ATLANTIS PRESS International Journal of Computational Intelligence Systems, Vol. 10 (2017) 1238–1249

KEEL 3.0: An Open Source Software for Multi-Stage Analysis in Data Mining

Isaac Triguero [1], Sergio González [2], Jose M. Moyano [4], Salvador García [2], Jesús Alcalá-Fdez [2], Julián Luengo [2], Alberto Fernández [2], Maria José del Jesús [5], Luciano Sánchez [3], Francisco Herrera [2]

[1] School of Computer Science
University of Nottingham, Jubilee Campus
Nottingham NG8 1BB, United Kingdom
E-mail: Isaac.Triguero@nottingham.ac.uk

[2] Department of Computer Science and Artificial Intelligence
University of Granada, Granada, Spain, 18071

[3] Department of Computer Science
University of Oviedo, Gijón, 33204, Spain

[4] Department of Computer Science and Numerical Analysis
University of Cordoba, 14071 Cordoba, Spain

[5] Department of Computer Science
University of Jaén, Jaén, Spain

Received 6 March 2017

Accepted 9 September 2017

Abstract

This paper introduces the 3rd major release of the KEEL Software. KEEL is an open source Java framework (GPLv3 license) that provides a number of modules to perform a wide variety of data mining tasks. It includes tools to perform data management, design of multiple kind of experiments, statistical analyses, etc. This framework also contains KEEL-dataset, a data repository for multiple learning tasks featuring data partitions and algorithms' results over these problems. In this work, we describe the most recent components added to KEEL 3.0, including new modules for semi-supervised learning, multi-instance learning, imbalanced classification and subgroup discovery. In addition, a new interface in R has been incorporated to execute algorithms included in KEEL. These new features greatly improve the versatility of KEEL to deal with more modern data mining problems.

Keywords: Open Source, Java, Data Mining, Preprocessing, Evolutionary Algorithms.

1. Introduction

Data Mining (DM) techniques [25] are widely used in a broad number of applications that go beyond the computer science field [41]. In order to ease the access to these models for people not directly related to computer science, many commercial and non-commercial software suites have been made available. The majority of the former are commercially distributed (e.g. SPSS Clementine, Oracle Data Mining or KnowledgeSTUDIO), but there is still a good number of open source tools. Among the existing open source applications, Workflow-based environments allow us to visually chain a number of DM methods together in a pipeline. The most used DM apps of this kind are: Weka [18], KNIME [1] and KEEL [2].

The KEEL software [5,4]*(Knowledge Extraction based on Evolutionary Learning) was originally developed as a tool mainly focused on the implementation of evolutionary algorithms and soft computing techniques for standard DM problems such as regression, classification or association rules, as well as data preprocessing techniques [23,14]. KEEL was launched in 2009 [5] and later upgraded in 2011 [4] as a non-commercial Java suite, so that, it could be on all major platforms. KEEL provides a simple GUI to design experiments with different data sets and computational intelligence algorithms in order to assess the behaviour of the algorithms. Moreover, it was designed with a two-fold goal: research and educational.

A screenshot of the main window of KEEL 3.0 is shown in Figure 1, where its main components are highlighted. This suite came along with KEEL-dataset†, a repository that includes standardized data set partitions for comparison purposes in the KEEL format. These can be used for comparison versus several algorithms of the specialised literature, as it also comprises some experimental results on these data sets. However, this platform is continuously evolving towards a more flexible and comprehensive tool that allows us to deal with new and more complex DM problems (namely semi-supervised learning [42], imbalanced classification [31] or multi-instance learning [6]).

Fig. 1. Screenshot of the main window of KEEL 3.0.

In this paper, we present the 3rd major release of KEEL, describing its new components. We release this version as an open source software under the terms of the GNU Public License GPLv3. The reasoning behind this is clear; Our implementation is publicly available for all the research community in the DM area, as well as non-specialised users that can take advantage of these models for their own applications. Furthermore, we support transparency, meritocracy, and community development [35,3]. The major contributions of this new release are:

- We have included four new modules focused on new problems and techniques. This considers a module for semi-supervised problems, imbalanced data, multi-instance learning and subgroup discovery [26].
- A package named RKEEL [32] has been recently created as a layer between R and KEEL, allowing users to execute KEEL functionalities from R.
- A new collection of data sets have been incorporated to the KEEL-dataset repository. We have added data sets for specific problems such as: Imbalanced, multi-label, semi-supervised, noisy and low-quality data.
- A new external GUI for nonparametric statistical analysis has been included. It allows users to analyse and compare their results from previous experiments on KEEL or any other source [15,21].
- Many new algorithms have been added to the original modules, including preprocessing and learning models.
- Documentation and project administration-wise have been greatly enhanced. We must refer to KEEL 3.0 on platforms such as GitHub and MLloss to foster external collaboration and usability. In addition, the API (javadocs) is now available to promote better integration with other tools.

