ISSN 1870-4069

A New Experimentation Module for the EPIC Software

Javier A. Hernández-Castaño[1], Yenny Villuendas-Rey[1], Oscar Camacho-Nieto[2], Carmen F. Rey-Benguría[3]

[1] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Mexico City, Mexico
[2] Instituto Politécnico Nacional, Secretaría de Extensión e Integración Social, Mexico City, Mexico
[3] Universidad de Ciego de Ávila, Centro de Estudios Educacionales "José Martí", Cuba
javierhc92@gmail.com, yenny.villuendas@gmail.com,
ocamacho@ipn.mx, carmenrb@sma.unica.cu

Abstract. In this paper, we introduce a new experimentation module for the recently developed EPIC software. EPIC is a tool for applying computational intelligence algorithms. The main advantages for our proposal concern the direct handling of mixed and incomplete data, the inclusion of several algorithms within the associative approach, and a very user-friendly graphical interface.

Keywords: computational intelligence, experimental tools, supervised classification.

1 Introduction

Intelligent Computing (IC) is an important branch of Computer Sciences, which has emerged recently as a scientific discipline [1, 2].

The introduction of IC is justified due to it bringing a computational solution to problems characterized as complex, due to the cost of obtaining the solutions, or due to the inexistence of exact solutions. For the development of new models and algorithms, it is necessary to compare them with respect to existing similar models; and for this task, several researching supporting tools have been developed. Among the most popular tools for supervised classification are WEKA [3, 4] and KEEL [5, 6]. Such tools hasten the researching process, as they include existing algorithms and procedures, and in some cases, they include ways to analyze the quality or performance of the algorithms under study. However, the researchers in the CI community suffer from numerous functionality insufficiencies exhibited in such tools.

The proposal of this research consists in the creation of a new experimentation module for the EPIC software [7]. With the inclusion of this new module, EPIC keeps the main functionalities and characteristics of the existent CI platforms and tools, and includes CI algorithms not considered by any other platform. In addition, the proposed module overcomes some of the deficiencies of the user interface shown by existing tools. EPIC software has a simple yet effective architecture, capable of fulfilling the needs of users, in particular the need of directly handling mixed and incomplete data,

pp. 243–252; rec. 2018-06-15; acc. 2018-08-23

without any data transforming or preprocessing, and the need of handling data belonging simultaneously to several decision attributes (multi-target classification). The main contribution of this paper is to develop a new module for EPIC, with a user interface to develop supervised classification experiments, which is friendlier and has more functionalities than the ones by other existing tools, such as WEKA and KEEL.

The rest of the paper is organized as follows. Section 2 details some of the previous works and Section 3 offers the description of the proposed module. Section 4 presents the discussion on the results obtained. Finally, the paper ends with some conclusions and future research suggestions.

2 Previous Works

From the point of view of the supervised classification, there are several tools to perform experiments. Among them, the mostly used are WEKA [3, 4] and KEEL [5, 6]. Both have a user interface, and allow the designing and execution of supervised classification experiments.

However, they have several disadvantages from the user point of view. Considering the above, and carrying out a deep analysis of both tools, we found that some of the drawbacks of WEKA experiment module are:

a. It does not allow the use of dissimilarity functions for mixed data descriptions (it only has distances, to be computed over real data).
b. It arbitrarily handles mixed and incomplete data (the architecture assumes that the feature values of instances are an array of doubles, and it converts the data to fulfill the architecture requirement).
c. It does not include any associative supervised classifier.
d. It does not allow other validation technique apart from Hold-Out and Cross Validation.
e. It does not cancel a single dataset while the experiment is running.
f. It does not serialize the results in a user-friendly way.
g. It does not serialize the results of each partition for each dataset.

On the other hand, although KEEL solves some the above mentioned drawbacks (a, d, g) the supervised classification experiment module of KEEL tool maintains drawbacks (b, c, e, f).

In order to solve the drawbacks 2, the new EPIC software [7] was developed. However, in its first release, it did not have a module for supervised classification experimentation. In this research, we develop such a module.

3 Supervised Experiments Module for EPIC

In this research, we decide to begin again, in order to develop an effective solution for the EPIC software, and to overcome the drawbacks shown by both WEKA and KEEL in their supervised experiment modules. We decide to use C# programming language, and the Integrated Development Environment (IDE) *Visual Studio Community* 2017, due to the facilities they offer to create a tool with a very user-friendly interface, and as it was the language used by EPIC developers. Despite C# not being a multiplatform language, we consider that its use will not represent a difficult, due to the widely extension of Windows operating system in Mexico and the rest of the world.

