Exploring the tree algorithms to generate the optimal detection system of students' stress levels

Yuni Yamasari[1], Anita Qoiriah[1], Naim Rochmawati[1], Aditya Prapanca[1], Agus Prihanto[1], I Made Suartana[1], Tohari Ahmad[2]
[1]Department of Informatics, Faculty of Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia
[2]Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received Dec 13, 2024
Revised Jun 8, 2024
Accepted Jun 25, 2024

Keywords:

Bot telegram
ID3
Stress
Student
Tree algorithm

ABSTRACT

The significant changes in the world of education after the coronavirus disease 2019 (COVID-19) pandemic have increased students' anxiety levels. This anxiety can trigger stress which can interfere with students' academic performance. Therefore, this condition is a critical problem that needs to be addressed immediately. However, researchers have not previously conducted much research to detect post-COVID stress levels. Apart from that, the existence of a system capable of carrying out this detection is still lacking. Therefore, this research focuses on building a system for detecting student stress levels. First, an exploration of the tree algorithm was carried out to find the most optimal method for recognizing student stress levels. Then a detection system is built using this optimal method. The research results show that the tree ID3 (Iterative Dichotomiser 3) algorithm achieves the highest accuracy value of 95% compared to other tree algorithms with the scenario of dividing training data into test data of 80%:20%. Moreover, this telegram bot-based detection system works well in recognizing three categories of stress, namely: light, moderate, and heavy stress based on black-box testing techniques.

Corresponding Author:

Yuni Yamasari
Department of Informatics, Faculty of Engineering, Universitas Negeri Surabaya
Surabaya, Jawa Timur 60231, Indonesia
Email: yuniyamasari@unesa.ac.id

1. INTRODUCTION

The end of the coronavirus pandemic has changed all aspects of life from previously limited to normal again. However, it is not easy for people to adapt to this situation because they are used to life during the coronavirus. This adaptation difficulty also occurs in the field of education [1]. Based on the decision of the Indonesian government through 3 ministers, namely: the Minister of Education and Culture, the Minister of Religion, and the Minister of Health (2021), it was stated that in line with the improvement of the coronavirus, the education system in Indonesia was implemented using a limited face-to-face model, with continued implementation and attention to health protocols. This change continues today, and many educational institutions in Indonesia have implemented full-time face-to-face learning.

In higher education, the significant changes due to coronavirus disease 2019 (COVID-19) have become a problem for students and educational staff. One of them is triggering anxiety with this significant change. Besides that, students' stress levels are also increasing due to very rapid changes in educational patterns and systems [2], [3]. For example, final-year students who previously did not receive optimal learning during the pandemic suddenly had to do face-to-face meetings for internships, and theses. Meanwhile, many students in the lower semester went through their initial learning period online and suddenly had to do face-to-face learning. Based on previous research, students' stress levels when taking virtual classes during the pandemic were 55% moderate and 30.2% high. Female students have a significant correlation with high levels of stress [4]. Other research analyses stress levels in online learning for all levels of school, specifically sports subjects. The results of the analysis show that junior- and senior-high-school levels have higher levels of stress compared to elementary school [5]. For the senior high school level, other research specializes in analyzing the relationship between stress levels and gender in the learning process during the pandemic. The research results show that there is no difference in the stress levels of women and men [6]. Furthermore, other research also measured stress levels in university students during the pandemic, and the results showed that the aspect with the highest category was the aspect of change [7]. The next research also focused on the causes of stress among students related to their difficulty managing time due to the COVID-19 pandemic. Furthermore, this research also seeks to provide solutions related to stress management [8].

Research on the relationship between stress levels and the pandemic has encouraged other researchers to focus on research on stress levels related to student learning after the pandemic. The research results showed an increase in study-related stress levels after the pandemic occurred [8]. Other research also focuses on students' psychological stress post-COVID. The analysis carried out is related to psychological pressure due to the online learning process during and after the pandemic. Students feel afraid because online learning can make them graduate late. Other research also focuses on the impact of online learning during the pandemic and post-pandemic. The research results concluded that there were extraordinary symptoms of stress regarding extreme fear of academic delays due to this learning model in Bangladesh [9].