This paper is organised as follows. Section 2 describes in detail the main novelties of this version. Section 3 compares KEEL 3.0 to other existing DM software. Section 4 discusses about the availability of KEEL 3.0 and its community. Finally, Section 5 concludes the paper.

* http://www.keel.es, http://mloss.org/software/users/keel, https://github.com/SCI2SUGR/KEEL

† http://www.keel.es/datasets.php

2. KEEL 3.0: New modules and features

This new version of KEEL improves upon the previous versions. From the first release of KEEL, a set of already well-established algorithms are present as classical learning methods in the software. KEEL 3.0 has now 514 algorithms integrated, consisting of 98 data preprocessing algorithms, 368 learning algorithms and 24 statistical test procedures.

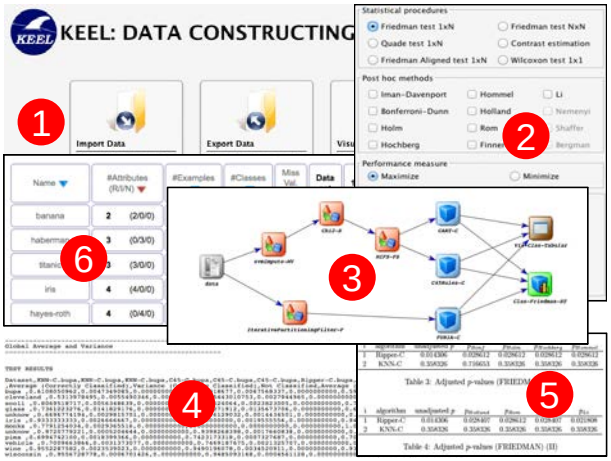

Fig. 2. Some KEEL 3.0 snapshots.

This section revises the main components added to KEEL 3.0, including new modules, new data sets, an interface to connect KEEL and R, as well as a case of study with one of the aforementioned modules. Figure 2 presents a graphic summary of KEEL 3.0, highlighting its main features: (1) data management section, including import/export, visualisations, editions and partitioning of data; (2) GUI for statistical analysis, with the statistical procedures described in [15,21]; (3) Flexible experiments configuration, showing an example of experiment illustrated by means of a flowchart, involving four data preprocessing algorithms (Chi2 discretizer, Mutual Information feature selection, SVM missing values imputation and Iterative Partitioning noise filter), three classifiers (CART, C4.5 Rules and FURIA), tabular visualisation of results and a Friedman test based analysis; (4) Output in CSV format of a tabular summary of results; (5) LaTeX output of statistical analysis; (6) Repository of data sets already partitioned, including data sets with missing values, noise, low quality data, imbalanced and semi-supervised classification, multi-instance and multi-label learning, regression and time series.

In the following subsections, we detail the functioning of these new features of KEEL 3.0. Nevertheless, the full description can be found in the new web-based Reference Manual at `http://www.keel.es/development.php`.

2.1. *New data mining scenarios*

Real world applications are demanding more flexible DM models that can deal with very challenging scenarios. Traditional supervised and unsupervised techniques may not fit well with the kind of data available in many of these applications. For this reason, KEEL now includes a number of extra modules that provide us with more advanced DM models for complex scenarios. In what follows, we discuss the four new modules added in KEEL.

2.1.1. *Imbalanced Learning*

In many supervised learning applications, we may run into the situation where there is scarcity of a particular class of samples. Taking binary problems as an example, this issue is typically known as the class imbalanced problem [31,10], in which positive data samples (usually the class of interest) are highly outnumbered by negative ones [29]. This issue brings along a series of difficulties such as overlapping, small sample size, or small disjunct.

The KEEL Software Suite accounts for this scenario of classification and it includes a complete framework for the experimentation of this type of problems. Several approaches have been designed to tackle this problem, which can be divided into two main alternatives: (1) internal approaches that create new algorithms or modify existing ones to take the class-imbalance problem into consideration and (2) external approaches that pre-process the data in order to diminish the effect of their class imbalance. In addition, cost-sensitive learning solutions incorporating both the data (external) and algorithmic level (internal) approaches assume higher misclassification costs for samples in the minority class and seek to minimise the high cost errors. Ensemble methods are also frequently adapted to imbalanced

domains, either by modifying the ensemble learning algorithm at the data-level approach to pre-process the data before the learning stage of each classifier or by considering to embed a cost-sensitive framework in the ensemble learning process.

