Received October 27, 2019, accepted November 15, 2019, date of publication November 27, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956136

Gentle Introduction to Artificial Intelligence for High-School Students Using Scratch

JULIAN ESTEVEZ, GORKA GARATE, AND MANUEL GRAÑA, JR., (Member, IEEE)
Computational Intelligence Group, University of the Basque Country (UPV-EHU), 20018 Donostia-San Sebastian, Spain

Corresponding author: Julian Estevez (julian.estevez@ehu.es)

This work was supported by the European Union's Horizon 2020 Research and Innovation Programme through the Marie Sklodowska-Curie Grant under Grant 777720.

ABSTRACT The importance of educating the next generations in the understanding of the fundamentals of the upcoming scientific and technological innovations that will force a broad social and economical paradigm change can not be overstressed. One such breakthrough technologies is Artificial Intelligence (AI), specifically machine learning algorithms. Nowadays, the public has little understanding of the workings and implications of AI techniques that are already entering their lives in many ways. We aim to achieve widespread public understanding of these issues in an experiential learning framework. Following a design based research approach, we propose to implement program coding scaffoldings to teach and experiment some basic mechanisms of AI systems. Such experiments would be shedding new light into AI potentials and limitations. In this paper we focus on innovative ways to introduce high school students to the fundamentals and operation of two of the most popular AI algorithms. We describe the elements of a workshop where we provide an academic use-create-modify scaffolding where students work on the Scratch partial coding of the algorithms so they can explore the behavior of the algorithm, gaining understanding of the underlying computational thinking of AI processes. The extent of the impact on the students of this experience is measured through questionnaires filled before and after participation in the workshop. Preliminary experiments offer encouraging results, showing that the workshop has differential impact on the way students understand AI.

INDEX TERMS Scratch programming, teaching AI fundamentals, public AI awareness.

I. INTRODUCTION

The widespread knowledge and application of scientific reasoning is a key social aspect in technologically advanced societies [13]. Therefore, it is of great importance that students find its elements in their curricula as early as possible. Nevertheless, high school is a good time because students are mature enough to cope with serious problems. Any effort invested by the education system to foster its mastering by the young students will undoubtedly pay off. It has been acknowledged that there is a quite widespread mistrust of science [1], [2], calling for a radical change in the way the younger generations are introduced to science and technology.

The education science community widely recognizes that school curricula must move on from traditional expositive classes to more informal, interactive, experimental, and collaborative learning environments. The experiential learning theory is the foremost important movement in this direction [8], [9]. The generally acknowledged fact is, however, that active participation of the students at the classroom is difficult to achieve. One of the reasons is the lack of appeal of presentations to the students. In order to grasp their attention, teachers should make use of more attractive resources, such as fun facts, interactive interfaces, games, social interaction, observation and discussion of real life problems, novelty, etc. This is specially true of high school students, which are digital natives. Computational science seems to be a specially fertile ground for such kinds of innovations, such as the interactive graphical programming environment Scratch[1] designed by MIT researchers [29] in order to introduce K-12 students to programming in a playful mode.

The associate editor coordinating the review of this manuscript and approving it for publication was Chin-Feng Lai.

[1] https://scratch.mit.edu

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

A. ARTIFICIAL INTELLIGENCE AND COMPUTATIONAL THINKING

Artificial Intelligence (AI) is a breakthrough technology that is quickly transforming our society, economy and jobs. Some popular examples of AI applications are social networks, driverless cars, chatbots, voice assistants, internet search engines, robot stock traders, etc. These systems can be embedded in physical machines or be stand-alone software agents acting in the digital space. Their astounding promised capacity to change our lifestyle calls for society and politicians to underastand and regulate the functions and limits of these devices [15]. In an pioneering move, British Parliament has recommend the education on AI topics starting from high school [36], in order for future generations to cope with coming technological and social challenges. This political statement has no parallel in our country, Spain, or other European countries yet, declaring that it is highly necessary to improve technological understanding by the public, enabling citizens to navigate an increasingly digital world, and to contribute to the debate about how AI should, and should not, be used. Critical thinking allowing the public to go beyond the mythical thinking represented by Hollywood blockbusters is a growing social need.

Computational thinking (CT) underlies AI develpments as one of the fundamental approaches to problem solving. CT is viewed as an important ingredient of STEM curricula in primary and secondary education [20], closely related to mathematics which has received a strong impulse to improve the curricula and pedagogic resources for primary and high schools [28] linking with appealing new technologies, such as robotics [20]. CT was proposed originally as the relation between computer programming and thinking skills [21], so that computer programming would enhance the children performance across disciplines. More modern definitions of CT elaborate on the idea that problems and solutions can be proposed in terms of an information processing agent, including some precise definitions of CT related skills. In this setting, some researchers worked on the definition of visual tools that facilitate teaching and learning [22], giving rise to Scratch and similar visual programming environments [29]. Since early studies, AI has been recognized as a relevant part of the CT curricula [38], [41], [43], often using game based approaches [39]. However, the literature on actual research about teaching AI at high school or before is scarce [40]. On the other hand, the Scratch platform has been proved to be a solid tool for the acquisition of CT [37]. It allows to build programs by the graphical interactive composition and editing of blocks controlling the actions of different actors. This approach to building algorithms make programming fairly easy for beginners.

B. PAPER AIMS AND CONTRIBUTIONS

Our long term aim is to help educating the general public and specially the younger generation on the power and limitations of AI as a breackthrough technology that is already having an strong impact on their lives. Design based research (DBR) [3], [5]–[7] provides the most general methodological framework for this endeavor. In this direction, we work on the development of a collection of simple educational exercises aimed to promote understanding and learning of AI at high schools using Scratch as the transmission medium. Full fledged DBR pedagogical projects are long term efforts involving big pedagogical research teams, carrying out several iterations of the problem formulation and solution design, and have a direct connection with the teaching professionals that are in direct contact with the students. In this regard, we have already contacted some teachers with a positive interest in the topic focusing on a specific segment of the student population. However, our work reported here represents a first iteration of a long term effort that would involve more high schools and the training of teachers to carry out the workshop pedagogical scaffolding.