This large amount of research indicates that detecting stress levels is a crucial problem that must be resolved. For this reason, with the development of various scientific disciplines, researchers are exploring methods to find the most optimal method. Several methods that have been explored by previous researchers can be categorized into two categories: supervised learning and unsupervised learning algorithms. Previous research exploring supervised learning methods includes Random forest and regression models for stress detection using public AffectiveROAD data [10] and tree method for stress analysis [11], [12]. Other researchers use methods that are categorized as unsupervised learning. For example, the popular K-means method, density-based spatial clustering of applications with noise (DBSCAN), is intended for stress monitoring [13]. Then, the clustering method is also used to detect high levels of stress aimed at health [14].

Specifically in the education field, several methods have also been applied to overcome this stress problem. For example, the regression method is used to analyze stress levels in student learning [15] and to predict six stress scales in students [16]. Machine learning (ML) can also be applied to stress prediction [17], [18]. The classification and regression tree method is applied to detect student stress during the exam [19], [20]. Then, the text mining method is implemented to find key attributes that influence the stress and mental health of taekwondo student-athletes [21]. Furthermore, data mining methods can be applied to look for stress triggers in student-athletes when they face psychological stress in adjusting to campus life [22] and characteristics of stress levels in engineering students [23]. Other research also applies data mining, especially one-class classification, to diagnose stress levels based on four levels (low, medium-low, medium-high, and high) so that decreasing student performance does not happen [24]. Data mining, especially the K-means method, can be applied to determine stress levels at the senior high school level [25]. Data mining techniques are also applied to assess and predict levels of psychological stress in students [26].

Overall, the methods explored are from various scientific disciplines, including artificial intelligence (AI), ML, and data mining. However, only a few previous studies focused on system or application development. Detecting a person's stress level will be very useful if it is built into a system that is easy for users to use. For example, an expert system for determining stress levels during learning at the high school level is website-based [27]. This website-based system has several weaknesses, one of which is the lack of direct involvement from users because notifications are not real-time. Regarding the detection of stress levels, a system based on a Telegram or Android bot platform could be a solution that needs to be considered. With the Telegram bot, users can detect their stress levels without having to install the application again. Apart from that, the automatic chat feature makes users feel like they are having a consultation or test with a psychology expert. Telegram itself is a short messaging application that is currently popular among the public. The advantage of Telegram is that it provides an API that can be used by the wider community, such as this Telegram bot. This bot can reply to short messages automatically with its AI feature [28]. To run the bot, users only need to use Telegram without having to install other applications.

Therefore, this research focuses on building a stress level detection system using the Telegram bot platform, that continues our previous research. This system is intended so that students understand and know what attitude to take according to their psychological state. Further, to find the best performance of the system, we explore tree algorithms. In our case, the best algorithm to detect the level of stress is ID3. Then, the ID3 method is used to produce trees for creating patterns from the rules. Model ID3 is trained on a dataset comprising various stress indicators, allowing for more accurate and tailored assessments.

Furthermore, our approach involves the design and development of a Telegram bot-based system capable of assessing student stress levels in real-time. Leveraging ID3 methodology, the system analyzes user input to generate personalized stress assessments and recommend appropriate interventions. By utilizing the Telegram bot platform, we ensure widespread accessibility and ease of use, enabling students to receive support without the need for additional applications. So, with this detection system based on a telegram bot, students can find out early about their psychological condition and can immediately find the right solution so that this condition does not interfere with their learning achievement.

2. MATERIAL AND METHOD

In this chapter, we explain questionnaire preparation and the techniques employed for processing questionnaire results and data labeling. Next, the proposed methodology is described, consisting of modeling and development algorithm. Additionally, the architecture of the stress level detection system is detailed, showcasing the steps involved in data processing and system implementation.

