Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 4 (2011) 1761–1770

Procedia Computer Science

Prediction Time Assessment in a DDDAS for Natural Hazard Management: Forest Fire Study Case☆

Andrés Cencerrado, Ana Cortés, Tomàs Margalef

Departament d'Arquitectura de Computadors i Sistemes Operatius. Escola d'Enginyeria. Universitat Autònoma de Barcelona. 08193 Bellaterra (Barcelona), Spain.

Abstract

This work faces the problem of quality and prediction time assessment in a Dynamic Data Driven Application System (DDDAS) for predicting natural hazard evolution. In particular, we used forest fire spread prediction as a case study to show the applicability of the methodology. The improvement on the prediction quality when using a two-stage DDDAS prediction framework has been widely proved. The two-stages DDDAS has a first phase where an adjustment of the input data is performed in order to be applied in the second phase, the prediction. This paper is focused on defining a new methodology for prediction time assessment under this kind of prediction environments by evaluating, in advance, how a certain combination of simulator, computational resources, adjustment strategy, and frequency of data acquisition will perform, in terms of prediction time. Since the time incurred in the hazard simulation is a crucial part of the whole prediction time, we have defined a methodology to classify the simulator's execution time using Artificial Intelligence techniques allowing us to determine upper bounds for the DDDAS prediction time depending on the particular input parameter setting. This methodology can be extrapolated to any DDDAS for predicting natural hazards evolution, which uses the two-stage prediction scheme as a working framework.

Keywords:
DDDAS, Data Uncertainty, Forest Fire Spread Prediction, Classification Techniques

1. Introduction

A natural hazard is an unexpected or uncontrollable natural event of unusual intensity that threatens people's lives or their activities. Unfortunately, the losses caused by natural hazards are increasing dramatically. Therefore, in order to mitigate the tragic consequences of such disasters, it is interesting to be able to take urgent decisions while the natural catastrophe is taking place. For this purpose, a lot of interdisciplinary research has been carried out to provide models/simulators to the community for evaluating in advance the natural hazard evolution. However, model-related issues aside, many simulators lack precision on their results because of the inherent uncertainty of the data needed to define the state of the system environment. This uncertainty is due, basically, to the difficult in gathering precise data

☆This research has been supported by the MICINN-Spain under contract TIN2007-64974

Email addresses: acencerrado@caos.uab.es (Andrés Cencerrado), ana.cortes@uab.es (Ana Cortés), tomas.margalef@uab.es (Tomàs Margalef)

doi:10.1016/j.procs.2011.04.191

at the right places where the disaster is taking place. So, in many cases, the simulators have to work with interpolated, outdated, or even absolutely unknown data values. Another key point to be considered when dealing with an ongoing disaster is the time incurred in providing evolution prediction results. In order to be useful, any evolution prediction of an ongoing hazard must be delivered as fast as possible for not being outdated. Consequently, we come up with the binomial *urgency-accuracy*.

From the urgency point of view, one must rely on computational resources with a high computational power. On the other hand, when approaching the accuracy aspect of the prediction, we can focus on either improving the underlying model or adjusting the input model parameters in order to overcome their uncertainty and, therefore, obtain better results. To overcome the just mentioned input uncertainty problem, we have developed a two-stage prediction strategy, which, first of all, carries out a parameter adjustment process by comparing the results provided by the simulator and the real observed disaster evolution. Then, the underlying simulator is executed taking into account the adjusted parameters obtained in the previous phase, in order to predict the evolution of the particular hazard for a later time instant. A successful application of this method mainly depends on the effectiveness of the adjustment technique that has been carried out. In this sense, our research group has developed several solutions for input parameters optimization, all of them characterized by an intensive data management: use of statistical approach based on exhaustive exploration of previous fires databases [6], application of evolutionary computation [10], calibration based on domain-specific knowledge [5], and even solutions coming from the merge of some of the above mentioned [9]. Since all these approaches perform the calibration stage in a data driven fashion, they all match the Dynamic Data Driven Application Systems paradigm [18, 19, 20].

In particular, we have developed this prediction scheme using forest fire as a study case and it has been demonstrated that the above mentioned adjustment techniques contribute to improve the quality of the fire spread prediction. However, it has to be taken into account that in these kind of urgent situations, a successful prediction is not only determined by the accuracy of the results: it is also necessary to seriously consider the time restrictions. For this purpose, we introduce a new methodology to characterize each element of the proposed DDDAS prediction process, with the aim of being able to design a tool for prediction time assessment during an emergency management. As in the case of the quality aspect, we have used forest fire spread prediction as study case.

