

Article

Using LSTM to Identify Help Needs in Primary School Scratch Students

Luis Eduardo Imbernón Cuadrado * , Ángeles Manjarrés Riesco  and Félix de la Paz López 

Department of Artificial Intelligence, Universidad Nacional de Educación a Distancia, 28040 Madrid, Spain; amanja@dia.uned.es (Á.M.R.); delapaz@dia.uned.es (F.d.I.P.L.)

* Correspondence: limbernon3@alumno.uned.es

Featured Application: This work can be used to build a model capable of identifying when a student needs help while performing block-based programming exercises, and providing personalized pedagogical help, or an automatic system that can guide the student when he/she is stuck.

Abstract: In the last few years, there has been increasing interest in the use of block-based programming languages as well as in the ethical aspects of Artificial Intelligence (AI) in primary school education. In this article, we present our research on the automatic identification of the need for assistance among primary school children performing Scratch exercises. For data collection, user experiences have been designed to take into account ethical aspects, including gender bias. Finally, a first-in-class distance calculation method for block-based programming languages has been used in a Long Short-Term Memory (LSTM) model, with the aim of identifying when a primary school student needs help while he/she carries out Scratch exercises. This model has been trained twice: the first time taking into account the gender of the students, and the second time excluding it. The accuracy of the model that includes gender is 99.2%, while that of the model that excludes gender is 91.1%. We conclude that taking into account gender in training this model can lead to overfitting, due to the under-representation of girls among the students participating in the experiences, making the model less able to identify when a student needs help. We also conclude that avoiding gender bias is a major challenge in research on educational systems for learning computational thinking skills, and that it necessarily involves effective and motivating gender-sensitive instructional design.

Keywords: distance calculation; block-based programming language; LSTM model; teaching with Scratch; ethics in AI



Citation: Imbernón Cuadrado, L.E.; Manjarrés Riesco, A.; de la Paz López, F. Using LSTM to Identify Help Needs in Primary School Scratch Students. *Appl. Sci.* **2023**, *13*, 12869. <https://doi.org/10.3390/app132312869>

Academic Editors: Yu Liang, Wenjun Wu and Ying Li

Received: 30 October 2023

Revised: 16 November 2023

Accepted: 28 November 2023

Published: 30 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the last decade, online learning environments have gained attention, even more so since the start of the lockdowns in 2020 caused by the coronavirus (SARS-CoV2 or COVID-19) pandemic [1]. Additionally, Computer Science (CS) is experiencing exponential growth, and Scratch, a graphical block-based programming language, is a widely-used approach to improving students' Computational Thinking (CT) skills in primary school [2,3].

In an educational environment, and even more so in primary education, feedback is an important factor, as it allows students to evaluate their progress in learning. Identifying student's mental blocks and providing personalized feedback can be difficult when teachers and learners are separated by space and time, such as in online learning settings [4]. As concluded in [5], automatic feedback provided by the software can improve students' performance in programming activities.

How far a student is from finding the solution to an exercise is a key factor in knowing if he/she needs help. For this purpose, there are methods that calculate the distance between the student's workspace and the given solution of the exercise at hand. However, though different existing distance calculation approaches take into account the name and

the position of the program blocks, they ignore other attributes like the block family, or the value of the block inputs. For this reason, in [6], we proposed a novel method for calculating distances in block-based programming languages.

One of the most important elements of the learning process is knowing when to ask for help. When a student has exhausted his/her capacity for understanding and still cannot find a solution for the problem he/she is facing, he/she has to ask for help from a more competent person. However, some students avoid asking for help because they think they are incompetent and not smart enough [7]. This can prevent the student from progressing, causing him/her to lose motivation and interest in the subject. Therefore, identifying the moment when a student needs help to complete the task he/she is carrying out is a crucial aspect of effective learning.

In Ref. [6], the defined distance between the student's workspace and the solution varies depending on the difficulty of the exercise and the student's knowledge of Scratch. So each student will need help at different points in time. In Ref. [8] it is concluded that investigating the time and circumstances in which a learner requests help in an intelligent tutor system allows us to understand the type of help they need, and therefore provide high-quality hints. However, we did not find in the literature any model for the identification of help needs in block-based programming languages. This leads us to propose a help identification mechanism based on a time-series model and on user interactions in block-based programming languages. In the last few years, time-series models have seen increasing use in forecasting [9] and are widely implemented in education for identifying key elements at a certain moment, for example to analyse the concentration of students during English courses [10], or for predicting school absenteeism of autistic students [11].

Until recently it was not common to integrate ethical considerations into the development processes of artificial intelligence applications. However, nowadays experts agree on the importance of embedding ethical requirements into the design of artificial intelligence systems [12]. One of the key ethical concerns in education is discrimination based on gender and towards minority groups. An educational system that has a bias can cause student frustration and lead to loss of interest in the subject, which can in turn lead to academic failure. In recent years, researchers have focused their efforts on improving the ethical frameworks of artificial intelligence, and implementing them in different areas, including in education.

The work presented here was carried out in the context of the project Affective Robot Tutor Integrated Environment (ARTIE), in which we are developing an environment for configuring an affective robot tutor, to be used to teach block-based programming in primary school education. This environment must be able to identify when a student needs help, what their emotional state is, and how the robot should act according to each situation and emotional state [13,14]. Therefore, the robot is "affective" because it deals with or arouses emotions. The objective of this work is to create an artificial intelligence model capable of identifying the need for help of a primary school student, while performing Scratch exercises. This objective involves contributions in different research areas, in particular in the field of intelligent educational systems for teaching block-based programming and, in general, in the field of intelligent educational systems that provide automatic feedback, since the results of this work can be generalised to apply to other disciplines. The application of LSTM architectures in this domain, as well as the function proposed in [6] to evaluate the distance between the student's workspace and a given block-based programming exercise, are innovative.

The rest of the paper is structured as follows. In Section 2, we present the research context and the technological choices, we explain the user experience carried out to evaluate our proposal, we detail the distance calculation for block-based programming languages that we have used, and we explain the construction of the help identification model. In Section 3 we report on the results of our research. In Section 4 we include a discussion of our work and outline some future research directions suggested by our experience. Finally,

in Section 5 we draw conclusions from the results and discussion, and propose future work to expand on aspects of this research.

2. Materials and Methods

All the code developed for the different services to carry out the user experiences and for the implementation of the model presented in this work is available in open source at the following GitHub link: <https://github.com/orgs/ARTIEROCKS/repositories> (accessed on 27 November 2023).

2.1. Research Context and Technological Choices

In this section, we present the current state of the research areas of interest for this paper. Such areas include block-based programming interfaces in primary school education, distance calculation methods, time-series based models, and ethics in education.

2.1.1. Block-Based Programming Interfaces in Primary School

Learning programming is beneficial for primary school children, providing them with tools to improve problem-solving, as observed in [15] where students solve arithmetic and geometric sequences and series using block-based programming codes instead of using algebra. It also has other benefits such as improvement in efficiency and learning, and an increase of enthusiasm, motivation, a sense of fun and CT [16,17].

It has been observed in previous studies that, in order to increase student's achievements and abilities, programming should start at earlier stages than high school [18]. There is not much research involving primary school learners, which has been considered a critical stage to develop an interest in this subject. Block-based programming interfaces are useful in reducing the complexity of programming, with respect to text-based programming, and they are also effective at creating interest in programming at primary school [19].

Our experiment is based on Scratch, which is a free web-based programming tool focused on young people that allows for the creation of games, interactive stories, and animations [3]. This tool has been used in many CT research projects in which primary school children were involved, such as [2,3,20].

2.1.2. Distance Calculation Methods in Block-Based Programming Languages

In order to identify how "far" or "close" a block-based program is from the solution to a programming problem, we have to measure in some way the distance between the two. Block-based programming programs can be transformed into Abstract Syntax Trees (ASTs).

There are several methods that can be used to measure the "closeness" between two different ASTs, such as pg-gram implemented in [21], Tree Edit Distance (TED) applied in [22], or Tree Inherited Distance (TID) as defined in [23].

Though these methods measure the distance between two ASTs, they only consider two dimensions: the position and the label of the node. In block-based programming, as we will further explain, there are other dimensions that should be part of the distance calculation.

2.1.3. Time-Series Based Models

Time series data are samples taken over time from a real process. Many time series are temporal dependent and non-stationary [24], and constitute a discipline of data modeling where past values of a given variable are analyzed to predict the future values of this variable [9].

One of the most popular methods for processing time series is Recurrent Neural Networks (RNNs) [25]. This method is widely used for Time Series Classification (TSC), for example in forecasting [26,27], as RNNs are able to preserve information between neurons over the time line [28].

There are different types of architecture in RNNs, but the most common and currently used are LSTM and Gated Recurrent Unit (GRU) [25].

LSTM is a solution of the problem of vanishing gradient and gradient explosion in the training of RNNs. This architecture has two gates: an “input gate” and a “forgetting gate”. The “input gate” is used to add information to the neurons, and the “forgetting gate” decides which information is important and should be stored and which information to forget.