2.1. Data preparation

This sub-chapter describes about questionnaire preparation, data collection and data labelling. For questionnaire, our research utilizes an 8-question questionnaire derived from prior research. Then, the questionnaire is distributed to respondents who are students at our university via Google Forms. There are 4 answer options in this questionnaire, namely strongly agree (SS), agree (S), disagree (TS), and strongly disagree (STS) with the scale as presented in Table 1.

After the data collection process through questionnaires is complete, the labeling process is done because this research employs a supervised learning method requiring labeled data.

Labeling is conducted in two steps:

a) Conversion of answers to numerical values as per Table 1.
b) Score calculation based on the Table 2 formula, utilizing the respondent's total score, mean, and standard deviation.

Where x, μ, and SD are the total score of the respondent's answers, mean and standard deviation, respectively.

Table 1. Respondent rating scale

Scoring scale	Score
Strongly disagree (STS)	1
Disagree (TS)	2
Agree (S)	3
Strongly agree (SS)	4

Table 2. Three-tier categorization for data labeling

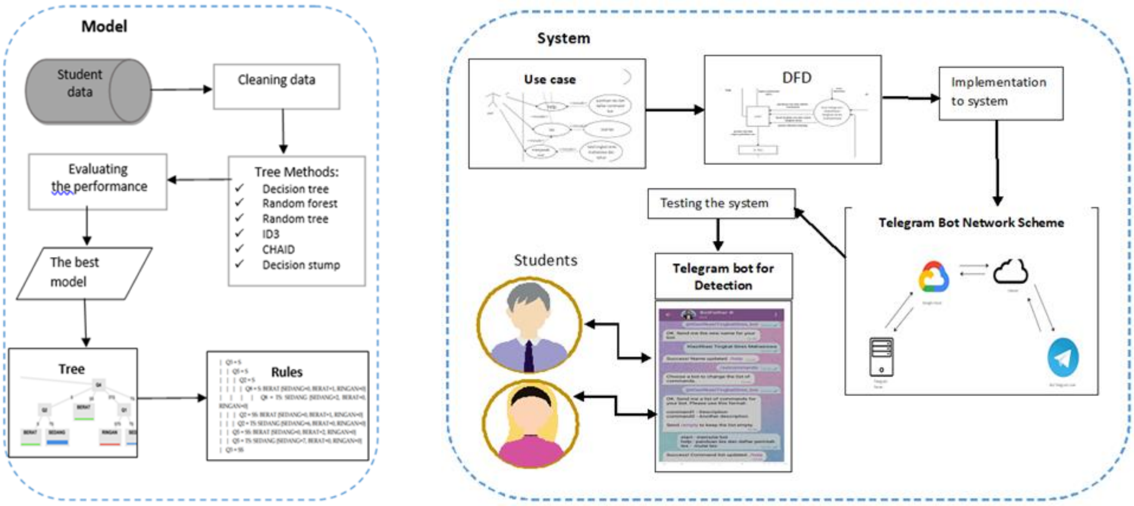
Category labels	Formulation
Ringan (light)	x<(μ-SD)
Sedang (moderate)	$(\mu - SD) <= x < (\mu + SD)$
Tinggi (heavy)	$x >= \mu + SD)$

2.2. The proposed architecture

This sub-section describes the proposed architecture of stress level detection, consisting of two main parts: modeling and system building, as shown in Figure 1. Figure 1(a) shows that the modeling algorithm has many steps. In detail, after student data is ready, the next step is data cleaning, which cleans the data so there are no duplicates and no noise. Then, the mining process is executed on the student data using many classifiers. They are categorized as the tree family, namely: ID3, random forest, random tree, CHAID, decision tree, and decision stump. We explore tree algorithms because of the popular algorithms in educational data mining [29], [30]. Exploration with many methods is aimed at finding the most optimal model performance. The built models are evaluated using percentage split techniques and several performance measures to measure optimality. For performance measures, this study uses accuracy, recall, and precision measures.

