ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2025, Vol. 86, No. 5, pp. 417–431.

Regression Models for the AI Gaming Chatbot for Learning Programming Based on Wordle-Type Puzzles

A. N. Varnavsky

National Research University "Higher School of Economics", Moscow, Russia
e-mail: avarnavsky@hse.ru

Received November 29, 2024
Revised January 9, 2025
Accepted January 14, 2025

Abstract—Programming is one of the most important skills of the 21st century. However, for many students, learning programming is a rather complex process. In such cases, it is important to maintain the interest and involvement of students in the learning process. It is believed that digital games can solve this problem. One of the types of games that are well suited for the field of computer science are puzzles, which are aimed, among other things, at developing cognitive abilities. The purpose of the work is to develop models, an algorithm for the operation and structure of a game chatbot with artificial intelligence for learning programming using Wordle-like puzzles. Game Wordle was chosen due to its worldwide popularity and adapted as a gaming chatbot for use in learning programming. Artificial intelligence in the chatbot is necessary to control the appropriateness and appropriate time of its use, as well as adaptively forming the difficulty level of puzzle. Based on the data collected as a result of using a non-intelligent gaming chatbot, regression models of the influence of student indicators on the level of interest and difficulty of puzzles offered by the gaming chatbot were built. The developed models formed the basis of the algorithm of operation and structure of the gaming chatbot with artificial intelligence. When using an intelligent gaming chatbot, it is possible to further train the models and adjust the previously obtained coefficient values.

Keywords: gaming chatbot, learning to program, artificial intelligence

DOI: 10.31857/S0005117925050046

1. INTRODUCTION

For many students, learning to program is quite a difficult process. This is due to the fact that solving programming problems requires not only knowledge of programming languages, but also appropriate levels of cognitive abilities (logical, critical, computational thinking). Low levels of such abilities often create difficulties in learning and lead to a decrease in student engagement and motivation.

The use of educational computer games is one of the methods for maintaining student interest in learning in various disciplines and areas [1], including programming [2]. Cognitive abilities and functions (working memory, processing speed, executive functions, etc.) can be significantly improved by using games [3], and the specific effect depends on the types of games used [4]. Thus, logical games and puzzle games are well suited for teaching computer science [5].

Traditionally, Parsons programming puzzles [6], which are randomly jumbled code fragments that the learner must arrange in the correct order to obtain a working program, are well suited for teaching programming at the initial stage. They are effective for mastering basic syntactic structures and programming logic, but their capabilities are limited and they are not sufficiently focused on developing more complex cognitive skills.

At the same time, word puzzles, which are widely used in teaching foreign languages and medicine, are practically not used in teaching programming, despite their enormous potential for developing cognitive abilities [7]: vocabulary, working memory, cognitive processing, strategies for retrieving information from working memory, reasoning and thinking abilities [8].

Effective programming directly depends on the knowledge of the syntax of the programming language, vocabulary related to specific terminology, syntactic rules [9]. Even a minor deviation from the syntax can lead to compilation or execution errors, requiring significant time spent on debugging. Moreover, unclear and imprecise use of terminology in the code significantly complicates its understanding both by the author and other developers, increasing the cost and time spent on supporting and upgrading the software [10]. This is especially critical in large projects where the code is written and supported by many people, and unclear vocabulary can lead to code incompatibility, loss of time, and even to the occurrence of errors leading to a failure of the entire system. Given the potential of word puzzles, it is necessary to develop the direction of their use in teaching programming and computer science. Moreover, it is necessary to search for options for implementing word puzzles that would be interesting and exciting for students of different levels, while developing their skills.

One approach to creating educational games is to adapt well-known computer games (for example, the adaptation of Super Mario in [11]). In recent years, a fairly popular word game is Wordle (https://www.nytimes.com/games/wordle/index.html). Wordle is a word puzzle game in which players must guess a word in several attempts. Scientific works appear that study strategies, the process of solving puzzles, the complexity of words, and the behavior of players in this game [12–14]. Therefore, the principle of the Wordle game can be adapted to create a puzzle game for recognizing program code for use in the process of teaching programming. Given the growing popularity of chatbots in various fields, including education [15], as well as the text format of the Wordle game, the adaptation can be implemented in the form of a chatbot.

The gaming chatbot with puzzle tasks can be embedded with models that will make it intelligent, for example, in terms of generating tasks with an appropriate level of difficulty and interest. This will increase the efficiency of its use, in particular due to the fact that interest affects engagement and motivation, and the difficulty of tasks can affect interest. Similar gaming chatbots were no found in the literature.

The aim of the work is to develop models, an algorithm for the operation and structure of the gaming chatbot with artificial intelligence for learning programming using like Wordle-like puzzle. To achieve this aim, it is necessary to collect data on the use of the first version of the non-intelligent gaming chatbot, on the basis of which regression models can be built. These models will form the basis for the algorithm for the operation and structure of the new version of the gaming chatbot.

2. RELATED WORK

2.1. Use of Educational Games

Although some studies have not found any positive effects from using educational games, most scientists consider games to be a very effective and inexpensive teaching method that increases interest in the training course, student engagement, academic performance, and knowledge assessment after using games [17, 18]. Games can influence various aspects, including affective and motivational elements, behavioral changes, knowledge acquisition, and the development of cognitive and social skills [19].

There are computer games for teaching computer science and programming, and their number has been steadily increasing in recent years [5, 20–22]. There are games for different levels of education, although games for teaching programming to beginners and schoolchildren predominate. Among the existing games, puzzles and logic games are a separate area, since they are well suited for teaching computer science and allow developing thinking skills and logical thinking [23]. This is manifested in the improvement of such indicators as the speed of information processing, the volume of working memory, cognitive control, argumentation and problem solving.

Word puzzles are a fairly popular type of game used in educational practice, especially in teaching foreign languages and medicine. Their integration with courses, in combination with other learning tools, helps to reduce the cognitive load on students, stimulate motivation, improve academic performance [16], effectively develop general knowledge [25] and cognitive abilities [7, 24]. However, word puzzles are not very common in programming education.