GRU is more general than LSTM and simplifies both the mathematical model and the parameter mechanism [25].

When there is a limited amount of historical sequence data, problems such as gradient explosion or gradient disappearance may occur when processing a long-term sequence of samples. This fact causes instability in the model, making it unable to converge to the optimal result. LSTM, with a strong feedback mechanism solves the problem of the long-term dependency in the Recurrent Neural Network (RNN) [10].

In education, LSTM has been implemented in different domains. For example, in [10,11], LSTM has been implemented together with a Multilayer Perceptron (MLP) and a Convolutional Neural Network (CNN) to increase their accuracy. In [28] a new framework based on the LSTM algorithm was proposed for performing personalized and adaptive learning, giving better results than the Deep Knowledge Tracing (DKT) method. And in [29], a method based on LSTM helps programming students by providing the next word following an incomplete program.

When a student is carrying out an exercise, it is crucial to identify the moment when the student will need help, therefore time is a dimension to take into account. Because we have a limited dataset, that may cause the gradient vanish or gradient explosion problem, we will implement a LSTM architecture.

RNNs have different configurations, as is described in [30]:

- One-to-one (Figure 1): One input and one output are suitable for classification tasks, but not for time series models;
- One-to-many (Figure 2): This model converts an input into a sequence;
- Many-to-many (Figure 3): This model is a sequence to sequence generator;
- Many-to-one (Figure 4): This model is suitable for a prediction or classification from a sequence.

In this study, the input is a sequence of data and the output is a boolean that indicates if a student needs help or not, therefore we will implement a many-to-one configuration.

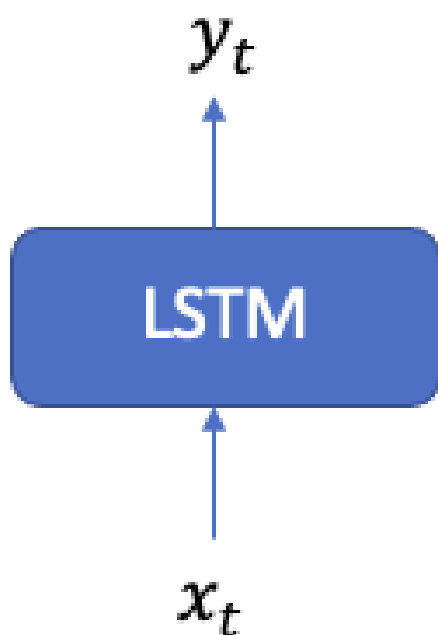


Figure 1. LSTM one-to-one.

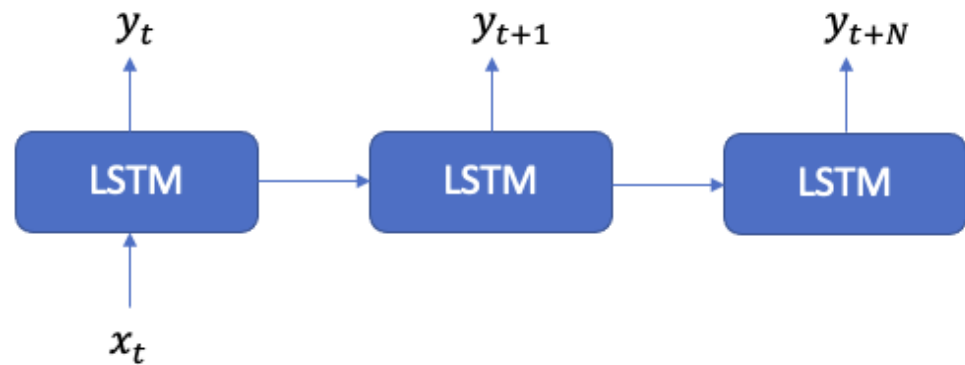


Figure 2. LSTM one-to-many.

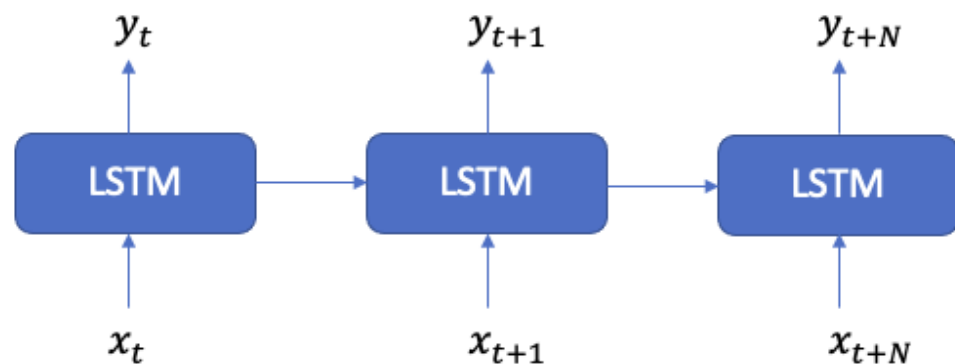


Figure 3. LSTM many-to-many.

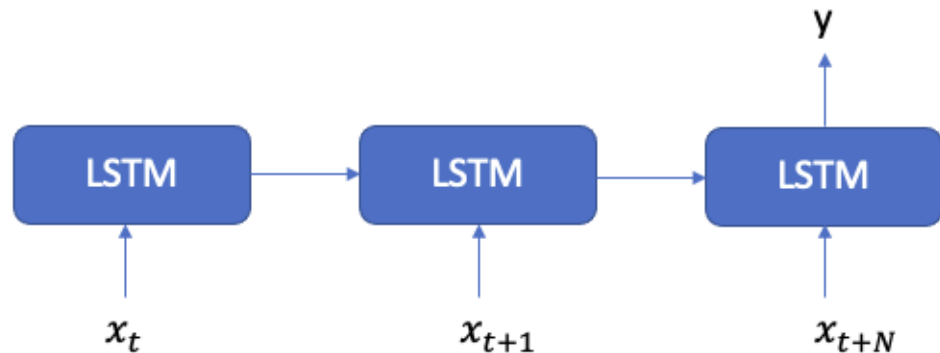


Figure 4. LSTM one-to-many.

2.1.4. Ethics in Education

Greater use of technology in education has led to an increase in the data available for training AI models. These datasets sometimes contain sensitive information. The lack of attention to ethical aspects during the design of these models has produced errors, such as bias in machine learning models, that could have a negative impact on people, industry, or society. Ref. [31] highlights a disproportionate lack of representation of black women in commercial datasets used for facial recognition. Hence, all the models fully trained with these datasets also have the same bias. Today, the biggest challenges that AI is facing in education are its huge implementation costs, ethics, transparency, and security [32].

Relatedly, we expect technology to treat students fairly, so that it does not favor some students over others. This will only be possible if an ethical design in AI is correctly implemented. We need to know how to differentiate between doing ethical things and doing things ethically. Ethical intentions, like good intentions, do not always lead to ethical designs [12].

Within the ethical aspects of education, one of the most analyzed topics is related to bias, usually aligned to gender discrimination and discrimination against minority groups. At present, there are many stereotyped educational environments, that cause negative effects on individuals belonging to minority groups. These negative effects could produce a decrease in performance of students of these groups while they are interacting with such environments.

Educational technologies are often developed with a strong unconscious gender bias, focused on the male gender. In this sense, gender must be a very important factor when designing educational technologies, in order to make them more inclusive and egalitarian [33].

Whereas AI has been integrated into education for more than a decade, the concern around ethics in this area is quite recent. Despite this, today, a large number of researchers dealing with AI in education recognize the importance and value of ethics [12].

According to [34], educational AI is divided into three large groups, within each of which there are ethical aspects that must be considered:

- Tools focused on student assistance: they help students to understand the subject being studied and increase their motivation. An incorrect categorization of students by groups can lead to presenting students with tasks of inappropriate difficulty;
- Tools focused on teacher assistance: they help teachers to focus their time and effort on identifying which students need more help. An incorrect diagnosis about which students need more help can lead the teacher to focus on those who, in fact, require less attention;
- Tools focused on educational administration: they relate to problems found in the classroom with the didactic material. In these tools, errors usually come from an incorrect identification of the student profile or from an incorrect evaluation of the didactic material.

According to [12], the AI education community needs to acknowledge the value of developing an ethical framework and practical guides. Currently, many courses include the study of ethical frameworks for the promotion of ethical AI systems in education. This is the case in [32] in primary school education, or [31] in high school education, which implemented Developed AI Literacy (DAILy), incorporating ethics into the teaching of AI, with the aim of raising awareness among students about the importance of ethical aspects within this discipline. According to the general consensus, there are five fundamental principles for ethics in AI [35,36]:

- Beneficence: technology must be beneficial to humanity;
- No maleficence: technology must be prevented from causing harm, particularly focused on privacy violations;
- Autonomy: technology should empower people's autonomy, allowing them to choose what they delegate to technology and what not;
- Justice: technology must be fair, avoiding any kind of discrimination;
- Explicability: technology must be transparent, understandable, and interpretable by people, so that they understand the logic that the AI has followed, and are thus able to identify and correct errors.