For the standard supervised classification experiment module, we had the following requirements:

a. The module must include supervised classifiers within the associative approach.
b. The module must include several validation techniques.
c. The user can select the desired validation technique (k-fold cross validation, k-fold stratified cross validation, 5x2 cross validation, 5x2 stratified cross validation or Distribution optimally balanced stratified cross validation).
d. The user can select the desired dataset, either in .ARFF or .Dat format
e. The user can select the desired supervised classifier and to freely configure all of the parameters.
f. The module must serialize the results of each classifier, over each partition of each dataset, including the real and assigned labels for each instance, as well as the corresponding confusion matrix. Such results must be provided in a compatible, friendly way.
g. The module must serialize a summary file, including the average results (of all partitions) of the datasets and classifiers, according to several performance measures, suitable for both balanced and multi-class imbalanced scenarios. Such results must be provided in a compatible, friendly way.
h. The module must allow the user to be able to cancel, at any time, the execution of the experiment only in a desired dataset, without affecting the execution of the experiments in the other datasets.

Considering those requirements, we designed a user interface. In order to update the module with the potential inclusion of other validation techniques and supervised classifiers, we use the *Visual Studio Community* 2017 IDE functionalities of Assembly to create at execution time, all of the related user-interface controls, such as buttons, labels, combo boxes, and so on.

To make an efficient use of computational resources, we program the module using asynchronous threads. It also allows us to cancel the execution of the experiment only in the desired datasets, as well as to show the user the overall progress of the experiment. Figure 1 will show the first user interface of the Standard Supervised Classification Experiment developed. Note, *a* and *b* requirement are fulfilled.

The module (Figure 2) include eight classifiers of the associative approach: HACT [8], CHAT-OHM [9], Gamma [10], Gamma with Differential Evolution (GammaED) [11], NAC [12], NAC with Differential Evolution (NACED) [13], SNDAM [14] and ACID. It also includes the ALVOT classifier [15] from the logical Combinatorial Approach to Pattern Recognition. Neither WEKA nor KEEL includes such classifiers.

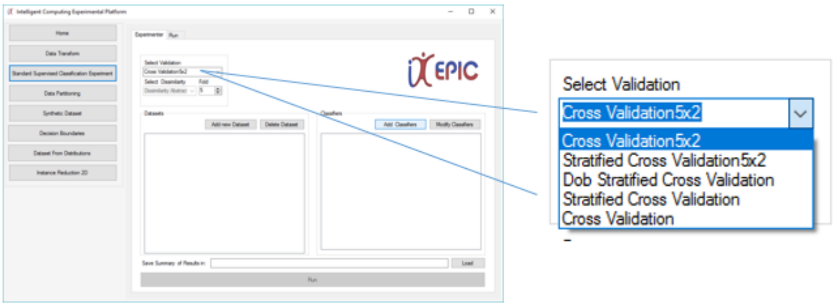

EPIC
Select Validation
Cross Validation5x2
Cross Validation5x2
Stratified Cross Validation5x2
Dob Stratified Cross Validation
Stratified Cross Validation
Cross Validation

Fig. 1. Validation procedures available in the module.

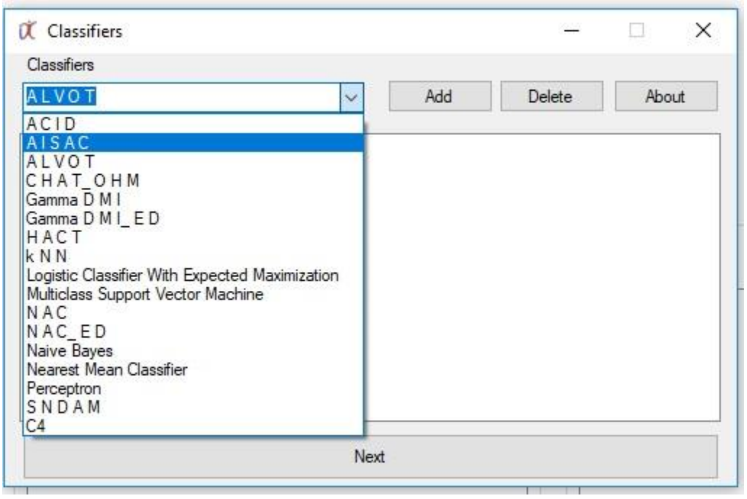


Fig. 2. Classifiers available in the module.