In recent years, one of the most popular word puzzles is the Wordle game. This game is intended for entertainment, but it is possible to note the appearance of a fairly large number of scientific works studying this game, the solution process, options for enumeration and search for a solution, etc. [12–14]. Therefore, it is possible to consider the direction of adapting this game, as applied to the educational process, in particular for use in teaching programming.

2.2. AI Chatbots

Puzzle games can be implemented in a variety of forms, including a chatbot. This format may work well for some text/word puzzles. Basic chatbots use a rule-based approach [26]. In contrast, AI chatbots use machine learning models.

AI chatbots are one type of conversational AI application. According to forecasts, the global conversational AI market size is expected to grow at a CAGR of 21.8%, reaching $18.4 billion by 2026 (https://www.marketsandmarkets.com). Conversational AI has the potential to become a mainstream technology in almost all market verticals (https://www.emarketer.com/content/tech-catches-up-conversational-ai-ambitions).

Overall, the benefits of AI chatbots include providing a natural language, user-friendly interface that can improve the user experience [27]. Using AI chatbots can reduce stress and help users avoid information overload [28], providing users with a unique and more personalized experience. Such chatbots can influence users' emotions in a way that conventional enterprise systems cannot [29].

AI conversational systems can exhibit personality and have anthropomorphic characteristics [30], better match user and organizational contexts, and are able to continually evolve by learning from the content of requests and how users interact with them. Therefore, the continued use of AI systems is important for their continued improvement, since they are usually based on machine learning technologies. Since the continuous improvement of an AI chatbot depends on its use, it is important to consider user behavior and engage usage patterns [31].

Currently, the development of AI chatbots is focused on the development of algorithm training methods and approaches related to customization, anthropomorphism [32], personalization [33], improving algorithms to obtain more accurate results [34] and expanding functionality, including, for example, sentiment analysis [35]. There are proposals to create a new human-machine intelligence [36].

Examples include AI chatbots that are used in the workplace [37], in business processes [38], in organizing effective interactions with users and clients [39], in training [40]. Such chatbots are more effective due to the automatic detection of user needs [41, 42] and the formation of user-oriented behavior [43]. For example, in education, AI-powered chatbots can increase users' interest in learning mathematics [44] and reading [45].

2.3. Formulation of the Task

In general, there are very few educational gaming chatbots described in the literature, and even fewer educational gaming chatbots with artificial intelligence. [46] can be used as an example.

Each described chatbot has its own logical/scenario or AI/ML behavior model. Given the originality of the task, such models are not suitable for building an intelligent gaming chatbot with puzzles like Wordle and need to be developed. To do this, we will use the non-intelligent gaming chatbot and conduct an experiment with it. The data collected as a result of the experiment will be used to create machine learning models to control the behavior of the intelligent gamimg chatbot. The models will control the level of difficulty of tasks and assess the feasibility of using a chatbot at a given stage of learning a programming language by predicting interest in the task. The models can be further trained based on new data obtained when using an intelligent gaming chatbot. Based on the obtained models, we will create an algorithm for the operation and structure of the gaming chatbot with artificial intelligence for teaching programming using Wordle-like puzzles.

3. GAMING CHATBOT WITH PUZZLE FOR LEARNING PROGRAMMING

The non-intelligent gaming chatbot "Identify the Python command" for the Telegram messenger with text puzzle similar to the Wordle game [47] has been created, which is designed to teach a text programming language, in particular Python (Fig. 1), and has the following game mechanics.

1. The aim of each task is to recognize a line of a certain program in several steps-attempts. Initially, the line of the program is presented as a sequence of symbols $*$, the number of which is equal to the length of the line (with spaces).

2. Simplifications in the formation of the task are carried out in relation to the names of variables, values of numbers, strings, lists, dictionaries and other structures. Thus, variables are changed and used in the order a, b, c, ..., numbers — in the order 1, 2, 3, ..., symbols and strings — in the order 'a', 'b', 'c', ..., mathematical operations — in the order '+', '−', '$*$', lists, tuples, dictionaries, etc. are simplified to 1 element.

3. The text entered at each step is checked for compliance with the length of the program line, the syntax rules of the language and the rules of PEP8 (in the case of Python), and a check is also made for the presence of already open and known symbols. In case of violations, the student is informed about the discrepancy and incorrectness of the input, and in case of absence, a line with open symbols is given, followed by a hint after the # symbol. The hint is given as a set of symbols sorted in ascending order of ASCII code.

4. Open symbols and a hint after each attempt are formed based on the result of a symbol-by-symbol comparison of the entered text with the required line of the program according to the same rules as in the Wordle game: if the symbols match, i.e. are in the correct places, these symbols appear instead of $*$, and if the symbols are in other positions, i.e. are not in their places, these symbols appear in the hint.

5. The maximum number of correct attempts given to complete the task is 20.

6. There are 3 levels of tasks of varying difficulty and different scoring, the ability to change and move between levels. When completing at least 3 tasks at the current level, the level can be changed to the next one. The level can always be changed to a lower one. When completing task in 5 or fewer attempts, on the first level 10 scores are awarded, on the second level 20 scores are awarded, on the third level 30 scores are awarded. If the task is completed in more attempts, the number of scores received decreases.

7. The number of characters in the task depends on the level: the higher the level, the more characters in the used line of the program. Program lines with widespread functions, commands, methods are used at levels 1 and 2, less common ones are used at level 3.