KEEL includes all of these approaches in this new module. Figure 3 summarises the three main contributions of this module:

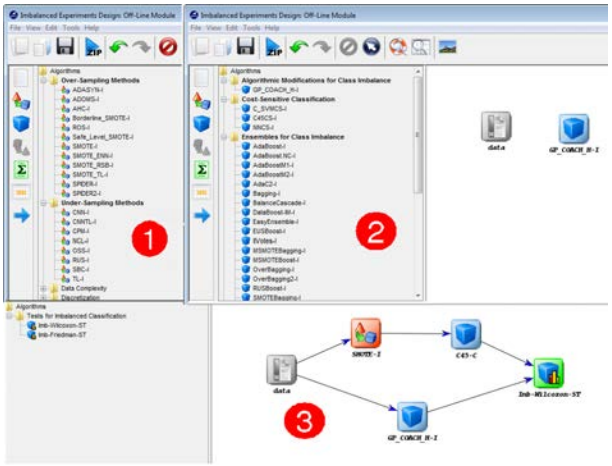

Fig. 3. Imbalanced learning module Main Characteristics: (1) Under-Sampling and Over-Sampling Models, (2) Imbalanced Learning Algorithms, (3) Tailored Statistical tests.

1. **Preprocessing techniques**: Apart from the existing preprocessing techniques included in the original KEEL Experiment section, this module includes two new categories: Over-Sampling Methods and Under-Sampling techniques. These preprocessing techniques may be later connected to standard data mining models.

2. **Methods**: KEEL provides tailored algorithm for the class-imbalanced problem. It contains the state-of-the-art in ensemble learning [20] and cost-sensitive classification.

3. **Visualisation and Statistical Tests**: As in the case of standard classification, KEEL includes a number of visualisation and statistical utilities. For this module, these have been modified to take into account the imbalanced problem. Specifically, it uses geometric mean and area under ROC curve as more appropriate performance measures for this scenario.

2.1.2. *Semi-Supervised Learning*

The Semi-Supervised Learning (SSL) paradigm has attracted much attention in many different fields where it is easier to obtain unlabelled than labelled data because it typically requires less effort, expertise and time-consumption. In this context, traditional supervised learning is limited to using labelled data to build a model. Nevertheless, SSL is a learning paradigm concerned with the design of models in the presence of both labelled and unlabelled data.

KEEL 3.0 includes a dedicated module to deal with this kind of problems. Similarly to the standard experiments design module, it allows the user to create experiments on a work-flow fashion. It also grants us the possibility of testing the transductive and inductive capabilities of these methods. This module already includes a great number of self-labelled techniques for SSL, from classical models such as Self-Training and Co-Training to state-of-the-art techniques as Tri-Training or Democratic Co-Learning. All of those methods were experimentally evaluated in [37]. The addition of tailored preprocessing techniques for this scenario [36] as well as other families of methods such as Graph-based models [13] is still in progress. Nevertheless, the software is ready to add both kind of techniques.

2.1.3. *Multi-Instance Learning*

Multiple instance learning (MIL) [27] is a generalisation of traditional supervised learning. In MIL, training patterns called bags are represented as a set of feature vectors called instances. Each bag contains a number of non-repeated instances and each instance usually represents a different view of the training pattern attached to it. There is information about the bags and each one receives a special label, although the labels of instances are unknown. The problem consists of generating a classifier that may correctly classify unseen bags of instances. The key challenge in MIL is to cope with the ambiguity of not knowing which instances in a positive bag are actually positive examples, and which ones are not. In this sense, a multiple instance learning problem can be regarded as a special kind of supervised

learning problem with incomplete labelling information.

Although MIL is a relatively recent learning framework, the MIL setting has produced numerous and interesting applications in different domains which have improved considerably the previous results achieved with other learning frameworks due to greater flexibility in their representation.

This new version of KEEL adds a module to perform MIL experiments. This module includes 9 algorithm from the state-of-the-art [6]. As in the case of SSL, no preprocessing techniques are available at the moment, but the module is ready to integrate such kind of algorithm when available.

2.1.4. *Subgroup discovery*

Subgroup discovery (SD) [26,9] consists of extracting interesting rules with respect to a target variable. SD is a problem somewhere halfway between predictive and descriptive induction. Its goal is to generate single and interpretable subgroups to describe the relations between independent variables and a certain value of the target variable. Since this kind of techniques will work on exactly the same kind of input data as standard classification, this module has been directly incorporated as a section of the experiment design module. Thus, all the preprocessing techniques included in KEEL can be used prior to the application of the subgroup discovery models. In addition, these techniques do not only make predictions but they also provide descriptive rules and a set of quality measures as output [11].

2.2. *New Data preprocessing and Learning techniques*

KEEL is a project continuously in expansion. Therefore, existing modules such as the standard experiments design increase their number of algorithms very rapidly. We enumerate the new additions as follows:

- **Preprocessing**: An entire set of data preprocessing algorithms have been added, including the state-of-the-art in discretisation [24,34] and noise filtering [19]. Additionally, newly developed algorithms in instance selection, feature selection and missing data imputation have also been incorporated [22].
- **Learning**: New families of models such as lazy learning (including an exhaustive set of Fuzzy k nearest neighbours methods [16]) and associative classification as well as new implementations for rule induction and decision trees, statistical learning, association rule mining and fuzzy based learning [28,17] are now available.

