Analyzing the ability of an ITS to identify causes of errors through step-by-step mutation: Quasi-experiment

Alan S. de Oliveira
Federal University of Rio Grande do Norte
Natal – RN, Brazil
alandeoliveirasantana@gmail.com

Eduardo H. S. Aranha
Federal University of Rio Grande do Norte
Natal – RN, Brazil
eduardoaranha@dimap.ufrn.br

Thiago Reis da Silva
Federal Institute of Education of Maranhão
São Raimundo das Patos - MA, Brazil
thiago.reis@ifma.edu.br

***Abstract*— Making mistakes is a natural process in the learning of any human being. However, it does not mean that they should not undergo interventions. In this sense, understanding the cause of errors can help teachers address students' limitations. In this sense, this work evaluates an ITS focused on analyzing answers to mathematical questions, capable of processing and identifying common errors, providing personalized feedback on these errors, and having features that enable learning through reflection on these mapped errors. To evaluate the study, a quasi-experiment was conducted that provided access to the ITS to a diverse group of users who performed activities through the ITS interface. This study involved a number of 113 unique user records who took a test through the support of the ITS containing 10 questions based on templates that randomized the variables of the proposed problems, as well as the order of these questions. The results showed that the tool is capable of identifying a satisfactory amount of errors based on its knowledge base. Through this error base, the tutor was able to provide personalized feedback and also accompanied assistance to help students better understand their mistakes. In addition, it was also noticed that when the student performed the activities with approximate assistance from the tutor, helping in the resolution after errors via step-by-step guidance, students tended to make fewer mistakes and consequently, learning may have occurred through this method.**

Keywords*— *Intelligent tutoring systems - ITS; Mutation; Step-By-step; Math; Error

I. Introduction

The use of computing in education is not new; many research teams have studied the positive impacts and challenges of using digital systems in educational practices, particularly due to the potential benefits these systems can provide [1].

In this context, systems like ITS (Intelligent Tutoring Systems) have been researched and applied in different contexts, such as distance education [2], language teaching [3], and mathematics education [4], for example. Particularly in mathematics, ITS aim to develop solutions or aids for daily activities, such as assisting in word problem solving, monitoring student learning, and developing extracurricular activities, among others [5].

Despite this diversity of applications, in general, the systems seek to act in the teaching and learning of students, supporting teachers' decision-making processes through personalized feedback, which can help both teachers and students understand errors more precisely and find solutions to encountered difficulties.

Thus, various solutions can be applied in building tutor models to meet the needs of their application, including the use of similar solutions, known as worked examples, step-by-step resolution, and complementary methodology applications, in order to create ways to meet the demands of these ITS requirements.

Therefore, the objective of this work is to present and evaluate the application of a new module for mathematics-focused ITS, linked to the application of software testing techniques known as mutant analysis, thus creating a module that collaborates with the standard modules of ITS—pedagogical, student, interface, and domain—in order to identify and categorize errors, as well as provide personalized feedback based on this categorization.

To guide this work towards its objectives, research questions were developed and applied to the study, namely:

- **RQ1**: How can an ITS use error patterns and mutant analysis to identify student mathematical errors?
- **RQ2**: What is the capability to detect student errors based on known error patterns and mutant analysis?
- **RQ3**: What are the benefits and limitations of identifying student errors based on error patterns and mutant analysis?

II. Literature Review

A. Intelligent Tutoring Systems

Intelligent Tutoring Systems (ITS) are computational systems that operate in education with multiple objectives, highlighting their ability to interact with students, track proposed tasks, provide feedback, among other functionalities.

ITS are widely studied by academia and industry, allowing validation of many of their concepts and approaches, making this field of study present in the academic reality of teachers and students, as in the successful case of the ITS "Assistments", a math-focused ITS that has been on the market for quite some time and has numerous works on its performance in education, such as the works of [6] and [7].

ITS are characterized by having modules that constitute them, with the main ones being [8]:

Student module: This module is responsible for collecting and storing information about the student, such as their learning

history, preferences, and cognitive profile. It can also monitor the student's progress and adapt the teaching content according to the student's individual needs.

Pedagogical module: The pedagogical module is responsible for providing instructions to the student, including the selection and sequencing of learning activities, the delivery of educational content, and the assessment of the student's performance. It uses pedagogical strategies to promote effective and adaptive learning.

Domain module: This module contains the specific knowledge of the domain being taught, such as mathematics, sciences, or languages. It includes information about concepts, rules, procedures, and relevant examples for the subject in question. The domain module is essential for customizing teaching according to the specific context of the content.

Interface module: The interface module is responsible for the interaction between the tutoring system and the student. It provides a user interface that allows the student to access educational content, receive feedback, ask questions, and interact with the system efficiently and intuitively. A good user interface can facilitate the student's learning experience and promote participation and engagement.

There are tutors that use more modules or describe them with another name, but generally these four modules are the main ones that make up ITS.

B. Mutant analysis

Mutant analysis is a widely used technique in the software testing field to evaluate the quality and effectiveness of test suites. This technique involves introducing small controlled changes, known as mutants, into the program's source code, and then executing tests on these mutants [9]. The objective is to determine the tests' ability to detect the changes introduced in the mutants, thereby identifying potential faults or deficiencies in the test suites [9].