To address requirements *c* and *d*, the module automatically checks if the folder contains the files corresponding to the validation technique selected. If not, it shows an error message. Otherwise, it includes the desired dataset. An important advantage of this module, with respect to the ones by WEKA and KEEL, is that it allows including, in the same experiment, datasets in .ARFF format and datasets in .Dat format.

For requirement (*e*), the interface has the user controls according to the data types of the corresponding parameters. For example, if the parameter is a real number, the user interface will create a "Numeric Up Down" component, on the contrary, if the parameter is a dissimilarity function (for instance, for ALVOT classifier) the user-interface will create a "Combo box" component, and will fill this component, at execution time, with all the available classes having the IDissimilarity interface. It is important to highlight that the procedure for creating user-interfaces for parameter configuration is recursive (Figure 3).

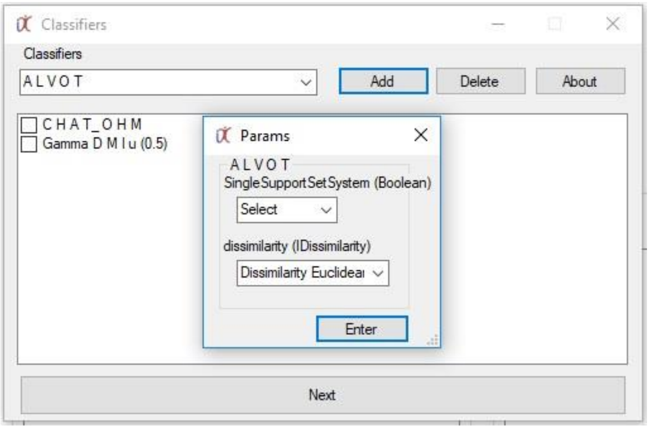
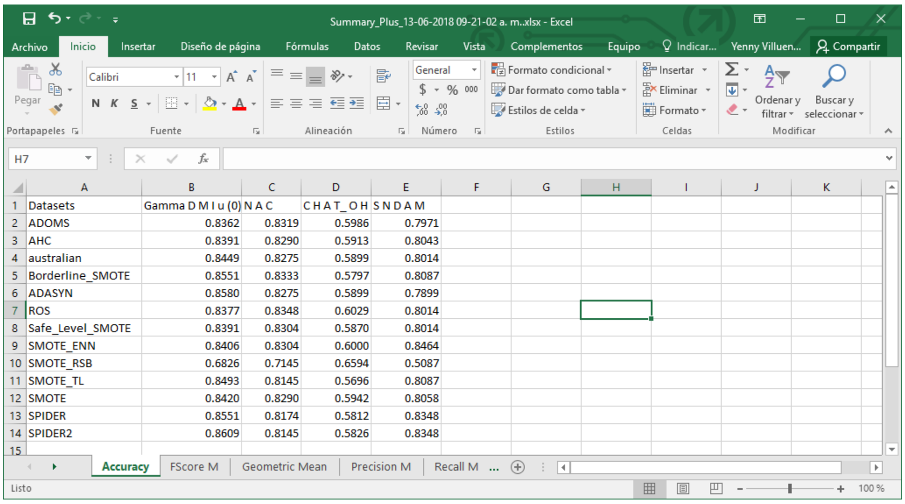


Fig. 3. Parameters of the classifiers detected at runtime.

For serializing requirements (*f* and *g*) the module creates a folder named “Results” in the same folder of each dataset. Into such folder, the module saves a file for each classifier. These files are .xlsx files, compatible with Microsoft Excel and Open Office. For each partition, the file has two sheets: one stating real and assigned labels for each test instance, and the other with the corresponding confusion matrix.