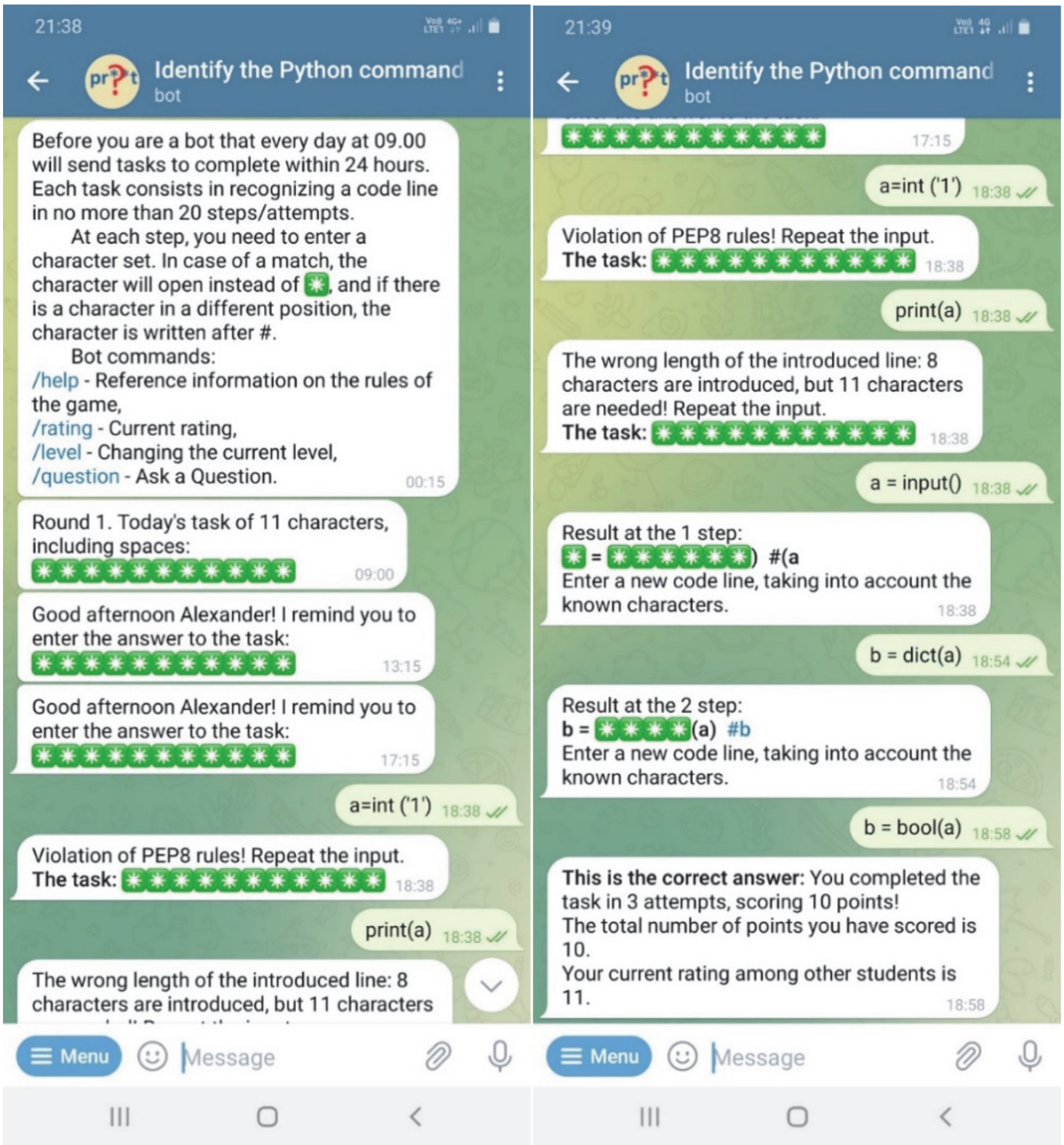

Fig. 1. Operation of the gaming chatbot "Identify the Python command".

8. A rating is maintained and after completing the task, the number of scores scored for the current task, the total number of scores and the student's current position in the rating are displayed.

The gameplay of "Identify the Python command" is as follows.

1. Every day at 09.00 the bot sends a task. The duration of the task within 24 hours is not limited in any way.

2. The bot tracks the task completion, the entered attempts and, if there are none, sends a reminder about the need to complete the task twice a day.

3. Text is entered in the Telegram input line, no keyboards other than the standard one for entering text on a smartphone are used.

4. There is a functionality for answering students' questions.

5. The student's actions are recorded, statistics are collected on working with the chatbot: determining the number of steps taken when completing each task, points scored, changing the level.

6. Interaction with the gaming chatbot is aimed at supporting the development of hard skills (memorization and recall of commands, functions, methods of the programming language), as well as cognitive skills (working memory, attention and logical thinking) [48].

4. MODELS DEVELOPMENT

4.1. Conducted Experiment

The experiment involved 60 first-year IT students studying the Python programming language as part of a two-semester course. Most of the students had no programming experience when they began their studies at the university. The experiment itself took place in the second semester.

The experiment used a non-intelligent game chatbot "Identify the Python command" and 2 questionnaires. The indicators in the questionnaires were assessed using a 5-point Likert scale.

Stages of the experiment.

1. Filling out the first questionnaire and indicating self-assessment of the level of writing programs in Python and the complexity of the course.
2. Using the chatbot for 10 days outside the classroom and solving one puzzle per day. Automatic collection of data on the progress and results of solving tasks for each student.
3. Filling out the second questionnaire and indicating the interest and difficulty of the tasks, sensation of involvement of working memory, attention and logical thinking when solving tasks.

4.2. Developed Models

To build the models, we used three data sources: the first questionnaire, the second questionnaire, and statistics on students' work with the gaming chatbot.

The following indicators were obtained from the first questionnaire:

- Self-assessment of Python coding skill (SAWP);
- Assessment of Python course difficulty (ACD).

The following indicators were obtained from the second questionnaire:

- Self-assessment of sensation of working memory involvement in solving tasks (SAWM);
- Self-assessment of sensation of logical thinking involvement in solving tasks (SALT);
- Self-assessment of sensation of remembering Python function in solving tasks (SARF);
- Perceived difficulty of tasks (PDT);
- Interest in tasks (PIT).

The following indicators were obtained from the student work statistics:

- Final level of tasks in the gaming chatbot (FLT);
- Number of steps/attempts spent on solving a separate task (NS);
- Average number of steps spent on solving tasks (ANS);
- Final number of points for completing tasks (SRT).

