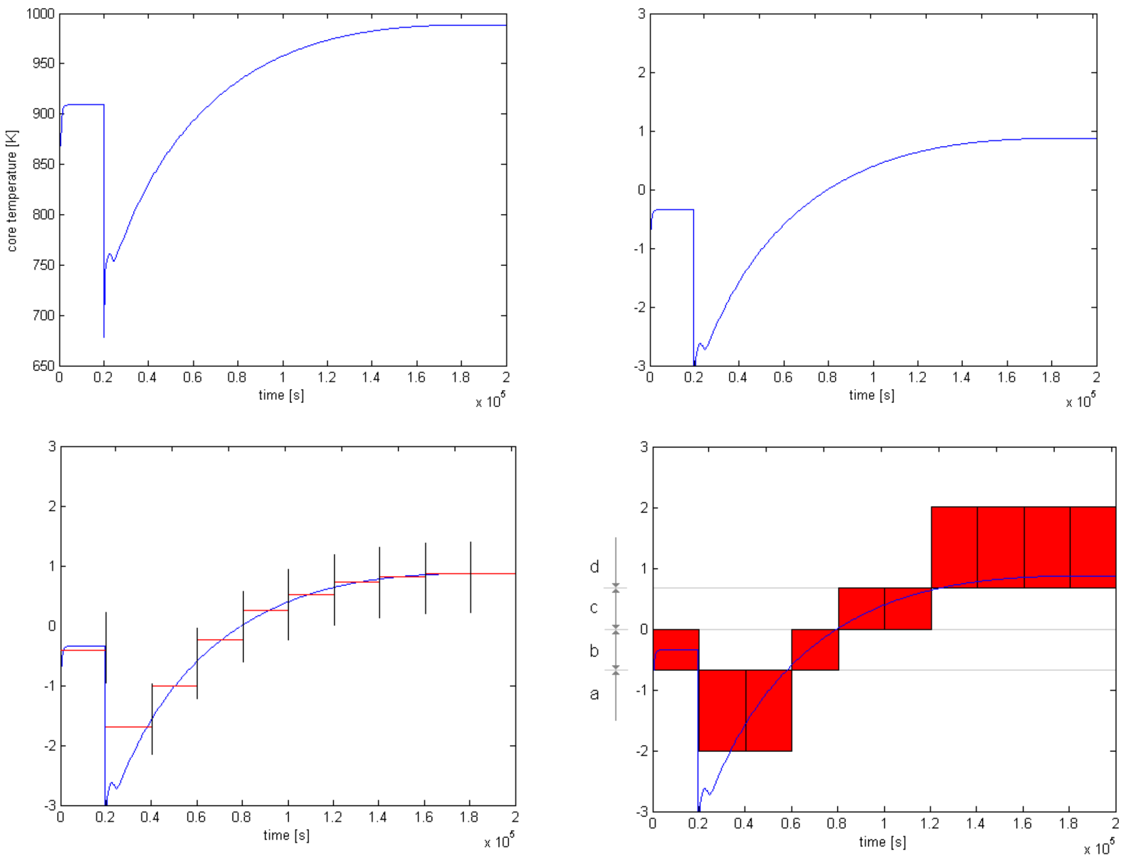


Figure 4. Application of the SAX algorithm [13] for a nuclear transient [9]: raw data (top left), data normalized (top right), temporal discretization (bottom left) and symbol sequence generation (bottom right)

Regarding the issue of identifying rapid changes of state variables, a solution is to recursively analyze the rate of change of the covariance matrix computed in that interval (as shown in [15]). The rationale is to choose time intervals such that the rate of change of the covariance matrix eigenvalues is below a fixed threshold (see Algorithm 2). As input, the minimum and a maximum length of the time interval are required in order to preserve accuracy and avoid extremely long phrases.

From our experience, the choice of the optimal values of n and α is case dependent. In our application we aim to choose values of n and α such that no two scenarios are represented by the same phrase.

Figure 5 shows an example of time adaptive discretization of a scenario characterized by 2 variables.