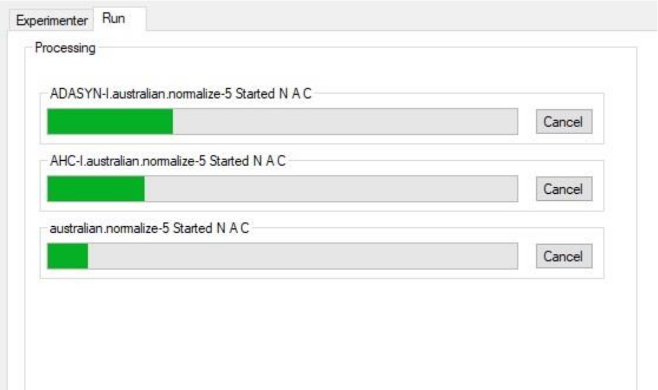


Datasets	Gamma D M I u (0)	NAC	CHAT_OH	SNDAM
ADOMS	0.8362	0.8319	0.5986	0.7971
AHC	0.8391	0.8290	0.5913	0.8043
australian	0.8449	0.8275	0.5899	0.8014
Borderline_SMOTE	0.8551	0.8333	0.5797	0.8087
ADASYN	0.8580	0.8275	0.5899	0.7899
ROS	0.8377	0.8348	0.6029	0.8014
Safe_Level_SMOTE	0.8391	0.8304	0.5870	0.8014
SMOTE_ENN	0.8406	0.8304	0.6000	0.8464
SMOTE_RSB	0.6826	0.7145	0.6594	0.5087
SMOTE_TL	0.8493	0.8145	0.5696	0.8087
SMOTE	0.8420	0.8290	0.5942	0.8058
SPIDER	0.8551	0.8174	0.5812	0.8348
SPIDER2	0.8609	0.8145	0.5826	0.8348

Fig. 4. Results obtained in the Summary of the performance measures.

In addition, at the end of the experiment, the module saves a summary file (Figure 4). This file is also a .xlsx file, and has five sheets, one for each performance measure: accuracy, F-Score M, Geometric Mean, Precision m and Recall M measures. All these measures are given in a tabular form, with datasets in the rows and supervised classifiers in the columns. This summary allows the user to quickly report the results, and to easily include graphics, figures and other Microsoft Excel related elements.

For canceling requirement (*h*) the module has a user interface showing the current progress of the experiment (Figure 5) and it allows canceling a single dataset at any time (Figure 6).

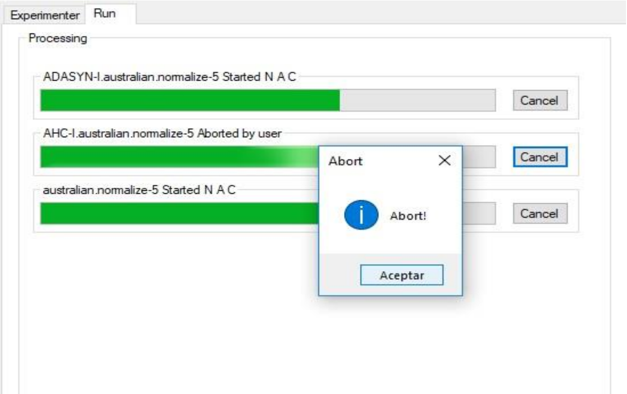


ADASYN-I.australian.normalize-5 Started N A C

AHC-I.australian.normalize-5 Started N A C

australian.normalize-5 Started N A C

Fig. 5. Execution of an experiment.



ADASYN-I.australian.normalize-5 Started N A C

AHC-I.australian.normalize-5 Aborted by user

australian.normalize-5 Started N A C

Abort!

Fig. 6. Cancelling a dataset.

This functionality keeps the experiment working without the canceled dataset, and saves time and effort to the users.

4 Results and Discussion

In this section, we define several dimensions and indicators in order to evaluate the proposed module for supervised classification experiments. We consider four dimensions, from the user point of view. Each of these dimensions has several indicators of usability. Table 1 shows the dimensions and their corresponding indicators, while Table 2 addresses the corresponding comparison.

We consider four dimensions: Versatility, Execution, Serialization of results and Interpretation of results. All of them have several indicators. We emphasize from the user point of view, for as researchers, we want a tool easy to use, user-friendly, as well as being intuitive and at the same time robust. For the research process, it is very good to have summaries; have all of the information available, and to store it for further use, as well as to be able to easily navigate for the information.

