

# Explainable Artificial Intelligence for Kids

**Jose M. Alonso**

Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS)  
 Universidade de Santiago de Compostela, Santiago de Compostela, Spain  
 josemaria.alonso.moral@usc.es

## Abstract

Artificial Intelligence (AI) is part of our everyday life and has become one of the most outstanding and strategic technologies of the 21st century. Explainable AI (XAI in short) is expected to endow AI systems with explanation ability when interacting with humans. This paper describes how to provide kids with natural explanations, i.e., explanations verbalized in Natural Language, in the context of identifying/recognizing roles of basketball players. Semantic grounding is achieved through fuzzy concepts such as tall or short. Selected players are automatically classified by an ensemble of three different decision trees and one fuzzy rule-based classifier. All the single classifiers were first trained with the open source Weka software and then natural explanations were generated by the open source web service Expli-Clas. The Human-Computer Interaction interface is implemented in Scratch, that is a visual programming language adapted to kids. The developed Scratch program is used for dissemination purposes when high-school teenagers visit the Research Center in Intelligent Technologies of the University of Santiago de Compostela.

**Keywords:** Explainable AI, Natural Language Generation, Human-Computer Interaction, Decision Trees, Fuzzy rule-based Classifiers.

## 1 Introduction

The European Commission emphasizes the importance of Explainable Artificial Intelligence (XAI in short) in order to develop an AI coherent with European values: *“to further strengthen trust, people also*

*need to understand how the technology works, hence the importance of research into the explainability of AI systems”* [10]. This is in accordance with the European General Data Protection Regulation [15], approved in 2016 and effective since May 2018, which remarks that European citizens have the “right to explanation” of decisions affecting them, no matter who (or what AI system) makes such decision.

In consequence, XAI is attracting attention of researchers in many varied areas such as Ethics [9], Law, Psychology, Engineering or Computer Science [14]. Moreover, XAI is in the core of human-centric computing applications, e.g., decision-support and recommender systems for e-Health or e-Learning [20]. One of the main challenges of XAI is how to build conversational agents able to provide humans with semantic grounding, persuasive and trustworthy interactive explanations [1, 6]. Explanations are expected to be presented as a narrative/story in Natural Language (NL) because this aids human comprehension. Moreover, they should be communicated to users through multi-modal (graphical + textual) interactive interfaces.

Prof. Zadeh made many highly valuable contributions to the Fuzzy Logic field and beyond, e.g., the definition of fuzzy sets [22], the concept of linguistic variables and their application to approximate reasoning [23], the paradigm of computing with words [24], or the computational theory of perceptions [25]. Many of these contributions were pioneer ideas and/or challenging proposals with a lot of potential to be fully developed later by other researchers [2]. Nowadays, XAI is a prominent and fruitful research field where many of Zadeh’s contributions can become crucial if they are carefully considered and thoroughly developed. It is worth noting that, as described in the bibliometric study made in [5], about 30% of publications in Scopus related to XAI, dated back to 2017 or earlier, came from authors well recognized in the Fuzzy Logic field. This is mainly due to the commitment of the fuzzy community to produce interpretable

fuzzy systems, since interpretability is deeply rooted in the fundamentals of Fuzzy Logic [4].

This paper presents the xai4kids dissemination initiative which is developed at the Research Center in Intelligent Technologies (CiTIUS) of the University of Santiago de Compostela. CiTIUS organizes periodically (about once per month) thematic workshops with the twofold goal of (1) making Science closer to society and (2) motivating kids to opt for STEM<sup>1</sup> careers. The xai4kids initiative is aimed for groups of about 20 high-school teenagers (in the range from 6 to 17 years old) and it takes about 30 minutes. Each session starts with a brief but motivating introduction to AI and its applications. Then, students play with the application described in this paper. Then, they are asked about the rationale they think is behind the provided explanations. Then, they are taught how the explanations are actually generated. The session ends with a brief brainstorming about how to enhance the application.