The workshop focuses on two specific tasks: data clustering, and artificial neural networks learning. The software workshop is designed for students 16-17 years old. The chosen algorithms are specifically designed or adapted considering the mathematical background of those students. After the workshop, the students should be able to understand, play and eventually code more sophisticated versions of these algorithms. This paper is a follow up and improvement over work reported in [44], [45]. We report on its experimental evaluation with a group of high school students, assessing the improved understanding of the technology and its implications after the hands-on workshop was carried out. Before and after the workshop, students must fulfill a questionnaire to check the level of achievement of the planned objectives. The two concrete contributions of the paper towards the answering the main research question are: (1) the design of a learning intervention grounded on experiential learning theory (ELT) approach to teach CT topics, and (2) its validation with a population of high-school students to see the impact on their learning of AI concepts and also the understanding of societal implications of AI.

C. PAPER CONTENTS

The article is divided into the following sections. Section II, presents some methodological background for the study. Section III gives the research question of the study, along with the concrete questions that will provide some light on the effect of the worshop. Section IV describes the workshop design and the mathematics underlying the Scratch coding exercises. Section V details the methodological aspects of the workshop. Section VI presents the results of the pilot workshop realization. Section VII provides a discussion of the results. Finally, Section VIII gives our conclusions, limitations of the experiment, and some future directions of work.

II. METHODOLOGICAL BACKGROUND

Design based research (DBR) aims to develop learning pedagogical interventions through a series of refinements of the experiments, focusing the target population, research questions, and intervention implementations, such as software tools, appropriate scaffolding, etc. [5], [7]. DBR loong term efforts usually leave a trail of pedagogical tools that have been develped and tested along the several iterations of the process. Lately, these tools are becoming mostly software implementation and applications for a variety of devices that can be handy even for purposes different from the DBR project original aim. The focus of DBR is on the evolving design of the tools and processes. Consequently, DBR projects are long in time and often broad in scope [6]. We think that our own work as reported in this paper may be part of a increasingly needed DBR approach to educate the public on the potentials and pitfalls of AI.

Experiential learning theory (ELT) is based on the works of Kolb and Kolb [8]. The essence of ELT is the iteration over the cycle of concrete experience, observation, abstraction of concepts, and active design of experiences. In this regard, ELT proposes and evolving generation of knowledge that fits quite well with DBR view of the pedagogy evolution. Naturally, ELT has been broadly applied learning science, increasingly using computer simulations as the main tool that allows repeteability without tearing equipment [9].

Computational thinking (CT) can be defined as the process of recognizing aspects of the sorrounding world that can be formulated in computational terms in order to apply tools and techniques from computer science to understand and reason about both natural and artificial systems and processes [10], [30], [31]. From a pedagogical perspective, computational tools facilitate an experiential approach to the learning of mathematics and science through the use of simulation and software based scaffolding [32]–[34], but CT needs its own tools and methodological approach. It is quite evident that ELT is the most appropriate pedagogical framework for guiding the student in the maturation of its own CT skills, and the use of visual programming tools sucha as Scratch [29] greatly facilitates this process [10]. Software to enhance CT skills should be easy for a beginner, i.e. the learning curve to create working programs must be not too steep, but the tool should also be powerful and extensive enough to satisfy the needs of advanced programmers. Graphical programming environments are intuitive to use and allow early experiences to focus on designing and creating, avoiding issues of program syntax correctness. Several programming environment tools fit these criteria to varying degrees. Scratch is today one of the most popular of these programming environments, and it has proved to be very effective for the engagement and motivation of young students with no programming experience [10], [35]. A very positive feature of program coding is that it allows a variety of ELT aproaches in a general framework of use-modify-create experiences. Students can learn the code properties by runing the code over diverse data sources, or by introducing small modifications and observing the perturbed behavior of the algorithm. Finally, they can use pre-existing codes as building blocks for the creation of new programs with emergin properties.

The robotics paradigm has been widely recognized as a proper instrument for honing student CT skills [30] since the pioneering works of Seymour Papert [21] already in the 1970s laying much of the groundwork for using robots in the classroom. Studies generally show that robotics generates a high degree of student interest and engagement and promotes interest in math and science careers [26]. Specifically, robotics workshops have been used to teach coding to young students [20], [24], [25]. Lego robots have been pionering the development of easy graphical interfaces that allow children to code the control software of the robots that they build [19] managing the sensors and the actuators in the environment, sustaining an ELT approach to robotics and CT.

There is a wide consensus in computer science literature that it is quite difficult to teach the basics of AI [18], due to (a) the lack of a unified methodology, and (b) the fact that AI emerges as a mixture of many other disciplines involving skills ranging from very applied thinking to extremely formal reasoning [19]. For instance there are specific prerequisite knowledge areas needed to understand machine learning, such as statistics and probabilistic modeling [27]. Robotics can be also useful to teach AI. For example, [17] uses a low-cost robot platform to teach students how a neural network is trained to solve a robotic navigation problem. However, the boundaries between robotics and AI concepts may need some clarification and enforcement [23], because students may be distracted by the strong attraction of robotics gadgetry. In this article we follow a non-robotics-based approach to teaching AI fundamentals, introducing mathematic foundations of some algorithms and guiding the students to experiment them with the help of Scratch in an ELT oriented strategy.

III. RESEARCH QUESTION

Our endeavor is to have an impact on the students understanding of the potential impact of AI technology in their lifes through the experimentation with computational constructs. We can summarize this aim into the following research question:

Does the computational experience with basic AI algorithms affect the perception of the capability of AI to affect the lives of common people?

We have prepared a questionary with 6 questions to be filled before and after the workshop realization aimed to assess the effect of the workshop on the student's perception of the AI impact in society. The questionaire contains two open-ended questions, and 4 questions that are answerd selecting a degree in a 5 degree Lickert scale, from *totally disagree* to *totally agree*. The latter questions allow a quantitative measure of the achieved effect.

1) "Describe shortly what you think Artificial Intelligence is." We expect the intervention to shift the responses of the students to this open-ended question from vague references to robotics to more specific references to data processing algorithms and the ability to model the world for predictive purposes.