These methods in the family tree are explored to build models and then the performance of the models is measured using percentage split as its evaluation technique to produce optimal models. There are two critical steps: generating a tree and transforming it into rule patterns. The tree generation step utilizes the highest-performing model to build a hierarchical tree structure, followed by the extraction of rule patterns for easier interpretation. These rule patterns are then converted into if-then forms to facilitate implementation.

Subsequently, the stress level detection system is designed based on these rules. Before they are implemented to build a detection system, the use case and DFD are defined early. Development is carried out based on Telegram bots as shown Figure 1(b). There are 2 methods for building a Telegram bot, namely webhooks and long-polling [31]. Several researchers are interested in building Telegram bots with their webhooks because this technique has advantages compared to long polling, for example: real-time response, efficiency and performance, greater control over logic, and so on [32], [33].



(a) (b)

Figure 1. The proposed architecture of stress level detection (a) modeling algorithm and (b) development algorithm

This Telegram Bot-based stress level detection begins with the display of a welcome message to the user. Then the user will be directed to see the help menu to read and understand the use of the bot first, but the user can also see other menus via the menu list display on the menu button. Users can see the test menu to start detection by answering questions. Each question will be displayed one by one, and the user must answer according to the question being displayed. When all the questions have been answered, the Telegram bot detecting student stress levels will carry out an analysis of the user's answers. So, the results of the stress level and solutions to the stress level experienced by the user will be displayed immediately after all the questions have been answered. Regarding the input and output involved in the system, the first process is starting the bot, the input data of which is a command start, and the output data is a welcome message. Then, the next process is help, where the input data is a help command and the output data is a test guide and a list of commands that apply to the Telegram bot. For the test process, the input data is test command input data and test answer data, and the output data is test questions. After all questions are answered, the stress level detection results and solutions appear. The features in this telegram bot-based stress level classification system for students appear to be interconnected with each other. An overview of user interaction with the system is explained through this image, where users can connect to the system through several features of the telegram bot for classifying stress levels in students, namely through the start, help, test, and question-answering features.

Then, the system will respond by displaying a welcome greeting, test guide, and questions as well as test results and solutions from the user. The next step observes the data flow in detecting the stress level, so we use the data flow diagram. Relate to the building of telegram bots, this research classifies students' stress levels using the Webhook method. This method allows the application to perform a task automatically after being triggered by certain conditions (events) and uses hosting from Google. The telegram server containing the script is placed on hosting which in this research uses Google Cloud. This is so that Telegram bots can be accessed by users using the internet network.

After the system is successfully created, system testing is carried out. To test the system, testing is carried out using the black box testing method. Testing using a black box testing method like this is carried out to test the functionality of the system being created. The system will be tested whether it can run according to the specified flow or not. In this test, several user acceptances must be achieved for each feature to meet the requirements to pass the test. At this stage, a test scenario is also created in the form of a test case that must be carried out by the user during the testing stage. Some user acceptance in testing this system includes:

i) The telegram bot can respond to commands from users
ii) The telegram bot can display test questions
iii) The telegram bot can display the results of the user's stress level classification
iv) The telegram bot can provide suggestions for what the user should do according to the level of stress the user is experiencing.

If this testing result is successful, then the system can be used by students as users. However, if the test is not successful, the system will be revised until the testing result is successful.

3. RESULTS AND DISCUSSION

This chapter explains the results of data collection until the data is ready to be mined. Then the description continued with the exploration result of the tree methods on student data. Implementing the results to build a Telegram chatbot and testing the Telegram bot system.

3.1. Student data description

This research involves 102 students as respondents. Each respondent answers a questionnaire consisting of 8 questions. However, among the 102 respondents' data, 2 respondents did not complete the questionnaire. Thus, the respondent data that is further processed in this research amounted to 100 respondent data. The summary results of respondents' answers to each questionnaire question are presented in Table 3.