Mutants are created by applying mutation operators to the original source code, resulting in modified versions of the program, known as mutants [10]. These changes may include, for example, modifying arithmetic operators, inverting logical conditions, removing instructions, or introducing deliberate errors into the code [10]. Each mutant represents a potential fault in the original program.

After creating the mutants, tests are executed on both the original program and each generated mutant. If a test fails on the original program but passes on a mutant, it indicates a failure in the test suite, as the mutant was not “killed” by the test, meaning the change introduced in the mutant was not detected by the test [11]. On the other hand, if a test passes on the original program but fails on a mutant, it indicates that the test was able to “kill” the mutant, meaning it detected the introduced change [11].

III. A MUTATION-BASED STI

This section proposes an ITS based on mutant analysis. The ITS was developed using a pedagogical model, performing the teaching and learning strategies of the ITS; a domain model, containing the knowledge of the covered contents; a graphical interface module of an existing interactive tutor; and a module focused on performing mutation analysis.

The student interacts with the ITS by answering a mathematical problem, providing only the final answer. The pedagogical module with the domain module solves the problem automatically, step by step, checking at the end if the answer is equal to the one provided by the user. Currently, the ITS supports solving first-degree equations. However, its modeling allows other types of mathematical problems to also be used in its domain module. Nevertheless, due to factors addressed during the presentation of this study, this topic was used because it is simple to implement and reaches a large audience.

After creating the step-by-step solution, the pedagogical module delivers this data to the mutant module, which has a knowledge base of common math errors reduced to their smallest form, such as sign change and arithmetic operation errors. Once the module receives the step-by-step solution, it simulates known error patterns by mutating one of the solution steps and checking if the new answer equals the student's. This process repeats until all mutations have been applied to each step. Notably, the following steps will not receive mutations when a mutant is applied to a step. In this way, only one mutant will be applied for each instance of answer verification.

For each simulated error, the mutation module can provide the pedagogical module with the cause of the student’s error and feedback for her and her teacher. In addition, if the student is unsatisfied with the feedback, the ITS can provide a step-by-step solution to the problem, leading the student to the correct answer at the end of the follow-up.

In this way, the ITS is divided into two major interaction ways. One is called Model A (identifying the mutant list), which lists possible student errors. The other is called Model B (step-by-step analyses), which performs a step-by-step problem resolution. These two models can interact individually or in a combined way. The system also logs all student actions for auditing purposes.

The identified error base included errors of sign change, arithmetic operation errors (division, multiplication, addition, and subtraction), errors in exponentiation and radication, and also errors in the order of operations [12]. All of these are considered irreducible errors and were implemented in the ITS. However, it should be noted that the model allows for the creation and mapping of other errors, as well as the combination of these errors to identify more complex mistakes.

It is noteworthy that there are studies focused on identifying student errors in mathematics, as in [13]. However, the created database is distinguished by using irreducible errors, that is, the smallest possible error, which can, in turn, be combined to create more complex errors.

For this study, the ITS was modified to simulate a 10-question test, each question being randomly selected from a math question database (first-degree equations). During the interaction, the student automatically receives feedback (correct or incorrect solution) with information about possible errors (mutations). If the error made by the student is in the list, she can identify it and move on to the next question. Otherwise, the

Identify applicable funding agency here. If none, delete this text box.

student can perform a step-by-step interaction with the IST to identify her error.

During the interaction with the prototype, the student will answer 10 questions about first-degree equations. For each question, the student will receive a textual problem and can provide an answer. After providing the answer, the prototype will construct the solutions step-by-step and perform mutant analysis, creating mutations and identifying their relation to the student's answer. It will then provide the list of identified mutations as feedback to the student.

Following this, the student will be asked if they wish to go through the step-by-step solution, which is an optional stage related to the functions of Model B. If the student agrees, the system will present the same problem solved step-by-step by the prototype, offering options related to the identified or simulated mutations if no mutations are found in the current step.

In this second interaction, still within the same question, the goal is to allow the student to resolve the problem with the support of the prototype, enabling more precise identification of potential difficulties and the location of errors. After completing these interactions, the system will load the next question or end the interaction if all 10 questions have been answered.

Thus, note that the following figure, Figure 1, represents this mentioned flow, starting with the presentation of the problem, receiving the user's response, accessing the internal ITS modules and the proposed models to perform the error identification process. If successful, the next step is to proceed with the step-by-step process, remaining in a cycle of questions and answers until the student completes the task. This second interaction is important to validate the steps the student performed in the previous process, thereby allowing a more direct identification of the error cause.

IV. Methodology

This study evaluates the proposed ITS and aims to capture data that provide insights into its usability and error identification capabilities. Moreover, it sought to identify any features in the data that may indicate improvement in student learning and/or teaching.

To this end, the first step is to have the ITS prototype ready in an online environment. This application model was chosen to extend the reach of the ITS, allowing a more significant number of participants.

The participants' profiles were then defined since they should be connected to the research scope, knowledge base, and questions implemented in the ITS. Thus, the selected profile was students enrolled in or who have already completed the first year of high school, as the type of problems that the ITS could address were related to first-degree equations, content typically introduced during the first year of high school.