This work is part of a more ambitious project, which consists of determining in advance, how a certain combination of natural hazard simulator, computational resources, adjustment strategy, and frequency of data acquisition will perform, in terms of execution time and prediction quality. However, in order to approach the problem in an organized way, we first introduce, in this paper, how to characterize the core of the DDDAS for the case of forest fire spread prediction. As it is well known, the execution time of the underlying simulator depends on the specific setting of the input parameters. For this reason, decision trees were used to obtain an upper bound for the simulator execution time by previously classifying it according to the input parameter setting. The proposed classification scheme has been carried out considering two different fire spread simulators in order to validate the classification strategy with different setup conditions.

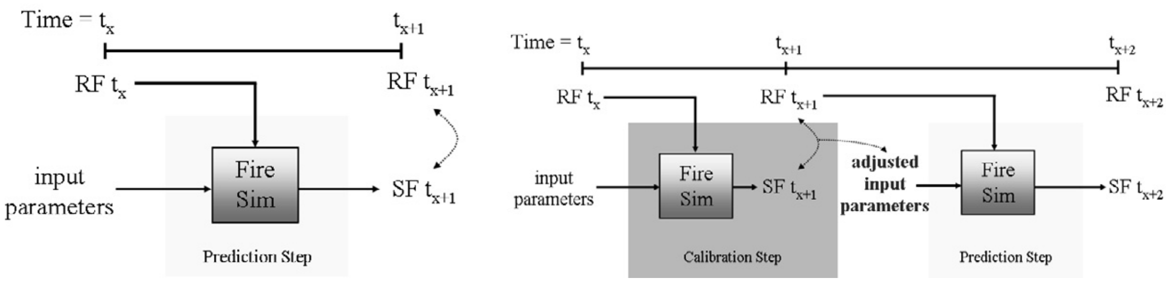
This paper is organized as follows. In the next section, an overview of how the two-stages DDDAS for forest fire spread prediction is given. In Section 3, we expose how this framework could be generalized to any natural hazard, and the methodology to evaluate the prediction time assessment is described. In Section 4, the experimental study is reported and, finally, the main conclusions are included in Section 5.

2. DDDAS for Forest Fire Spread Prediction

In the field of physical systems modelling, specifically forest fire behavior modeling, there exist several tools for mitigating damages caused by them such as fire propagation simulators [11, 12, 13, 14, 15], based in some physical or mathematical models [1, 2].

These simulators need certain input data, which define the characteristics of the environment where the fire is taking place, in order to evaluate its future propagation. This data usually consists of the current fire front, terrain topography, vegetation type, and meteorological data such as humidity and wind direction and wind speed.

Some of this data could be retrieved in advance and with noticeably accuracy, as, for example, the topography of the area and the predominant vegetation types. However, there is some data that turns out very difficult to obtain with reliability. For instance, getting an accurate fire perimeter is very complicated because of the difficulties involved



(a) Classic prediction (b) Two-stage prediction

Figure 1: Prediction Methods

in getting, at real time, images or data about this matter. Other kind of data sensitive to imprecisions is that of meteorological data, which is often distorted by the fire itself. However, this circumstance is not only related to forest fires, but it also happens in any system with a dynamic state evolution throughout time (e.g. floods [25], thunderstorms [26, 27], etc.). These restrictions concerning uncertainty in the input parameters, added to the fact that these inputs are set up only at the very beginning of the simulation process, become an important drawback because as the simulation time goes on, variables previously initialized could change dramatically. This may mislead results of simulations. In order to overcome these restrictions, we need a system capable of dynamically obtaining real time input data in those case that is possible and, otherwise, properly estimating the values of the input parameters needed by the underlying simulator.

The classic way of predicting forest fire behaviour, summarised in Figure 1(a), takes the initial state of the fire front (RF = real fire) as input as well as the input parameters given for some time t_x. The simulator then returns the prediction (SF = simulated fire) for the state of fire front at a later time t_{x+1}.