2.3. *KEEL-dataset*

The KEEL-dataset repository‡ was established to provide data sets in KEEL format so that the integration with KEEL would be straightforward. Figure 4 shows a snapshot of the currently existing data sets in KEEL-dataset repository. It includes experimental studies and results on many of those data sets. Nowadays, more than 900 data sets are available for download, covering all the modules existing in KEEL 3.0, and more. Specifically, we now include specific data sets and results for: Imbalanced, multi-label, SSL, noisy and low-quality data.

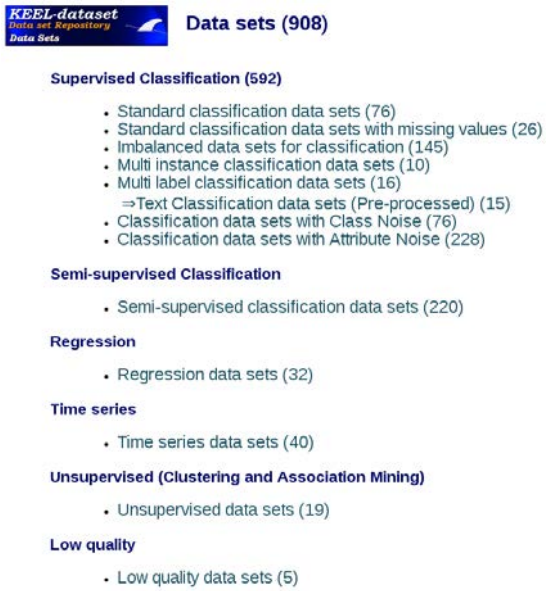

Fig. 4. KEEL-dataset.

‡ `http://keel.es/datasets.php`

2.4. A dedicated module for statistical analyses

Since the previous version, KEEL included a statistical library to analyse the results of the algorithms. These analyses were limited to be performed within the design of experiments module. In this way, only those algorithms included in this particular experiment were statistically evaluated. To make our statistical library even more flexible, KEEL 3.0 now provides a separated module for non-parametric statistical tests [15,21]. More information about this kind of statistical test can be found at `http://sci2s.ugr.es/sicidm/`.

This module takes as input a file in CSV format or the user can manually introduce their results or copy-paste them on the cells of an interactive table. Figure 5 shows how the module looks like.

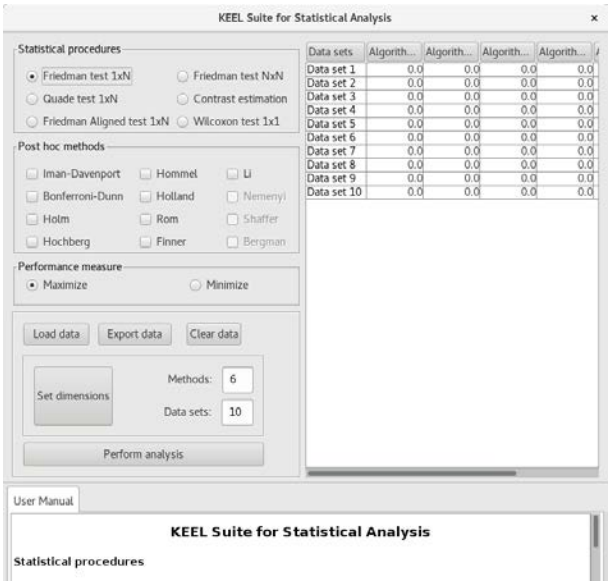

Fig. 5. Non-parametric statistical test Module.

There are a number of statistical procedures for 1-to-N and N-to-N comparisons, respectively. These tests come together with post hoc models to characterise the differences detected by the statistical tests.

Depending on the characteristics of the problem considered, it is possible to perform the statistical test for maximisation and minimisation purposes. This feature allows us to determine if the results have been obtained from a maximisation problem (e.g. using accuracy in supervised classification problems) or from a minimisation problem (e.g. using mean squared error in regression problems).

The resulting statistical analyses are summarised in a report which is provided in Latex format.

2.5. RKEEL:Run KEEL 3.0 with R

R is an interactive programming language and an environment that provides a wide variety of methods for statistics, classification, association rules, regression, linear and non-linear models, graphical representation of data, and so on as packages. One of the greatest virtues of R is that it can be easily extended through packages. These packages can be downloaded from different repositories, being CRAN the main repository. In this way, researchers have available a wide variety of functionalities and algorithms already programmed and tested without needing to know how they are made.