The most used formats for supervised classification datasets are .ARFF format (designed by the WEKA team) and the .Dat format (designed by the KEEL team). Also it is possible to convert .ARFF into .Dat and vice versa in the KEEL software (not in WEKA). These conversions lead to unnecessary computation and storage costs, particularly for big data. Thus, we consider as an important issue to be able to handle, simultaneously datasets for each format, increasing usability and versatility of the supervised classification experimental modules.

Table 1. Dimensions and indicators of module usability.

Dimensions	*Indicators*
1. Versatility	1.1 Simultaneous handling of both .ARFF and .Dat datasets 1.2 Sampling procedures for classifier validation 1.3 Use of specific (serialized) partitions as a hole 1.4 Evaluation of datasets stored independently of the module
2. Execution	2.1 Automatic parametrization of classifiers 2.2 Cancelation of a desired dataset 2.3 Visualization of the experiment progress
3. Serialization of results	3.1 Serialization of the classification results for each classifier on each partition (real vs assigned labels) 3.2 Serialization of the confusion matrix for each classifier on each partition 3.3 Organization of the serialized information
4. Interpretation of results	4.1 Automatic calculation of average (by partition) performance measures 4.2 Performance measures 4.3 Intuitive interpretation of results 4.4 Existence of global (classifiers vs datasets) summaries of performance

Another important issue for supervised classification experiments is the validation (or data partition) procedure. There are several well-known procedures, such as Leave One Out, Hold Out and Cross Validation. For the last two, there are stratified versions, able to divide the data considering class distribution. In literature, stratification is a “must be” for experimentation, with special relevance over imbalanced data.

In addition, due to the computational complexity of several algorithms, and the high volume of some datasets, the possibility of cancelling the execution of the experiment only in the desired datasets, without losing all the experiment is very important. Neither WEKA nor KEEL offers such functionality to the user. In both cases, if the user wants to cancel the execution of the experiment in a single dataset, it must cancel the entire experiment, and must start over, losing all the computations made so far. On the other hand, the proposed module for EPIC software, allows the user to cancel a desired dataset, and the experiments continues executing in the remaining data. In addition, the final performance summaries are not affected by the cancellation. We believe that this is a huge improvement for researchers and students who use the tools for experimentation.

Table 2. Evaluation of the proposed module with respect to others.

Indicator	*WEKA*	*KEEL*	*EPIC*
1 Versatility			
1.1 Simultaneous handling of both .ARFF and .Dat datasets	No	No	Yes
1.2 Sampling procedures for classifier validation	2[*]	3[†]	5[‡]
1.3 Use of specific (serialized) partitions as a hole	No	Yes	Yes
1.4 Evaluation of datasets stored independently of the module	Yes	No	Yes
2 Execution			
2.1 Automatic parametrization of classifiers	Yes	Yes	Yes
2.2 Cancelation of a desired dataset	No	No	No
2.3 Visualization of the experiment progress	Partial	No	Yes
3 Serialization of results			
3.1 Serialization of the classification results for each classifier on each partition (real vs assigned labels)	No	Yes	Yes
3.2 Serialization of the confusion matrix for each classifier on each partition	No	Yes	Yes
3.3 Organization of the serialized information	Bad	Bad	Good
4 Interpretation of results			
4.1 Automatic calculation of average (by partition) performance measures	No	No	Yes
4.2 Performance Measures (by partition)	49[§]	2[**]	5[††]
4.3 Intuitive interpretation of results	No	No	Yes
4.4 Existence of global (classifiers vs datasets) summaries of performance	No	No	Yes

Another important issue is to store the real and assigned labels for a dataset, and the corresponding confusion matrix. Such information is very useful for data analysis, and

[*] Hold-Out and k-fold Cross Validation

[†] 5x2 Cross Validation, k-fold Cross Validation and k-fold Distribution Optimally Balanced Stratified Cross Validation

[‡] 5x2 Cross Validation, 5x2 Stratified Cross Validation, k-fold Cross Validation, k-fold Stratified Cross Validation and k-fold Distribution Optimally Balanced Stratified Cross Validation

[§] In the authors' opinion, most of them are useless. In addition, WEKA computes several measures (such as Area under ROC Curve) under circumstances where it cannot be computed (for instance in multi-class datasets).

[**] Accuracy and Area under ROC Curve (the last for 2-classs datasets only, as it should be)

[††] Accuracy, F Score M, Geometric Mean of Recall, Precision M and Recall M

having a confusion matrix allows the researcher to compute almost all desired performance measures, due to they are based on this matrix.