Algorithm 2: Adaptive Time Discretization

```
1: Input: maximum value for the rate of change of the eigenvalues of the
   covariance matrix, $\dot{\lambda}_{Max}$; minimum and maximum length of the time
   discretization, i.e., $t_{Min}$ and $t_{Max}$
2: Divide the time scale in intervals having length $t_{Max}$
3: Evaluate the covariance matrix of the data points contained in that interval
   and determine its eigenvalues $\lambda_i$
4: Determine the highest eigenvalue relative variation $\dot{\bar{\lambda}}$
5: If $\dot{\bar{\lambda}} > \dot{\lambda}_{Max}$ then split the interval into 2 intervals of equal length
6: Repeat Steps 3 and 4 for all intervals
```

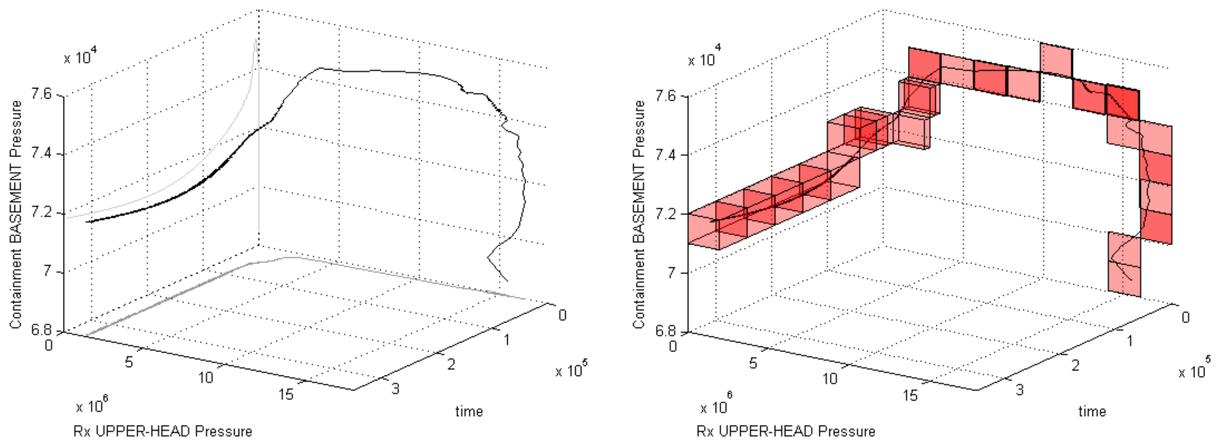


Figure 5. Adaptive time discretization of multi-dimensional scenarios; a 2-variable case (containment and reactor pressure vs. time) is shown: original data (left) and corresponding discretization (right)

2.2 Discrete Data Conversion

The discrete type of data includes the timing of events. An example of discrete data is given in Fig. 6. Following a loss of off-site power, diesel generators are only available for a limited time and, thus, the Emergency Core Cooling System (ECCS) temporal functionality is limited. Conditions of core uncovered are reached briefly after ECCS is no longer available.

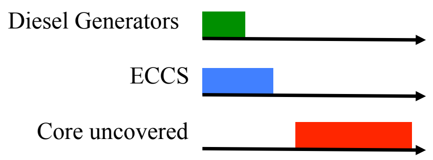


Figure 6. Example of temporal discrete events for a Loss of Offsite Power scenario

What is needed is an algorithm able to convert the set of discrete data into a symbolic form that preserves duration, order and coincidence of events. In this respect, the TSKR algorithm [14] offers a flexible way to solve this problem.

A few definitions are need before presenting the actual algorithm:

- *Tone*: Tone is the elemental component of time interval analysis. It is described by the triple $\{\sigma, s, e\}$ where σ is the symbol associated to the time interval $[s, e]$.
- *Chord*: A chord pattern describes a time interval where $k > 0$ tones coincide
- *Phrase*: A phrase is a sequence of $k > 1$ non overlapping chords

TSKR is a hierarchical language for expressing temporal knowledge in symbolic data. The term hierarchical refers to the fact that the symbolic conversion is performed in three levels (see Fig. 7) that describes the concept of duration (through tones), coincidence (through cords) and order (through phrases). The algorithm (see Algorithm 3) is summarized as follows:

Algorithm 3: TSKR Algorithm

```
1: Mining aspects: given a set of $d$-dimensional time series, select and label $k$ aspects
2: Mining tones: generate a series of tones from the $k$ aspects generated in Step 1
3: Mining marginally interrupted tones: find and filter small temporal gaps in order to avoid generation of chords having small temporal intervals
4: Mining chords: given $k$ tones, generate a set of $k$ chords
5: Mining phrases: generate a symbolic interval ordered sequence representing occurrences of the set of $k$ chords determined in Step 4
```

Note that a set of k not overlapping tones still produces k chords, each of them containing a single tone.