We assume that the PDT perceived difficulty of tasks is influenced by SAWM, SALT, SARF, FLT (Fig. 2). This influence can be explained as follows. If a student was at 2 or 3 levels, he faced more difficult tasks, compared to those who were at 1 or 2 levels and were not at 3. Therefore, such students will obviously rate the tasks as more difficult, i.e. FLT influences PDT. However, it should be taken into account that if a student cannot solve a task while at a level, he will most likely move to a lower level. Working memory and logical thinking play a key role in solving various problems. To successfully complete tasks, it is necessary to remember Python functions and commands and apply logical reasoning at each step (attempt) when open symbols appear. Working memory will

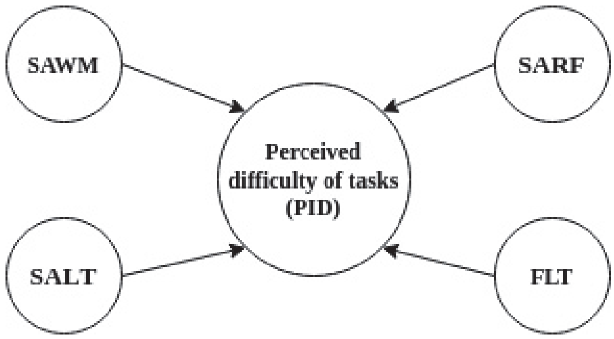

Fig. 2. Model of the influence of indicators on the perceived difficulty of tasks.

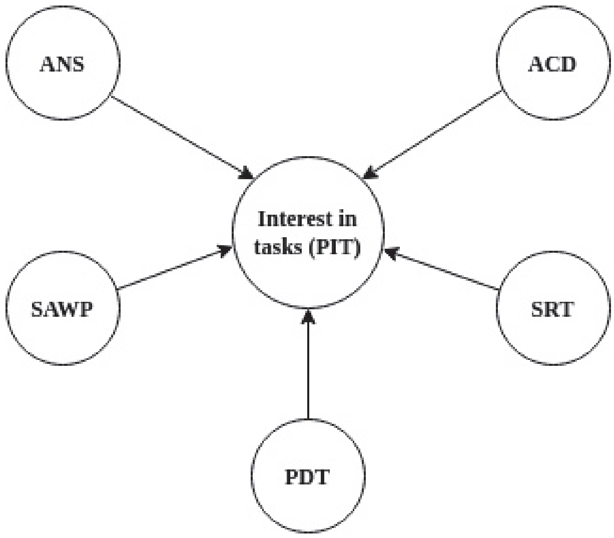

Fig. 3. Model of the influence of indicators on the interest in tasks.

ensure the process of solving tasks, so SAWM will influence PDT. It should be noted that simple tasks, unlike complex ones, do not exert a significant load on working memory. In addition to working memory, the student's level of logical thinking and the number of previously memorized Python functions and commands will affect the perceived difficulty of the tasks. With a low level of logical thinking and a low number of memorized functions, the student may experience difficulties in solving even problems of average complexity, and the tasks will be perceived as difficult. Hence the influence of SARF and SALT on PDT [49].

We assume that PIT interest in tasks is influenced by ANS, SAWP, PDT, ACD, SRT (Fig. 3). This influence can be explained as follows. A large number of attempts spent on solving a task may indicate both high task complexity and the use of brute force. Both of these options negatively affect the interest in the tasks being performed. Therefore, ANS affects PIT. To successfully write code, students must know and remember the functions of the Python language. Students who rate their programming skills highly usually know a sufficient number of functions and have developed logical and algorithmic thinking. Therefore, solving problems will be more interesting for them than for students who are unsure of their programming abilities, are not familiar with a sufficient number of functions, and do not have developed logical and algorithmic thinking. Therefore, SAWP affects PIT. If students spend a lot of time solving complex problems, especially if they have a low level of working memory and logical thinking, this can cause some discomfort and reduce the level of interest. Therefore, PDT affects PIT. The use of gamification tools and digital games can reduce the

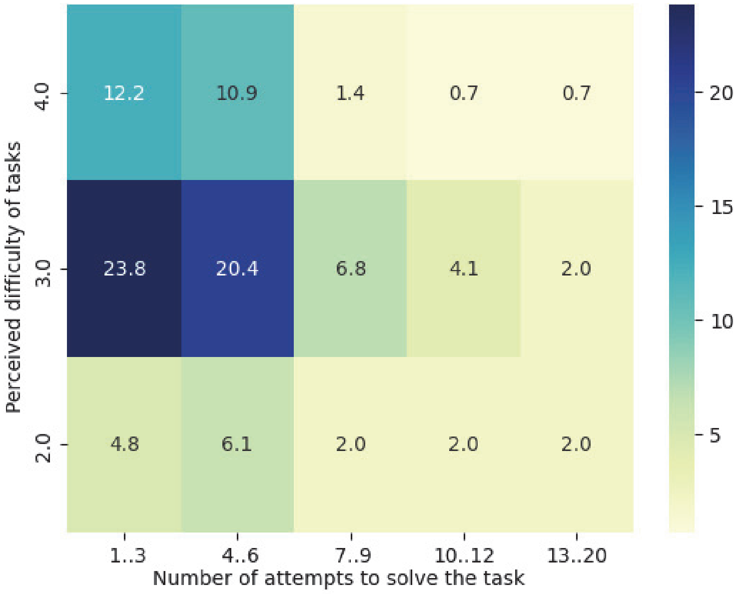

Fig. 4. Heat map of the frequency of occurrence of combinations of perceived task difficulty and the number of attempts to solve tasks.

cognitive load in complex courses. Therefore, ACD affects PIT. The more points a student scored as a result of working with a game chatbot, the more engaged he was and the more interesting it was for him to work with the chatbot. Thus, SRT affects PIT [49].

We conducted a regression analysis and built models of the impact of the parameters on the interest and perceived difficulty of the gaming chatbot. To assess the importance and contribution of each parameter to the value of the dependent variable, standardization was used with a mathematical expectation of 0 and a standard deviation of 1. The dependent variables were not standardized.

