SDC 8. Data pre-processing.

To optimise the performance of the different learning algorithms used in the data processing stage, standard pre-processing methods such as data cleaning and data discretization were applied.

Firstly, those players who did not complete all the neuromuscular tests for any reason (six players) were removed. This exclusion criterion was based on the fact that if a player had not completed a neuromuscular test a large number of features would be absent and this might have a negative impact on the performance of the models generated. Furthermore, four players were also removed because they left their respective teams before the follow up procedure was completed. Secondly, an investigation regarding the presence of outliers was carried out using boxplots and the detected outliers were removed. The third step consisted of looking for missing data. To address this issue, frequency tables and diagrams were built. Thus, missing data were replaced by the mean value of the corresponding feature of the specific level of play (1^{st} or 2^{nd} B divisions) of the players. For example, if a 1^{st} division player did not report his height for any reason, then the average value of his counterpart 1^{st} division players was inputted. It should be noted that none of the features reported a percentage of missing data and / or outliers higher than 5%. The SPSS Statistical software (V21.0) was used to carry out these data cleaning processes.

After having applied the above-mentioned data cleaning methods, an imbalance (showing an imbalance ratio of 0.26) and high dimensional data set comprised of 86 soccer players (instances) and 229 potential risk factors (features) was created.

The final step comprised the discretization of the continuous features as this has been shown to be an effective measure to improve the performance of several classifiers [4]. Thus, continuous features were discretized applying the unsupervised discretization

algorithm available in the well-known Weka (Waikato Environment for Knowledge Analysis) Data Mining software and using the equal frequency binning approach (three intervals). We selected three intervals in order to reflect taxonomy of low, moderate and high scores that might make the final models more comprehensible. In those features that the graphical representation of the data allow the authors to suggest alternative cut-off values, a comparative analysis was run in order to identify the discretization approaches (algorithm vs. authors visual inspection) that displayed the best predictive ability. The approach reporting the better predictive results was used for the discretization of each feature. Consequently, lower extremity ROM and isokinetic angle of peak torque (APT) features as well as both the reciprocal knee flexion to knee extension ratios and bilateral knee flexion and extension ratios were discretized using the graphical representation of the data as a guide; whereas the remaining features were discretized using the Weka unsupervised discretization algorithm (Supplementary files 1-7).

Data processing

Part of the taxonomies for external (oversampling) and internal (ensembles) methods for learning with imbalanced data sets proposed by Elkarami et al. [5] and Galar et al. [7] were used to build models for predicting HSI in professional soccer players. Thereby, the algorithms of each of the above mentioned families (oversampling and ensembles) that showed the best goodness scores in the latter mentioned studies were used to train models. The model with the highest validity metrics was considered the best for predicting HSI based on the current data set.

To achieve founded conclusions, three decision tree algorithms were selected to be used in the oversampling and ensemble methodologies as base classifiers: J48, which is an algorithm for generating a pruned or unpruned C4.5 decision tree [8]; ADTree, which is

an alternating decision tree [6]; and SimpleCart, which implements minimal cost-complexity pruning. Hence a decision tree is a set of conditions organized in a hierarchical structure [1]. An instance is classified by following the path of satisfied conditions from the root of the tree until a leaf is reached, which will correspond with a class label.

All the decision trees selected were made cost sensitive to minimize the cost of misclassification of the minority class by using the filter cost sensitive classifier algorithm available in Weka workbench. Thus, the training data were reweighted according to the costs assigned to each class. The set up of the definitive cox matrix was based on the best performance reported after testing all the possibilities. For the sake of brevity and the lack of space, the codes of the algorithms used in this study are not presented. Instead, only the names of the algorithms have been specified and the reader is referred to the original sources. Furthermore, all the classification algorithms used are available in the Weka Data Mining software.