The raw data above must be converted into numerical form so that labeling can be carried out. Labeling is done by calculating the total value of respondents' answers, then categorization is carried out using a formula categorization of 3 levels as explained in the materials and methods chapter. Before carrying out calculations, the answer options (Likert scale) must be converted into numbers first. Then the scores from each respondent are added up. The following are the values for each answer option: strongly agree=4, agree=3, disagree=2, strongly disagree=1.

Then the scores are added up and the maximum and minimum scores are found from the existing data. So that the maximum and minimum scores obtained are as follows:

$$Min\ score\ =\ lowest\ score\ scale\ *\ number\ of\ questions$$
$$=\ 1\ *\ 8\ =\ 8$$
$$Max\ score\ =\ highest\ score\ scale\ *\ number\ of\ questions$$
$$=\ 4\ *\ 8\ =\ 32$$

Then the mean and standard deviation are calculated:

$$\mu\ =\ (Max.score\ +\ Min.score$$
$$=\ (32+8)/2\ =\ 20$$
$$SD = \frac{Max.score\ -\ Min.score}{6}$$
$$=\ (32-8)/6\ =\ 4$$

So, the categorization of student stress level data is as follows:

Light: $x\ <\ 1$
Moderate: $16\ <=\ x\ <\ 24$
Heavy: x >= 24

Where the value of x is the total score obtained by each respondent. So, in calculating the case of stress level in students, the following results are obtained: the results of labeling 100 respondents were as follows: 66 respondents experienced moderate levels of stress, 28 respondents had heavy levels of stress, and only 6 respondents had light levels of stress. Part of the labeling results are presented in Table 4.

Table 3. Composition of respondents' answers to each questionnaire question

Question	Strongly agree	Agree	Disagree	Strongly disagree
1. I felt pressured by my parents' demands for me to graduate immediately.	14	45	36	7
2. My friends always invite me to play, so I feel like I can't focus on doing my assignments or my thesis.	7	29	59	7
3. I had difficulty sleeping and often stayed up late.	32	37	30	3
4. I am experiencing financial difficulties or my financial situation is limited.	20	43	29	9
5. I found it difficult to understand the material presented by the lecturer.	9	55	36	1
6. I find it difficult when I have to look for literature and references for assignments or theses.	12	53	33	3
7. I feel like I can't play or rest freely.	12	45	37	7
8. I feel uncomfortable at home because the noise makes me less focused.	12	42	37	10

Table 4. Part of the labelling process results

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Sum	Label
4	2	4	4	4	4	3	3	28	*Berat* (heavy)
2	2	2	2	3	3	3	3	20	*Sedang* (moderate)
2	2	3	4	3	3	3	3	23	*Sedang* (moderate)
4	2	3	3	3	3	2	3	23	*Sedang* (moderate)
3	3	2	3	3	2	3	2	21	*Sedang* (moderate)
...	...	...	...	...	...	...	...	...	...
3	2	4	3	3	3	3	3	24	*Berat* (heavy)
1	1	2	2	3	2	1	3	15	*Ringan* (light)
2	2	4	1	3	2	4	2	20	*Sedang* (moderate)
3	3	4	4	3	3	3	3	26	*Berat* (heavy)
2	2	3	3	3	3	2	2	20	*Sedang* (moderate)

3.2. Exploring tree algorithms

After labeling, the classification method is applied to the student data. There are 6 classifiers applied, namely: ID3, random forest, random tree, CHAID, decision tree, and decision stump. This exploration has the goal of generating the best model. Therefore, after the model is built, performance evaluation needs to be carried out. The evaluation technique used in this research is percentage split. Where data is divided into 2: train data and test data. Split data is also carried out in several experiments (scenarios), namely: by dividing the train data and test data as follows: 60%:40%, 65%: 35%, 70%:30%, 80%:20%, and 85%:15%.