To assess the perceived difficulty of the gaming chatbot tasks, a linear regression with a significance level of $p < 0.01$ was obtained:

$$PDT = a_{SAWM} \cdot SAWM + a_{SARF} \cdot SARF + a_{SALT} \cdot SALT + a_{FLT} \cdot FLT + a_0^{PDT}, \tag{1}$$

where the weighting coefficients take the following values:

$$a_{SAWM} = 0.63^{***}, \quad a_{SARF} = -0.52^{**}, \quad a_{SALT} = -0.41^{**}, \quad a_{FLT} = 0.23^{*},$$
$$a_0^{PDT} = 3.07^{***} \ ({}^{***}{-p} < 0.001, {}^{**}{-p} < 0.01, {}^{*}{-p} < 0.1).$$

The coefficient of determination of the model was 0.82 (adjusted — 0.74).

To assess the interest in the gaming chatbot tasks, a linear regression with a significance level of $p < 0.001$ was obtained:

$$PIT = a_{ANS} \cdot ANS + a_{SAWP} \cdot SAWP + a_{PDT} \cdot PDT + a_{ACD} \cdot ACD + a_{SRT} \cdot SRT + a_0^{PIT}, \tag{2}$$

where the weighting coefficients take the following values:

$$a_{ANS} = -0.68^{**}, \quad a_{SAWP} = 0.64^{**}, \quad a_{PDT} = -0.47^{*},$$
$$a_{ACD} = 0.44^{*}, \quad a_{SRT} = 0.27^{*}, \quad a_0^{PIT} = 3.73^{***}.$$

The coefficient of determination of the model was 0.86 (adjusted — 0.78).

In the model (1), the SAWM, SALT, SARF indicators change little over a short period of time. The FLT indicator characterizes the final level of tasks in the chatbot. With different FLT values, different values of the number of steps to solve tasks are observed. So, with $FLT = 2$, the average value of $ANS = 5.4$, and with $FLT = 3$, the average value of $ANS = 5.0$.

In the model (2), the SAWP, ACD indicators change little over a short period of time. The SRT indicator characterizes the final number of points scored for completing tasks in the chatbot. This indicator is correlated with ANS, and the Pearson correlation coefficient is −0.43. The PDT indicator used in the model (2) is the output of the model (1).

We constructed a heat map that shows the frequency of occurrence of various combinations of perceived task difficulty and the number of attempts used to solve the tasks. We plotted the frequencies on the heat map as a percentage of the total number of tasks solved by students (Fig. 4).

The heat map showed that the highest frequencies of combinations of PDT values and the number of attempts to solve tasks are observed among students who solved tasks in 1–6 attempts and for whom PDT is medium or high. An increase in the number of attempts is observed among students for whom the perceived difficulty of the gaming chatbot tasks was medium.

Therefore, it is possible to propose to use models (1) and (2) in the task issuance control algorithm based on the ANS value.

4.3. Limitations of the Developed Models

Despite the statistical significance of the obtained results, it is necessary to take into account a number of limitations associated with the developed models.

- Limited sample of students. The experiment was conducted on a limited sample of students (60 people), which may not reflect the diversity of educational contexts and different groups of students. Students with different levels of training, motivation and previous experience of interacting with chatbots, programming in Python, using traditional puzzles may demonstrate different reactions to tasks, which limits the generalizability of the results.
- Limited set of factors. The models do not take into account demographic factors of students, the influence of external factors. These factors can affect the perception of difficulty and interest in tasks, but they were not included in the analysis. In addition, time parameters for completing tasks in the chatbot, such as duration or frequency of attempt input, were not used.
- Subjectivity of assessments. The perception of difficulty and interest in tasks are subjective and may vary among different students. The models are based on self-assessments by participants, which may introduce biases related to individual preferences.
- Linear dependencies. The assumption of a linear relationship between the parameters and perceived difficulty and interest in tasks may be an oversimplification. Real relationships may be more complex and nonlinear, which requires further research and more complex models.
- Long-term effects. The experiment does not take into account the long-term effects of using the gaming chatbot, such as the impact on academic performance, motivation, and the sustainability of interest in the learning process over time. This limits the ability to draw conclusions about the long-term effects of introducing such technologies into the educational process.

5. STRUCTURE OF THE GAMING CHATBOT WITH ARTIFICIAL INTELLIGENCE

The effectiveness of using the gaming educational chatbot in learning programming depends on a number of factors related to the characteristics of the students. These factors include:

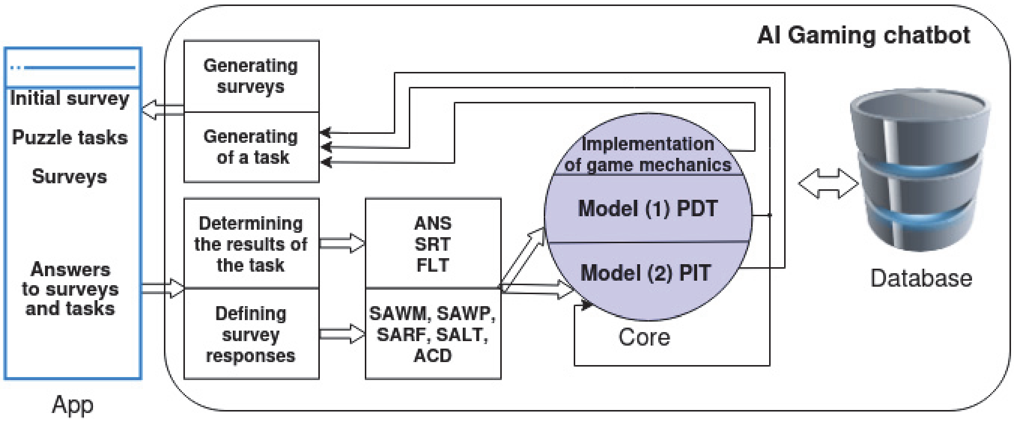

Fig. 5. Structure of the gaming chatbot with artificial intelligence.

- Programming proficiency: students with basic knowledge and experience with a programming language (in this case, Python) will benefit more from using the chatbot.
- Task-solving strategies and trajectories: students ability to effectively analyze tasks, plan solution steps, and use adequate algorithms is directly related to their success in solving tasks.