It is worth noting this initiative is aligned with the vision of the European Higher Education Area (EHEA)<sup>2</sup> which has the origin in the so-called Bologna Process (started in 1999). The EHEA comprises 48 countries which agree to and adopt reforms on higher education with the aim of increasing staff and students' mobility as well as facilitating employability in Europe. In the age of big data and AI, our society demands more and more employees with technical skills. Thus, there is a need for promoting STEM careers since the earliest.

The rest of the manuscript is organized as follows. Section 2 introduces two related software: (1) the Scratch coding environment which is used by kids worldwide and (2) the ExpliClas research tool which we have recently developed to generate XAI systems ready to provide users with explanations in NL automatically extracted from data. Section 3 presents the architecture of the XAI system we have developed with the assistance of both Scratch and ExpliClas. Section 4 goes in depth with an illustrative use case where this XAI system explains in NL how to recognize the role of basketball players in terms of their statistics. Finally, Section 5 concludes the paper.

## 2 Material and Methods

### 2.1 Scratch

Scratch<sup>3</sup> is a project of the Lifelong Kindergarten Group at the Massachusetts Institute of Technology (MIT) Media Lab. It is available in more than 40 lan-

<sup>1</sup>STEM stands for Science, Technology, Engineering and Mathematics.

<sup>2</sup><http://www.ehea.info/>

<sup>3</sup><https://scratch.mit.edu/>

guages and used worldwide in schools of more than 150 different countries. Kids can learn to code in Scratch and share their creations with others in an online community. As a side effect, they learn strategies for solving problems, designing projects, and communicating ideas. Accordingly, there are students who learn with Scratch from elementary school to college (Scratch is designed especially for ages 8 to 16) and across heterogeneous disciplines (not only STEM but also linguistics, art, social studies, and so on). ScratchED<sup>4</sup> gathers an online community of educators who share their initiatives.

The Scratch editor can be used online and offline. It has a user-friendly interface (see Fig. 1) that makes coding a very intuitive task. Several predefined "blocks" are available to code procedures (related to movements, events, control actions, etc.) by drag and drop. Additional blocks can be defined by the user what makes Scratch a very simple but powerful programming tool. There are also extensions to connect Scratch programs to specific hardware and/or software.



Figure 1: Scratch editor.

### 2.2 ExpliClas

ExpliClas<sup>5</sup> is a web service coded in Java. Namely, it includes a REST API developed with the Jersey framework<sup>6</sup> and a web client developed with ReactJS<sup>7</sup>.

<sup>4</sup><http://scratched.gse.harvard.edu/>

<sup>5</sup><https://demos.citius.usc.es/ExpliClas/>

<sup>6</sup>RESTful Web Services in Java: <https://jersey.github.io>

<sup>7</sup>A JavaScript library for building user interfaces: <https://reactjs.org>

ExpliClas provides users with multimodal (textual + graphical) explanations related to Weka<sup>8</sup> classifiers [21]. Notice that Weka is a well-known open source Data Mining project, led by researchers affiliated to the University of Waikato (New Zeland), and with a huge community of users and developers worldwide. Several benchmark datasets (e.g., iris, wine, etc.) are pre-loaded (see Fig. 2) while other datasets can be uploaded by the user. Four Weka classifiers (J48, RepTree, RandomTree and FURIA) are currently available. Two kinds of explanation are provided: (1) Global and Local Explanation of the selected classifier (which is trained by Weka with 10-fold cross-validation); and (2) Explanation of the confusion matrix (regarding both training and test data).

Explanations are multimodal because they comprise graphical visualization along with sentences in NL. Branches of decision trees (J48, RepTree and RandomTree) are first translated into crisp rules from the root to the leaves. Then, decisions are verbalized and justified in terms of the fired rules (no matter the nature of the selected classifier). In the case of FURIA, several rules can be fired with different activation degrees what makes easier handling naturally imprecision and uncertainty. In the case of crisp decision trees, ExpliClas analyzes potential alternative branches when data values are close to the split values in the trees. In order to guarantee semantic grounding, global semantics is enforced (no matter if the selected algorithm is either crisp or fuzzy) by means of defining beforehand strong fuzzy partitions with three linguistic terms (e.g., small, medium, large) for each decision variable. Then, either split values in decisions trees or fuzzy intervals in FURIA are interpreted in terms of the closer linguistic terms previously defined. Those terms are used to verbalize decisions in the form of textual explanations. The interested reader is kindly referred to [3] for further details about ExpliClas.