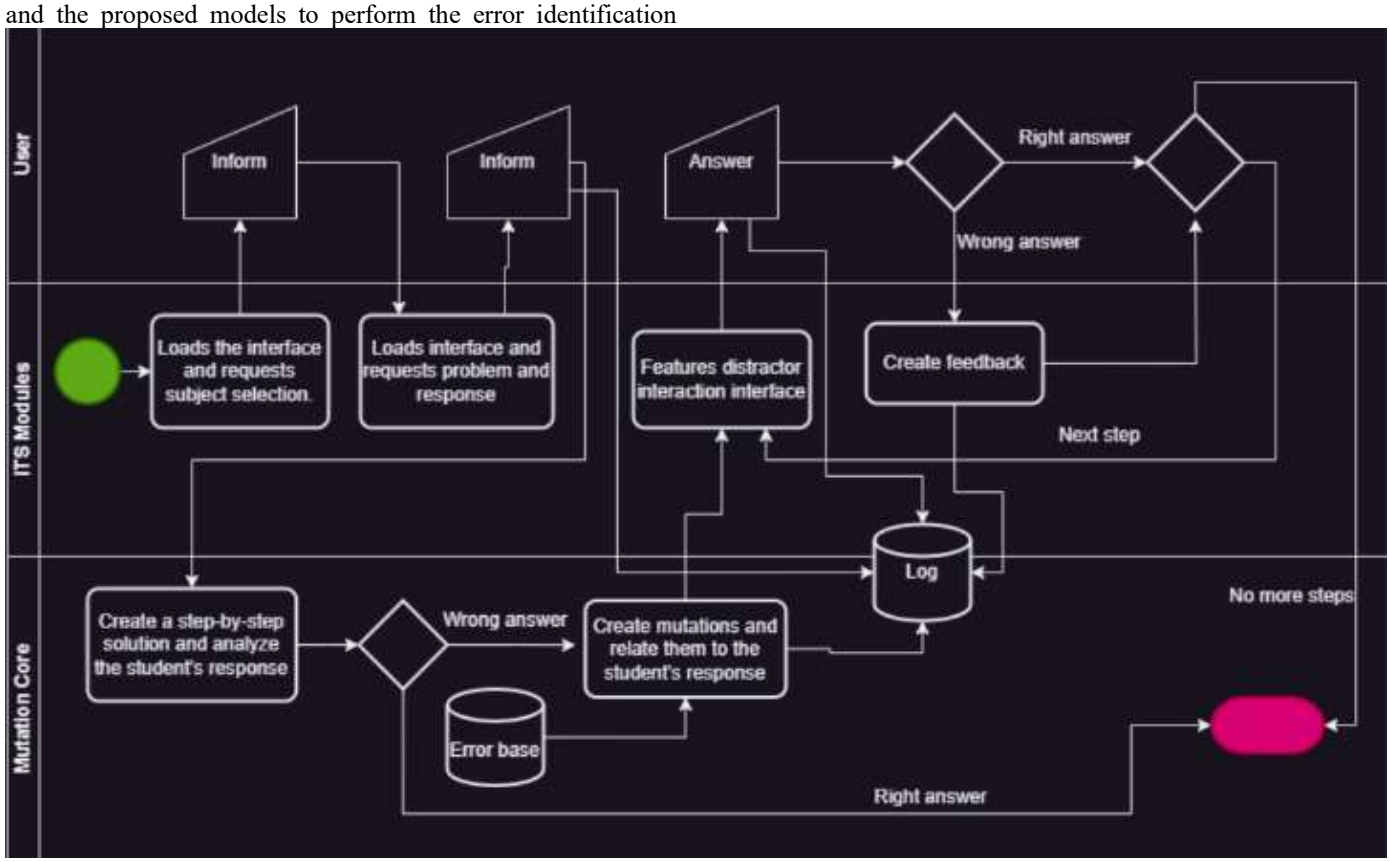

Fig. 1. Flow model.

It is emphasized that the proposed ITS model is conceptually capable of addressing different types of mathematical problems and was structured as a generic error identification methodology. Using a simpler error base, the model can handle more complex problems through the combination of errors or the insertion of more advanced concepts. However, a more diverse audience was sought, which was found in the concepts of first-degree equation, since this is a subject learned during the early years of secondary education and is also connected to numerous everyday activities. This also justifies the selection of such a broad audience.

Although content is introduced in a specific high school period, its basic concepts can also be seen in other subjects as part of the solution. In addition to the above, this content is also generally reviewed in pre-calculus courses. Therefore, the audience selected for the prototype consisted of students from different educational levels, workers, retirees, and anyone who has had contact with the content addressed by the ITS. Because the ITS can be applied in different situations, such as practice tests and reviews, it can support the mentioned audience.

Thus, the next step after defining the audience was to establish the means of dissemination, as it was necessary to send access to the ITS in a specific manner, avoiding access by individuals outside the target audience. This study decentralized access, meaning laboratories or meetings in controlled environments were not used. Instead, participants were given complete geographic and time freedom to complete tasks with the ITS.

Some precautions and recommendations were mentioned to the participants to prevent the access links from being sent to or used by individuals outside the defined profile. However, this threatens the study's validity, as there cannot be complete control over who accessed the ITS.