Comparing the simulation result SF from time t_{x+1} with the advanced real fire RF at the same instant, the forecasted fire front tends to differ to a greater or lesser extent from the real fire line. One reason for this behaviour is that the classic calculation of the simulated fire is based upon one single set of input parameters afflicted with the before explained insufficiencies. To overcome this drawback, a simulator independent data-driven prediction scheme was proposed to optimize dynamic model input parameters [4]. Introducing a previous calibration step as shown in Figure 1(b), the set of input parameters is optimized before every prediction step. The solution proposed come from reversing the problem: how to find a parameter configuration such that, given this configuration as input, the fire simulator would produce predictions that match the actual fire behavior. Having detected the simulator input that better describes current environmental conditions, the same set of parameters, could also be used to describe best the immediate future, assuming that meteorological conditions remain constant during the next prediction interval. Then, the prediction becomes the result of a series of automatically adjusted input configurations.

This strategy works under the hypothesis that the environmental conditions are stable throughout the adjustment and calibration steps. This actually does not happen in most cases. For this reason, new techniques had to be introduced to overcome this disadvantage, so that the system is able to dynamically acquire data if there have been detected sudden changes in the initial conditions [9].

Previous works proposed several calibration techniques, which made the problem of fire spread prediction to fit the DDDAS paradigm, rather than the classic prediction scheme such as [7, 9, 10]. Despite the reported works were focused on the forest fire case, the two stages DDDAS for forest fire spread prediction described in figure 1(b) could be extrapolated to any kind of natural disasters. Figure 2 shows a general scheme for a two-stage DDDAS for natural hazard management. In the following section, we shall describe a methodology to perform the prediction time assessment under this prediction framework.

3. Prediction Time Assessment

As stated in Section 1, when dealing with emergency simulation, it is extremely necessary to maximize the result of the *urgency-accuracy* binomial. This goal is oriented to provide the personnel in charge of making decisions about

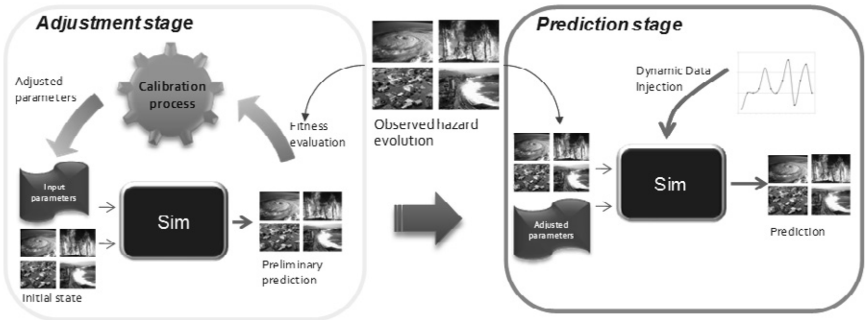


Figure 2: General two-stages DDDAS for natural hazard prediction evolution

how to face an ongoing emergency, with intelligent tools able to evaluate, in advance, how a certain combination of simulator, computational resources, adjustment strategy, and frequency of data acquisition will perform, in terms of execution time and prediction quality. In order to bound the problem, we work under certain assumptions:

- We focus on those emergencies where the corresponding simulators present high input-data sensitivity.
- We assume scenarios where the computational resources are dedicated. Currently, we are working on adapting tools that allow urgent execution of tasks in distributed-computing environments, e.g. SPRUCE [21].
- We rely on the two-stage DDDAS prediction strategy.

Taking into account these premises and bearing in mind the scheme shown in figure 2, we can define three levels of prediction time assessment: Simulator level assessment (SLA), Adjustment level assessment (ALA) and Prediction level assessment (PLA).

3.1. Simulator level assessment (SLA)

Prediction time assessment at this level must be done independently on the underlying simulator (natural hazard) and the particular setting of their input parameters. The main objective at this level is to define a simulator-independent methodology to determine a clustering classification of the simulator execution time, where each cluster has associated an upper bound for the execution time depending on the values of the input parameters. This process is carried out in an *offline* way and will be widely explained later on in this paper. Since this characterization process depends on the executable platform, different simulator characterizations will be performed for each available computational resource.

3.2. Adjustment level assessment (ALA)

This level corresponds to estimate the prediction time increase due to the calibration strategy used in the *Adjustment stage*. As we have previously mentioned, there exist several calibration strategies that have been demonstrate to be useful for improving the prediction quality of a hazard evolution. Each one of this optimization schemes must be modeled independently of each other because the way of performing is quite different. As it could be observed in figure 2, there is a tight relation between the results obtained at SLA with this level because SLA is inside ALA, therefore, ALA is directly proportional to SLA.