2) "The level of technical knowledge required to understand AI is too high for most of the population." This question points in the direction of the student belief that the social awareness on the workings of AI can be truly informed. We expect that the intervention will move the students to believe that AI can be understood for a wide fraction of the population.
3) "If an algorithm can predict the result of a football match, it is quite probable that it will predict the beginning of the 3rd World War." The question points in the direction of the student's ability to detect the hype in the marketing campaings selling AI backed products. We expect that the intervention will move the student to believe that upgrading of the AI systems to solve very complex ill defined problems is not easily achieved.
4) "In less than 10 years, a *Terminator* kind of AI will be developed, which will threat the Humanity." The question points to the sheer difficulty/imposibility of translating movie blockbuster mythology to reality. We expect that the students will be moved to disagree.
5) "As a user, the legal regulation of the devices using AI will affect my life." The question points to the need of AI regulation, we expect that the intervention will move the student to agree.
6) "Describe shortly which are the main dangers of AI, if you know any of them." The open-ended question is directed also to debunk the movie mythological thinking. We expect that the intervention will change the student opinion from catastrophic scenarios to more realistic scenarios such as the control of the personal information.

Questions 1 and 6 allow free elaboration of the student thoughts, aiming at focusing his/her attention on the topic. Summarizing, we consider that the workshop has a positive effect if the students responses move from agreement or indiference to disagreement in questions 2-4, and from disagreement or indiference to agreement in question 5.

IV. WORKSHOP DESIGN

The short term objective of the hands on-workshop is that the students become aware through construction and experimentation of the working behind intelligent systems, achieving an intuitive knowledge of their properties and limitations. Our long term objective is to educate the public to identify the complexity and long-term consequences of AI avoiding simplistic propaganda views such as the rising of an all-powerful malignant AI visualized by movie blockbusters. We evaluate the extent that the short term objective serves to our long term objective by means of questionnaire response evaluations. We have selected specific algorithms which are prototypical ilustrations of the most salient computational approaches underlying AI implementations. The algorithms are simple enough so they can be implemented in Scracth, and the students can understand them to point of making some numerical experiments. Also, the algorithms are poweful enough to produce non-trivial results which capture the imagination of the students:

- K-means : is the most popular unsupervised learning algorithm, which aims to discover aggrupations of objects which may be similar enough to have some meaning.
- Artificial neural network (ANN): with the recent surge in interest in deeplearning architectures, ANNs have gained prominence in the AI literature. Also it is an exellent example of supervised learning approaches where the system learns a map from the input features to the classification/regression result from the data.

The basic mechanisms of unsupervised (K-means) and supervised (ANN) machine learning are the cornerstones of modern AI, so their intuitive understanding brings light on the power and limitations of AI. For instance, being data driven, all methods are conditioned to the available data, even when the learning method includes the generation of data samples for positive or antagonistic learning. We apply the use-modify-create scaffolding usually applied in progamming language teaching. We have also followed a backward design approach [46] to identify the minimal working elements in order to achieve the required pedagogical results.

A. K-MEANS

The K-means algorithm, proposed in 1967 by MacQueen [4], is the most popular unsupervised learning algorithm solving the data clustering problem. The motivation of selecting K-means for presentation to the students is three-fold:

- Its simplicity: it is the simplest of all the unsupervised learning algorithms and has a clear intuitive interpretation of its results.
- Its popularity: despite its simplicity and some robustness issues (such as the sensitivity to initial conditions) it is the most used clustering algorithm in practice.
- It is well known: the algorithm has been explored and tested from many points of view, so that its behavior is well understood by the community, which creates a strong agreement on claimed results.

Applying K-means aims to show the students how items in a big dataset can be automatically organized into categories, even while new items are being added. The underlying mathematical technique for creating as many clusters as we want is the minimum square error rule. Starting from a given cloud of points, in this activity the student familizarizes with the Scratch code that finds K clusters in the cloud of points. In order to visualize the clustering, the points will be assigned colors corresponding to the clusters. Each data point is assigned to the cluster corresponding to the closest cluster center of mass (see Figure 1), ensuring that the achieved clustering has the minimal intracluster variance (aka reconstruction error) given the actual distribution of the cluster centers. A second phase of the iterative K-means is the recomputation of the cluster centers given the actual assignment of points to clusters. The K-means algorithm alternates both phases

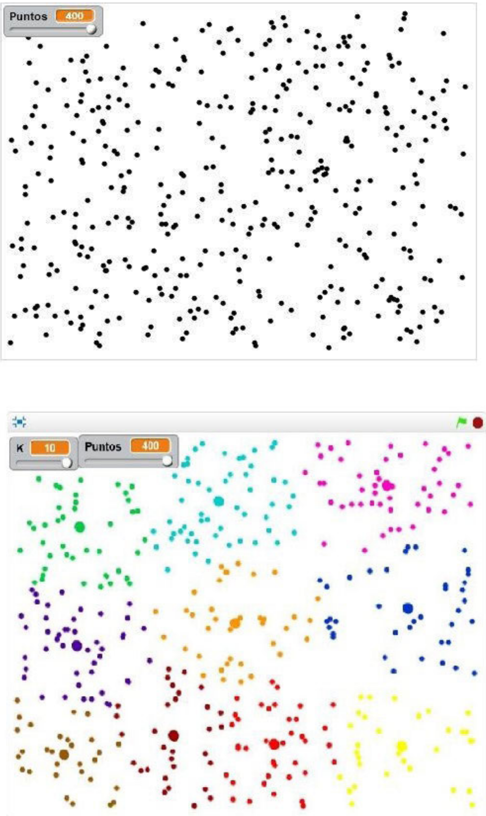

FIGURE 1. Initial randomly generated cloud of points (up), and color coded clustering of the points according to a given set of cluster centers visualized as bigger dots. K = 10 (bottom).

until convergence to a situation where no more changes of the cluster centers happen. For the experimental workshop we have restricted the coding experience to the first phase.