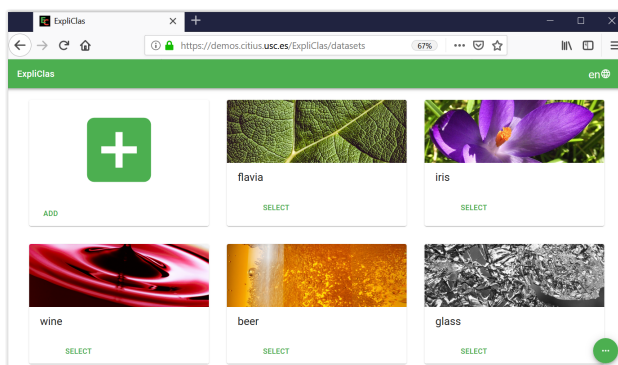


Figure 2: ExpliClas Pre-loaded Datasets.

<sup>8</sup>The Waikato Environment for Knowledge Analysis: <https://www.cs.waikato.ac.nz/ml/weka/>

## 3 Architecture of the xai4kids System

### 3.1 The grey-box classifier ensemble layer

We have built a classifier ensemble which is made up of the four Weka classifiers already integrated into ExpliClas. All single classifiers can be deemed as grey-box classifiers. The first three classifiers are different implementations of decision trees (belonging to the `weka.classifiers.tree` package) while the fourth one is a fuzzy classifier (belonging to the `weka.classifiers.rules` package):

- **J48** is the Weka class for generating C4.5 decision trees [17].
- **RepTree** is a fast implementation of C4.5 trees using information gain along with backfitting reduced-error pruning.
- **RandomTree** builds C4.5 trees that consider  $K$  randomly chosen attributes at each node.
- **FURIA** is the Weka class that implements the Fuzzy Unordered Rule Induction Algorithm [13].

The ensemble output class is computed by majority voting. In case of a tie, the output pointed out by the pair of classifiers with the highest joint accuracy is selected. In case of several winner classifiers, we select the shortest explanation among all provided. Notice that the linguistic layer described below is in charge of generating one NL explanation associated to the output given by each single classifier. Of course, all these technical details are transparent to the kids who interact with the xai4kids systems through the interactive interface described in Section 3.3.

### 3.2 The linguistic layer

A recent review of the Natural Language Generation (NLG) state of the art [11] shows NLG as a well-known area within the Natural Language Processing (NLP), Computational Linguistics and Artificial Intelligence fields. In addition, the NLG pipeline (Macro Planner + Micro Planner + Surface Realizer) proposed by Reiter and Dale [19] is cited as the most popular one. That is the reason why ExpliClas implements this pipeline with a multilingual linguistic realization (English<sup>9</sup>, Spanish<sup>10</sup> and Galician<sup>11</sup>) which is a mix of templates and NL text dynamically generated.

<sup>9</sup>SimpleNLG [12] github: <https://github.com/simplenlg/simplenlg>

<sup>10</sup>SimpleNLG-ES [18] github: <https://github.com/citiususc/SimpleNLG-ES>

<sup>11</sup>SimpleNLG-GL [7] github: <https://github.com/citiususc/SimpleNLG-GL>

Accordingly, the NL explanations in the xai4kids system are first automatically generated by ExpliClas (where the NLG pipeline is coded in Java) and then ported to the Scratch programming language to be communicated to kids via the interactive interface described in the next section. It is worth noting that ExpliClas is run by an expert who is also in charge of porting the generated explanations to Scratch. Let us remark again that all the technical details are transparent for kids who only have to concentrate in playing with the interactive interface.