3.3. Prediction level assessment (PLA)

At this level one can rely either on dynamic data injection to the system or not. A pure DDDAS will take into account data injection at real time and this is the way that the DDDAS for forest fire spread prediction has been designed in its advanced form. However, in a preliminary version, the dynamic data injection was not considered and it was based in the working hypothesis that the environmental conditions keep constant from the calibration stage to the prediction stage. For this reason, the PLA methodology has been designed in a two step fashion, first of all we will determine a standard methodology for the prediction stage without real time data injection and, afterwards, the PLA's characterization will be performed, taking into account data gathering frequency and data source. The aim consists of reaching the capability to determine the probability distribution that indicates which percentage of prediction improvement has *historically* been obtained in the cases where the data was acquired with a certain frequency, and from a certain data sources. This characterization level, as in SLA, relies on a massive stadistic study. Thus, we can assess in advance the probability of improvement the dynamic data injection process may produce in the prediction, without the need to modify the underlying simulator.

It is important to notice that in the characterization of the simulator, we focus on the execution time as a ”classification criteria”, whereas the quality of prediction is the factor taken into account when characterizing the adjustment stage (ALA). This is because the quality of the initial prediction given by the simulator has no influence over the final prediction. Nevertheless, the execution time of each calibration technique is directly proportional to the execution time of the simulator. Hence, in order to estimate both accuracy of prediction and time needed to perform it, the study of these aspects is carried out in this way.

In the next section, an empirical study concerning the method followed for the Simulator Level Assessment is detailed and the obtained results are analyzed.

4. Experimental Study

As stated above, the fact of having well characterized each simulator we deal with, in terms of execution time, becomes crucial to validate the proposed methodology.

This matter may be tackled by means of taking the strategy of carrying out large sets of executions of the underlying simulator, and then analyzing its behavior from the obtained results. However, this fact may not be trivial in certain cases. While it is easy to detect that the application presents a high sensitivity to certain input parameters, even in an intuitive way, some of them produce a behavior of the simulator that turns out hard to predict. Figures 3 and 4 show examples of each case, respectively. In the former, one can observe that the dimension of the map to be simulated has a direct influence on the execution time (as it was bound to happen), whereas, in the latter, it can be noticed that the relation between execution time and wind direction is not so clear (this *anomaly* is reported in [16]), and even it becomes odder when combining variations in wind direction with variations with vegetation type.

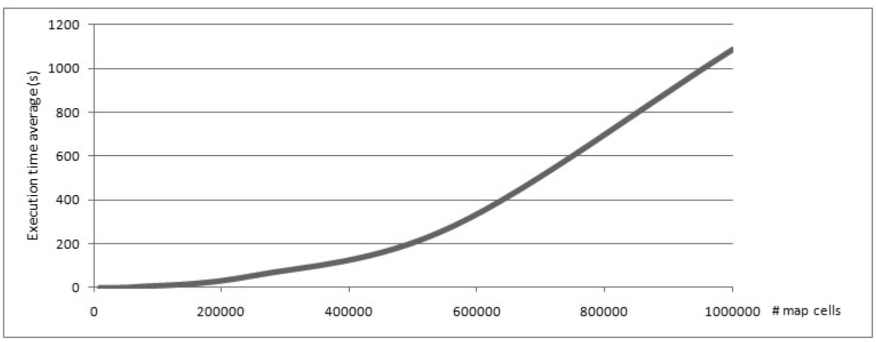


Figure 3: Execution time as a function of number of cells.

Currently, this characterization is fulfilled by means of carrying out large sets of executions (on the order of tens of thousands) counting on different initial scenarios (different input data sets), and then, applying knowledge-extraction techniques from the info they provide. We record the execution times from the experiment, and then we establish a classification of the input parameters according the elapsed times they produced. At this moment, we are capable to apply machine learning techniques to determine classification criteria and, therefore, given a new set of input parameters, to be able to estimate how much the execution will last.

This learning process is carried out *offline*, i.e. the classification rules are established prior to the hazard occurrence. This way, at the moment of the urgency management, we only have to apply the classification technique, which involves a negligible cost of computational time.