Of the five points cited [34] highlights autonomy and explicability as elements that must play a central role. The integration of ethics in the teaching of AI is a fundamental element to increase the awareness of the population about the impact of AI in industry and society [31].

Regarding ethical aspects in the field of education, our aim is to achieve an AI model capable of identifying when a student needs help, avoiding gender discrimination as much as possible. To achieve this goal, we trained two models and compared the results: the first model was trained with the gender attribute, and the second model was trained without the gender attribute.

With regard to the ethical aspects of our research, we aim to:

- Use an ethical approach to the user experiences conducted to collect data for our research, taking into account the standards of the Universidad Nacional de Educación a Distancia (UNED) Research Ethics Committee and the Guide for research ethics committee members of the Council of Europe (<https://www.coe.int/en/web/bioethics/guide-for-research-ethics-committees-members>, accessed on 27 November 2023).
- Respect the five standard fundamental ethical principles to guide the development of AI applications, with a focus on developing affective robot tutors that support autonomous learning in a transparent way by acting as a support and not as a substitute for the human teacher (avoiding dehumanisation of teaching), and that do not act in a discriminatory way and take into account the idiosyncrasies of each student.

As far as the work presented in this article is concerned, our aim is to achieve an AI model capable of identifying when a student needs help, avoiding gender discrimination as much as possible. As a first step towards achieving this goal, to identify any gender bias, we trained two models and compared the results: the first with the gender attribute, and the second without.

It is not our intention in this paper to address in depth the elimination of gender bias in intelligent educational systems for learning computational skills, but to take a first step in that direction by highlighting the existence of such bias and the influence of considering the gender attribute as a training parameter in the models. We recognise the difficulty of addressing this multifaceted issue.

There is a wealth of academic literature, as well as various reports from international organisations, on the gender digital divide, which is of great concern at this historical moment of the digital revolution [37–43]. Although progress has been made in reducing the gender gap in education, the fact remains that women’s participation in ICT is not growing in proportion to women’s education in general. This is particularly true in the case of AI, whose disruptive potential makes women’s participation essential. Occupational segregation by gender is increasing, so that in Spain, for example, only 0.5% of AI professionals are women.

Among the causes of this gap, there is agreement in the literature on the importance of the educational factor. Ref. [37] points to girls’ lack of self-esteem/self-confidence/self-sufficiency, ignorance of their own needs/interests, imposed codes of conduct/stereotypes, fewer learning opportunities for girls inside and outside educational settings, and lack of recognition that they are capable of the same achievements but have different ways of learning. Many studies agree on the need to promote gender equality through education, culture, and communication. Educational institutions need to identify gender barriers in the STEM ecosystem and promote digital literacy for women at all levels of education with a proactive and personalised gender approach.

2.2. User Experience with Scratch

The purpose of the user experience is to collect data to assist in the development of an intelligent model for identifying, through the use of machine learning techniques and the block-based programming language distance calculation, when a primary school student has need of pedagogical assistance.

Since our ultimate goal is to help elementary students in a real learning environment while performing Scratch exercises, the data collection must be done in a real learning environment. To this end, we reached a collaboration agreement with two learning centers, and chose 10 Scratch exercises, designed and validated by primary school technology teachers, of varying degrees of difficulty for students to perform, with the goal of learning the basics of programming and computational thinking. The realization of the user experiences took a total of three months, and a total of four teachers specialized in teaching technology to primary school students participated in them.

Since not all students have the same knowledge of Scratch, they will need help at different points of the process, so the model must be customized according to the knowledge level. To evaluate this, we also designed three level assessments, also of varying degrees of difficulty, that must be completed prior to executing the aforementioned exercises. These assessments measure the student's prior knowledge of Scratch.

The students who participated in these user experiences were very aware that during the experiment they should ask for help only when they needed it. In order for students to be able to ask for help, and for this request to be recorded in the database, modifications were made to the Scratch project to enable a button for students to ask for help and to display hints about the exercise they were currently working on. In order to clarify the procedure of the sessions, an explanatory session was held prior to the user experiences to explain the purpose of the research and the importance of requesting help when it was really needed. In addition, the teachers supervised the development of these experiences and also validated the students' requests for help.

As described in Section 2.1, the application of AI ethical principles must play a central role in education. In this study, in addition to building a model capable of identifying when a student needs help, we relied on ethical methods.

In user experiences, two of the most critical ethical elements are biases (discrimination by gender and by minority groups), and transparency. To avoid any bias due to gender or minority status, participation in user experiences was offered to any elementary school student in the collaborating centers, without any kind of discrimination based on gender, race, or social group. Despite this, we are aware that given the low number of participants in the user experiences, there is no equity in the representation of different races, social groups, or minority groups. In this study we have just focused on gender, trying to ensure that both genders are represented as equally as possible. In addition, to provide maximum transparency, parents or legal guardians and students who wanted to participate in the experiences received an informed consent form that included a description of the benefits of the study and its possible benefits and risks for the student, a detailed description of the study itself and of the data protection policy, and finally a channel through which to ask questions. Parents or legal guardians can request at any time the withdrawal of a student from the experience, as well as the desire to keep or delete the data already collected. Likewise, a certificate was provided by the Research Ethics Committee of the UNED.

In the case of this research, the Research Ethics Committee of the UNED required the declaration of a "data treatment activity", which in our case referred to the automated treatment of academic data of individuals under the age of 14. This declaration implies the commitment to keep the data strictly for the time necessary to fulfill the purpose for which it was collected and to determine any responsibilities that may arise from this purpose and from the processing of the data.

Another stipulation of the Ethics Committee is that the user experience does not cause any harm or discomfort to the participants. In this research, the user experiences were carried out during school hours, in the case of the collaborating school Los Peñascales (<https://www.colegiolospenascales.com>, accessed on 27 November 2023), and in the case of the company Rockbotic (<https://rockbotic.com>, accessed on 27 November 2023), during already scheduled after-school courses. Furthermore, they could be carried out both in person and remotely, since the tool was accessible via the web. The competences exercised during the user experiences were those of the training program that the students were following, and the duration of each session varied between 25 min and 1 h, coinciding with the duration of the usual classes.

In addition, at the beginning of the experience at each center, the students received an introductory talk about the research project in which they would be participating. The talk explained the details of the scientific method, the models of artificial intelligence, and the potential of this project within artificial intelligence and education. At the end of the talk, all the students' questions were answered. The opportunity to participate in a research project in the field of artificial intelligence contributed to the motivation of participating students

and their confidence in intelligent technologies, as well as to the development of their computational thinking skills, which could be very useful for their professional future.

The Scratch tool was adapted to facilitate the recording of student profiles and the actions performed during the completion of the exercises through keyboard and mouse interaction. It was designed to be as easy to use as possible and was accessed through a login with a username, password, and student number. The student numbers were assigned by the teacher. In addition, at the end of the sessions, the codes were randomly renamed by an automatic system to ensure that there was no record of the origin of the data.

The Research Ethics Committee also supervised the data-storage procedures. All data collected was stored exclusively in a remote database (never on the local computers used by the students in the schools or academies) in order to be able to retrieve and analyze it throughout the analysis phase. The server hosting the database also hosted the web platform used by the teachers and students. Once the research is completed, the data will be deleted from the server and stored on a hard drive in the researchers' custody.

The committee also requires a commitment that any publications resulting from the research will avoid any explicit reference to student data that could uniquely identify students through cross-referencing. Therefore, these future publications will present only anonymized, aggregated data and will never include images of students, unless their legal guardians have given their explicit consent to do so. The data collected will only be used for the purposes of this research and will not be shared with third parties. Finally, the data will only be kept for the legal period and for the time necessary to fulfill the purposes of the study. Finally, on the recommendation of the Committee, the researchers commit themselves to communicate the results of the research in which they participate to the collaborators.

A summary of the user experience process is described below:

- We first sent the informed consent to the parents or legal guardians of the students. Before the students participated, both parents or legal guardians had enough time to consider participation;
- Then, the students whose parents had authorized their participation started by filling out a questionnaire.
- After that, they started the experience by carrying out the assessments. During them, they could not request any kind of help to evaluate the real prior knowledge of Scratch, but they could exit from the assessment at any time. Once the students had completed or exited from all the assessments, the knowledge level was set;
- Finally, the students could start performing the Scratch exercises in their preferred order. In those exercises they could ask for help and also switch to another exercise at any time.