Regarding dissemination, a cloud service was used, allowing ITS code to be run online and synchronized with a Git repository. Thus, any code modifications and log retrieval were performed through synchronization via Git commands between the repository and the server. With the external access link obtained, invitation messages were sent to potential participants.

The participant group was connected to social media groups, third-party projects, and university and high school classes in public schools. In addition to the researchers of this study, some high school teachers also assisted in distributing the access link to individuals with the defined profile.

An important step was the capture of user data. Thus, when a user accessed the link, the ITS generated a unique hash code for task execution. Therefore, a new hash code would be generated if the same user made a new access. Additionally, no forms of data capture were applied, making the user experience as clean as possible and presenting only the functions of the ITS in its interface.

As previously mentioned, each interaction generated a logged action, and from these data, the information used to evaluate the ITS was captured. However, as mentioned, no sensitive user data, such as age or grade level, were captured. In the context of task execution, it was decided not to use complementary interfaces to make the experience as close as possible to that of a student who would use the ITS to review first-degree equation contents.

Regarding the application and user interaction format, the simulated format was chosen. As the ITS has a database of questions that can be used procedurally, 10 specific questions were defined with their variables modified according to their variation template for each access. Thus, when a new access was made, a new simulated test with 10 questions was structured for that access. Therefore, each user's access had unique elements, such as the order of the questions and the variables present in the problems. It is worth noting that the same statement was not repeated within a given test. Only the order of the 10 questions and the variables within their statements were modified.

For this study, the combined format of the ITS modules was used. That is, after answering incorrectly and requesting help from the ITS, a list of possible errors is shown, and then an interaction is made to walk through the solution steps, making the experience more complete but with the same information capture results.

Data were captured from each user interaction, recording both correct and incorrect responses along with the ITS's feedback, such as error identification and step-by-step guidance. The log included details like identification code, server IP, access date and time, and action performed. Specific data were recorded depending on the action type, with commas used to separate this information. Figure 2 illustrates an example of how actions were logged.

Fig. 2. Example of log entry.

The initial preprocessing involved separating and structuring this information into CSV files. The First Dataset provided a general overview by focusing on the data from the action field.

Further preprocessing included removing IP addresses and date-time information to present a cleaner view, as well as filtering out some actions, such as step-by-step problem-solving interactions, to concentrate on data relevant to the study. This led to the creation of the Second Dataset.

The Third Dataset was derived from the second dataset and contained only statistically relevant data, including student errors, correct answers, whether the ITS identified errors, and details about the step-by-step process. Interactions without actions were excluded.

Hence, the Fourth Dataset consisted solely of statistical summaries, removing user identification and retaining information such as mean, sum, standard deviation, variance, and median. The third and fourth datasets were primarily used for analysis, while the first and second datasets were consulted for validation purposes.

Finally, the data analysis stage utilized all the datasets generated in the preprocessing phase to address the research questions of this study, thus guiding the analysis process.

The analysis utilized the quantitative data from the databases, particularly those related to statistical values, and subsequently related them to data from other datasets when relevant. This relevance occurred when the data seemed questionable, requiring validation from the original data. Relationships between the statistical and original data were also explored to understand the relationships that generated these data.

V. RESULTS

This section presents the research findings and correlates these data with the research questions. These are the main findings of the study:

- The ITS identified 70 out of 387 (18.09%) student errors in the first interaction with the questions, showing a list of possible errors.
- In the second step (step-by-step), a total of 133 answers were registered, with just 32 errors using the mutant list of errors identified.
- A total of 419 errors were identified in this study.
- In the first step, identifying the mutant list, a total of 207 answers were correct. The total number of correct and incorrect answers in this step was 594.
- In the second step (step-by-step), a total of 133 answers were registered, with 101 answers correct.
- Among the questions where errors were recorded, between 1 and approximately 20 errors associated with a single response were listed.
    - The average number of mutants generated per question was 3, with some questions presenting 1 identified error, while others presented up to 18 identified errors, for example.
    - The main errors identified were related to mathematical operation mistakes and sign errors.

The next sections discuss the observed data that supports the presented findings.

A. Data Summary

The total records log amounted to 12,024 lines of entries, of which 2,004 correspond to unique action records within the ITS interface. In addition to these data, there were 165 accesses to the test profile, where 52 did not engage in activities with the ITS. Therefore, 113 user identifications were considered for the data collection, and the results are presented below.

The first stage of interaction with a question involves answering it correctly or incorrectly. There were 594 responses recorded among the 10 questions presented for each access ID. Out of this total, 207 were correct responses, concluding the question and initiating the next one, while 387 were incorrect.

Among the 387 incorrect responses, the ITS could not associate 317 with a present error in implementing common errors, while 70 were identified correctly with the mutant list.

Thus, 11.78% (70 of 594 answers) of all responses were mapped and related to one or more errors with the mutation technique, and specifically, among the incorrect responses, 18.09% (70 of 387 errors) were correctly mapped.

The data related to this stage of interaction with the ITS presented a considerable rate of variation and dispersion, with the standard deviation and variance of the question accuracy data being 2.69 and 7.18, respectively.