Although there are several data oversampling methods, we used one of the most popular methodologies that is the classic synthetic minority oversampling technique (SMOTE) [2]. The main concept behind SMOTE is to create new minority class examples by interpolating several minority class instances that lie together for oversampling the training set. With this technique, the positive class is oversampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbours. Three different levels of balance in the training data were analysed (25:75; 40:60; 50:50) and the best in term of predictive ability was reported. Additionally, the interpolations that are computed to generate new synthetic data were made considering the 5-nearest neighbours of minority class instances using the Euclidean distance.

Regarding ensemble learning algorithms, the algorithm families designed to deal with skewed class distributions in data sets were included: Boosting-based and Bagging-based. The Boosting-based ensembles that were considered in the current study were SMOTEBoostM1 [3] and RUSBoost [9]. With respect to Bagging-based ensembles, it was included from the OverBagging group, OverBagging (which uses random oversampling) and SMOTEBagging [10].

Finally, the behaviour of some specific combination of class-balanced ensembles with cost-sensitive base classifiers was also studied. The final cox matrix set up was based on the best performance reported after testing all the possibilities.

The following table summarizes the list of algorithms grouped by families and also shows the abbreviations that have been used along the experimental framework and a short description of them.

Algorithms used in the data processing phase

Cost-Sensitive base classifiers		
Abbr.	*Method*	*Short Description*
J48	J48	Algorithm for generating a pruned or unpruned C4.5 decision tree
SCart	SimpleCart	Algorithm for implementing minimal cost-complexity pruning
ADTree	ADTree	Alternating decision tree
Resampling techniques		
Abbr.	*Method*	*Short Description*
CS-SMT	SMOTE	Each cost-sensitive decision tree applied on data set previously pre-processed with Smote

Boosting-based Ensembles **with a cost-sensitive base classifier**		
Abbr.	*Method*	*Short Description*
CS-SBOM1	SmoteBoost	AdaBoost.M1 with Smote in each iteration and with an asymmetric classification cost matrix in the base classifier
CS-RUS	RusBoost	AdaBoost.M2 with random undersampling in each iteration and with an asymmetric classification cost matrix in the base classifier
Bagging-based Ensembles **with a cost-sensitive base classifier**		
Abbr.	*Method*	*Short Description*
CS-OBAG	OverBagging	Bagging with oversampling of the minority class and with an asymmetric classification cost matrix in the base classifier
CS-SBAG	SmoteBagging	Bagging where each bag´s Smote quantity varies and with an asymmetric classification cost matrix in the base classifier

References

1. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and regression trees*. Wadsworth & Brooks. Monterey, CA 1984
2. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. *J Artificial Intelligence Res* 2002;16:321-357
3. Chawla N, Lazarevic A, Hall L, Bowyer K. SMOTEBoost: Improving prediction of the minority class in boosting. *Paper presented at the European Conference on Principles of Data Mining and Knowledge Discovery* 2003:107-119
4. Ekbal A. Improvement of prediction accuracy using discretization and voting classifier. *In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on IEEE* 2006;2:695-698
5. Elkarami B, Alkhateeb A, Rueda L. Cost-sensitive classification on class-balanced ensembles for imbalanced non-coding RNA data. In: *Proceedings of the Student Conference (ISC), 2016 IEEE EMBS International* 2016:1-4
6. Freund Y, Mason L. The alternating decision tree learning algorithm. In: *Proceedings of the icml* 1999;99:124-133
7. Galar M, Fernandez A, Barrenechea E, Bustince H, & Herrera F. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* (Applications and Reviews) 2012;42:463-484
8. Quinlan JR. Learning decision tree classifiers. *ACM Computing Surveys* (CSUR) 1996;28:71-72
9. Seiffert C, Khoshgoftaar TM, Van Hulse J, Napolitano A. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 2010;40:185-197

10. Wang S, Yao X. Diversity analysis on imbalanced data sets by using ensemble models. *Paper presented at the Computational Intelligence and Data Mining 2009. CIDM'09* IEEE Symposium on; 2009:324-331