From the first interaction with Scratch up to the end of the exercises, all the interactions were recorded in a database with the aim of analyzing them later. All this information was made anonymous by previously giving a random number to each participant. The stored information of this experiment is hereunder listed:

- Student data:
 - Student number: the student number was randomly assigned to the student;
 - Gender: provided by the student in the opening questionnaire and stored in numerical format (1 for boy and 2 for girl);
 - Mother tongue: provided by the student in the opening questionnaire. It is a boolean field that indicates if the student's mother tongue is Spanish or not. Given that the exercises and the user experiences were carried out in Spanish, we think that the variable "language" may affect whether a student understands the statements of the exercises and the teacher's explanations. Therefore, this variable could influence the number of help requests made by the students;
 - Age: provided by the student in the opening questionnaire;
 - Competence: set by the aforementioned starting assessments.

- Exercise data:
 - Name: exercise name;
 - Description: exercise description;
 - Skills: each exercise may help to improve students' CT skills to different degrees. These skills and their degrees (which are numerical values from 0 to 10) were set by experts;
 - Evaluation: indicates if the exercise is an evaluation level exercise or not.
- Solution distance data: family distance, element distance, position distance, input distance, total distance (explained in Section 2.3).
- Interaction data:
 - Date time: date and time of the interaction;
 - Request help: indicates if the student requested help or not;
 - Seconds help open: when a student requested help, this parameter indicates how many seconds the help popup was open before the student closed it;
 - Last login: date and time of the student's last login.
- Workspace data:
 - Elements: block elements that the student currently has in their workspace. This information is then used to calculate the distance between the student workspace and the solution.

2.3. Distance Calculation for Block-Based Programming Languages

In programming learning activities there may be multiple solutions for a single task. The learning progress can be measured by calculating the distance between the student's workspace and the best candidate solution, i.e., the solution that is closest to the student's workspace according to the distance measure. Below we present a normalized version of the distance function presented in [6].

The elementary blocks of a block-based programming language are divided into families (Figure 5) that group a common functionality, as we can see from example in Figure 6. Elementary blocks can be combined into higher-level blocks. These blocks can also have an input, and each input can contain one or many fields. When designing a program in a block-based programming language, blocks can be combined into a single block for a sequential execution, as shown in Figure 7, or in different groups for a parallel execution, as illustrated in Figure 8.



Figure 5. Block families in Scratch.

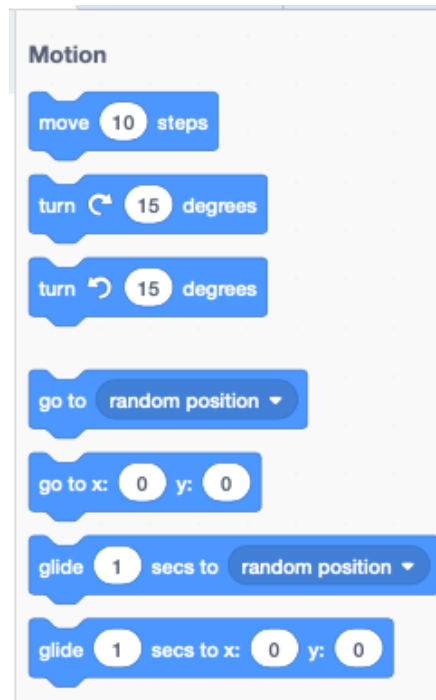


Figure 6. Movement family blocks in Scratch.



Figure 7. Single element block in Scratch.



Figure 8. Multiple blocks of elements in Scratch.

In summary, the dimensions that influence the distance in a block-based programming language are:

1. Block family;
2. Block;
3. Block position;
4. Block inputs values.

2.3.1. Block Family Distance Calculation

Let F_w be the block family set of the student's workspace, and F_s the block family set of the exercise solution. The difference between both family groups is fixed as the symmetric difference between the set F_w and the set F_s .

We consider the correct family set as the block families in the student's workspace, that are also present in the exercise solution; and the incorrect family set as the block families in the student's workspace that are not present in the exercise solution. The distance calculation is carried out as follows:

Let: $F_i = \{family \mid family \in F_w \Delta F_s\}$, the incorrect family set.

If x is a set of elements, and $n(x)$ is the function returning the number of elements in x , the distance between the families of the solution and the families of the student's workspace is defined as:

$$d_f = n(F_i) \quad (1)$$

2.3.2. Block Distance Calculation

As mentioned previously, blocks are grouped by families, which means that blocks are elements of these families. We consider the correct block set as the block elements in the student's workspace, that are also present in the exercise solution; and the incorrect block set as the block elements in the student's workspace that are not present in the exercise solution.

Let:

B_w , the set of blocks in the student's workspace.

B_s , the set of blocks in the solution.

$B_i = \{block \mid block \in B_w \Delta B_s\}$, the incorrect block set.

The distance calculation between the solution block set and the student's workspace block set is defined by:

$$d_b = n(B_i) \quad (2)$$

2.3.3. Block Position Distance Calculation

As noted above, blocks can be organized sequentially, establishing an order for their execution. We must check that the correct blocks of the student's workspace are executed in the proper order. To do this, we compare the position of each block in the student's workspace with the position of the same block in the solution.

Let $p_s(block)$, $p_w(block)$, be the positions of the blocks in the solution and in the workspace, respectively, where we assume that the position is a natural number higher than zero.

The distance calculation for the correct block sets is carried out by subtracting from the position of each block in the student's workspace, the position of that same block in the solution. We use the term correct block position for the position of a block in the student's workspace that matches the position of the same block in the exercise solution; and the term incorrect block position for the position of a block in the student's workspace that does not coincide with the position of the same block in the exercise solution. Therefore, the positional distance of the correct block sets is calculated as follows:

$$d_{pc} = \sum_{block \in B_s \cap B_w} |p_s(block) - p_w(block)| \quad (3)$$

The positional distance of the incorrect block sets B_i is also taken into account in a similar way. To do this, the value of each position of the blocks belonging to B_i is added, as follows:

$$d_{pi} = \sum_{block \in B_s - B_w} p_s(block) + \sum_{block \in B_w - B_s} p_w(block) \tag{4}$$

The final positional distance will be the sum of the correct block position distance and the incorrect block positional distance:

$$d_p = d_{pc} + d_{pi} \tag{5}$$

2.3.4. Block Input Values Distance Calculation

As mentioned in Section 2.3, blocks can have alphanumerical and numerical inputs. The calculation of the input distance between the input values is performed in two steps: In a first step, we compare the input values of the correct blocks in the student’s workspace and the solution, and in a second step, we add the input values for the incorrect blocks.

Because there can be both alphanumerical and numerical inputs, we defined two different ways to calculate the distance depending on the input type.

Let $i_s(block)$, $i_w(block)$ be the inputs of a block in the solution and in the workspace, respectively. The calculation of the input-value distance between the student’s workspace and the solution in the correct set of blocks is done in two different ways. In the case of alphanumerical inputs, for different values we simply consider that the distance is 1. In the case of numerical inputs, we calculate the numerical difference between the input value of the student’s workspace and that of the solution.

$$d_{ic} = \sum_{block \in B_s \cap B_w} \begin{cases} \frac{|i_s(block) - i_w(block)|}{|i_s(block)| + |i_w(block)|}, & \text{if the input is numerical} \\ 1, & \text{if the input is alphanumerical, and } i_w(block) \neq i_s(block) \end{cases} \tag{6}$$

As in Equation (6), the calculation of the input value distance in the incorrect block set is also done in two different ways as follows:

$$d_{iis} = \sum_{block \in B_s - B_w} \begin{cases} i_s(block), & \text{if the input is numerical} \\ 1, & \text{if the input is alphanumerical} \end{cases} \tag{7}$$

$$d_{iiw} = \sum_{block \in B_w - B_s} \begin{cases} i_w(block), & \text{if the input is numerical} \\ 1, & \text{if the input is alphanumerical} \end{cases} \tag{8}$$

The final input value distance is calculated as the sum of the input value distance in the correct block set (d_{ic}), and the input-value distance in the incorrect block set ($d_{iis} + d_{iiw}$):

$$d_i = d_{ic} + d_{iis} + d_{iiw} \tag{9}$$

2.3.5. Total Distance

As outlined above, there is an order of relevance in the parameters when determining the distance between the workspace and the solution.

So, the total distance is calculated as the weighted sum of Equations (1), (2), (5) and (9).

The total distance between the student’s workspace and the solution is defined as follows:

$$d_t = d_f + \left(\frac{1}{2} \cdot d_b\right) + \left(\frac{1}{4} \cdot d_p\right) + \left(\frac{1}{8} \cdot d_i\right) \tag{10}$$

2.4. Help Identification Model Development

Once the user experiences defined in Section 2.2 were completed, we used the collected data to develop a model that can determine when learners need help. This was achieved through feature selection and analysis of the data, and development and training of the model, as shown in Figure 9.