Perception of difficulty task is a key factor influencing interest in the chatbot. Perception of difficulty is influenced by:

- Cognitive abilities: the volume of working memory and the level of logical thinking play an important role in students ability to cope with tasks.
- Objective difficulty of task: the difficulty of the tasks should be adequate to the level of students preparation.
- Solution trajectory: the number of steps required to solve a problem can affect the perception of its difficulty.
- Knowledge of programming language functions and commands.

Based on this, we can conclude that the use of the gaming chatbot by beginners in programming is associated with a certain risk. And a certain "threshold" level of student preparation is required, including:

- Knowledge of basic Python functions and commands.
- Skills in reading and writing program code.

Students with a low level of working memory, logical thinking, or insufficient knowledge of Python commands may face increased difficulty of tasks, which will negatively affect their interest in the chatbot. Therefore, when introducing the gaming chatbot into the educational process, it is important to take into account the individual characteristics of students and provide a "threshold" level of preparation for its effective use.

It is possible to propose a structure of the gaming chatbot with models for managing Wordle-like puzzle for learning programming. Here, based on the results of the survey and data transfer to the chatbot, an assessment of the level of difficulty and interest of the tasks can be made, showing the feasibility of using the chatbot at the current moment in time. Accordingly, the chatbot, depending on the received values, will monitor the feasibility of use, and in case of feasibility, depending on the strategy used and the completion of tasks, it will select puzzles of the appropriate level (Fig. 5).

We presented the algorithm of the intelligent gaming chatbot (Algorithm 1) in the form of pseudocode.

Algotithm 1. Intelligent chatbot behavior

```
Input  : Models (1) and (2), survey and task completion results
Output: Chatbot usage recommendation, puzzle
Setting PDT_l and PIT_l thresholds.
Determining SAWM, SARF, SALT, SAWP, ACD values based on testing and survey
  results.
Using models (1) and (2) and determining initial values PDT_0 and PIT_0.
if PDT_0 > PDT_l or PIT_0 < PIT_l then
    Issuing a recommendation about undesirable and premature use of the chatbot at this
      point in time.
    Exit.
else
    Issuing the first puzzle.
    Monitoring the solution of the first puzzle.
    Determining NS.
end
if Puzzle solved then
    Based on NS and models (1), (2), the values of PDT, PIT are determined and a new
      puzzle is generated taking these values into account.
    As puzzles are solved, ANS is determined.
    The values of PDT, PIT are recalculated and each new puzzle is formed in accordance
      with them.
else
    Formation of a new puzzle.
end
```

6. CONCLUSION

Wordle-like puzzles can be used at the stage of learning programming. In general, they are quite an interesting educational tool for students. However, they should be used when the level of students' competencies is not low. In this case, solving puzzles will be exciting and useful. Also, this type of puzzles may not be very interesting for students with a low level of RAM and logical thinking.

Therefore, the introduction of machine learning models into the gaming chatbot, taking into account the student's indicators and the progress of the task, will allow the chatbot to predict the perceived difficulty of tasks and interest in them. Due to this, it is possible to adapt the behavior of the chatbot to the student's level in order to make the difficulty of the tasks optimal, and the interest — maximum.

The developed machine learning models formed the basis for the algorithm of operation and the structure of the game chatbot with artificial intelligence. Based on the developed structure, a new version of the game chatbot was implemented. Testing of the resulting chatbot showed great interest in its use among students studying programming.

Some parameters of the models (FLT, ANS, SRT, PDT) can be determined by the chatbot based on the results of completing the task and using the model 1. However, for some parameters (SAWM, SARF, SALT, SAWP, ACD), it is necessary to conduct surveys at least at the initial stage of connecting and launching the chatbot and, preferably, adjust them as the chatbot is used by more frequent surveys. In this regard, it is of interest to replace subjective indicators with objective ones and search for models that do not contain subjective indicators, or in which these indicators are expressed through the parameters of task completion. This will be the main direction of further work.

In our future work, we also plan to expand the sample and take into account other indicators in the models, in particular, demographic data of students and their experience with chatbots, Python programming and solving traditional puzzles. We plan to consider nonlinear models of the influence of indicators on the perceived difficulty and interest in tasks.

FUNDING

The publication was prepared within the framework of the Academic Fund Program at the HSE University (grant no. 22-00-052).

REFERENCES

1. Mayer, R.E., Computer Games in Education, *Ann. Rev. Psychol.*, 2019, vol. 70, pp. 531–549. https://doi.org/10.1146/ANNUREV-PSYCH-010418-102744

2. Zhan, Z., He, L., Tong, Y., et al., The effectiveness of gamification in programming education: Evidence from a meta-analysis, *Computers and Education: Artificial Intelligence*, 2022, vol. 3, p. 100096. https://doi.org/10.1016/J.CAEAI.2022.100096

3. Nouchi, R., Taki, Y., Takeuchi, H., et al., Brain Training Game Boosts Executive Functions, Working Memory and Processing Speed in the Young Adults: A Randomized Controlled Trial, *PLoS One*, 2013, vol. 8, p. e55518. https://doi.org/10.1371/JOURNAL.PONE.0055518

4. Huang, V., Young, M., and Fiocco, A.J., The Association Between Video Game Play and Cognitive Function: Does Gaming Platform Matter?, *Cyberpsychol Behav Soc Netw*, 2017, vol. 20, pp. 689–694. https://doi.org/10.1089/CYBER.2017.0241

5. Harteveld, C., Smith, G., Carmichael, G., et al., A design-focused analysis of games teaching computer science, *Proceedings of Games+Learning+Society*, 2014, pp. 1–8.

6. Parsons, D. and Haden, P., Parsons Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses, in *Tolhurst D. and Mann S. (eds.) Proceedings of the 8th Australasian Conference on Computing Education (Vol. 52)*, 2006, pp. 157–163. Australian Computer Society, Inc.