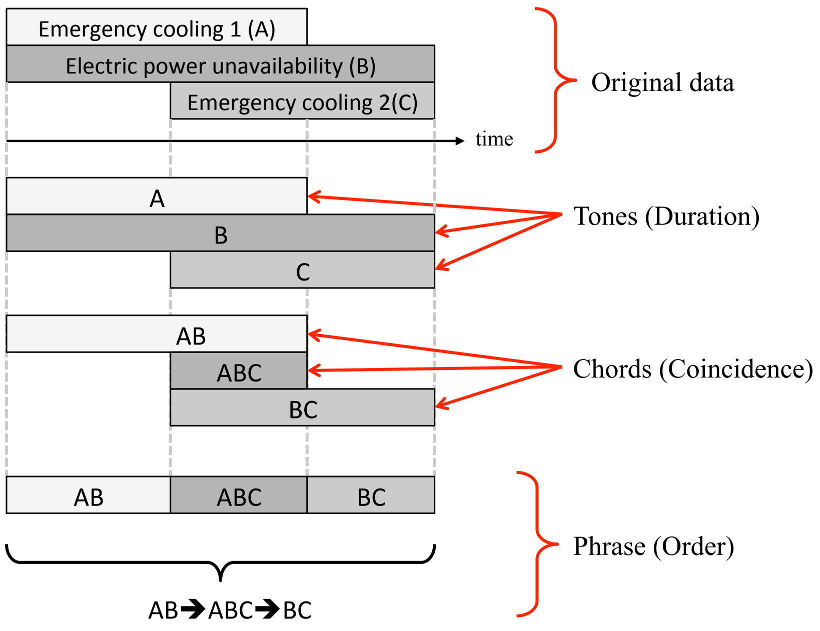


Figure 7. Generation of a phrase from a sequence of tones [14]

3 TIME SERIES ANALYSIS ALGORITHM

At this point, it has been shown how the SAX (see Section 2.1) and TSKR (see Section 2.2) algorithms can generate symbolic sequences, i.e. phrases, from continuous and discrete type of data respectively. The scope of the Time Series Analysis Algorithm is to merge these two symbolic sequences into a single one. This process comes naturally if the sequence generated by SAX is considered as a single multivariate tone.

Thus, when both continuous and discrete data have been converted though a series of tones, it is possible to repeat Algorithm 3 (i.e., mining marginally interrupted tones, mining chords and mining phrases) in order to obtain the overall symbolic conversion for both sets of data (see Fig. 8).

4 APPLICATIONS

Sections 2 and 3 have shown how it is possible to symbolically represent both sequence of events and temporal profiles of state variables. Sections 4.1 and 4.2 show how such a representation can be used to analyze data generated by DPRA methodologies.

For all the applications, a basic requirement is the definition of a dissimilarity measure (also known as distance metric). Most of these measures are defined over real-valued data (e.g., Euclidean or the generic Minkowsky distance [16]). New dissimilarity measures for symbolic data have been recently introduced [17], however, a comparison of these measures is not within the scope of this paper. In this paper we used tested the MINDIST function [13] and the Compression-Based dissimilarity Measure (CDM) [17] which is based on the concept of Kolmogorov complexity.