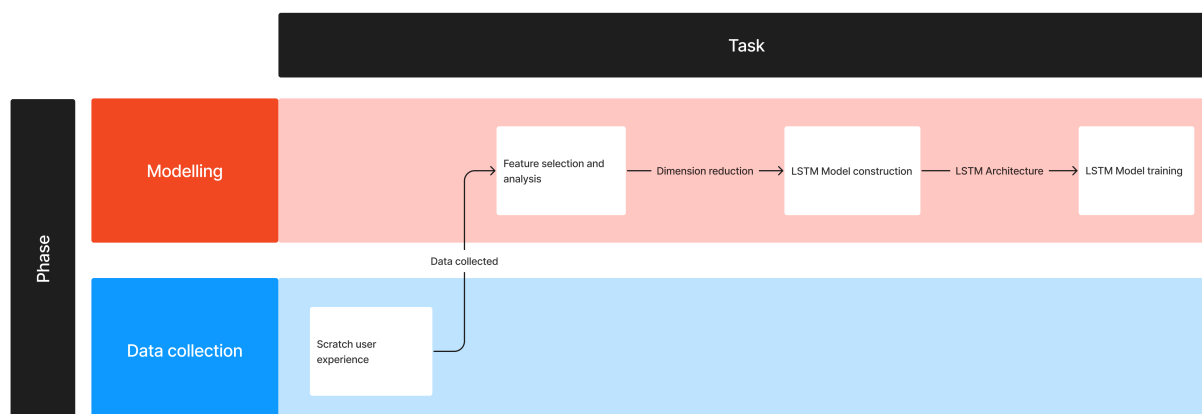


Figure 9. LSTM model building flow.

2.4.1. Feature Selection and Analysis

To the list of features described in Section 2.2, we added the calculated feature `total_seconds`. This feature measures the time between the first interaction that occurs during the completion of an exercise, and the current interaction. The purpose of this feature is to know how much time the student has spent on the current exercise.

Feature selection is performed in two stages: the first stage consists of eliminating those features that do not improve the precision of the model, and the second stage consists of removing those characteristics that can pose an ethical risk. The first stage of feature selection was performed by combining an initial manual selection, based on our own knowledge of the features, with an automatic feature selection. In the manual selection, we discarded the following features: `exercise_is_evaluation`, `exercise_name`, `exercise_description` and `last_login`. In the second stage, `student_gender` was manually removed, because it can cause the system to take into account the student's gender to give more or less support, leading to a bias. At this stage, the effect on the accuracy of the model of removing the gender was analysed. For the automatic feature selection, we implemented a filter method based on Pearson correlation. This method filters out all features that are not relevant by using a correlation matrix. In our study, we filtered all the features with a correlation above or equal to 0.85. The correlation matrix heatmap result is shown in Figure 10.

Figure 10, shows that gender is slightly correlated with the attributes `exercise_skill_logical_thinking`, `exercise_skill_flow_control`, `exercise_skill_user_interactivity` and `exercise_skill_information_representation`. It also shows that the mother tongue attribute has almost no correlation with any attribute, and that there is also a correlation between the student competencies and the different skills. Once features were filtered, we analyzed them by calculating the feature impact, using the SHapley Additive exPlanations (SHAP) framework [44]. The results show which features are driving model decisions, and they were obtained as follows:

1. Takes a sample of records from the training data;
2. Computes SHAP values for each record in the sample, generating the local importance of each feature in each record;
3. Computes global importance by taking the average of $\text{abs}(\text{SHAP values})$ for each feature in the sample;
4. Normalizes the results.

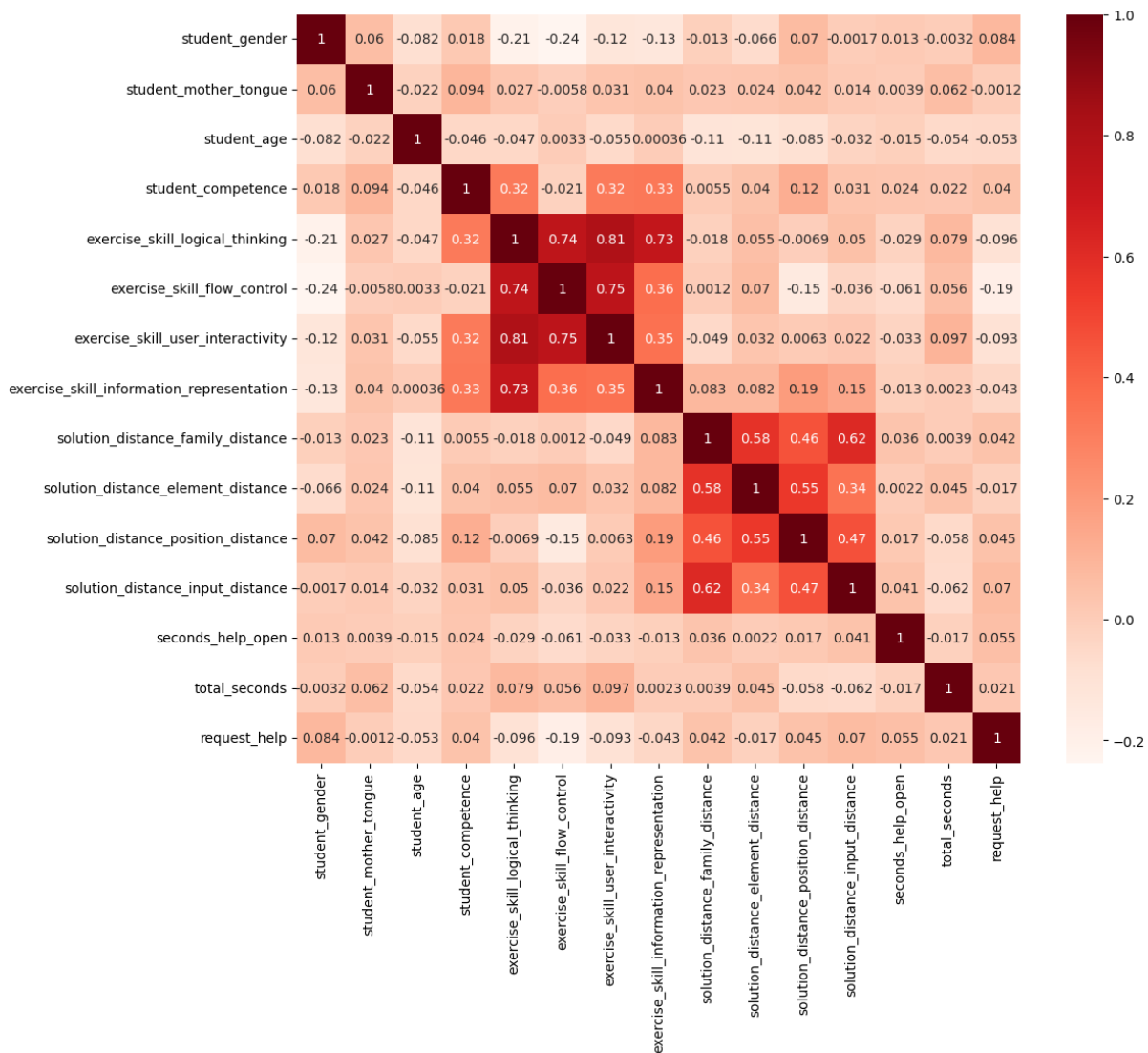


Figure 10. Correlation Matrix Heatmap.

Figure 11 shows the feature impact results. Notice that total_seconds, the different skills, and the student_competence have a big impact on the model, which means that these variables are important when determining if a student needs help. Additionally, the gender and student_mother_tongue attributes have a very low impact.

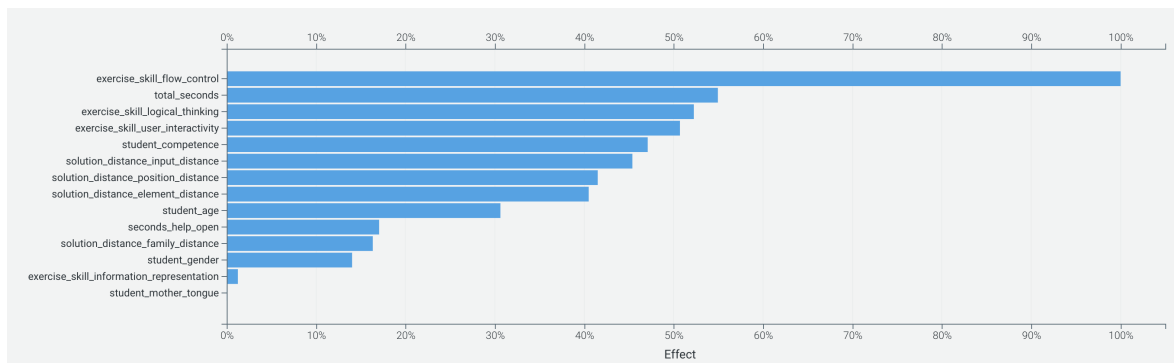


Figure 11. Feature impact.

2.4.2. Model Construction

For the construction and training of the model, the dataset in the correlation matrix shown in Figure 10 was used. Since we are looking for the model to predict whether the student needs help or not, we use the variable “request_help” as the target variable, and we look for the output of the model to be a boolean value.