B. ARTIFICIAL NEURAL NETWORK

The basic idea behind an artificial neural network (ANN) is to simulate lots of densely interconnected brain cells inside a computer so you can get it to learn things, recognize patterns, and make decisions in a human-like way. Its most appealing characteristic is the robust learning algorithm based on the minimization of the error measured as the distance between the desired response and the one provided by the ANN corresponding to a given input pattern. Learning consists of a sequence of presentations of input patterns, computation of the ANN response, computation of the response error relative to the known desired response, and correction of the ANN parameters (aka weights) according to a rule that minimizes the error (aka back-propagation of the error). The programmer just needs to design the ANN structure (number of outputs, inputs, hidden layers, and units *per* hidden layer) and code some simple rules involving additions, multiplications and derivatives implementing the back-propagation rule. Current ANNs are based on the perceptrons developed in the 1950s and 1960s by Frank Rosenblatt [14], inspired on earlier

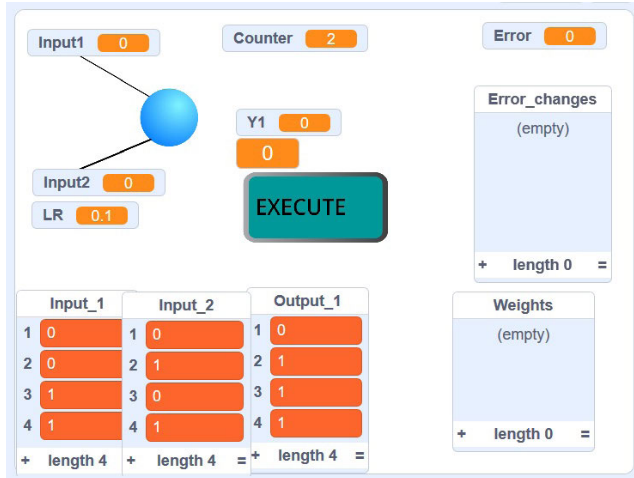

FIGURE 2. Interface for the ANN based on an AND gate.

work done by Warren McCulloch and Walter Pitts [11]. It is important to note that ANNs are (generally) software simulations: they are implemented by programming conventional serial computers.

Habitually, ANNs use back-propagation type algorithms which require the usage of derivatives. However, the target students of this workshop exercise (16-17 years old) do not have this mathematical operation in their curricula yet. As a consequence, the ANN training is carried out by application of a least-square algorithm and a logic activation function, which are within the scope of the actual mathematical background of the students. Necessary formulas are presented to them without sophisticated mathematical elaborations.

For this workshop, we propose a simplified ANN with two inputs and an output neuron. This ANN is trained with an AND logic gate in an iterative way. Figure 2 visualizes the ANN structure and the actual values of the variables and weights during an step of a simulation. At each learning iteration, students will see the change in values of the ANN weights. Next, and OR logic gate will be used, and as a consequence, students will observe how adjusting parameters change with this new structure. Next subsection explains the workings of the AND based ANN.

1) NAÏVE SINGLE NEURON OPERATION AND TRAINING

The neuron activationN_1 is computed multiplying the two ANN inputs $Input_1$ and $Input_2$ by the corresponding connection weights as follows:

$$N_1 = Input_1 \cdot W_1 + Input_2 \cdot W_2 \tag{1}$$

The neuron output Y_1 is computed as follows:

$$Y_1 = \begin{cases} 1, & N_1 \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The error is defined as the difference between the desired output *Desired_output* and the neuron's output

$$Error = Desired\_output - Y_1 \tag{3}$$

Each time the function is executed, the algorithm updates the weights and the bias using the gradient descent rule, until the output Y_1 converges to the desired output *Desired_output*. The updated value of the first weight will be the sum of its previous value and the product of the first input, the learning rate (LR), and the error

$$W_{1,new} = W_1 + Input_1 \cdot LR \cdot Error \tag{4}$$

Similarly, the updated value of the second weight will be the sum of its previous value and the product of the second input, LR, and the error.

$$W_{2,new} = W_2 + Input_2 \cdot LR \cdot Error \tag{5}$$

C. SCRATCH IMPLEMENTATIONS

Here we present the Scratch templates implementing both algorithms.

1) K-MEANS

The algorithm is implemented using one main block and three function blocks of Scratch code. The function blocks are called `NewDataSet`, `KMeans` and `ColourPoints`. The student must finish the specification of the block called `KMeans`. The main block creates the cluster centers by random generation following an uniform distribution in the data space. The set of cluster centers contains K random points with X coordinates between $(-230, 230)$ and Y coordinates between $(-170, 170)$. The block `NewDataSet` will create the point cloud containing N points with X coordinates between $(-230, 230)$ and Y coordinates between $(-170, 170)$ (the block is visualized in Figure 3). The block `KMeans` contains the core of the clustering algorithm. It must calculate the distance of each of the N points to each K mass center, and assigns the point to the the nearest mass center. This is the algorithm that students must complete. Finally, block `ColourPoints` graphs the clouds of points and the set of cluster centers, using the colour palette given by vector `Clusters`.

2) ANN WITH AND LOGIC GATE

The algorithm is implemented using a main block, which initializes the data, and two blocks, `Neuron` and `ExecuteButton`. The user interface visualization for this example is shown in Figure 2. The students must code the equation that updates the weight W_2 of the neural network. For guidance, equation for the update of W_1 is already implemented in the Scratch file (the block is visualized in Figure 5). Once the exercise is completed, the students will train the neuron the set of data of an AND logic gate, so that they see the evolution of weights and the output error, as seen in Figure 2.

FIGURE 3. Sequence of Scratch blocks that create the cloud of points for the K-means experiment.

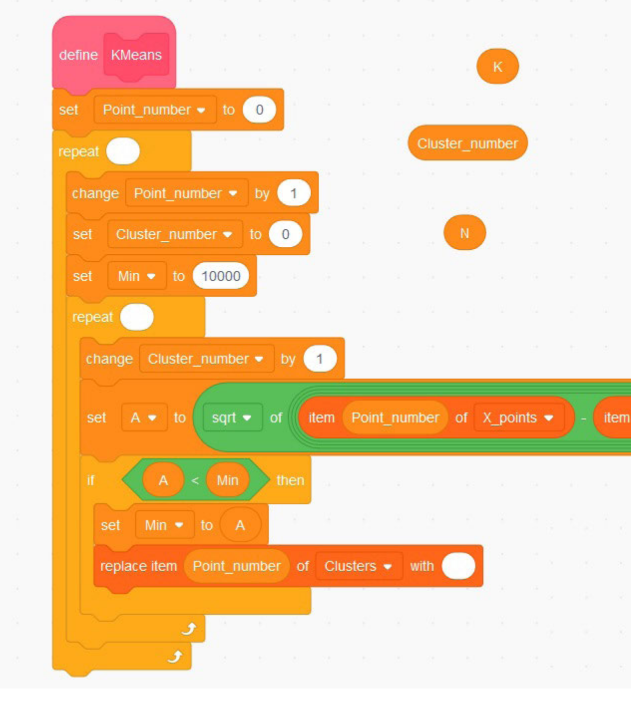

FIGURE 4. Task 1. Unfinished function template that must be completed by the students. Its functionality is to assign the cloud points into K clusters.