Each access linked to a unique hash ID could make N attempts to answer the question even without accessing the step-by-step process, as the interface allowed the submission of consecutive responses. However, the submission closes after a correct response is given. Thus, it was possible to map how many IDs completed the 10 questions correctly. This number was 4 out of the 113 unique IDs, followed by 3 with 9 correct responses, 1 with 8 correct responses, 4 with 7, 2 with 6, 2 with 5, 2 with 4, 5 with 3, 11 with 2, 37 with 1, and finally, a high number of 95 with 0 correct responses. Table 1 presents this result.

Some outliers need to be highlighted regarding "Unidentified Errors". As mentioned, students could repeat the same problem N times before getting it right or requesting the ITS to walk through the question step by step.

TABLE I. RESULTS OF CORRECT AND INCORRECT ANSWERS TO THE QUESTIONS

	Total records	**Percentage**	**Standard deviation**	**Variance**
Correct	207 out of 594	34,8%	2,68	7,18
Identified Errors	70 out of 594	11.8%	1.32	1.74
Unidentified Errors	317 out of 594	53.4%	4.33	18.73

Thus, 6 IDs were recorded to have a high number of errors associated with values greater than 10. The values were 20, 13, 12, 22, 27, and 11, representing 33.12% of the unidentified errors, totaling 105 out of 317 records. These outliers are the main contributors to the dispersion seen in this indicator's standard deviation and variance. Table 2 presents the overall results.

The next result is related to the step-by-step process. Since all incorrect steps are based on construction from mutations, these errors, treated as distractors, are linked to specific causes. Thus, whenever a user made a mistake in the step provided by the ITS, this error was associated with a specific cause.

A total of 133 entries related to responses during the step-by-step interaction with the users were recorded. Linked to this result, 101 records were correct responses, while incorrect responses totaled 32 records. The question with the most records was question 1, and the question with the fewest records was question 3.

The data presented contrasts with the data presented in the first part of the user interaction with the ITS, as there were more errors than correct responses in the direct problem-solving part,

representing 34.8% accuracy. However, when users go through the step-by-step process, the number of errors significantly decreases, with the percentage of correct responses in this step representing 75.94%.

Notably, the higher number of activities in step 1 is explained because the API used to generate the step-by-step process in most problems resolved in just one step, justifying this finding. However, up to 5 steps were recorded in the activities performed by the users. It is also worth noting that the questions were not ordered in a standardized manner since the order in which they appeared was randomized for each unique access, thus distributing the number of steps in the results while still maintaining this characteristic as prominent in the results.

Another interesting fact is that a progression in the reduction of activities during the questions and in the steps was expected, but the data show that this did not occur. As mentioned, question 3 had the lowest acceptance rate for the step-by-step process, with 6 activity records. However, the second lowest was question 10, with 7 records, and the third was question 5, with 9 activity records, demonstrating the dispersion in quantity and lack of logical standardization, a fact corroborated among the two questions with the most step-by-step records, question 1 with 25 records, and question 9 with 19 records.

TABLE II. OVERVIEW OF THE RESULTS OF CORRECT AND INCORRECT ANSWERS TO THE QUESTIONS

	Correct	Identified error	Unidentified error	Total
Question 1	22	5	34	61
Question 2	22	7	31	60
Question 3	24	6	30	60
Question 4	18	8	33	59
Question 5	22	8	29	59
Question 6	17	7	35	59
Question 7	21	8	30	59
Question 8	18	5	36	59
Question 9	24	7	28	59
Question 10	19	9	31	59
Total	207	70	317	594

The same reduction mentioned was expected among the steps. Step 1 had a total of 66 records, of which 48 were correct and 18 were incorrect. Step 2 had 30 records, 20 correct and 10 incorrect. Step 3 presented a total of 17 records, 16 correct and 1 incorrect. Step 4 had 19 records, 16 correct and 3 incorrect. Finally, only 1 record was observed in step 5, which was a correct response.

The Table 3 presents the results obtained from the step-by-step interactions.

B. Findings

The following data presents findings derived from the data obtained in this study.

Based on the data presented in Tables 2 and 3, a statistical test was conducted to identify relevance related to whether there was learning between the stages of direct response and the step-by-step stage.

TABLE III. DATA FROM THE STEP-BY-STEP PROCESSES

Step / Question	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total
Step 1 correct	7	4	3	6	2	4	4	5	10	3	**48**
Step 1 incorrect	1	7	0	3	1	0	2	0	4	0	**18**
Step 2 correct	5	2	1	1	3	1	1	3	1	2	**20**
Step 2 incorrect	2	0	2	1	0	2	0	2	1	0	**10**
Step 3 correct	3	2	0	1	2	1	1	3	2	1	**16**
Step 3 incorrect	0	0	0	0	0	1	0	0	0	0	**1**
Step 4 correct	4	3	0	1	1	0	2	3	1	1	**16**
Step 4 incorrect	2	0	0	0	0	1	0	0	0	0	**3**
Step 5 correct	1	0	0	0	0	0	0	0	0	0	**1**
Step 5 incorrect	0	0	0	0	0	0	0	0	0	0	**0**
Total	**25**	**18**	**6**	**13**	**9**	**10**	**10**	**16**	**19**	**7**	**133**

To perform this statistical test, Analysis of Variance (ANOVA) was chosen. ANOVA is a statistical technique used to compare the means of three or more different groups to determine if there are significant differences between them [14]. ANOVA assesses the variability within each group compared to the variability between the groups, allowing us to determine if the observed differences between the means are statistically significant [14].