Given that all the student actions are recorded over time and reflect when a student needs help, the model implementation is based on a RNN, making use of LSTM. Since we have many inputs (interactions in Scratch over time), and just one output (whether the student needs help or not), we applied a many-to-one architecture, as shown in Figure 4. In order to identify the architecture model that best fits the problem and the current data, we conducted some experiments with different parameters. In these experiments, we combined the number of LSTM layers (one or two LSTM layers), and none, or a few dropout layers (with a value of 0.5). Taking into account that the model must identify if a student needs help or not in a particular moment, we were faced with a binary problem, hence, we implemented a binary cross entropy loss function with an adam optimizer in each experiment. The different model architectures we tested are shown in Figures 12 and 13.

The following parameters were used for the training of the different architectures:

- LSTM units (in each LSTM layer): 256;
- Activation function of the LSTM layers: sigmoid;
- Return sequences: true;
- Dropout layer value (in each dropout layer if applicable): 0.5;
- Dense layer units: 1;
- Dense layer activation function: sigmoid;
- Dense layer dropout function: binary_crossentropy;
- Dense layer optimizer: adam;
- Dense layer metrics: binary_accuracy;
- Training batch size: 1;
- Number of training epochs: 50;
- Percentage of training data: 70%;
- Percentage of validation data: 30%.

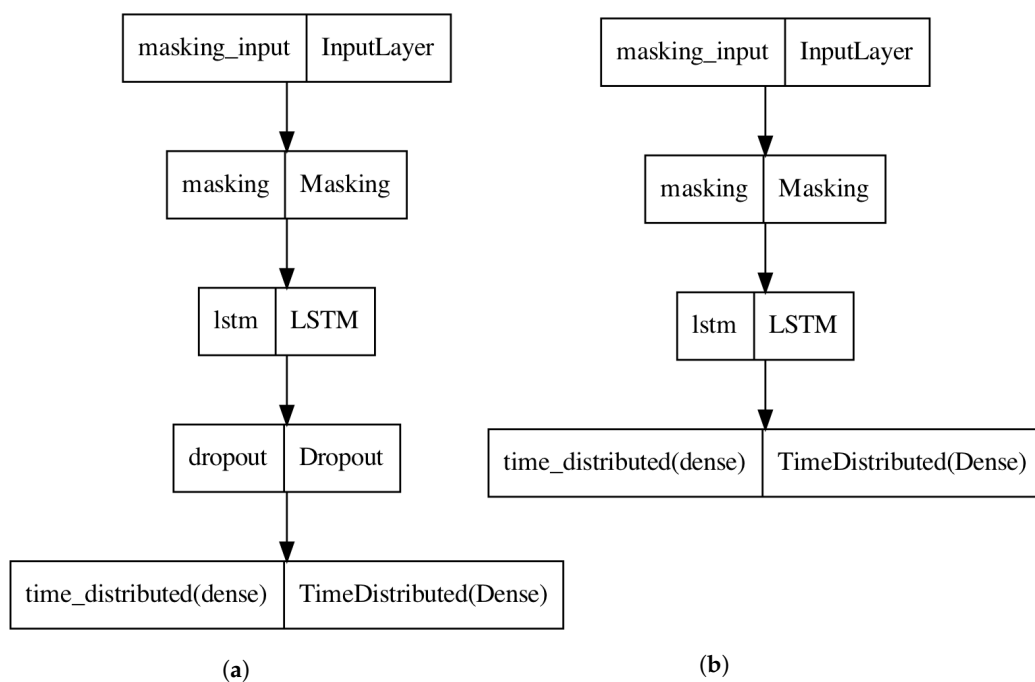


Figure 12. Architectures with 1 LSTM layer. (a) 1 LSTM layer with dropout layer; (b) 1 LSTM layer with no dropout layers.

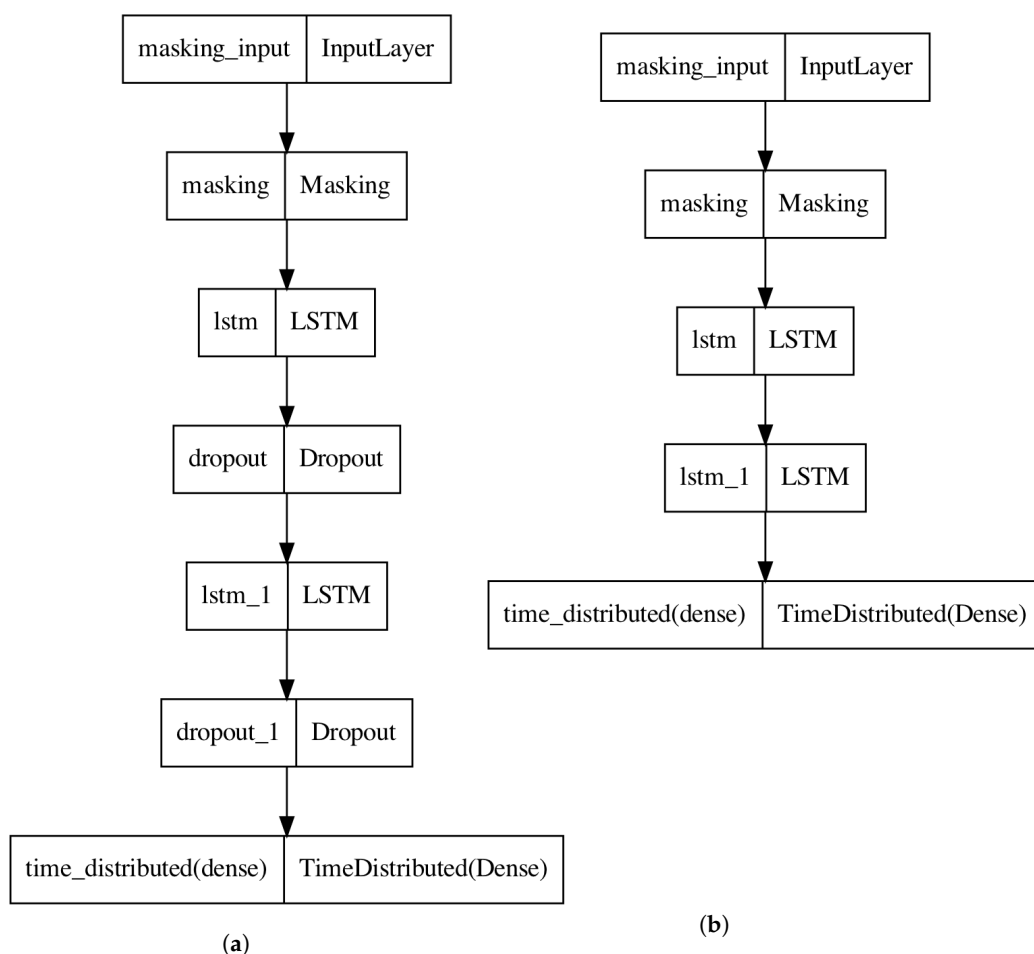


Figure 13. Architectures with 2 LSTMs layers. (a) 2 LSTMs layer with dropouts layer; (b) 2 LSTMs layer with no dropout layers.

During the user experiences described in Section 2.2, we collected 11,790 interactions, taken from 82 students, who generated 695 help requests. In Table 1 we present the data collected.

Table 1. Data collected.

Interactions	Help Requests	Students	Average Age	Girls	Boys
11,790	695	82	13.3	30	52

Because each user performs many interactions in a single exercise and day, grouping these interactions by user, date, and exercise, we finally obtained 316 time series. Therefore, we have an average of 2.19 help requests per time series, and 37.31 interactions per time series. Finally, for the training of our model we used 70% of the time series, and 30% for validation, as shown in Table 2.

Table 2. Dataset distribution.

Phase	Time Series	Help Requests	Students	Average Age	Girls	Boys
Training	221	450	52	13.5	16	36
Validation	95	245	30	13	14	16

Finally, in order to analyze whether gender is a bias in the models, it was decided to train the four architectures shown in Figures 12 and 13 twice. In the first training run the dataset includes the gender of the students, while in the second run it does not.

3. Results

After executing the different training runs (including and excluding gender) of the four architectures shown in Figures 12 and 13, the results were divided into two different tables. Table 3 shows the results of the training model including the gender. In the majority of the architectures we exceeded 99% accuracy. Table 4 shows the results of the training model without the gender feature. It can be observed that in all the architectures we exceeded 91% accuracy.

Table 3. Model training results with gender feature.

LSTM Layers	Dropout Layers	Loss	Binary Accuracy	Val Loss	Val Binary Accuracy
1	False	0.017185	0.94459	0.0071762	0.99144
1	True	0.019589	0.94277	0.0077519	0.99249
2	False	0.017146	0.94621	0.0078189	0.98975
2	True	0.020001	0.94273	0.0085079	0.99266

Table 4. Model training results without gender feature.