### 3.3 The interactive interface layer

We have implemented a simple multi-modal dialogue system for kids. Neither NLP nor NLG techniques are applied in Scratch. Explanations are generated beforehand by ExpliClas. In addition, communication is always started and guided by the xai4kids system. Hence, kids only have to type quick answers (e.g., y=yes or n=no) or select items in the screen. Additionally, they are asked to rate the goodness of the provided explanations (in a 5-points Likert scale) and to write brief alternative explanations in free text format (to be processed later for research purpose) only if assessments are poor (smaller than or equal to 3).

The xai4kids system runs in a laptop that is connected to a TV screen in order to make easier the interaction with groups of kids. It is worth noting the textual messages printed in the screen can also be communicated as voice messages that must be recorded beforehand and played through the TV speakers. However, neither voice recognition nor voice generation techniques are applied yet. Anyway, the combination of textual messages, sound and images through a TV is likely to make the human-computer interaction more natural and appealing for kids. In the case of handicapped or very young kids, the keyboard of the laptop is directly handled by the researcher who leads the interactive session.

## 4 Illustrative Use Case

This section illustrates the use of the xai4kids system in the context of an interactive game for identifying/recognizing roles of basketball players.

First of all, we have created the basketball-players classification dataset<sup>12</sup>. It is made up of 80 samples corresponding to four classes (Point Guard, Shooting Guard, Small Forward, Center) which are linked to 13 attributes (Height, Blocks, Rebounds, Assists, Points, Personal Fouls Made, Personal Fouls Received, Free

<sup>12</sup><https://gitlab.citius.usc.es/jose.alonso/basketballplayers-dataset>

Throws Percentage, 2-points Field Goals Percentage, 3-points Field Goals Percentage, Turnovers, Steals, and Global Assessment). The dataset is perfectly balanced with 20 samples belonging to each class. Numerical values associated to each sample correspond to statistics available online at the website of the Spanish Basketball League ACB<sup>13</sup>. For each player, we have taken statistics related to season 2017-2018.

Then, we have uploaded this dataset to ExpliClas and generated the four grey-box classifiers that are available. Rows 1-4 in Table 1 summarize the goodness of these classifiers. Accuracy computes the ratio of correctly classified samples. Interpretability of trees is measured in terms of their number of leaves and tree size. The number of FURIA rules is also reported in the last column of the table. The fifth row reports the quality indexes for the ensemble classifier in the core of xai4kids. In addition, the last three rows in the table are just for comparison purposes. They report quality indexes for three black-box algorithms provided by Weka (notice that we kept the default parameters suggested by Weka when learning the related classifiers):

- **SMO**<sup>14</sup> implements the John Platt's sequential minimal optimization algorithm for training a support vector classifier [16].
- **MultilayerPerceptron**<sup>15</sup> uses backpropagation in a neural network with sigmoid nodes.
- **RandomForest**<sup>16</sup> builds an ensemble of RandomTrees.

Algorithm	Accuracy (%)	Leaves	Size	Rules
J48	57.5	12	23	-
RepTree	72.5	5	9	-
RandomTree	70	21	41	-
FURIA	70	-	-	8
xai4kids	75	-	-	-
SMO	68.75	-	-	-
MultilayerPerceptron	77.5	-	-	-
RandomForest	78.75	-	-	-

Table 1: Quality indexes of Weka classifiers for the basketball-players dataset (10-fold CV).

The ensemble classifier xai4kids is more accurate than any of the single grey-box classifiers but not so accurate as the most accurate black-box classifier that is RandomForest. Fortunately, the lack of accuracy is compensated with the explanation ability of our system.