7. Murphy, M., Spillane, K., Cully, J., et al., Can Word Puzzles be Tailored to Improve Different Dimensions of Verbal Fluency? A Report of an Intervention Study, *J. Psychol.*, Elsevier Publ., 2016, vol. 150, pp. 743–754. https://doi.org/10.1080/00223980.2016.1182887

8. Friedlander, K.J. and Fine, P.A., Fluid Intelligence is Key to Successful Cryptic Crossword Solving, *Journal of Expertise*, 2020, vol. 3, pp. 101–132.

9. Juricic, V. and Radoševic, M., Puzzle-like programming languages in teaching programming, in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 — Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 542–546. https://doi.org/10.239/r/mipro.2019.8757192

10. Fakhoury, S., Ma, Y., Arnaoudova, V., et al., The Effect of Poor Source Code Lexicon and Readability on Developers Cognitive Load, in *2018 IEEE/ACM 26th International Conference on Program Comprehension (ICPC)*, Gothenburg, Sweden, 2018, pp. 286–28610.

11. Wassila, D. and Tahar, B., Using serious game to simplify algorithm learning, in *2012 International Conference on Education and e-Learning Innovations, ICEELI 2012*, 2012, pp. 1–5. https://doi.org/10.1109/ICEELI.2012.6360569

12. Liu, Y., Tan, J., Yang, Z., et al., Application Study of Collaborative Multi-Model in Wordle Game, in *2024 5th International Conference on Machine Learning and Computer Application (ICMLCA)*, 2024, pp. 591–596.

13. Zhang, Z., Wang, S., and Sun, D., A Study of Wordle Games Based on Mathematical Models, in *2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT)*, 2024, pp. 813–817.

14. Wang, S., Zhou, G., Zhang, Z., and Liu, J., Exploring Wordle: A Data-Driven Analysis of Gameplay Dynamics, Mathematical Modeling, and Educational Implications, in *Fourth International Conference on Applied Mathematics, Modelling, and Intelligent Computing (CAMMIC 2024)*, 2024.

15. Cunningham-Nelson, S., Boles, W., Trouton, L., and Margerison, E., A review of chatbots in education: Practical steps forward, in *30th Annual Conference for the Australasian Association for Engineering Education (AAEE 2019): Educators Becoming Agents of Change: Innovate, Integrate, Motivate*, Engineers Australia, Brisbane, Queensland, 2019, pp. 299–306.

16. Khamparia, A. and Pandey, B., Effects of visual map embedded approach on students learning performance using Briggs-Myers learning style in word puzzle gaming course, *Computers & Electrical Engineering*, 2018, vol. 66, pp. 531–540. https://doi.org/10.1016/J.COMPELECENG.2017.12.041

17. Setiawan, H. and Phillipson, S., The effectiveness of Game-Based Science Learning (GBSL) to improve students academic achievement: A meta-analysis of current research from 2010 to 2017, *REID (Research and Evaluation in Education)*, 2019, vol. 5, pp. 152–168. https://doi.org/10.21831/REID.V5I2.28073

18. Karakoç, B., Eryılmaz, K., Turan Özpolat, E., et al., The Effect of Game-Based Learning on Student Achievement: A Meta-Analysis Study, *Technology, Knowledge and Learning*, 2020, vol. 27, pp. 207–222. https://doi.org/10.1007/S10758-020-09471-5

19. Connolly, T.M., Boyle, E.A., MacArthur, E., Hainey, T., and Boyle, J.M., A systematic literature review of empirical evidence on computer games and serious games, *Comput. Educ.*, 2012, vol. 59, pp. 661–686. https://doi.org/10.1016/J.COMPEDU.2012.03.004

20. Vahldick, A., Mendes, A.J., and Marcelino, M.J., A review of games designed to improve introductory computer programming competencies, in *Proceedings — Frontiers in Education Conference, FIE*, 2014, pp. 1–7. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/FIE.2014.7044114

21. Miljanovic, M.A. and Bradbury, J.S., A review of serious games for programming, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11243 LNCS, pp. 204–216. https://doi.org/10.1007/978-3-030-02762-9_21/COVER

22. Giannakoulas, A. and Xinogalos, S., A Review of Educational Games for Teaching Programming to Primary School Students, in *Handbook of Research on Tools for Teaching Computational Thinking in P-12 Education*, 2020, pp. 1–30. https://doi.org/10.4018/978-1-7998-4576-8.ch001

23. Falkner, N., Sooriamurthi, R., and Michalewicz, Z., Puzzle-based learning for engineering and computer science, *Computer*, 2010, vol. 43, pp. 20–28. https://doi.org/10.1109/MC.2010.113

24. Toma, M., Halpern, D.F., and Berger, D.E., Cognitive Abilities of Elite Nationally Ranked SCRABBLE and Crossword Experts, *Applied Cognitive Psychology*, 2014, vol. 28, pp. 727–737. https://doi.org/10.1002/ACP.3059

25. Hambrick, D.Z., Salthouse, T.A., and Meinz, E.J., Predictors of crossword puzzle proficiency and moderators of age-cognition relations, *Journal of Experimental Psychology: General*, 1999, vol. 128, pp. 131–164. https://doi.org/10.1037/0096-3445.128.2.131

26. Makasi, T., Nili, A., Desouza, K.C., and Tate, M., A Typology of Chatbots in Public Service Delivery, *IEEE Software*, 2022, vol. 39, pp. 58–66. https://doi.org/10.1109/MS.2021.3073674

27. Marikyan, D., Papagiannidis, S., Rana, O.F., et al., "Alexa, let's talk about my productivity": The impact of digital assistants on work productivity, *Journal of Business Research*, 2022, vol. 142, pp. 572–584. https://doi.org/10.1016/J.JBUSRES.2022.01.015

28. Kimani, E., Rowan, K., McDuff, D., Czerwinski, M., and Mark, G., A Conversational Agent in Support of Productivity and Wellbeing at Work, in *2019 8th International Conference on Affective Computing and Intelligent Interaction, ACII 2019*, 2019, pp. 332–338. https://doi.org/10.1109/ACII.2019.8925488