Thus, two groups were defined. The first linked to the total correct and incorrect responses made in the first interaction with the ITS. The second group comprised all errors and correct responses in the step-by-step phase. To do so, four vectors were generated for each of the mentioned pieces of information. Thus, the data related to errors in the first step were composed of the sum of the identified and unidentified errors from each question. The column of errors and correct responses from the step-by-step phase was also formed by the combination of all errors and correct responses, respectively, from each question, composing two arrays from this data. The data used were as follows:

Correct_answers_direct = [22, 22, 24, 18, 22, 17, 21, 18, 24, 19]

Wrong_answers_direct = [39, 38, 36, 41, 37, 42, 38, 41, 35, 40]

Correct_answers_step = [20, 11, 4, 9, 8, 6, 8, 14, 14, 5]

Wrong_answers_step = [5, 7, 2, 4, 1, 4, 2, 2, 5, 0]

Thus, the ANOVA test was conducted for two groups, comparing the data of correct responses in the direct response

stage and correct responses in the step-by-step stage, and also the same test for error cases. The ANOVA results for the difference in correct responses before and after the step-by-step process are as follows:

F-statistic: 37.897472924187724;

P-values: 8.199255254011708e-06;

Descriptive statistics for correct answers before step-by-step:

Variance: 5.81;

Mean: 20.7;

Standard Deviation: 2.41039415863879;

Median: 21.5.

Descriptive statistics for correct answers after step-by-step:

Variance: 21.889999999999997;

Mean: 9.9;

Standard Deviation: 4.678675026115834;

Median: 8.5.

The value of the F-statistic is a statistical test that indicates whether there is a significant difference between the means of the groups. In this case, since the value of the F-statistic is relatively high, it indicates that there is a significant difference in the differences of correct responses before and after the step-by-step process. The P-value is the probability of observing a test statistic equal to or more extreme than that observed in the data, assuming the null hypothesis is true. A very small P-value (usually less than 0.05) suggests that the null hypothesis can be rejected. In this case, since the P-value is very close to zero, it suggests that the difference in means of the groups is statistically significant.

On the other hand, the ANOVA for the difference in errors before and after the step-by-step process presented the following results:

F-statistic: 1264.4648829431462;

P-values: 3.945290346352622e-18.

Descriptive statistics for errors before step-by-step:

Variance: 4.81;

Mean: 38.7;

Standard Deviation: 2.1931712199461306;

Median: 38.5.

Descriptive statistics for errors after step-by-step:

Variance: 4.159999999999999;

Mean: 3.2;

Standard Deviation: 2.0396078054371136;

Median: 3.0.

The F-statistic value is very high, indicating an extremely significant difference in errors before and after the step-by-step process. The P-value is also very close to zero, indicating that the difference in means of the groups is highly unlikely to be due to chance. Figures 3 and 4 respectively present a boxplot on the results of the comparison between the correct and incorrect responses in the direct response and step-by-step stages.

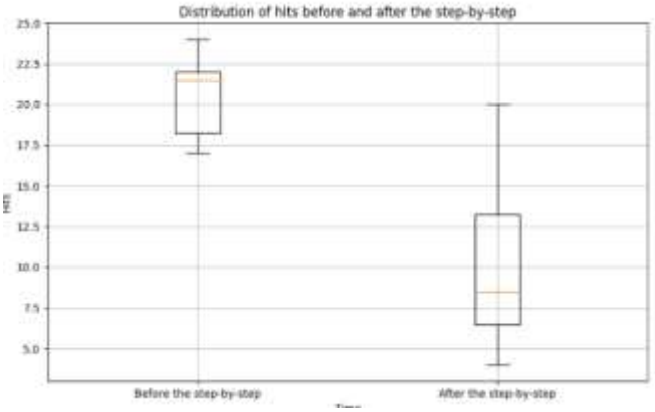

Fig. 3. Distribution of correct answers.

In summary, both for correct answers and errors, the ANOVA results strongly suggest that there is a significant difference in means between the groups before and after the step-by-step process. This suggests that there was indeed learning after the error identification step, which is an important finding suggesting that engaging in the ITS- guided step-by-step process can lead to significant gains in overall student performance, something that can be further explored through testing and training activities.

Regarding the ITS's ability to identify error causes and provide personalized feedback, it was observed that the system is indeed capable of associating causes for errors in direct responses through mutation processes. Additionally, errors generated via distractors seem to be effective in creating valid alternatives. This assertion is supported by the fact that the ITS provides immediate feedback related to the error cause both in the direct response interaction and during the step-by-step process, fostering opportunities for students to understand where they are going wrong and subsequently generate knowledge synthesis from this identification.

Another factor that supports this assertion is the reduction of errors in the step-by-step stage. Through analysis relating the results to error identification data, it was observed that in most of the questions where automatically identified feedback was provided through the use of mutants, students did not repeat the same error in subsequent attempts.

C. Limitations

Despite not being a planned stage during the study, some users provided feedback after using the ITS regarding issues faced in the interface and interactions with the ITS. These data were spontaneously reported and promptly recorded in order to make improvements in the interface module and other modules.