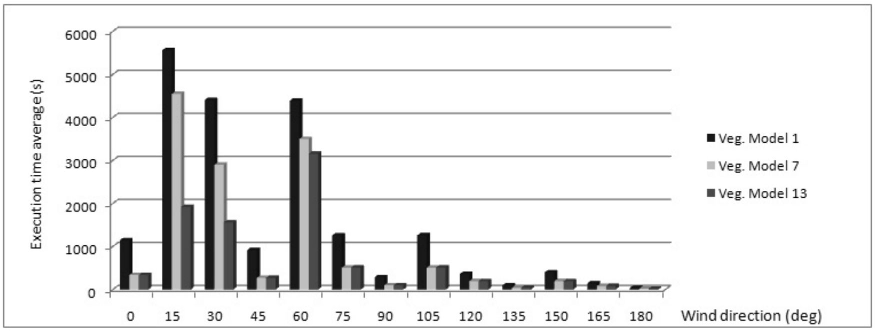


Figure 4: Variations in execution time according to variations in wind direction and vegetation type.

This fact highlights the need to base on complex criteria in order to successfully classify the input data sets according to the execution time they will cause. Consequently, we rely on the field Artificial Intelligence to reach such an objective. Specifically, this experimental study shows the results obtained from the use of decision trees as classification technique.

4.1. Test bed description

For validation purposes, we have used two different forest fire spread simulators in the experimental study: HFire [13] and fireLib [15]. HFire is a spatially explicit fire spread model that was developed for modeling fires in chaparral environments in 2001. FireLib is a C function library for predicting the spread rate and intensity of free-burning wildfires, developed in 1996. Both of them are based on the Rothermel fire model [1] to determine the direction and magnitude of the maximum rate of spread. Because of the specific features of each simulator, the simulated scenario slightly differs in each case as are subsequently listed:

- HFire:
  - Domain: The domain studied in the case of HFire was the Santa Monica Mountains National Recreation Area (SMM) in southern California, which topografy details are provided in [17].
  - Simulation duration: In the case of HFire a 10-day simulation was carried out in every execution.
  - Ignition point: When using HFire, it was approximately the ignition point of the well known 1996 Calabasas fire in California (also provided in [17]).
- FireLib:
  - Domain: For the characterization of fireLib, an artificial 1001x1001 cells map was used (cells width and height: 100 feet). In both cases, the indicated topography remained constant for all the executions.
  - Simulation duration: FireLib simulations end once the fire reaches one edge of the map.
  - Ignition point: The ignition point in the case of fireLib was the central cell of the map.

Apart from these peculiarities, the rest of input parameters were the same in both cases. Specifically, Table 1 shows the assigned probability distributions for each type of input. As regards wind speed and direction, the chosen distributions and their associated parameters were the ones used in [24], based on statistical analysis of data from weather stations in the area of SMM. The vegetation models correspond to the 13 standard Northern Forest Fire Laboratory (NFFL) fuel models [3].

Once established the distribution of each input parameter, a set of 38750 different combinations of input data sets was generated, and the simulations of each scenario for each simulator were performed.

As regards the computational platform, all the experiments carried out in this work were done on a cluster of 32 IBM x3550 nodes, each of which counting on 2 x Dual-Core Intel Xeon CPU 5160, 3.00GHz, 4MB L2 cache memory (2x2) and 12 GB Fully Buffered DIMM 667 MHz, running Linux version 2.6.16.

Input	Distribution	μ,σ	Min,Max
Vegetation model	Uniform	—	1,13
Wind Speed	Normal	12.83,6.25	—
Wind Direction	Normal	56.6,13.04	—
Dead fuel moisture	Uniform	—	0,1
Live fuel moisture	Uniform	—	0,4

Table 1: Input parameters distributions description.

4.2. Preliminary conclusions

Figure 5 depicts the histogram obtained from the execution of the test bed using HFire simulator. As it can be observed, there exist some execution time intervals which assemble most of the execution instances. Nevertheless, the important matter concerning these results is that HFire shows a very regular behavior, so the whole execution time interval is short enough to discard a classification process. Hence, when using HFire as a fire spread simulator, we can assume the worst case (executions will last approximately 29 seconds) for the characterization of the whole prediction process.