Figure 3 shows the decision tree generated by Rep-Tree. It is the most accurate but also the simplest

<sup>13</sup><http://www.acb.com/>

<sup>14</sup>[weka.classifiers.functions.SMO](http://weka.classifiers.functions.SMO)

<sup>15</sup>[weka.classifiers.functions.MultilayerPerceptron](http://weka.classifiers.functions.MultilayerPerceptron)

<sup>16</sup>[weka.classifiers.trees.RandomForest](http://weka.classifiers.trees.RandomForest)

classifier among the four ones generated by ExpliClas. It is worth noting that ExpliClas provides the user with alternative solutions for a given sample and the fired branch is highlighted in either green (right output) or red (wrong output). At the bottom of the figure there is a button (“Show Explanation”). When this button is clicked a panel with textual explanation is displayed. For example, Figure 4 shows the confusion matrix coming out of 10-fold cross-validation along with the explanation panel. In this case, we selected “Global Explanation” which summarizes the classifier behavior, including the list of classes, a comment related to accuracy and one or more comments related to the confusion among classes. Additional details are provided when selecting “Local Explanation” or “Matrix Explanation”.

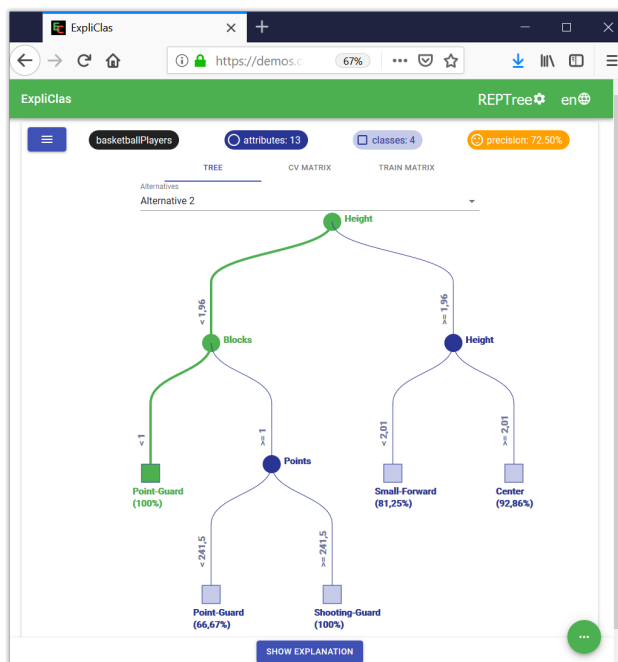


Figure 3: RepTree visualization in ExpliClas.

Figure 5 shows the local explanation associated to FURIA for the same sample that activated the tree in Figure 3. In addition to the textual explanation that is verbalizing the information contained in the winner rule, a bar graph shows the activation degree associated to each class. It is also possible to visualize the membership functions related to each rule premise.

Once we had built all single classifiers and the related local explanations with ExpliClas, then we ported all this information to Scratch. Figure 6 shows how the xai4kids project looks like in the Scratch offline editor. We can distinguish three main panels:

- **The execution panel.** On top of the left part of the editor, the execution panel displays the run-

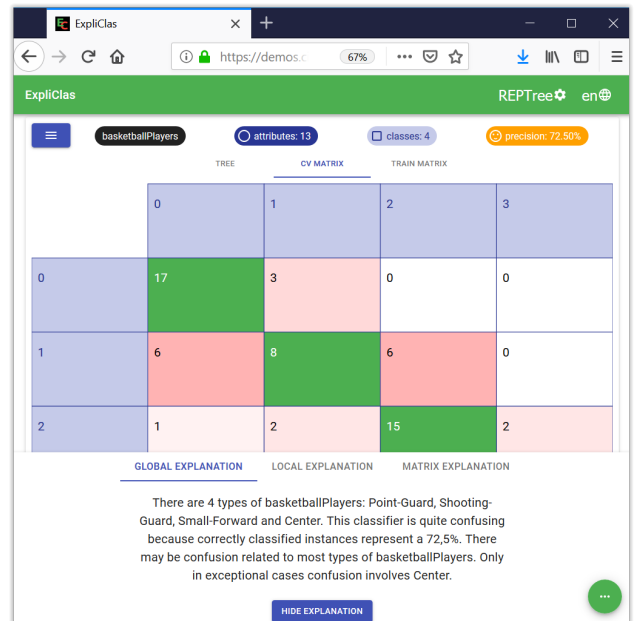


Figure 4: RepTree global explanation and confusion matrix in ExpliClas.

ning scene (this is the welcome scene in the case of Figure 6) of the xai4kids program. Scratch programs are not compiled. Instructions are just interpreted and executed on the fly. The execution begins when clicking the green flag that is on top of this panel. It can be aborted at any time just by clicking the red button.