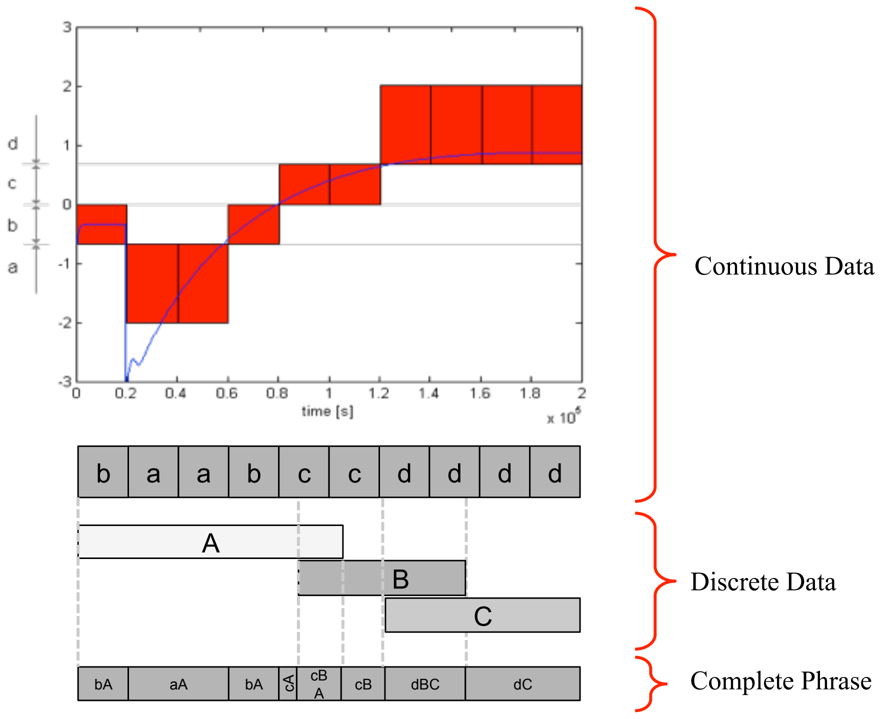
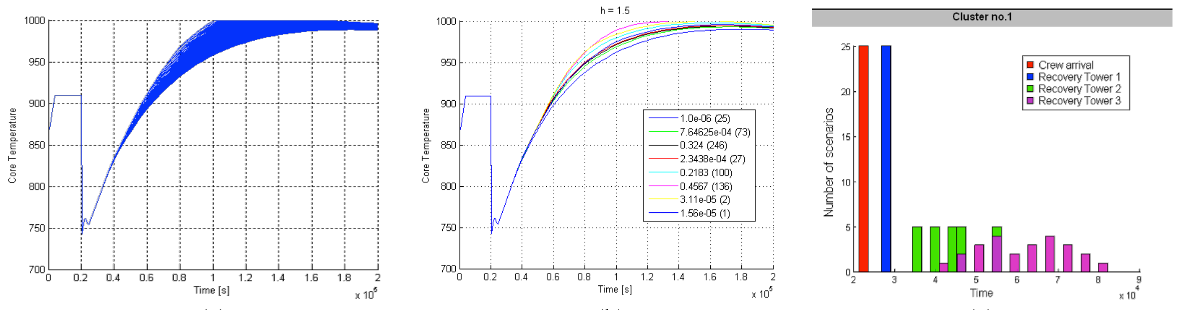


Figure 8. Symbolic conversion of both continuous and discrete data

4.1 Clustering

Typically, DPRA methodologies aim to evaluate the impact of uncertainties on system response during accident scenarios by simulating system dynamics under different sequences and timing of events. In order to identify the correlations between system dynamics and sequence/timing of events, advanced data mining algorithms are required.

Clustering algorithms offer a solution to this problem by grouping scenarios having similar temporal profiles into clusters. In [9], the continuous data is chosen to represent each scenario in the clustering process; the discrete data is considered after the clustering process when, for each cluster, the distribution of the timing of events is analyzed (see Fig. 9).



(a) (b) (c)

Figure 9. Clustering applied to a set of scenarios [9] by considering only continuous data: original data (a), cluster obtained (b) and distribution of timing of events (c) for cluster 1 (blue line)

The symbolic conversion methodology shown in Sections 2 and 3 offer an alternative as a pre-processing tool before clustering to represent each scenario.

Several tests have been performed using the datasets presented in [9,18,19]. Preliminary results have shown a drastic reduction in memory requirements to store such data (see Table 1). Such reduction has positively affected also the time required to cluster such data.