LSTM Layers	Dropout Layers	Loss	Binary Accuracy	Val Loss	Val Binary Accuracy
1	False	0.015799	0.9574	0.04275	0.91122
1	True	0.013381	0.95972	0.047054	0.91095
2	False	0.015337	0.95727	0.045792	0.91122
2	True	0.013264	0.94273	0.045034	0.91116

4. Discussion

In this research, we used the distance calculation presented in [6], and we implemented a first prototype model for predicting when a primary school student will need help while performing block-based programming exercises.

One important motivation for this model is to ensure that the students will be properly helped when they are stuck. To achieve this goal, the model must be able to correctly identify the moment when a student has a mental block while he/she is carrying out Scratch exercises.

During the user experiences, despite the fact that we sought to have an equal number of boys and girls, and to have a significant number of students whose mother tongue is different from Spanish, the reality is that there was an imbalance in the gender of the participants, as well as a large majority of students whose mother tongue is Spanish. User experiences were designed taking into account ethical aspects such as transparency and avoiding any kind of discrimination. In this first approach, we focused on trying to have a significant number of boys and girls, and studying the influence of gender, being mindful of its relevance to learning in the field of computer science, as studies like [45] show. In this sense, we intend to conduct larger experiments in the future. Even with such data, we consider that this first approach, which includes the data collection with the distance-calculation attribute and the predictive help model, is relevant, because there is no mention in the literature of repositories containing user interactions with block-based programming software, and solution distance metrics may help to build the suggested model.

Looking at the calculated impact of the different features in Figure 11, it can be seen that the most important features are the student's skills, the total time, and the distance calculations. This means that the distance calculation introduced in Section 2.3.2 is a relevant feature that helps to improve the accuracy, reaching more than 40% impact. Our predictive model trained with the data collected from user experiences in a real learning environment obtained 99.2% accuracy if we consider the gender attribute in training, and 91.1% accuracy if we exclude the gender attribute in training. Despite the impact of gender being relatively low, as shown in Figure 11, it can be seen in Tables 3 and 4 that this attribute

influences the accuracy of the model by around 8%. All this seems to indicate that gender may be a relevant attribute. Also, in Figure 10, gender is inversely correlated with the different skills, which means that, in general, in our experiment girls had a lower score in these skills. The small amount of data collected, and the high precision of the model when gender is present, makes us suspect that the model is overfitting, leading to it not properly identifying when a student requires help, due to gender, and this fact could cause frustration and loss of interest in the subject. In addition, we believe that in our experiment variables correlated with gender, such as learning style, additional skills, etc., were lacking.

5. Conclusions and Future Work

The model that has been presented has 99.2% accuracy taking gender into account and 91.1% accuracy without taking gender into account. This could indicate that the gender present in the dataset may be introducing an overfitting into the model.

By including the gender variable, given the lower representation of the female gender in the student body participating in our experiments, the trained models will not generalise adequately and their predictions will be biased by the specific characteristics of the participating girls, who had low computational thinking skills in the experiments conducted. By removing the gender variable, predictions of competence in completing the exercises would correlate with computational skills without gender bias.

We can conclude from the above that there are hidden variables, determinants of low computational competence, that it would be interesting to make explicit in our models; variables that, in the current context, define women's idiosyncrasies but do not imply a lower real capacity of women for computational thinking. These variables would be related, for example, and as indicated in [37], to having been taught computational skills through ineffective and demotivating instructional designs.

From what can be observed in this first approximation, students' skills, total time, and distance calculations could be relevant in determining whether a primary school student needs help while performing block-based programming exercises.

Since the distance calculations shown are common to all block-based programming languages, and the results are not specific to Scratch, nor to elementary education, this same study can be replicated with any other block-based programming tool and for any educational level.

Therefore, this work could serve as a starting point to investigate models to identify when a primary school student (even at other levels) gets stuck during block-based programming exercises. This may allow for more personalized interventions when they are really needed, through tutor interventions, or help systems, improving the students' learning experience and increasing their motivation.

In future work, we intend to perform new user experiences to collect more data with more participants; we will also strive to have an equal number of boys and girls, and to explore feature selection methods, as well as RNN architectures to improve the accuracy of the model. In addition, for these models, more performance metrics, such as precision, recall, and f1 score, as well as statistical techniques to measure the loss of accuracy resulting from shuffling, will be taken into account. We will also refine and validate the model in a real learning environment, and a more in-depth analysis of the impact of the different attributes in determining whether a student needs help will be conducted. Finally, this work will also support the ARTIE project [13], introduced in Section 1, so that the system will be able to automatically identify when a primary school student needs help during block-based programming exercises, and send the appropriate pedagogical intervention to the affective tutor robot.

Author Contributions: Conceptualization, L.E.I.C.; Methodology, L.E.I.C.; Software, L.E.I.C.; Validation, L.E.I.C., Á.M.R. and F.d.I.P.L.; Formal analysis, L.E.I.C.; Investigation, L.E.I.C.; Resources, Á.M.R. and F.d.I.P.L.; Writing—review & editing, Á.M.R. and F.d.I.P.L.; Visualization, L.E.I.C.; Supervision, Á.M.R. and F.d.I.P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MCIN/AEI/10.13039/501100011033 and, as appropriate, by “ERDF A way for making Europe”, by the “European Union” or by the “European Union Next Generation EU/PRTR” grant number PID2020-115220RB-C22.

Institutional Review Board Statement: All subjects gave their informed consent for inclusion before they participated in the study. The study was conducted in accordance with the Declaration of Helsinki and the protocol was approved by the Ethics Committee of UNED (10/02/2022 and Project identification code: 1-ETSIINF-2021).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: On request: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

Acknowledgments: The authors also wish to thank the staff of Rockbotic and Los Peñascales primary school of Las Rozas (Madrid) for their collaboration, in particular David Moreno from Rockbotic and Ignacio Martín from Los Peñascales, the regular teachers of the students who took part in the experiments, as well as the children who collaborated in this research during the sessions. Finally, we want to express our gratitude to Hipoo, the company that provided us with the tools to train the model.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ARTIE	Affective Robot Tutor Integrated Environment
ASTs	Abstract Syntax Trees
AI	Artificial Intelligence
CS	Computer Science
CT	Computational Thinking
CNN	Convolutional Neural Network
DAILY	Developing AI Literacy
DKT	Deep Knowledge Tracing
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
RNN	Recurrent Neural Network
SHAP	SHapley Additive exPlanations
TED	Tree Edit Distance
TID	Tree Inherited Distance
TSC	Time Series Classification
UNED	Universidad Nacional de Educación a Distancia

References

- Uddin, I.; Imran, A.S.; Muhammad, K.; Fayyaz, N.; Sajjad, M. A Systematic Mapping Review on MOOC Recommender Systems. *IEEE Access* **2021**, *9*, 118379–118405. [\[CrossRef\]](#)
- Jiang, B.; Li, Z. Effect of Scratch on computational thinking skills of Chinese primary school students. *J. Comput. Educ.* **2021**, *8*, 505–525. [\[CrossRef\]](#)
- Fagerlund, J.; Häkkinen, P.; Vesisenaho, M.; Viiri, J. Computational thinking in programming with Scratch in primary schools: A systematic review. *Comput. Appl. Eng. Educ.* **2021**, *29*, 12–28. [\[CrossRef\]](#)
- Howard, N.R. How Did I Do?: Giving learners effective and affective feedback. *Educ. Technol. Res. Dev.* **2021**, *69*, 123–126. [\[CrossRef\]](#)
- Cavalcanti, A.P.; Barbosa, A.; Carvalho, R.; Freitas, F.; Tsai, Y.-S.; Gašević, D.; Mello, R.F. Supporting Teachers Through Social and Emotional Learning. *Comput. Educ. Artif. Intell.* **2021**, *2*, 100027. [\[CrossRef\]](#)
- Cuadrado, L.-E.I.; Riesco, A.M.; de la Paz López, F. A first-in-class block-based programming language distance calculation. In *Lecture Notes in Computer Science; Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 13259, pp. 423–432. [\[CrossRef\]](#)
- Kamil, K.; Sevimli, M.; Aydin, E. An investigation of primary school students’ self regulatory learning skills. In *Proceedings of the 7th International Conference on Education and Education on Social Sciences, Taipei, Taiwan, 15–17 June 2020*. [\[CrossRef\]](#)