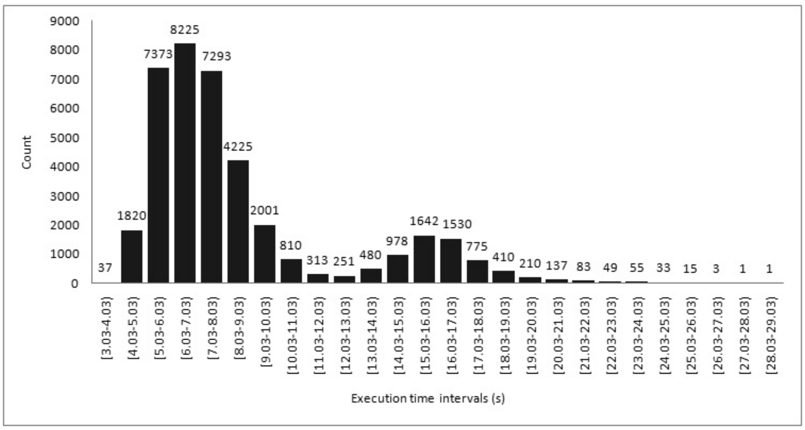


Figure 5: Histogram of execution times using HFire.

This behavior contrasts with the one obtained from the fireLib simulator. As one can see in Figure 6, the variance on the simulation time is very noticeable. The great majority of the executions are located under the 2500 seconds threshold, but there were several executions that lasted more than 30000 seconds, and even more than 50000 seconds.

From the point of view of emergency prediction, it is crucial to have the question of execution time under control, so we may deal with cases that drastically slow down the prediction process. An elapsed time prediction for a simulator execution with an error on the order of thousands of seconds would be prohibitive, so, from cases like this one, there arises the need to be able to predict how the simulator is going to behave and, therefore, the need to use an efficient classification technique.

4.3. Empirical evaluation

In order to respond to this need, the experimental study carried out in this work consisted of use decision trees as the classification method, to be able to estimate, in advance, the execution time of fireLib, given a new unknown set of input parameters.

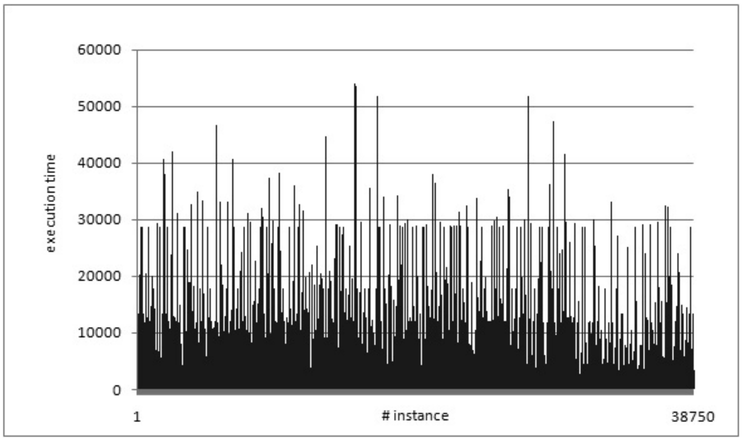


Figure 6: Execution times using fireLib.

The decision trees used in this research were the generated by the C4.5 algorithm [23], specifically, the J48 open source Java implementation of the C4.5 algorithm in the Weka [22] data mining tool. The data obtained from the 38750 executions was used as a training set, and 1000 new instances were generated (according to the distributions shown in Table 1) to be used as a test set.

The number of classes, and the execution time intervals they represent, were determined taking into account where our work is framed, i.e. the intervals chosen for each class are those that in a real emergency situation would matter (it has no sense, for example, to classify by intervals of 10 seconds when predicting forest fire spread). They are:

- Class A: ET $\leq$ 900 seconds.
- Class B: 900 seconds < ET $\leq$ 1800 seconds.
- Class C: 1800 seconds < ET $\leq$ 3600 seconds.
- Class D: 3600 seconds < ET $\leq$ 7200 seconds.
- Class E: 7200 seconds < ET.

Where *ET* stands for execution time.

		Predicted Class				
		A	B	C	D	E
Real Class	A	669	14	4	2	0
	B	17	72	9	4	0
	C	2	12	72	12	4
	D	5	6	14	24	5
	E	0	3	2	12	36

Table 2: Correspondence between real and predicted classes.