Initially, problems were reported during the step-by-step stage where users needed to choose binary options indicating whether the alternative was incorrect or correct. In some

executions, the system temporarily froze and did not display the options, which was promptly addressed.

Another issue faced by users was also in the step-by-step interface, but this time with the distractor alternatives. For some users, it was not clear that they needed to click on the alternatives to select them, so this information was promptly added in a prominent manner.

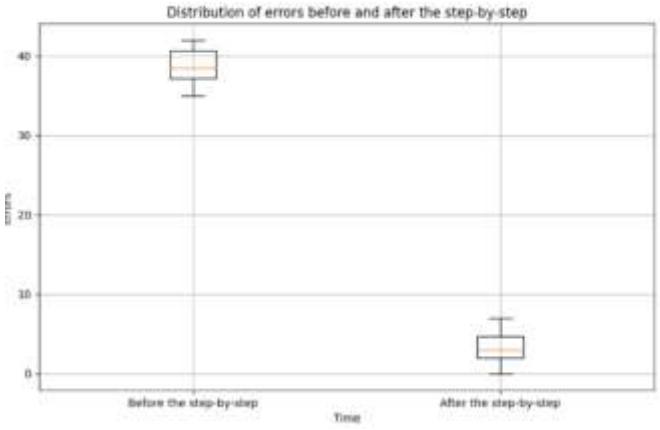

Fig. 4. Example of log entry.

Another difficulty encountered by users was the lack of the statement during the step-by-step stage, which was also promptly adjusted.

Some users reported that certain types of responses were not generating feedback because the ITS was unable to identify their causes, despite apparently being simple and within the scope of the mutation module and error database. Subsequently, there was an opportunity to make some modifications based on the reports, validating the modifications with experts, making the ITS able to identify more sources of errors. However, this implementation was carried out after the experiment.

It is worth noting that some limitations were self-imposed to avoid excessive processing times, such as in the case of calculation errors, which had significant limitations in possibilities. However, a user also reported that the ITS was unable to identify a simple calculation error and associate it with a calculation error. Upon analyzing the log of this action, it was noticed that the error occurred due to the maximum and minimum values used as reference for the implementation applied in the study. Furthermore, this implementation only took into account the addition and subtraction of values to the variable modified in the step.

For example, if the answer is 6, it could vary from 1 to 11. If it were -3, it could vary from -8 to 2. Note that in some cases, the error may be related to multiples, as in the case of 10, which can obtain errors such as 100 or 20 in its responses, and this was the reported case. Thus, also subsequently, this modification was implemented, using an algorithm that takes into account multiples of 10 and some logical values, such as adding a 0 to numbers also divisible by 10.

Regarding the percentage of errors mapped with the mutation list, it is emphasized that issues in the experiment version may have actively contributed to the reduction of this value. After analyzing the results and user feedback, improvements were made and new tests conducted during the writing of this study showed approximately 45 to 80 percent accuracy in constructing lists of possible errors from mutant analysis.

Among these issues, the main ones were related to user inputs, where correct answers were often mapped as incorrect and thus could not be identified. Additionally, incorrect answers in formats not supported by the presented prototype version did not allow the correct association of the response with the mutation list. These errors were mapped and corrected, significantly increasing the success rate in creating the mapped error lists.

D. Answers to the research questions

RQ1: How can an ITS use error patterns and mutant analysis to identify student mathematical errors? As observed, using a system that applies mutations based on known errors not only allows for identifying potential error locations but also associating causes, providing personalized feedback, reports, among other tools that assist students and teachers in understanding and resolving encountered difficulties. Thus, the proposed approach offers a comprehensive method for error identification and potential assistance mechanisms in teaching and learning.

RQ2: What is the capability to detect student errors based on known error patterns and mutant analysis? As observed, the implementation of the presented ITS has limitations in error identification, many of which are related to the implementation rather than the proposed model. It is noteworthy that the proposed module utilizes a common error database and can be complemented with tree-like hierarchies, the use of AI to enhance error identification and its causes, and the association of error classes and situations, which can be supplemented by other modules such as emotion recognition, among others. Thus, this study concludes that the model is efficient and has the potential to be expanded with mapped enhancements, providing a better error identification capability than observed, showcasing the potential of using mutant analysis techniques and identifying common error patterns in ITS.

RQ3: What are the benefits and limitations of identifying student errors based on error patterns and mutant analysis? It was observed that by combining the two models, learning may have occurred, which is one of the potential major benefits of the proposal. However, it was also observed that standardizing errors atomically is a relevant method for error identification in step-by-step problem-solving, as it allows for error isolation and better understanding by students and teachers of the moment of error and its potential causes, providing a reflection base for students to overcome difficulties from the precise identification of the error location.

Another important point is the use of the mutant analysis method for creating distractors and also error classification based on mutations. This method provides the computer with an additional layer of analysis, much deeper and more complex, allowing ITS to provide better feedback and reports. Thus, it is understood as a module applied to ITS that interacts with other modules, particularly the pedagogical and domain modules.

The limitations of this project were also mapped, highlighting the construction of a solid error database. This study previously built an error database that was used; however, the scarcity of complete databases that can be used in academia is evident, demonstrating the need to create common and public databases to provide an easier replication possibility of this model. Another limitation specific to the study is the implementation of ITS, which considers the available tools and project requirements.