- **The sprite panel.** It is just below the execution panel. The characters playing a role in a Scratch program are called sprites. In xai4kids, there is a main sprite that is Giga who leads the conversation with the user. In addition, it is in charge of making classifications and providing explanations to the user. The panel also includes a sprite for each basketball player to consider. When selecting one sprite, then the related information is displayed on the right hand side of the screen.
- **The programming panel.** It includes all tools needed to visually (just with drag and drop actions) code a Scratch program. It includes three different views that are changed by selecting the right tag (Scripts, Costumes, Sound) on top. Scripts are created as a mix of blocks which represent different kinds of programming instructions (e.g., if-then, control loops, etc.). The user can build up new blocks as a combination of previous ones and use them in a similar way to functions in other programming languages. It is also possible to define Scratch extensions that are aimed at creating new blocks beyond the available ones. It is

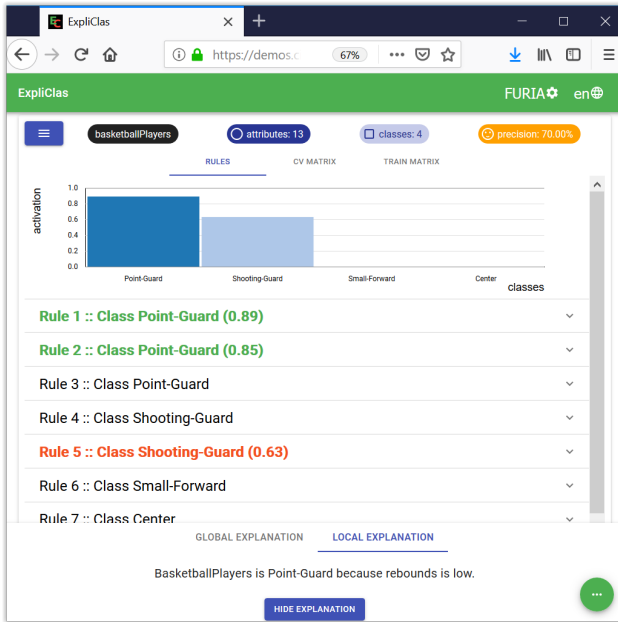


Figure 5: FURIA visualization and local explanation in ExpliClas.



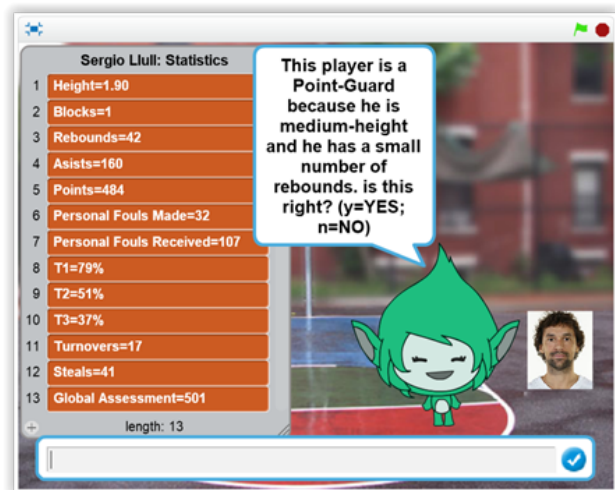
Figure 6: xai4kids project in Scratch.

worth noting that the current version of xai4kids does not use any extension. Everything (even the classifiers and the explanations) is programmed just by making use of the blocks pre-defined in the Scratch editor.

Figure 7(b) shows an example of execution of the xai4kids Scratch script. Once the user selects one basketball player among all available (see Figure 7(a)), then a panel with the summary of statistics is displayed on the left of the screen while the picture of the player goes to the right. Notice that the pictures of the other players are hidden. The result of classification along with the related explanation are shown in the center of the screen. In this case, we have selected the player related to the data sample that was



(a) xai4kids screenshot with the available players.