RKEEL [32] is a recently published package at CRAN repository [33], which provides an interface to execute from R some preprocessing, classification, regression and association rule algorithms integrated in the KEEL software tool. Thus, R developers can now take advantage of the wide variety of algorithm available in KEEL.

The RKEEL package has the following main structure: the *KeelAlgorithm* class implements the main methods and properties of any KEEL algorithm to be interfaced. Then, the *ClassificationAlgorithm*, *PreprocessAlgorithm*, *RegressionAlgorithm* and *AssociationRulesAlgorithm* classes, which inherit from the base *KeelAlgorithm* class, implement each one the main features of a KEEL classification, preprocessing, regression and association rules algorithm, respectively. Any interfaced algorithm from KEEL must inherit from its corresponding class. Furthermore, the RKEEL package depends on other R packages to ensure its performance, such as XML [30], R6 [12], doParallel [7], foreach [8], gdata [40] and rJava [39]. Also, RKEEL needs at least Java version 8 installed on the computer to run the KEEL algorithms.

As a case of study, we show how to use in R all the functionality offered by RKEEL to work with association rules as a particular example (Figure 6). First, the user has to install and load the RKEEL package (lines 2-3). Then, one of the included data

sets in RKEEL can be loaded with the *loadKeelDataset* method (line 6). The "?" character shows the help for a specific algorithm or method (line 9) and, in order to create an association rules algorithm, the name of the algorithm with the data set between parenthesis is necessary (line 11).

```
1  > #Install and load the package
2  > install.packages("RKEEL")
3  > library(RKEEL)
4  >
5  > #Load data set
6  > dat <- loadKeelDataset("iris")
7  >
8  > #Show algorithm help
9  > ?FPgrowth_A
10 > #Create the algorithm object with the
       dataset
11 > algorithm <- FPgrowth_A(dat)
12 >
13 > #Run algorithm
14 > algorithm$run()
15 >
16 > #Rule set in arules format
17 > algorithm$rules
18 > #Show rules
19 > algorithm$showRules(2)
20 > #Get interest measures
21 > algorithm$getInterestMeasures()
22 > #Add new interest measure
23 > algorithm$addInterestMeasure("YuleY","
       yulesY")
24 > #Sort rule set by interest measure
25 > algorithm$sortBy("yulesY")
26 >
27 > #Export rules with interest measures in
       CSV format
28 > algorithm$writeCSV("myrules")
29 > #Export rules with interest measures in
       PMML format
30 > algorithm$writePMML("myrules")
31 > #Generate .tex file with measures table
32 > algorithm$writeKeelLatexTables("table")
```

Fig. 6. Example of use in R.

To run the algorithm, the user simply has to call the *run* method of the object (line 14). Once the algorithm is executed, the user can check the number of rules of the generated rules set (line 17), show the rules (line 19) or get the interest measures of the different rules (line 21). Also, RKEEL allows us to add new interest measures to the rules set (line 23). Furthermore, the user can sort the rules by an interest measure with the *sortBy* method (line 25), as well as export the rules and results in CSV and PMML formats (lines 28-30). Finally, the interest measures of the different rules can be exported as tex file (line 32).

2.6. *Case Study in KEEL 3.0*

The purpose of this section is to illustrate the use and potential of the new KEEL modules. To do so, we have developed a study with several methods included in the new imbalanced classification module (see Figure 7 (1)) using ten data sets available in the KEEL-dataset repository[§] for the experiment.

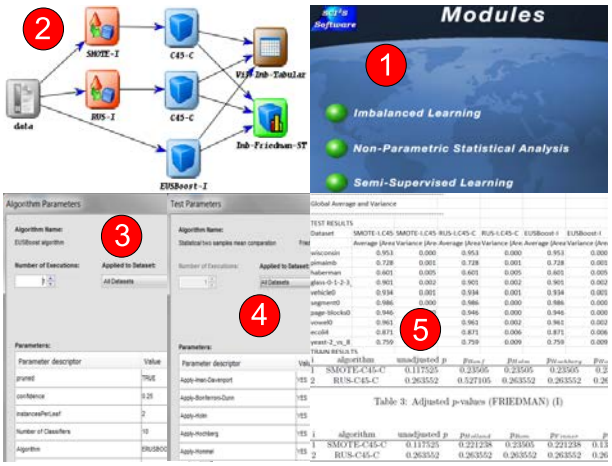

Fig. 7. Some snapshots taken from KEEL for this study case.