V. METHODS

A. PARTICIPANTS

We have enrolled 37 students of the technological branch of a local high school[2] (16-17 years old, 40% girls 60% boys) to

[2]http://www.summa-aldapeta.com/castellano/

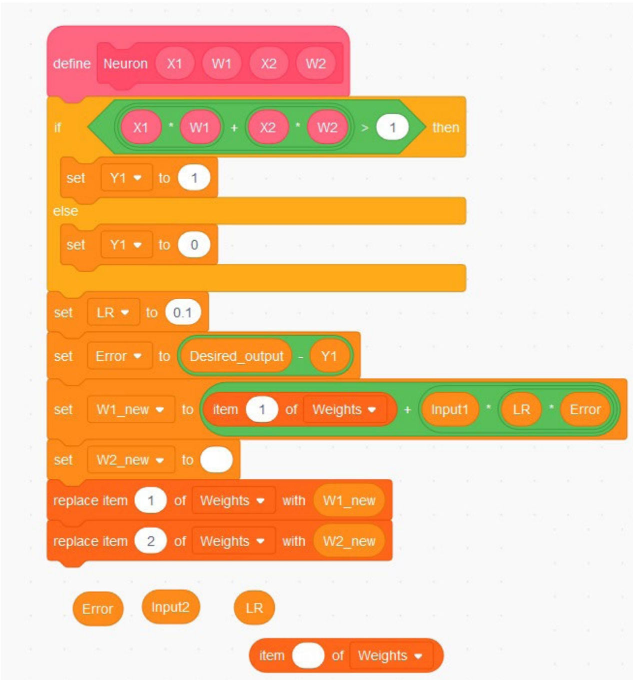

FIGURE 5. Task 2. Unfinished block `Neuron`**.**

participate in the workshop. The workshop has been carried out once outside the formal curricula of the local school. Students had a prior acquatance with Scratch through preparatory lectures of exercises (2 hours per week in the previous 2 weeks). Students have technological and mathematical maturity according to the teachers opinion to enter the workshop. The workshop is a one time experiment, because of limitations of student time availability. Besides,repeating the same protocol will not add much to the student knowledge. The files containing the Scratch implementations used in the workshop are available at `https://gitlab.com/juleste`. For third party realization of the workshop, it is convenient that the students previously acquire some practice of coding in Scratch.

B. WORKSHOP PROTOCOL

The workshop of Artificial Intelligence has been based on the usage of the educational tool Scratch. Scratch is a visual programming language, and its online community is targeted primarily at children and young students [10]. Using Scratch, users can create online projects and develop them into almost anything by using a simple block-like interface. When they are finished, students can share their projects and discuss their creations with each other.

For all algorithms presented in Section IV, the task of the students is to fill in the white gaps left among the lines of code. That is, we use a "use-modify-create" approach to teach computer programing where the students do not need to create the algorithm or write the whole code themselves; a template of the code is provided. Moreover, the pieces of code to use will be available on the screen. Students work in pairs. The workshop is structured in the following steps:

- At the beginning, the teachers gave a short explanation of 15 minutes about AI and the objective of the workshop, with a twofold aim: first, to demystify Artificial Intelligence; second, to understand some simple mathematics underneath the computations. Teachers will provide them with some written theoretical background about the algorithms and the instructions for the exercises (K-means and neural networks)
- Students had forty minutes to finish the codes (20 minutes for K-means and 20 minutes for the ANN), eventually executing the applications to assess the working of the algorithms. Teachers were at hand to assist whenever necessary.
- First, students received the Scratch file related to K-means exercise. Once all the groups finish, they will receive the neural network file.
- Finally, at the end of each algorithm, the teachers will provide the students with the finished proposed answers in Scratch software for the students to check them.

C. DATA COLLECTION METHOD

The method to validate the degree of achievement of the planned objectives in the workshop, is a questionnaire given to each participant student. The questions have been already presented and discussed in section This questionnaire was answered before and after the workshop, so that the authors could compare the results. No personal information was collected in any way, each participant questionnaire was identified by random number uncorrelated to any personal identifier.

D. DATA ANALYSIS METHOD

The open ended questions contain a natarrative that is not amenable to quantitative analysis, hence we carried out a qualitative assessment of their contents according to the critical lines pointed out in section III. The questions that are answered through a Likert scale valoration are processed as follows:

- Firstly, we visualize the pie charts of the responses before and after the workshop;
- Secondly, we build a two class contingency table as follows:
  - we gather the partial and complete agree responses into a single class of agree responses,
  - we do the same with the disagree responses,
  - we discard the indiferent responses.
- Finally, we compute the Fisher test over the two class contigency table [42] which gives an answer to the question: did the distribution of postive (agree) and negative (disagree) changed between pre and post-workshop answers?. The size of the effect is given by the odds ratio reported as part of the test in its R implementation.