(b) Example of XAI classification.

Figure 7: Running the xai4kids Scratch script.

classified by the tree in Figure 3 and the rules in Figure 5. It is a Point-Guard. There is agreement for Point-Guard between J48, RandomTree and FURIA. However, RepTree is for Shooting-Guard as first alternative and for Point-Guard as second alternative. Accordingly, the xai4kids program discards RepTree in this case. Thus, there are three plausible explanations: “The player is Point-Guard because...”

- “... he is medium-height and he has a small number of rebounds.” (J48)
- “... he has a small number of rebounds, he has a high percentage of free throws, he has a medium percentage of 2-point field goals, and he scores a large number of assists.” (RandomTree)
- “... he is medium-height and he has a small number of rebounds.” (FURIA)

Then, xai4kids provides the user with the shorter explanation among the plausible ones. Notice that by chance J48 and FURIA provides the same explanation for this example but all explanations may be different for other cases. Having four classifiers ready to provide the user with plausible explanations endows xai4kids with plenty of diversity and naturalness. Actually, in case the same player is selected more than once by the same user then the system is likely to choose another explanation apart from the shorter one (i.e., with a probability initially set to 90%). The reason to do this is to introduce variability in language and as a result to avoid repetitive robot-like unnatural dialogues.

Then, the user is asked to assess the goodness of the provided explanation in a 5-point Likert scale. If assessment is smaller or equal than 3, then an alternative explanation is required. Finally, the results of each session can be stored in a log plain text file for further analysis and future enhancements of the program.

## 5 Conclusions and Future Work

This paper has presented the xai4kids initiative. A couple of demo videos are available at:

<https://youtu.be/7jfYYudPUZY>

<https://youtu.be/tKgW2hoNC9Y>

It is worth noting that the Scratch studio project, including the source code of the xai4kids system described in the previous section is shared to the Scratch community at:

<https://scratch.mit.edu/projects/303945261/>

As our illustrative example deals only with male basketball players, it may be deemed as biased and too boy-oriented. We would like to highlight this was done by purpose in order to show the effects of bias to kids. Actually, we are surrounded by many examples of gender bias that affects our everyday life. As described in [8], everything is designed for a stereotype of men who has a height of 1.76m and a weight of 77kg. In our example, obviously, if we had asked our XAI system to classify a basketball woman with a height of 1.91m the system is likely to say she is a Small Forward even if she is really a Center. This is due to the fact that men are in average taller than women. To fix this issue it is needed to re-design the XAI system from scratch with gender as one of the key features to consider. Moreover, we should add to the dataset new examples of female basketball players related to all the four classes under consideration.

As future work, we plan to enhance the interactive interface with a more natural and inclusive dialogue system, which is free of any bias. Firstly, we will build

a new dataset including as many examples of women as men. Then, we will re-build the XAI system. In addition, we will enhance current explanations to make them more narrative in accordance with the feedback provided by kids in a series of sessions to be carried out during next months. Secondly, we will adopt a *fabula* model as well as argumentation schemes in order to make explanations even more natural and persuasive. Finally, explanations will be enriched with causal relations and counterfactual facts.

## Acknowledgement

Jose M. Alonso is *Ramón y Cajal* Researcher (RYC-2016-19802). In addition, this research was funded by the Spanish Ministry of Economy and Competitiveness (grants RTI2018-099646-B-I00 and TIN2017-90773-REDT) and the Galician Ministry of Education (grants ED431F 2018/02, ED431C 2018/29 and “accreditation 2016-2019, ED431G/08”). All grants were co-funded by the European Regional Development Fund (ERDF/FEDER program).