The data flow and results from the methods and statistical techniques are shown in Figure 2 (2). In this example, we have used the data preprocessing algorithms SMOTE (based on oversampling) and RUS (based on undersampling) with the classifier C4.5, and the current ensemble classifier EUS-Boost that uses an evolutionary undersampling approach as preprocessing method in a boosting procedure with the classifier C4.5. We have used a 5-fold cross validation procedure and the parameters of the methods have been selected according to the recommendation of the corresponding authors within each proposal, which are the default parameter settings included in KEEL, although they can be adjusted by clicking twice on the node (see Figure 2 (3)). Once the experiment has been run, by including the output nodes tabular visualisation and Friedman test,

§ http://www.keel.es/imbalanced.php

we can find several summary results files including both the confusion matrices and performance values with the AUC, and statistical analyses of the former results (Figure 2 (4) shows the configuration window of test node). By the analysis of the results presented in Figure 2 (5), we can highlight that EUS-Boost outperforms the remaining methods in terms of average AUC for the test data. Furthermore, the Friedman test shows that there are significant differences among the observed results and the hypothesis of equality is rejected by the post-hoc tests. This experiment can be downloaded following the link `http://www.keel.es/software/KEEL_caseStudy.zip`

3. KEEL compared to other DM tools

This section briefly summarises the main difference between KEEL 3.0 and the two most significant workflow-based DM software written in Java, KNIME and WEKA. A full report on KEEL vs. KNIME and WEKA can be found at `http://www.keel.es/documents/KEELcomparisonMLtools.pdf`.

We must point out that the ultimate objective of this study is to stress which are those characteristics that support the main strengths for each particular tool, rather than to establish a comparison for the advantages of one against another. Therefore, we have selected several features that will allow the reader to determine the major differences among the software tools, and then to categorise KEEL as a valuable alternative to these suites when other specific research requirements are needed.

Specifically, we distinguish four levels of support in these characteristics: none (N), basic support (B), intermediate support (I) and advanced support (A). If features do not have intermediate levels of support, the notation used is checked (✓) for supporting and non-checked (✗) for not supporting.

The following criteria are used for comparison purposes:

- **Off/On-line run of the experiment set up**. An On-line run implies that the tool interface and algorithm modules need to be in the same machine and the experiments are completely dependent on the software tool. An off-line run entails the independence of the experiments created with respect to the suite interface, allowing the experiment to be executed later on in other machines.
- **Pre-processing Variety**. This comprises the availability of discretisation, feature selection, instance selection and/or missing values imputation methods. The trend of most of the suites is to offer a good feature selection and discretisation set of methods, but they often neglect specialised methods of missing values imputation and instance selection. Usually, the contributions included are basic modules of replacing or generating null values and methods for sampling the data sets by random (stratified or not) or by value-dependence.
- **Learning Variety**. It is supported over main areas of DM, such as predictive tasks (classification, regression, anomaly/deviation detection), and descriptive tasks (clustering, association rule discovery, sequential pattern discovery). Additionally, we take into account several novel DM scenarios such as SSL, Imbalanced Classification and MIL.
- **Advanced Features**. This part includes some of the less common criteria incorporated for extending the functionality of a software tool.

  - *Post-processing techniques*, usually devoted to tuning the model learned by an algorithm.
  - *Meta-learning*, which includes more advanced learning schemes, such as bagging or boosting, or meta learning of the algorithm parameters.
  - *Statistical tests* for establishing comparisons of results. An advanced support of this property requires a complete set of parametric and non-parametric statistical tests; a basic support implies the existence of well-known standard statistical tests (such as t-test).
  - *Evolutionary Algorithms* (EAs) support indicates the integration of EAs into the DM areas that the software tool offers. A basic support of this feature implies the use of genetic algorithms in some techniques (usually, genetic feature selection). To upgrade the level it is necessary to incorporate EAs in learning or meta-learning models.

Table 1. Summary of characteristics of KEEL, KNIME, and WEKA software tools: run types and pre-processing variety.

Software	Run Type		Pre-processing variety				
	On-line	Off-line	Discretisation	Feature Selection	Instance Selection	Training Set Selection	Missing Values Imputation
KEEL	✓	✓	A	A	A	A	A
KNIME	✓	✗	I	A	B	N	B
WEKA	✓	✗	I	A	B	N	B

Table 2. Summary of characteristics of KEEL, KNIME, and WEKA software tools: learning variety.

Software	Learning Variety							
	Classif.	Regression	Clustering	Association Rules	Subgroup Discovery	Imbalanced Classification	SSL	MIL
KEEL	A	A	A	A	A	A	A	I
KNIME	A	A	A	A	N	N	N	N
WEKA	A	A	A	A	N	N	N	I

Table 3. Summary of characteristics of KEEL, KNIME, and WEKA software tools: advanced features

Software	Advanced Features					
	Post-processing	Meta - Learning	Statistical tests	EAs	Fuzzy Learning	Multi-Classifiers
KEEL	B	I	A	A	A	A
KNIME	N	N	I	B	B	N
WEKA	N	I	B	B	B	N

- *Fuzzy Learning Schemes* refer to the case that Fuzzy Rule Based Systems are included for the learning stage. We consider a basic support of if when at least one algorithm of this paradigm is included, whereas an advanced level is given to the tool with the most recent approaches.
- *Multi-classifiers* stands for the binarisation schemes, i.e. to carry out the learning of multiple class data set in a divide and conquer approach. This includes both One-vs-One and One-vs-All methodologies.