8. Wiggins, J.B.; Fahid, F.M.; Emerson, A.; Hinckle, M.; Smith, A.; Boyer, K.E.; Mott, B.; Wiebe, E.; Lester, J. Exploring Novice Programmers' Hint Requests in an Intelligent Block-Based Coding Environment. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE 2021), Virtual Conference, 13–20 March 2021; pp. 52–58. [\[CrossRef\]](#)
9. Alsharef, A.; Aggarwal, K.; Kumar, M.; Mishra, A. Review of ML and AutoML Solutions to Forecast Time-Series Data. *Arch. Comput. Methods Eng.* **2022**, *29*, 5297–5311. [\[CrossRef\]](#)
10. He, K.; Gao, K. Analysis of Concentration in English Education Learning Based on CNN Model. *Sci. Program.* **2022**, *2022*, 1489832. [\[CrossRef\]](#)
11. Jarbou, M.; Won, D.; Gillis-Mattson, J.; Romanczyk, R. Deep learning-based school attendance prediction for autistic students. *Sci. Rep.* **2022**, *12*, 1431. [\[CrossRef\]](#)
12. Holmes, W.; Porayska-Pomsta, K.; Holstein, K.; Sutherland, E.; Baker, T.; Buckingham, S.; Santos, S.O.C.; Rodrigo, M.T.; Cukurova, M.; Bittencourt, I.I.; et al. Ethics of AI in Education: Towards a Community-Wide Framework. *Int. J. Artif. Intell. Educ.* **2022**, *32*, 504–526. [\[CrossRef\]](#)
13. Cuadrado, L.-E.I.; Riesco, A.M.; de la Paz López, F. ARTIE: An Integrated Environment for the Development of Affective Robot Tutors. *Front. Comput. Neurosci.* **2016**, *10*, 77. [\[CrossRef\]](#)
14. Cuadrado, L.-E.I.; Riesco, A.M.; de la Paz López, F. FER in primary school children for affective robot tutors. In *From Bioinspired Systems and Biomedical Applications to Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 461–471. [\[CrossRef\]](#)
15. Ng, O.L.; Cui, Z. Examining primary students' mathematical problem-solving in a programming context: Towards computationally enhanced mathematics education. *Zdm -Math. Educ.* **2021**, *53*, 847–860. [\[CrossRef\]](#)
16. López, J.M.S.; Otero, R.B.; García-Cervigón, S.D.L. Introducing robotics and block programming in elementary education. *Rev. Iberoam. Educ. Distancia* **2020**, *24*, 95. [\[CrossRef\]](#)
17. Jen-I, C.; Mengping, T. Meta-analysis of children's learning outcomes in block-based programming courses. In *HCI International 2020—Late Breaking Posters*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 259–266. [\[CrossRef\]](#)
18. Demirkiran, M.C.; Hocanin, F.T. An investigation on primary school students' dispositions towards programming with game-based learning. *Educ. Inf. Technol.* **2021**, *26*, 3871–3892. [\[CrossRef\]](#)
19. Wang, J. Use Hopscotch to Develop Positive Attitudes Toward Programming For Elementary School Students. *Int. J. Comput. Sci. Educ. Sch.* **2021**, *5*, 48–58. [\[CrossRef\]](#)
20. Jo, Y.; Chun, S.J.; Ryoo, J. Tactile scratch electronic block system: Expanding opportunities for younger children to learn programming. *Int. J. Inf. Educ. Technol.* **2021**, *11*, 319–323. [\[CrossRef\]](#)
21. Obermüller, F.; Heuer, U.; Fraser, G. Guiding Next-Step Hint Generation Using Automated Tests. *Annu. Conf. Innov. Technol. Comput. Sci. Educ.* **2021**, *1293*, 220–226. [\[CrossRef\]](#)
22. Fahid, M.F.; Tian, X.; Emerson, A.; Wiggins, J.B.; Bounajim, D.; Smith, A.; Wiebe, E.; Mott, B.; Boyer, K.E.; Lester, J. Progression trajectory-based student modeling for novice block-based programming. In Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization (UMAP 2021), Utrecht Netherlands, 21–25 June 2021; pp. 189–200. [\[CrossRef\]](#)
23. Mlinaric, D.; Milasinovic, B.; Mornar, V. Tree Inheritance Distance. *IEEE Access* **2020**, *8*, 52489–52504. [\[CrossRef\]](#)
24. Yu, W.; Yong, I.; Mechefske, K.C. *Analysis of Different RNN Autoencoder Variants for Time Series Classification and Machine Prognostics*; Academic Press: Cambridge, MA, USA, 2021; Volume 149. [\[CrossRef\]](#)
25. Pudikov, A.; Brovko, A. Comparison of LSTM and GRU Recurrent Neural Network Architectures. In *Recent Research in Control Engineering and Decision Making*; Springer: Cham, Switzerland, 2021; Volume 337, pp. 114–124. [\[CrossRef\]](#)
26. Affonso, F.; Rodrigues, T.M.; Pinto, D.A.L. Financial Times Series Forecasting of Clustered Stocks. *Mob. Netw. Appl.* **2021**, *26*, 256–265. [\[CrossRef\]](#)
27. Bandara, K.; Hewamalage, H.; Hao, Y.; Kang, L.Y.; Bergmeir, C. Improving the accuracy of global forecasting models using time series data augmentation. *Pattern Recognit.* **2021**, *120*, 108148. [\[CrossRef\]](#)
28. Yongjun, M.; Wei, L. Design and Implementation of Learning System Based on T-LSTM. In Proceedings of the Advances in Web-Based Learning—ICWL 2021, Macau, China, 13–14 November 2021; pp. 148–153. [\[CrossRef\]](#)
29. Terada, K.; Watanobe, Y. Code completion for programming education based on deep learning. *Int. J. Comput. Intell. Stud.* **2021**, *10*, 78–98. [\[CrossRef\]](#)
30. Smagulova, K.; James, A.P. A survey on LSTM memristive neural network architectures and applications. *Eur. Phys. J. Spec. Top.* **2019**, *228*, 2313–2324. [\[CrossRef\]](#)
31. Zhang, H.; Lee, I.; Ali, S.; DiPaola, D.; Cheng, Y.; Breazeal, C. Integrating Ethics and Career Futures with Technical Learning to Promote AI Literacy for Middle School Students: An Exploratory Study. *Int. J. Artif. Intell. Educ.* **2022**, *33*, 290–324. [\[CrossRef\]](#)
32. Kazi, K.; Solutions, K.K.; Solapur, M.; Devi, S.; Sreedhar, B.; Arulprakash, P.; Kazi, K. A Path Towards Child-Centric Artificial Intelligence based Education. *Int. J. Early Child. Spec. Educ. (INT-JECS)* **2022**, *14*, 9915–9922. [\[CrossRef\]](#)
33. Santos, J.; Bittencourt, I.; Reis, M.; Chalco, G.; Isotani, S. Two billion registered students affected by stereotyped educational environments: An analysis of gender-based color bias. *Humanit. Soc. Sci. Commun.* **2022**, *9*, 249. [\[CrossRef\]](#)
34. Du Boulay, B. Artificial intelligence in education and ethics. In *Handbook of Open, Distance and Digital Education*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–16. [\[CrossRef\]](#)
35. Floridi, L.; Cows, J. A Unified Framework of Five Principles for AI in Society. *Harv. Data Sci. Rev.* **2019**, *1*, 1. [\[CrossRef\]](#)
36. Schiff, D. Education for AI, not AI for Education: The Role of Education and Ethics in National AI Policy Strategies. *Int. J. Artif. Intell. Educ.* **2022**, *32*, 527–563. [\[CrossRef\]](#)

37. Samuel, Y.; George, J.; Samuel, J. *Beyond Stem, How Can Women Engage Big Data, Analytics, Robotics and Artificial Intelligence?—An Exploratory Analysis of Confidence and Educational Factors in the Emerging Technology Waves Influencing the Role of, and Impact Upon, Women*; Leibniz Center for Informatics: Wadern, Germany, 2018.
38. Hajibabaei, A.; Schiffauerova, A.; Ebadi, A. Women, artificial intelligence, and key positions in collaboration networks: Towards a more equal scientific ecosystem. *arXiv* **2022**, arXiv:2205.12339.
39. West, M.; Kraut, R.; Ei Chew, H. I'd Blush if I Could: Closing Gender Divides in Digital Skills Through Education. 2019. Available online: <https://en.unesco.org/Id-blush-if-I-could> (accessed on 27 November 2023).
40. UNESCO. AI and Gender Equality: Key Findings of UNESCO'S Global Dialogue. Available online: <https://unesdoc.unesco.org/ark:/48223/pf0000387610.locale=es> (accessed on 27 November 2023).
41. ONTSI. Brecha Digital de Género. Available online: <https://www.ontsi.es/es/publicaciones/brecha-digital-de-genero-2022> (accessed on 27 November 2023).
42. UGT. Informe Agosto 2022. Available online: <https://www.fesmcutg.org/wp-content/uploads/2022/09/Resumen-Estadistico-242-AGO22.pdf> (accessed on 27 November 2023).
43. WEF. Global Gender Gap Report 2021. Available online: <https://www.weforum.org/publications/global-gender-gap-report-2021/> (accessed on 27 November 2023).
44. Lundberg, S.M.; Allen, P.G.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. 2017. Available online: <https://github.com/slundberg/shap> (accessed on 27 November 2023).
45. Kordaki, M.; Berdousis, I. Identifying Barriers for Women Participation in Computer Science. *Int. J. Educ. Sci.* **2020**, *2*, 5–20. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.