## References

- [1] A. Abdul, J. Vermeulen, D. Wang, B. Lim, M. Kankanhalli, Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda, in: CHI Conference on Human Factors in Computing Systems, Vol. 582, 2018, pp. 1–18.
- [2] J. M. Alonso, From Zadeh’s computing with words towards explainable Artificial Intelligence, in: R. Fuller, S. Giove, F. Masulli (Eds.), WILF2018 - 12th International Workshop on Fuzzy Logic and Applications, Springer, 2019, pp. 244–248.
- [3] J. M. Alonso, A. Bugarin, ExpliClas: Automatic generation of explanations in natural language for Weka classifiers, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2019, pp. 1–6.
- [4] J. M. Alonso, C. Castiello, C. Mencar, Interpretability of fuzzy systems: Current research trends and prospects, in: J. Kacprzyk, W. Pedrycz (Eds.), Handbook of Computational Intelligence, Springer, 2015, pp. 219–237.
- [5] J. M. Alonso, C. Castiello, C. Mencar, A bibliometric analysis of the explainable artificial intelligence research field, in: 17th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU), Vol. CCIS853, 2018, pp. 3–15.

- [6] O. Biran, C. Cotton, Explanation and justification in machine learning: A survey, in: IJCAI Workshop on Explainable AI, 2017, pp. 8–13.
- [7] A. Cascallar-Fuentes, A. Ramos-Soto, A. Bugarín, Adapting SimpleNLG to galician language, in: 11th International Conference on Natural Language Generation, ACL, 2018, pp. 67–72.
- [8] C. Criado-Perez, Invisible Women. Exposing Data Bias in a World Designed for Men, Chatto & Windus, 2019.
- [9] EU AI HLEG, AI ethics guidelines, <https://ec.europa.eu/digital-single-market/en/news/draft-ethics-guidelines-trustworthy-ai> (2018).
- [10] European Commission, Artificial Intelligence for Europe, Communication from the commission to the european parliament, the european council, the council, the european economic and social committee and the committee of the regions, Tech. rep., Brussels, Belgium, <https://ec.europa.eu/digital-single-market/en/news/communication-artificial-intelligence-europe> (2018).
- [11] A. Gatt, E. Kraemer, Survey of the state of the art in natural language generation: Core tasks, applications and evaluation, *Journal of Artificial Intelligence Research* 61 (2018) 65–170.
- [12] A. Gatt, E. Reiter, SimpleNLG: a realisation engine for practical applications, in: European Workshop on Natural Language Generation (ENLG), Athens, Greece, 2009, pp. 90–93.
- [13] J. Hühn, E. Hüllermeier, FURIA: an algorithm for unordered fuzzy rule induction, *Data Mining and Knowledge Discovery* 19 (3) (2009) 293–319.
- [14] T. Miller, Explanation in Artificial Intelligence: Insights from social sciences, *Artificial Intelligence* 267 (2019) 1–38.
- [15] Parliament and Council of the European Union, General data protection regulation (GDPR), <http://data.europa.eu/eli/reg/2016/679/oj> (2016).
- [16] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schoelkopf, C. Burges, A. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning*, MIT Press, 1998, pp. 185–208.
- [17] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [18] A. Ramos-Soto, J. Janeiro-Gallardo, A. Bugarín, Adapting SimpleNLG to spanish, in: 10th International Conference on Natural Language Generation, ACL, 2017, pp. 142–146.
- [19] E. Reiter, R. Dale, *Building natural language generation systems*, Cambridge University Press, 2000.
- [20] N. Tintarev, J. Masthoff, Explaining recommendations: Design and evaluation, in: F. Ricci, L. Rokach, B. Shapira (Eds.), *Recommender Systems Handbook*, Springer, 2015, pp. 353–382.
- [21] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2016.
- [22] L. A. Zadeh, Fuzzy sets, *Information and Control* 8 (1965) 338–353.
- [23] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning I, *Information Sciences* 8 (1975) 199–250.
- [24] L. A. Zadeh, From computing with numbers to computing with words - From manipulation of measurements to manipulation of perceptions 46 (1) (1999) 105–119.
- [25] L. A. Zadeh, A new direction in AI: toward a computational theory of perceptions, *Artificial Intelligence Magazine* 22 (1) (2001) 73–84.