Tables 1, 2, 3 collect the main characteristics of these DM software tools. From these tables, it is shown that the main differences between KEEL and the remaining software tools are related to the Run-Mode, preprocessing variety, and the novel learning schemes, i.e. mainly due to the modules for semi-supervised learning, and imbalanced classification. Additionally, the support for non-parametrical statistical tests, evolutionary algorithms, fuzzy learning, and multi-classifiers also allow us to establish a clear distinction of KEEL against WEKA and KNIME.

4. Availability and Community

As we stated before, KEEL is released under the terms of the GNU Public License GPLv3 as published by the Free Software Foundation. It requires no additional libraries and it does not need to be configured or installed. It can be used under Windows, Linux and MacOS X with Java version 7 or later.

To ease the use of KEEL, and obtain feedback and input from the DM community, we have adopted a number of mechanisms to make KEEL more broadly accessible.

- A new Getting Started section has been added to the reference Manual to help new users to quickly use the KEEL software. Please check the web Manuel at `http://www.keel.es/development.php`.
- We have added a new integration guide to allow researchers to add their own techniques to the KEEL platform. Please refer to the technical report [38] for more details.
- A public API documentation is essential to encourage developers to extend and modify our

methods. Therefore, the KEEL API documentation is now accessible at `http://www.keel.es/javadoc/index.html`

- **Google forum** is available at: `https://groups.google.com/forum/#!forum/keel` to encourage discussion about problems or issues found in KEEL.
- An **MLOSS project** page can be found at `http://mloss.org/software/users/keel/`.
- **GitHub**: The source code of KEEL 3.0 is now available on GitHub `https://github.com/SCI2SUGR/KEEL` (Figure 8). Here, users can report bugs, keep track of changes, and commit their own developments to the platform.

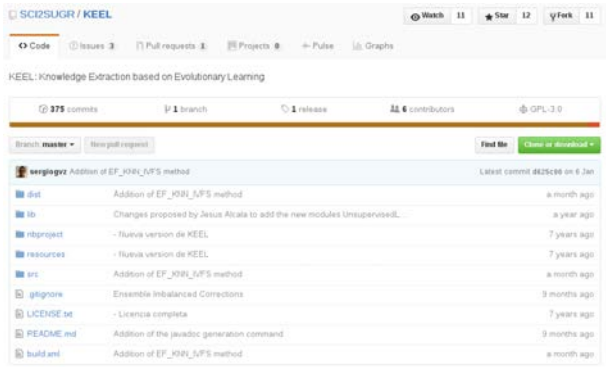

Fig. 8. KEEL 3.0 is on Github.

5. Conclusions

In this work, we have presented the main features of the new release (3.0) of the KEEL software. New advanced modules for novel DM problems such as SSL or MIL are now available, allowing practitioners and beginners to deal with a wider variety of scenarios. These new modules are continuously being updated and improved, including new algorithms.

KEEL has become a very rich DM tool kit that contains state-of-the-art algorithm in preprocessing and learning models as well as a number of add-ons for statistical evaluation, data management and so on. In addition, the way in which the data mining community can contribute to the project has been enhanced, including new guidelines for integration of algorithms, a new API, and a public version control repository.

Acknowledgements

We would like to acknowledge support for these projects from the Spanish Ministry of Education and Science (Grants TIN2014-57251-P, TIN2015-68454-R, TIN2014-56967-R). J.M. Moyano holds a FPU Grant FPU15/02948 from the Spanish Ministry of Education.

References

1. KNIME: The konstanz information miner, 2016. https://www.knime.org.
2. J. Alcalá-Fdez, R. Alcalá, and F. Herrera. A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. *IEEE Transactions on Fuzzy Systems*, 19(5):857–872, 2011.
3. J. Alcalá-Fdez and J. M. Alonso. A survey of fuzzy systems software: Taxonomy, current research trends, and prospects. *IEEE Transactions on Fuzzy Systems*, 24(1):40–56, 2016.
4. J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2011.
5. J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, and F. Herrera. KEEL: A software tool to assess evolutionary algorithms to data mining problems. *Soft Computing*, 13(3):307–318, 2009.
6. J. Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105, 2013.
7. Revolution Analytics and Steve Weston. *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*, 2015. R package version 1.0.10.
8. Revolution Analytics and Steve Weston. *foreach: Provides Foreach Looping Construct for R*, 2015. R package version 1.4.3.
9. M. Atzmueller. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.
10. P. Branco, L. Torgo, and R. P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.*, 49(2):31:1–31:50, August 2016.