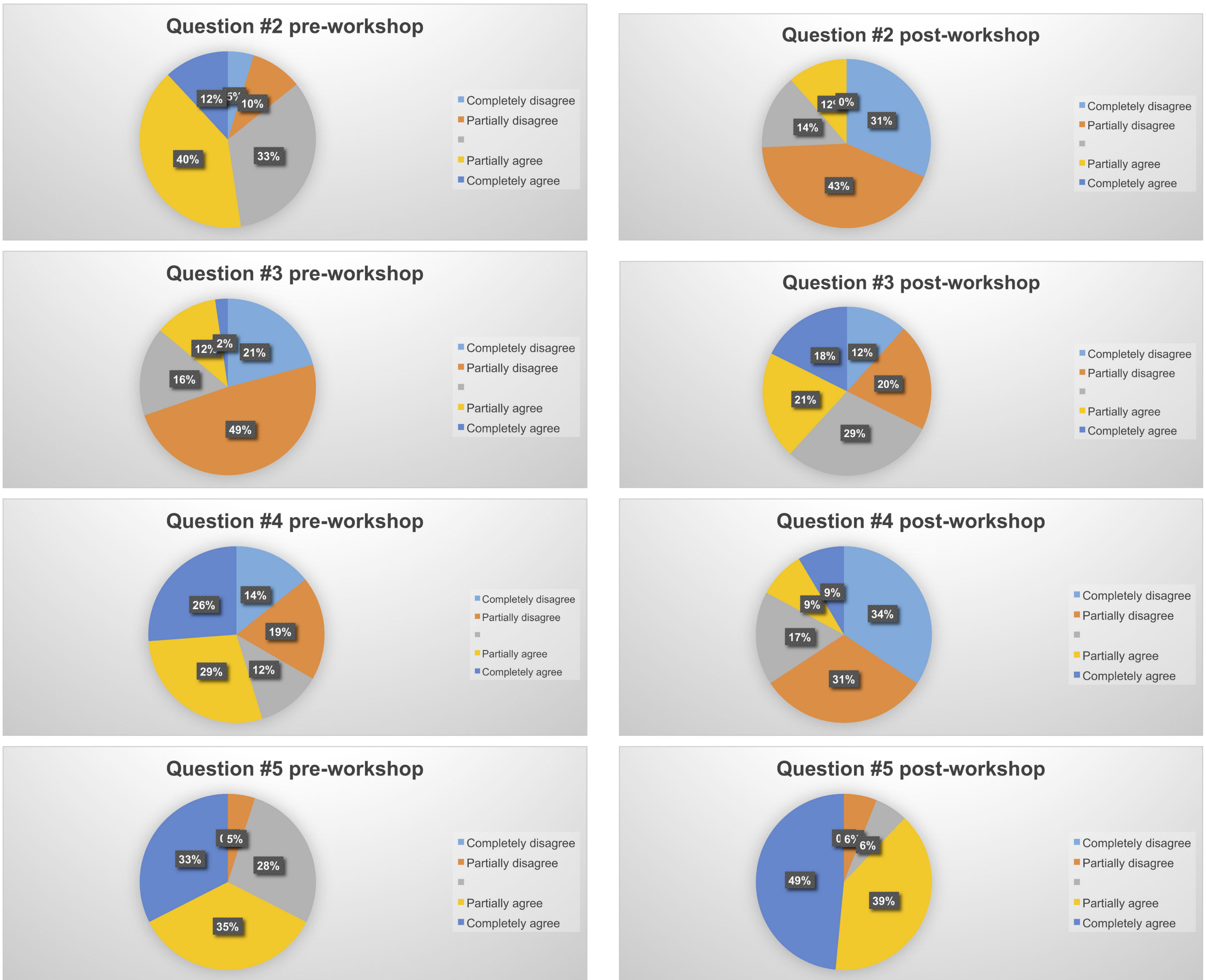

FIGURE 6. Distribution of responses to the Likert-scale questions pre (left) and post-workshop (right).

VI. EXPERIMENTAL RESULTS

First we will comment on the qualitative results from the open-ended questions of the questionnaire. Secondly, we report the quantitative results.

A. QUALITATIVE RESULTS

The answers of the open-ended questions show a good degree of qualitative impact of the workshop.

- Question 1: Answering this question in the pre-workshop phase, students tend to reply that AI is *the intelligence of the machines and robots*, while in the post-workshop phase, they tend to reply predominantly that *IA is the collection of algorithms to learn and predict things,* so that they appear to have acquired a more precise awareness of what AI really is.
- Question 6: In the pre-workshop phase, students tended to fill details with a high diversity of scifi movie worries, such as the slavement of humans by machines, robots becoming berseek, or the difference in physical strength between robots and humans. In the post-workshop phase, the variety of dangers was drastically reduced, and the responses were oriented towards the bad usage of data and its impact on privacy and freedom.

B. QUANTITATIVE RESULTS

For a qualitative appraisal of the distribution of the responses, we show in Figure 6 the pie charts with responses pre and post workshop, which include the indiferent responses in gray color. The effect of the workshop is more apparent on the indiferent responses that change sometimes by large percentages of the sample. We show in Tables 1, 2, 3, and 4 the percentages of agreement vs. disagreement before and after the workshop with questions 2, 3, 4, and 5, respectively. As we are discarding the indifferent responses, the rows do not add to the 100% of the entire population. The results of the Fisher test are as follows:

- Question 2 there is a significant difference on the answers pre and post workshop ($p<1e{-}5$, odds ratio $20{,}7>>1$). Before the workshop, answers seem to be random and equally distributed. After the workshop, students have acquired the confidence to be able to understand the workings of AI algorithms.

TABLE 1. Answers before and after the workshop to Question 2: The level of technical knowledge required to understand AI is too high for most of the population. P/C = partially / completely. Indifferent responses are discarded.

Situation	P/C Agree	P/C Disagree
Before	52%	15%
After	12%	74%

TABLE 2. Answers before and after the workshop to Question 3: If an algorithm can predict the result of a football match, it is quite probable that it will predict the beginning of the 3rd World War. P/C = partially / completely. Indifferent responses are discarded.

	P/C Agree	P/C Disagree
Before	15%	70%
After	39%	32%

TABLE 3. Answers before and after the workshop to Question 4: In less than 10 years, a *Terminator* kind of AI will be developed, which will threat the Humanity. P/C = partially / completely. Indifferent responses are discarded.

	P/C Agree	P/C Disagree
Before	65%	33%
After	18%	65%

TABLE 4. Answers before and after the workshop to Question 5: "As a user, the legal regulation of the devices using AI will affect my life.". P/C = partially / completely. Indifferent responses are discarded.