29. Gkinko, L. and Elbanna, A., Hope, tolerance and empathy: employees emotions when using an AI-enabled chatbot in a digitalised workplace, *Information Technology and People*, 2022, vol. 35, pp. 1714–1743. https://doi.org/10.1108/ITP-04-2021-0328/FULL/PDF

30. Bavaresco, R., Silveira, D., Reis, E., et al., Conversational agents in business: A systematic literature review and future research directions, *Computer Science Review*, 2020, vol. 36, article 100239. https://doi.org/10.1016/J.COSREV.2020.100239

31. Brandtzaeg, P.B. and Følstad, A., Why people use chatbots, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10673, pp. 377–392. https://doi.org/10.1007/978-3-319-70284-1_30

32. Fotheringham, D. and Wiles, M.A., The effect of implementing chatbot customer service on stock returns: an event study analysis, *Journal of the Academy of Marketing Science*, 2023, vol. 51, pp. 802–822. https://doi.org/10.1007/S11747-022-00841-2/TABLES/6

33. Ciechanowski, L., Przegalinska, A., Magnuski, M., and Gloor, P., In the shades of the uncanny valley: An experimental study of human-chatbot interaction, *Future Generation Computer Systems*, 2019, vol. 92, pp. 539–548. https://doi.org/10.1016/J.FUTURE.2018.01.055

34. Thakkar, J., Raut, P., Doshi, Y., and Parekh, K., Erasmus AI Chatbot, *International Journal of Computer Sciences and Engineering*, 2018, vol. 6, pp. 498–502. https://doi.org/10.26438/IJCSE/V6I10.498502

35. Sidaoui, K., Jaakkola, M., and Burton, J., AI feel you: customer experience assessment via chatbot interviews, *Journal of Service Management*, 2020, vol. 31, pp. 745–766. https://doi.org/10.1108/JOSM-11-2019-0341/FULL/PDF

36. Zhou, L., Paul, S., Demirkan, H., et al., Intelligence Augmentation: Towards Building Human-Machine Symbiotic Relationship, *AIS Trans. Hum.-Comput. Interact.*, 2021, vol. 13, pp. 243–264. https://doi.org/10.17705/1thci.00149

37. Gkinko, L. and Elbanna, A., The appropriation of conversational AI in the workplace: A taxonomy of AI chatbot users, *Int. J. Inf. Manage.*, 2023, vol. 69, p. 102568. https://doi.org/10.1016/J.IJINFOMGT.2022.102568

38. Rizomyliotis, I., Kastanakis, M.N., Giovanis, A., et al., "How mAy I help you today?" The use of AI chatbots in small family businesses and the moderating role of customer affective commitment, *J. Bus. Res.*, 2022, vol. 153, pp. 329–340. https://doi.org/10.1016/J.JBUSRES.2022.08.035

39. Chen, Q., Gong, Y., Lu, Y., and Tang, J., Classifying and measuring the service quality of AI chatbot in frontline service, *J. Bus. Res.*, 2022, vol. 145, pp. 552–568. https://doi.org/10.1016/J.JBUSRES.2022.02.088

40. Tam, W., Huynh, T., Tang, A., et al., Nursing education in the age of artificial intelligence powered Chatbots (AI-Chatbots): Are we ready yet?, *Nurse Educ. Today*, 2023, vol. 129, p. 105917. https://doi.org/10.1016/J.NEDT.2023.105917

41. Zhu, Y., Zhang, J., Wu, J., and Liu, Y., AI is better when I'm sure: The influence of certainty of needs on consumers' acceptance of AI chatbots, *J. Bus. Res.*, 2022, vol. 150, pp. 642–652. https://doi.org/10.1016/J.JBUSRES.2022.06.044

42. Ruan, Y. and Mezei, J., When do AI chatbots lead to higher customer satisfaction than human frontline employees in online shopping assistance? Considering product attribute type, *J. Retail. Consum. Serv.*, 2022, vol. 68, p. 103059. https://doi.org/10.1016/J.JRETCONSER.2022.103059

43. Janssen, A., Rodríguez Cardona, D., Passlick, J., and Breitner, M.H., How to Make chatbots productive — A user-oriented implementation framework, *Int. J. Hum. Comput. Stud.*, 2022, vol. 168, p. 102921. https://doi.org/10.1016/J.IJHCS.2022.102921

44. Lee, D. and Yeo, S., Developing an AI-based chatbot for practicing responsive teaching in mathematics, *Comput. Educ.*, 2022, vol. 191, p. 104646. https://doi.org/10.1016/J.COMPEDU.2022.104646

45. Liu, C.C., Liao, M.G., Chang, C.H., and Lin, H.M., An analysis of childrens interaction with an AI chatbot and its impact on their interest in reading, *Comput. Educ.*, 2022, vol. 189, p. 104576. https://doi.org/10.1016/J.COMPEDU.2022.104576

46. Johnson, C., Urazov, M., and Zanoli, E., Escapeling: A Gamified, AI-Supported Chatbot for Collaborative Language Practice, *Lect. Notes Networks Syst.*, 2022, vol. 349, pp. 141–148. https://doi.org/10.1007/978-3-030-90677-1_14/COVER

47. Varnavsky, A.N., Comparison of perception indicators of the gaming chatbot with puzzles tasks for teaching programming with perception indicators of other educational tools, *Informatics and education*, 2024, vol. 39(4), pp. 41–50. (In Russ.) https://doi.org/10.32517/0234-0453-2024-39-4-41-50

48. Varnavsky, A.N., Using the gaming chatbot to develop algorithmic and logical thinking skills, *Informatics and education*, 2024, vol. 39(2), pp. 48–58. (In Russ.) https://doi.org/10.32517/0234-0453-2024-39-2-48-58

49. Varnavsky, A.N., Models of Interest, Difficulty, and Perceived Usefulness of the Gaming Chatbot with Wordle Like Puzzles for Learning Programming, *Information technologies*, 2024, vol. 7, no. 30, pp. 372–381. (In Russ.) https://doi.org/10.17587/it.30.372-381

This paper was recommended for publication by A.A. Galyaev, a member of the Editorial Board