11. C. J. Carmona, P. González, M. J. del Jesus, and F. Herrera. Overview on evolutionary subgroup discovery: analysis of the suitability and potential of the search performed by evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(2):87–103, 2014.
12. Winston Chang. *R6: Classes with Reference Semantics*, 2015. R package version 2.1.1.
13. O. Chapelle, B. Schlkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010.
14. F. Coelho, A.P. Braga, and M. Verleysen. A mutual information estimator for continuous and discrete variables applied to feature selection and classification problems. *International Journal of Computational Intelligence Systems*, 9 - 4:726 – 733, 2016.
15. J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
16. J. Derrac, S. García, and F. Herrera. Fuzzy nearest neighbor algorithms: Taxonomy, experimental analysis and prospects. *Information Sciences*, 260:98 – 119, 2014.
17. A. Fernández, V. López, M.J. del Jesus, and F. Herrera. Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges. *Knowledge-Based Systems*, 80:109 – 121, 2015.
18. Eibe Frank, Mark A. Hall, and Ian H. Witten. *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*. Morgan Kaufmann, Fourth Edition, 2016.
19. B. Frénay and M. Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014.
20. M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(4):463–484, 2012.
21. S. García and F. Herrera. An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.
22. S. García, J. Luengo, and F. Herrera. *Data Preprocessing in Data Mining*, volume 72 of *Intelligent Systems Reference Library*. Springer, 2015.
23. S. García, J. Luengo, and F. Herrera. Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. *Knowledge-Based Systems*, 98:1 – 29, 2016.
24. S. García, J. Luengo, J. A. Sáez, V. López, and F. Herrera. A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning. *Knowledge and Data Engineering, IEEE Transactions on*, 25(4):734–750, 2013.
25. J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 3rd edition, 2011.
26. F. Herrera, C.J. Carmona, P. González, and M.J. del Jesus. An overview on subgroup discovery: foundations and applications. *Knowledge and Information Systems*, 29(3):495–525, 2011.
27. F. Herrera, S. Ventura, R. Bello, C. Cornelis, A. Zafra, D. Sanchez-Tarrago, and S. Vluymans. *Multiple Instance Learning. Foundations and Algorithms*. Springer, 2016.
28. G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning*. Springer, 2013.
29. B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.
30. Duncan Temple Lang and the CRAN Team. *XML: Tools for Parsing and Generating XML Within R and S-Plus*, 2015. R package version 3.98-1.3.
31. V. López, A. Fernández, S. García, V. Palade, and F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141, 2013.
32. J. M. Moyano and L. Sanchez. RKEEL: using KEEL in R code. *2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016*, pages 257–264, July 2016.
33. J. M. Moyano and L. Sanchez. *RKEEL: Using Keel in R Code*, 2017. R package version 1.1.21.
34. S. Ramírez-Gallego, S. García, H. Mourio-Talín, D. Martínez-Rego, V. Bolón-Canedo, A. Alonso-Betanzos, J.M. Benítez, and F. Herrera. Data discretization: taxonomy and big data challenge. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(1):5–21, 2016.
35. S. Sonnenburg, M.L. Braun, Ch.S. Ong, S. Bengio, L. Bottou, G. Holmes, Y. LeCun, K.-R. Müller, F. Pereira, C.E. Rasmussen, G. Rätsch, B. Schölkopf, A. Smola, P. Vincent, J. Weston, and R. Williamson. The need for open source software in machine learning. *Journal of Machine Learning Research*, 8:2443–2466, 2007.
36. I. Triguero, S. García, and F. Herrera. SEG-SSC: A framework based on synthetic examples generation for self-labeled semi-supervised classification. *IEEE Transactions on Cybernetics*, 45(4):622–634, April 2015.
37. I. Triguero, S. García, and F. Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, 2(2):245–284, 2015.
38. I. Triguero, S González, J.M Moyano, S. García, J. Alcalá-Fdez, J. Luengo, A. Fernández, M.J. del Je-

sus, L. Sánchez, and F. Herrera. Keel data-mining software suite: Integration of new algorithms, 2017. http://www.keel.es/documents/keel_extension.pdf.
39. Simon Urbanek. *rJava: Low-Level R to Java Interface*, 2015. R package version 0.9-7.
40. G. R. Warnes, B. Bolker, G. Gorjanc, G. Grothendieck, A. Korosec, T. Lumley, D. MacQueen, A. Magnusson, J. Rogers, and others. *gdata: Various R Programming Tools for Data Manipulation*, 2015. R package version 2.17.0.
41. I. Yoo, P. Alafaireet, M. Marinov, K. Pena-Hernandez, R. Gopidi, J.-F. Chang, and L. Hua. Data mining in healthcare and biomedicine: A survey of the literature. *Journal of Medical Systems*, 36(4):2431–2448, 2012.
42. X. Zhu, A.B. Goldberg, R. Brachman, and T. Dietterich. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.