	P/C Agree	P/C Disagree
Before	68%	5%
After	88%	6%

- Question 3, differences pre and post workshop are statistically significant ($p<1e-5$, odds ratio $0.17<<1$). Students have acquired the ability to contemplate the similarity between two seemingly quite different situations formulated as prediction problems.
- Question 4, differences in pre and post workshop responses are strongly significant ($p<1e-8$, odds ratio $7>>1$). Students have acquired a more realistic view of AI, far from movie culture.
- Question 5 was in fact quite unrelated to the workshop technical contents, on a more legal aspect, so responses of the students do show little variation after the workshop realization ($p = 0.7$, odds ratio 0.77). This variation is due to the change from indiferent to completely agree of 22% of the students, which is a very positive outcome of the workshop. Indiferent students became aware of the need to regulate AI due to their realization of its potential and shortcomings.

VII. DISCUSSION

Our experience with the workshop design and carrying out it with the enrolled students leads to a two-fold discussion. On one hand, we have developed some technical resources for the teaching of AI fundamentals that can easily used by teachers with minimal previous training due to the intuitive design and flexibility of visual programming in Scratch. This effort is parallel to many others in the elementary schools across the world [10]. We have found that the students have grasped the meaning of the code and the unerlying mathematical concents with ease, and that they become capable of some experimentation with new code after a few training sessions. The approach has shown to convey easily and effortessly sophisticated concepts involving the fundamentals of AI, paving alternative ways to learn machine learning [27] for the unsophisticated public. On the other hand, our work can be seen as a contribution to a broader DBR effort to educate the general public on the fundamentals of AI as a way to rise public awareness on the impact of AI on their lives, and to restore trust in this kind of technologies and the need to regulate them [1], [2]. In this regard, the contribution of the paper is that we have been able to show to some extent that the students carrying out the workshop effectively cahnge their minds regarding AI to become more realistic and aware on the potential benefits and pitfalls of AI. We think that this effect is difficult to achieve by other means, such as robotic based demostration of AI [23], because they do not convey a deep understanding of the workings of the AI fundamental algorithms.

VIII. CONCLUSION

The paper presents a proposal of a teacher-guided, easy to realize workshop on the introduction of AI fundamentals that can be performed at schools with Scratch. The mathematical description has been adapted for 16-17 year-old students with some technological/mathematical background. The tasks presented are scalable and adaptable: students can delve into the maths involved in the mathematical iterations or into the Scratch code itself, or even propose new neural networks to deal with other problems. The work can be extended to students of different ages. More AI algorithms can be added to the system for a richer variety of workshop experiences. The simplicity of the equations of K-means and ANN allow their implementation in other formats, such as MS Excel, which may be more familiar to some students than Scratch. Moreover, we expect that the students will realize that the mathematical knowledge acquired throughout the year will help them finishing the programming of automatic clustering and neuron networks.

The main purpose of the workshop is to help disseminate knowledge of AI fundamentals in order to promote social awareness and mature critical thinking about the impact of AI in our everyday life. We have made qualitative and quantitative evaluation of the the impact of the workshop on the students opinion, finding a strong effect, which encourages further research and continued working on the elaboration of more sophisticated materials.

Limitations of the study: (1) the size of the sample is small, which limits the generality of our conclusions from a quantitative point of view, (2) we have carried out one time the workshop so that further variability of the results may be expected from new experimentations, (3) the limited time of the experience does not allow for detailed and extensive student scaffolding, more time extensive experiences will require preparation of more sophisticated material.

As a future work, first, the authors plan to realize the AI workshop with a greater number of students from diverse high schools to measure the degree of the objective achievement in collaboration of pedagogical researchers. In fact, we are working on the realization at a massive scale in the framework of the science communication events at the University of the Basque Country. During this event students from local schools visit the science fair produced by the University of the Basque Country and participate in many activities. Secondly, alternative programming languages should be explored (GeoGebra, HTML, ShynnyApps) in order to implement more exercises, such as the usage of ANN for data prediction.