Table 1. Preliminary comparison results between numerical [9] and symbolic representation of three datasets generated by DPRA methodologies: memory requirements

Dataset	*Original data*	*Memory need (real-valued representation)*	*Memory need (symbolic representation)*
Level controller	2.3 MB	500 KB	0.2 KB
Aircraft crash scenario	7 MB	4 MB	600 KB
Zion Station Black-out (1)	362 MB	120 MB	31 KB
Zion Station Black-out (2)	3.2 GB	318 MB	1.9 MB

We performed a first clustering test of this algorithm using the aircraft crash dataset [9]. This dataset shown in Fig. 9(a) consists of 610 scenarios. Each scenario contains the temporal profile of one continuous variable (core temperature).

First, we normalized the original dataset (mean = 0 and standard deviation = 1). Afterwards, we create two datasets from the original dataset: real-valued and symbolically converted datasets. Since scenarios are very similar to each other (see Fig. 9(a)), for the symbolically converted dataset, large values of α and n were required: $\alpha = 12$ and $n = 35$. This is the reason why the memory reduction of this dataset (from 4 MB to 600 KB) was not as effective as for the other three data =sets shown in Table 1.

We used Mean-Shift [21] as clustering algorithm and we adapted it to act also on symbolic data by changing the functions that measure the dissimilarity between two symbolically converted scenarios. After obtaining the same number of clusters from both datasets, for each cluster, we compared the number of scenarios and the cluster membership (i.e., we checked that each cluster contains the same set of scenarios).

Table 2. Clustering results for the Aircraft crash scenario [9] using Mean-Shift [21] for the numerical and symbolically converted datasets

Cluster no.	*Mean-Shift (Real-valued)*	*Mean-Shift (Symbolic)*	*Cluster membership*
1	25	25	Identical
2	73	73	Identical
3	246	246	Identical
4	27	27	Identical
5	100	100	Identical
6	136	136	Identical
7	2	2	Identical
8	1	1	Identical

Table 2 shows that the set of clusters obtained from the symbolically converted dataset maintains the same cluster membership of the set of clusters obtained from the real-valued dataset. This implies that, after the symbolic conversion of the original dataset, it is still possible to maintain the same cluster structure obtained from the real-valued dataset.

4.2 Classification and Pattern Detection

A second test we have performed shows the improved performances for a typical classification problem using the K-Nearest Neighbors (*KNN*) [20] algorithm. In this respect, Table 3 summarizes that test for the same datasets used also in Section 4.1.

Such reductions (both in term of memory requirements and computational performances) are of relevance for diagnosis and prognosis methodologies when real-time measurements need to be continuously compared with sets of archived data (either generated by simulators or previously monitored and stored).

Table 3. Preliminary comparison results between numerical [9] and symbolic representation of three datasets generated by DPRA methodologies: time to compute K-Nearest Neighbors (KNN)

Dataset	Time to compute *KNN* (real-valued representation)	Time to compute *KNN* (symbolic representation)
Aircraft crash scenario	0.4±0.13 s	0.09±0.01 s
Zion Station Black-out (1)	4.1±0.75 s	0.2±0.03 s
Zion Station Black-out (2)	12±2.1 s	0.95±0.12 s

Another advantage of dealing with symbolic data is that analysis tools that are not defined for real-valued data, such as Markov models and decision trees, are available and can be directly applied to the obtained series of symbols [13].

5 CONCLUSIONS

This paper has shown a pre-processing methodology that can be applied to analyze the datasets generated by DPRA methodologies. This methodology performs the symbolic conversion of both the temporal profiles of state variables and the timing of events. Such conversion is performed by 1) symbolically converting both continuous and discrete data and 2) merging theses two sequences of symbols. Such conversions are performed in such way that duration, coincidence and order are preserved.

We have shown preliminary results for clustering and classification applications that are starting to be of interest in the DPRA arena. We have shown these applications since there is an emerging interest in diagnosis and prognosis methodologies that require fast computation of search algorithms such as *KNN*.

A wider testing of the algorithms presented in this paper is still underway. In particular, we are aiming to explore furthermore the data mining capabilities after the symbolic conversion is performed.

6 ACKNOWLEDGMENTS

The authors want to thank Cristian Rabiti and Andrea Alfonsi (at Idaho National Laboratory) for their constructive comments and suggestions in the development of the algorithms presented in this paper.