In addition, for analysis and reports, researcher need summarized information (for instance, the average results over the k-folds for each classifier over each dataset) of the desired performance measures. If the information is not summarized, there is a significant lack of time and effort to obtain such summaries. WEKA does not provide summarized files of results, just a single, big file with the information of the entire experiments. KEEL does provide the summaries, in an unstructured plain text file. For instance, to obtain a bar graphic of the summaries, it is necessary to copy the information, to format it, and then to organized into other program, such as Microsoft Excel, Numbers or similar. The proposed module gives the user a single .xlsx file, with five sheets, having the summary for the corresponding performance measure. From the user point of view, this is an important advance, gaining time and having the desired information without effort. Another aspect to highlight is that the information is well organized, intuitive, easy to use, and explain.

5 Conclusions and Future Work

This paper introduces a new module for standard experimentation on supervised classification, for the EPIC software. The proposal has several advantages with respect to the state- of- art, and includes significant contributions from the user point of view. In the future we will be working on other modules, for weighted supervised classification, as well as for data preprocessing.

Acknowledgments. The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas, Secretaría de Investigación y Posgrado, CIDETEC and SEIS), the Consejo Nacional de Ciencia y Tecnología, and Sistema Nacional de Investigadores for their economical support to develop this work.

References

1. Teti, R., Kumara, S.: Intelligent computing methods for manufacturing systems. Cirp Annals 46, pp. 629–652 (1997)
2. Mandal, J.K., Paramartha, D., Mukhopadhyay, S.: Advances in Intelligent Computing. Springer (2019)
3. Holmes, G., Donkin, A., Witten, I.H.: Weka: A machine learning workbench. In: Intelligent Information Systems. Proceedings of ANZIIS '94, pp. 357–361. IEEE (1994)
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD explorations newsletter 11, pp. 10–18 (2009)
5. Alcalá-Fdez, J., Sanchez, L., Garcia, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Computing 13, pp. 307–318 (2009)
6. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: data set repository, integration of algorithms and

experimental analysis framework. Journal of Multiple-Valued Logic & Soft Computing 17, (2011)

7. Hernández-Castaño, J.A., Camacho-Nieto, O., Villuendas-Rey, Y., Yáñez Márquez, C.: Experimental Platform for Intelligent Computing (EPIC). Computación y Sistemas 22, pp. 245–253 (2018)
8. Santiago-Montero, R.: Clasificador Híbrido de Patrones basado en la Lernmatrix de Steinbuch y el Linear Associator de Anderson-Kohonen. Centro de Investigación en Computación. Master Thesis. Instituto Politécnico Nacional. Mexico (2003)
9. Uriarte-Arcia, A.V., López-Yáñez, I., Yáñez-Márquez, C.: One-hot vector hybrid associative classifier for medical data classification. PloS one 9, e95715 (2014)
10. López-Yáñez, I., Sheremetov, L., Yáñez-Márquez, C.: A novel associative model for time series data mining. Pattern Recognition Letters 41, pp. 23–33 (2014)
11. Ramirez, A., Lopez, I., Villuendas, Y., Yanez, C.: Evolutive improvement of parameters in an associative classifier. IEEE Latin America Transactions 13, pp. 1550–1555 (2015)
12. Villuendas-Rey, Y., Rey-Benguría, C.F., Ferreira-Santiago, Á., Camacho-Nieto, O., Yáñez-Márquez, C.: The naïve associative classifier (NAC): a novel, simple, transparent, and accurate classification model evaluated on financial data. Neurocomputing 265, pp. 105–115 (2017)
13. Serrano-Silva, Y.O., Villuendas-Rey, Y., Yáñez-Márquez, C.: Automatic feature weighting for improving financial Decision Support Systems. Decision Support Systems 107, pp. 78–87 (2018)
14. Ramírez-Rubio, R., Aldape-Pérez, M., Yáñez-Márquez, C., López-Yáñez, I., Camacho-Nieto, O.: Pattern classification using smallest normalized difference associative memory. Pattern Recognition Letters 93, pp. 104–112 (2017)
15. Ruiz-Shulcloper, J., Ponce, E., López, N.: ALVOT, system of programs of voting algorithms for classification. Revista Ciencias Matemáticas (In Spanish) 7, pp. 41–67 (1986)