REFERENCES

[1] T. Arimoto and Y. Sato, "Rebuilding public trust in science for policy-making," *Science*, vol. 337, no. 6099, pp. 1176–1177, 2012.
[2] B. Haerlin and D. Parr, "How to restore public trust in science," *Nature*, vol. 400, p. 6744, p. 499, Aug. 1999.
[3] P. Bell, "On the theoretical breadth of design-based research in education," *Educ. Psychol.*, vol. 39, no. 4, pp. 243–253, 2004.
[4] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1. Berkeley, CA, USA: Univ. California Press, 1967, pp. 281–297.
[5] Design-Based Research Collective, "Design-based research: An emerging paradigm for educational inquiry," *Educ. Res.*, vol. 32, no. 1, pp. 5–8, 2003.
[6] T. Anderson and J. Shattuck, "Design-based research: A decade of progress in education research?" *Educ. Res.*, vol. 41, no. 1, pp. 16–25, 2012.
[7] M. W. Easterday, D. R. Lewis, and E. M. Gerber, "Design-based research process: Problems, phases, and applications," Int. Soc. Learn. Sci., Boulder, CO, USA, Tech. Rep. 111, 2014.
[8] A. Y. Kolb and D. A. Kolb, "Experiential learning theory: A dynamic, holistic approach to management learning, education and development," in *The SAGE Handbook of Management Learning, Education and Development*, S. J. Armstrong and C. V. Fukami, Eds. Newbury Park, CA, USA: SAGE, 2011, doi: 10.4135/9780857021038.n3.
[9] G. Falloon, "Using simulations to teach young students science concepts: An experiential learning theoretical analysis," *Comput. Educ.*, vol. 135, pp. 138–159, Jul. 2019.
[10] L. Zhang and J. Nouri, "A systematic review of learning computational thinking through scratch in K-9," *Comput. Educ.*, vol. 141, Nov. 2019, Art. no. 103607.
[11] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.
[12] M. A. Nielsen, *Neural Networks and Deep Learning*. San Francisco, CA, USA: Determination Press, 2015.
[13] "Scientific advice for policy making: The role and responsibility of expert bodies and individual scientists," OECD Publishing, Paris, France, Tech. Rep. 21, 2015.
[14] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, p. 386, 1958.
[15] P. Stone, R. Brooks, E. Brynjolfsson, R. Calo, O. Etzioni, G. Hager, J. Hirschberg, S. Kalyanakrishnan, E. Kamar, S. Kraus, and K. Leyton-Brown, "Artificial intelligence and life in 2030. One hundred year study on artificial intelligence: Report of the 2015-2016 study panel," Standford Univ., Standford, CA, USA, Tech. Rep., 2016.
[16] C. Woodford. *How Neural Networks Work—A Simple Introduction*. Accessed: Oct. 1, 2019. [Online]. Available: https://www.explainthatstuff.com/introduction-to-neural-networks.html
[17] L. Greenwald and D. Artz, "Teaching artificial intelligence with low-cost robots," in *Accessible Hands-On Artificial Intelligence and Robotics Education*, L. Greenwald, Z. Dodds, A. Howard, S. Tejada, and J. Weinberg, Eds. Menlo Park, CA, USA: AAAI Press, 2004, pp. 35–41.
[18] M. Hearst, "Improving instruction of introductory artificial intelligence," in *Proc. AAAI Fall Symp.* New Orleans, LA, USA: AAAI Press, Aug. 1995.
[19] D. Kumar and L. Meeden, "A robot laboratory for teaching artificial intelligence," *ACM SIGCSE Bull.*, vol. 30, no. 1, pp. 341–344, 1998.
[20] A. Eguchi, "Robotics as a learning tool for educational transformation," in *Proc. 4th Int. Workshop Teach. Robot., Teach. Robot., 5th Int. Conf. Robot. Educ.*, Padova, Italy, Jul. 2014, pp. 27–34.
[21] S. A. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*. New York, NY, USA: Basic Books, 1980.
[22] S. Rose, J. Habgood, and T. Jay, "An exploration of the role of visual programming tools in the development of young children's computational thinking," *Electron. J. e-Learn.*, vol. 15, no. 4, 297–309, 2017.
[23] A. N. Kumar, "Three years of using robots in an artificial intelligence course: Lessons learned," *J. Educ. Resour. Comput.*, vol. 4, no. 3, 2004, Art. no. 2.
[24] B. S. Barker and J. Ansorge, "Robotics as means to increase achievement scores in an informal learning environment," *J. Res. Technol. Educ.*, vol. 39, no. 3, pp. 229–243, 2007.
[25] M. Petre and B. Price, "Using robotics to motivate 'back door' learning," *Educ. Inf. Technol.*, vol. 9, no. 2, pp. 147–158, 2004.
[26] E. Mauch, "Using technological innovation to improve the problem-solving skills of middle school students: Educators' experiences with the LEGO mindstorms robotic invention system," *Clearing House*, vol. 74, no. 4, pp. 211–213, 2001.
[27] N. Lavesson, "Learning machine learning: A case study," *IEEE Trans. Educ.*, vol. 53, no. 4, pp. 672–676, 2010.
[28] B. Divjak and Z. Erjavec, "Enhancing *mathematics for informatics* and its correlation with student pass rates," *Int. J. Math. Educ. Sci. Technol.*, vol. 39, no. 1, pp. 23–33, 2008.
[29] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Scratch: Programming for all," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, 2009.
[30] S. Grover and R. Pea, "Computational thinking in K–12: A review of the state of the field," *Educ. Res.*, vol. 42, no. 1, pp. 38–43, 2013.
[31] Royal Society. (2012). *Shut Down or Restart: The Way Forward for Computing in UK Schools*. [Online]. Available: http://royalsociety.org/education/policy/computing-in-schools/report/
[32] M. Guzdial, "Software-realized scaffolding to facilitate programming for science learning," *Interact. Learn. Environ.*, vol. 4, no. 1, pp. 001–044, 1994.
[33] M. Eisenberg, "Output devices, computation, and the future of mathematical crafts," *Int. J. Comput. Math. Learn.*, vol. 7, no. 1, pp. 1–44, 2002.
[34] D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, and U. Wilensky, "Defining computational thinking for mathematics and science classrooms," *J. Sci. Educ. Technol.*, vol. 25, no. 1, pp. 127–147, 2016.
[35] D. J. Malan and H. H. Leitner, "Scratch for budding computer scientists," *ACM SIGCSE Bull.*, vol. 39, no. 1, pp. 223–227, Mar. 2007.
[36] *AI in The UK: Ready, Willing and Able?—Tech Report Select Committee on Artificial Intelligence*, House Lords, London, U.K., 2018.
[37] A. Repenning, D. Webb, and A. Ioannidou, "Scalable game design and the development of a checklist for getting computational thinking into public schools," in *Proc. 41st ACM Tech. Symp. Comput. Sci. Educ.*, Mar. 2010, pp. 265–269.
[38] J. Lažanský, V. Mařík, and O. Štěpánková, "Computers in artificial intelligence education," *Eur. J. Eng. Educ.*, vol. 17, no. 2, pp. 143–149, 1992.
[39] J. DeNero and D. Klein, "Teaching introductory artificial intelligence with Pac-Man," in *Proc. Symp. Educ. Adv. Artif. Intell.*, Jul. 2010, pp. 1–5.
[40] D. McArthur, M. Lewis, and M. Bishary, "The roles of artificial intelligence in education: Current progress and future prospects," *J. Educ. Technol.*, vol. 1, no. 4, pp. 42–80, 2005.
[41] N. Balacheff, "Artificial intelligence and real teaching," in *Learning From Computers: Mathematics Education and Technology*. Berlin, Germany: Springer, 1993, pp. 131–158.
[42] R. A. Fisher, "On the Interpretation of X^2 from contingency tables, and the calculation of P," *J. Roy. Stat. Soc.*, vol. 85, no. 1, pp. 87–94, 1922.
[43] E. McQuade and E. Petty, "Computers in engineering education," *Eur. J. Eng. Educ.*, vol. 9, no. 2, pp. 123–133, 1984.
[44] J. Estevez, G. Garate, J. M. L. Guede, and M. Graña, "Using scratch to teach undergraduate students' skills on artificial intelligence," 2019, *arXiv:1904.00296*. [Online]. Available: https://arxiv.org/abs/1904.00296
[45] J. Estevez, X. Elizetxea, and G. Garate, "Development of a workshop about artificial intelligence for schools," in *Proc. 13th Int. Conf. Soft Comput. Models Ind. Environ. Appl.* Cham, Switzerland: Springer, Jun. 2018, pp. 530–535.
[46] G. Wiggins and J. McTighe, Eds., "What is backward design?" in *Understanding by Design*. Alexandria, VA, USA: ASSN for Supervision & Curriculum, 2011, pp. 7–19.