7 REFERENCES

[1] N. Siu, "Risk assessment for dynamic systems: an overview," *Reliability Engineering and System Safety*, **43**, no. 1, pp. 43-73 (1994)

[2] RELAP5-3D Code Development Team, *RELAP5-3D Code Manual* (2005)

[3] R. O. Gauntt, *MELCOR Computer Code Manual, Version 1.8.5*, Vol. 2, Rev. 2., Sandia National Laboratories, NUREG/CR-6119

[4] D. I. Chanin, H. N. Jow, and J. A. Rollstin, *MELCOR Accident Consequence Code System (MACCS)*, Sandia National Laboratories, NUREG/CR-4691, Vols. 1-3, SAND86-1562 (1990)

[5] A. Hakobyan, T. Aldemir, R. Denning, S. Dunagan, D. Kunsman, B. Rutt, and U. Catalyurek, "Dynamic generation of accident progression event trees," *Nuclear Engineering and Design*, **238**, no. 12, pp. 3457-3467 (2008)

[6] K. S. Hsueh and A. Mosleh, "The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants," *Reliability Engineering and System Safety*, **52**, pp. 297-314 (1996)

[7] C. Rabiti, A. Alfonsi, J. Cogliati, D. Mandelli, and R. Kinoshita, "Reactor Analysis and Virtual Control Environment (RAVEN) FY12 REPORT," Tech. Rep. INL/EXT-12-27351, Idaho National Laboratory (INL) (2012)

[8] C. Rabiti, A. Alfonsi, D. Mandelli, J. Cogliati, and R. Martineau, "RAVEN as Control Logic and Probabilistic Risk Assessment Driver for RELAP-7," *in Proceeding of American Nuclear Society (ANS)*, **107**, pp. 333–335, San Diego (2012)

[9] D. Mandelli, A. Yilmaz, T. Aldemir, K. Metzroth, and R. Denning, "Scenario clustering and dynamic probabilistic risk assessment," *Reliability Engineering and System Safety*, **115**, pp. 146-160 (2013)

[10] U.S.NRC, *NUREG 1150 - Severe accident risks: an assessment for five U.S. nuclear power plants*, Division of Systems Research, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington, DC (1990)

[11] S. Guarro, M. Yau, and M. Motamed, "NUREG/CR-6465: Development of Tools for Safety Analysis of Control Software in Advanced Reactors," Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington, DC (1996)

[12] E. Zio and P. Baraldi, "Identification of nuclear transients via optimized fuzzy clustering," *Annals of Nuclear Energy*, **32**, no. 10, pp. 1068-1080 (2005)

[13] J. Lin, E. Keogh, S. Lonardi and B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms", *In proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, San Diego (2003)

[14] Morchen, F., "Time Series Knowledge Mining", PhD Thesis, Philipps-University Marburg, Germany (2006)

[15] D. Mandelli, A. Yilmaz, and T. Aldemir, "Clustering on manifolds: an application to scenario analysis using principal component analysis," *in Proceeding of American Nuclear Society (ANS)*, Washington, DC (2011)

[16] B. Mendelson, *Introduction to Topology*, New York (NY), USA: Dover Publications (1990)

[17] E. Keogh, S. Lonardi, C. Ratanamahatana, “Towards parameter-free data mining,” In proceedings of the 10th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 206-215 (2004)

[18] D. Mandelli, A. Yilmaz, and T. Aldemir, “Data processing methodologies applied to dynamic PRA: an overview,” *in Proceeding of PSA 2011 International Topical Meeting on Probabilistic Safety Assessment and Analysis*, CD-ROM, American Nuclear Society, LaGrange Park, IL (2011)

[19] D. Mandelli, A. Yilmaz, and T. Aldemir, “Scenario analysis and PRA: Overview and lessons learned,” i*n Proceedings of European Safety and Reliability Conference (ESREL 2011)*, Troyes (France) (2011)

[20] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley-Interscience Publication, (2000)

[21] K. Fukunaga and L. Hostetler, “The estimation of the gradient of a density function, with applications in pattern recognition,” *IEEE Transactions on Information Theory*, **21**, no. 1, pp. 32-40 (1975).

[22] E. Hofer, M. Kloos, B. Krzykacz-Hausmann, J. Peschke, and M. Woltereck, “An approximate epistemic uncertainty analysis approach in the presence of epistemic and aleatory uncertainties,” *Reliability Engineering and System Safety*, **77**, pp. 229--238 (2002)