Another limitation was the possibility of users guessing the answers. The tool, in its current implementation, is not capable of distinguishing guesses. However, it is proposed as future work to include a mechanism to identify such occurrences and address them both at the recording level and in the feedback provided by the ITS.

Another identified limitation was the broad audience. However, this limitation was mitigated by relating the concepts of first-degree equation, a topic initially covered in secondary education and also used in everyday life as well as in the professional lives of numerous professions. The questions addressed also aimed to present everyday facts, minimizing noise effects caused by participants' profile differences.

VI. Conclusions

This work presented and evaluated the proposal of a module based on mutant analysis applied to a mathematics-focused ITS. To evaluate its performance, a study was conducted with a specific audience linked to the knowledge of first-degree equations, knowledge linked to the domain module of the ITS.

The ITS was developed using the basic modules, student, domain, pedagogical, and interface, complemented by the module proposed in this study, the mutant module. The latter allows mutations to be made in the step-by-step resolution of the problem, associating the mutation inserted in the step with a probable error cause.

The study was applied remotely, with 113 users interacting with the system in an engaged manner. The results indicate that the so-called Model B and Model A can create an active learning environment, influencing student interaction with the ITS in real learning through reflection and understanding of errors made. However, further studies are suggested to validate these data. The ability of the ITS to identify errors was also observed. Even though the observed value was not very high, the mutant module demonstrated its ability to contribute to error identification positively.

Suggestions for future work are precisely linked to these points, improving the ITS interfaces and its feedback system, exploring other mathematical topics, both more straightforward and more complex, capturing participants' socioeconomic data for complementary performance analyses, evaluating the performance of Models A and B with groups of the same age range and knowledge, separately and jointly, and evaluating the performance of the proposed ITS in other disciplines, such as Portuguese and chemistry.

Another future work is to redo the current experiment using a specific class with a more homogeneous participant profile and also to implement a login system to identify unique users and their access instances.

Overall, the contribution to the study of tutors is observed by applying mutant analysis in the development of modules to support interaction and identification of difficulties encountered by students in solving mathematical problems.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] Smith, R. C. et al. 2023. A research agenda for computational empowerment for emerging technology education. Int. J. Child-Comput. Interact. 38, 100616 (2023).

[2] Alam, A. 2023. Harnessing the Power of AI to Create Intelligent Tutoring Systems for Enhanced Classroom Experience and Improved Learning Outcomes. In Intelligent Communication Technologies and Virtual Mobile Networks. Springer Nature Singapore, Singapore, 571–591.

[3] Crossley, S., Liu, R., & McNamara, D. 2017. Predicting math performance using natural language processing tools. In Proceedings of the Seventh International Learning Analytics & Knowledge Conference (LAK '17). Association for Computing Machinery, New York, NY, USA, 339–347. DOI: 10.1145/3027385.3027399.

[4] Kautzmann, T. R. & Jaques, P. A. 2019. Effects of adaptive training on metacognitive knowledge monitoring ability in computer-based learning. Comput. Educ. 129, 92–105.

[5] Ruan, S. et al. 2020. Supporting children's math learning with feedback-augmented narrative technology. In Proceedings of the Interaction Design and Children Conference. 567–580.

[6] Gong, Y., & Beck, J. E. 2015. Towards detecting wheel-spinning: Future failure in mastery learning. In Proceedings of the Second (2015) ACM Conference on Learning@Scale (L@S '15), 67–74.

[7] Xing, W., & Goggins, S. 2015. Learning analytics in outer space: a Hidden Naïve Bayes model for automatic student off-task behavior detection. In Proceedings of the Fifth International Conference on Learning Analytics and Knowledge (LAK '15), 176–183.

[8] Gamboa, H., & Fred, A. 2002. Designing intelligent tutoring systems: a Bayesian approach. In Enterprise Information Systems III, edited by J. Filipe, B. Sharp, and P. Miranda, 146–152. Springer Verlag: New York.

[9] Untch, R. H., Offutt, A. J., & Harrold, M. J. 1993. Mutation analysis using mutant schemata. In Proceedings of the 1993 ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA '93), 139–148.

[10] Wang, Z. et al. 2021. Prioritizing test inputs for deep neural networks via mutation analysis. In 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 397–409.

[11] Shen, Q. et al. 2021. A comprehensive study of deep learning compiler bugs. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '21), 968–980.

[12] Eduardo Aranha, Alan Santana, Thiago Reis da Silva, Nataly Lima Marinho, and Brenda dos Santos, "Construction of a Common Errors Database in Mathematics for Intelligent Tutoring System Development," in 2024 IEEE Frontiers in Education Conference (FIE), USA, 2024.

[13] Komang Tri Purnamayanthi, I Nengah Suparta, and Gde Suweken, "The Analysis of the Types of Student Error in Solving Mathematical Problems based on the Anderson's Taxonomy," in Edumatica: Jurnal Pendidikan Matematika, vol. 12, no. 01, 2022.

[14] Costa Santos, Leandro da, Leonardo Rolim Severo e Lindinalva de Alcântara Correia (2023). "Desafios ao engajamento acadêmico no ensino superior: uma análise a partir de posicionamentos sobre satisfação de estudantes". Em: Revista Internacional de Educação Superior 9, e023027–e023027 (ver p. 138).