The results of the application of decision trees to the test set are summarized in Table 2. Here, one of the main aspects to highlight is the prominence of the main diagonal, which means that perfect matches are predominant over the whole set of predictions. Furthermore, one can notice that the values decrease as one moves away from the main diagonal. Indeed, the worst possible cases (predict A when the real class is E, and vice-versa), never happened.

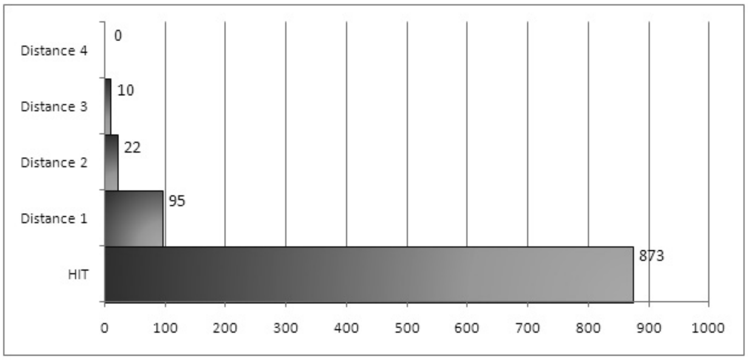


Figure 7: Classification accuracy.

Figure 7 shows the absolute values of the number of predictions that totally hit the real class, as well as the absolute values where the prediction had an accuracy determined by the distance between classes. A *Distance X* accuracy means that there are X-1 classes between the predicted class and the real class.

The most noticeable aspect when analyzing this graphic is that if we consider *Distance 1* as a good prediction accuracy, then the results obtained present a 96.8% of satisfactory classifications.

5. Conclusions

Natural hazard management is undoubtedly a relevant application area in which the DDDAS paradigm can play a very important role. As it has been proved in previous works, the application of this paradigm becomes crucial in order to improve the quality of the predictions given by the simulators. Particularly, the combination with the above exposed two-stage prediction method, contributes to relieve the input uncertainty problem and, therefore, enhancing the quality of prediction.

This work constitutes an essential part of a very ambitious project, which consists of determining in advance, how a certain combination of natural hazard simulator, computational resources, adjustment strategy, and frequency of data acquisition will perform, in terms of execution time and prediction quality.

Since we are dealing in the area of natural hazards management, it is absolutely necessary to take into account the time incurred for the prediction method. For this purpose, we have designed a methodology to assess in the urgency-accuracy binomial in each particular case.

Since the execution time of the simulations has a direct impact on the overall prediction time, it is necessary to characterize the behavior of each underlying simulator. As it is well known, the execution time of a particular simulator depends on the specific setting of the input parameters. However, as it has been exposed, it becomes hard to predict how certain variations on certain input parameters would affect the execution time. In this work, we approach such a challenge by means of Artificial Intelligence and Data Mining techniques. Particularly, in this work we present how we deal with simulators characterization by means of the use of decision trees as classification technique.

Since we have used forest fire spread prediction the study case, the experimental study has been done using two different forest fire spread simulator. The proposed classification scheme has been carried out considering HFIre and FireLib simulators and a huge set of input parameters combinations, in order to validate the classification strategy with different setup conditions.

The obtained results demonstrate that the use of decision trees as classification strategy is suitable for this research, obtaining up to 96.8% of satisfactory classification prediction, which represents a great advance, and allows us to tackle the subsequent steps of the proposed methodology.

References

[1] R. C. Rothermel. *A mathematical model for predicting fire spread in wildland fuels*, USDA FS, Ogden TU, Res. Pap. INT-115. 1972.

[2] R. C. Rothermel. *How to Predict the Spread and Intensity of Forest and Range Fires*, USDA FS, Ogden TU, Gen. Tech. Rep. INT-143, pp. 1–5. 1983.

[3] F. A. Albini. *Estimating wildfire behavior and effects*. Gen. Tech. Rep. INT-GTR-30. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 1976.

[4] B. Abdalhaq, *A methodology to enhance the Predction of Forest Fire Propagation*, PhD Thesis dissertation. Universitat Autònoma de Barcelona (Spain). June 2004.

[5] K. Wendt, A. Cortés and T. Margalef, *Knowledge-guided Genetic Algorithm for input parameter optimisation in environmental modelling*, Procedia Computer Science 2010, Volume 1(1), International Conference on Computational Science (ICCS 2010), pp. 1361–1369.

[6] G. Bianchini, A. Cortés, T. Margalef and E. Luque, *Improved Prediction Methods for Wildfires Using High Performance Computing A Comparison*, LNCS, Volume 3991, pp. 539–546, 2006.

[7] G. Bianchini, M. Denham, A. Cortés, T. Margalef and E. Luque, *Wildland Fire Growth Prediction Method Based on Multiple Overlapping Solution*, Journal of Computational Science, Volume 1, Issue 4, pp. 229–237. Ed. Elsevier Science. 2010.

[8] R. Rodríguez, A. Cortés, T. Margalef and E. Luque, *An Adaptive System for Forest Fire Behavior Prediction*, Proc. 11th IEEE Int'l Conf. Conference on Computational Science and Engineering (CSE 2008), IEEE Computer Society, pp. 275–282. 2008.

[9] R. Rodríguez, A. Cortés and T. Margalef, *Injecting Dynamic Real-Time Data into a DDDAS for Forest Fire Behavior Prediction*, Lecture Notes in Computer Science, Volume 5545(2), pp. 489–499, 2009.

[10] M. Denham, A. Cortés A. and T. Margalef, *Computational Steering Strategy to Calibrate Input Variables in a Dynamic Data Driven Genetic Algorithm for Forest Fire Spread Prediction*, Lecture Notes in Computer Science, Volume 5545(2), pp. 479–488, 2009.

[11] P. L. Andrews, *BEHAVE: Fire Behavior prediction and modeling systems - Burn subsystem*, part 1. General Technical Report INT-194. Odgen, UT, US Department of Agriculture, Forest Service, Intermountain Research Station. 1986.

[12] M. A. Finney, *FARSITE: Fire Area Simulator-model development and evaluation*, Res. Pap. RMRS-RP-4, Ogden, UT: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, 1998.

[13] M. Morais, *Comparing spatially explicit models of fire spread through chaparral fuels: a new algorithm based upon the Rothermel fire spread equation*. PhD Thesis, University of California, USA. 2001.

[14] A. Lopes, M. Cruz and D. Viegas *FireStation - An integrated software system for the numerical simulation of fire spread on complex toography*. Environmental Modelling and Software 17(3), pp. 269–285. 2002.

[15] FIRE.ORG - Public Domain Software for the Wildland fire Community. http://www.fire.org.

[16] fireLib User Manual and Technical Reference (online). http://www.fire.org/downloads/fireLib/1.0.4/doc.html.

[17] HFire Fire Spread Model homepage. http://firecenter.berkeley.edu/hfire/.

[18] F. Darema, *Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements*, ICCS 2004, LNCS 3038, Springer Berlin / Heidelberg, pp. 662–669. 2004.

[19] F. Darema, *Grid Computing and Beyond: The Context of Dynamic Data Driven Applications Systems*, Proceedings of the IEEE, Volume 93(3), pp. 692–697. 2005.

[20] Dynamic Data Driven Application Systems homepage. http://www.dddas.org.

[21] P. Beckman, S. Nadella, N. Trebon and I. Beschastnikh, *SPRUCE: A System for Supporting Urgent High-Performance Computing*, Grid-Based Problem Solving Environments, Volume 239/2007, pp. 295–311. 2007.

[22] G. Holmes, A. Donkin and I. H. Witten. *Weka: A machine learning workbench*, Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia. pp. 357–361. 1994.

[23] J. R. Quinlan. *Improved use of continuous attributes in c4.5*, Journal of Artificial Intelligence Research, Volume 4, pp. 77–90. 1996.

[24] R. E. Clark, A. S. Hope, S. Tarantola, D. Gatelli, P. E. Dennison and M. A. Moritz, *Sensitivity Analysis of a Fire Spread Model in a Chaparral Landscape*, Fire Ecology, Volume 4(1), pp. 1–13. 2004.

[25] H. Madsen and F. Jakobsen *Cyclone induced storm surge and flood forecasting in the northern Bay of Bengal*, Coastal Engineering, Volume 51, Issue 4, pp. 277–296. 2004.

[26] S. D. Aberson, *Five-day tropical cyclone track forecasts in the North Atlantic basin*, Weather and Forecasting, Volume 13, pp. 1005–1015. 1998.

[27] H. C. Weber, *Hurricane Track Prediction Using a Statistical Ensemble of Numerical Models*, Monthly Weather Review, Volume 131, pp. 749-770. 2003.