In this research, the performance of models is measured by 3 metrics, namely: precision, recall, and accuracy in Table 5. The experimental result shows that the highest performance is achieved by method ID3 (Iterative Dichotomiser 3). In addition, this performance is obtained using a division of train data and test data with a ratio of 80%:20%. The precision-, recall- and accuracy level are 83.33. 97.44, and 95%, respectively. In contrast, the lowest performance with precision and recall measures occurs in the model built using the random forest method. Respectively, the precision and recall values are 50% and 47.44% in the split data scenario of 80% training data and 20% testing data. Meanwhile, the lowest accuracy measure of 63% occurs in the model built using the decision stump method with a test scenario of 85% training data and 15% testing data.

Furthermore, the research also analyzed the three measures based on the average of all scenarios presented in Figure 2. The highest performance of the three measures was achieved by the ID3 method whose average values of precision, recall, and accuracy were 80.752, 85.188, and 91.2, respectively. On the other hand, the random forest method has the lowest average value for precision=54.344 and recall=64.24. Meanwhile, the lowest average accuracy value of 66.4 occurred in the decision stump method.

Based on overall performance measurements, the model built by the ID3 method achieves the highest performance. For this reason, the next process generates a tree using ID3 at this highest performance condition as shown in Figure 3. Then, the process continues with the transformation of the tree into rule patterns. This process generates rules as shown in Figure 4. These rules are coded in if-then-else form when building the detection system in the next chapter.

Table 5. Comparison of the precision value on all scenarios

	Split data				
	60%:40%	65%:35%	70%:30%	80%:20%	85%:15%
			Precision		
ID3	74.87	80.56	81.67	83.33	83.33
Random forest	70.6	66.57	70	50	56.3
Random tree	54.77	54.57	55.35	61.75	65.28
CHAID	54.63	60.25	61.59	65.56	65.28
Decision tree	52.14	57.69	58.73	65.56	65.28
Decision stump	54.77	54.57	55.35	61.75	65.28
			Recall		
ID3	71.68	78.55	81.67	97.44	96.60
Random forest	73.43	70.87	70	47.44	60
Random tree	53.96	54.2	55.83	75.64	81.57
CHAID	59.21	60.87	64.17	81.20	81.67
Decision tree	56.18	57.54	60	81.20	81.67
Decision stump	53.96	54.3	55.83	75.64	81.67
			Accuracy		
ID3	84%	91%	93%	**95%**	93%
Random forest	66%	65%	70%	70%	73%
Random tree	84%	80%	80%	75%	80%
CHAID	66%	71%	73%	75%	73%
Decision tree	64%	68%	70%	75%	73%
Decision stump	66%	65%	68%	70%	63%

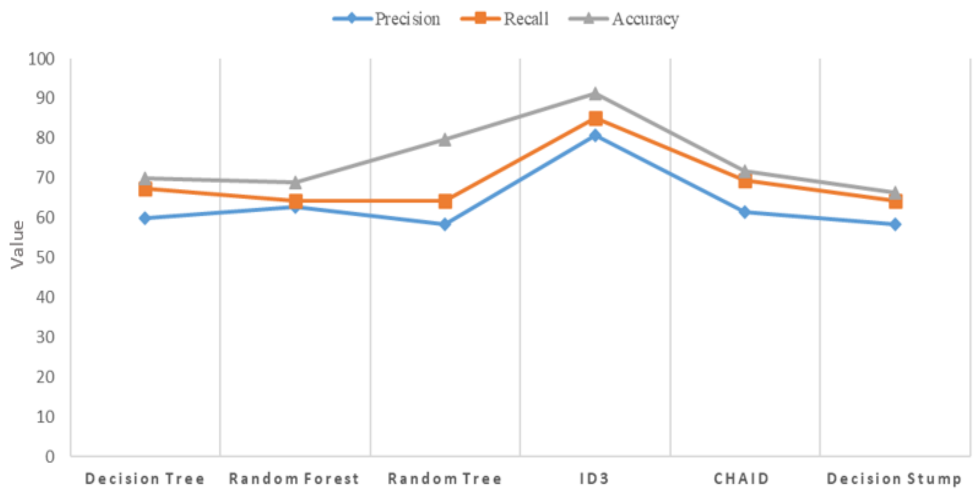


Figure 2. Comparison of the average of three metrics on all models

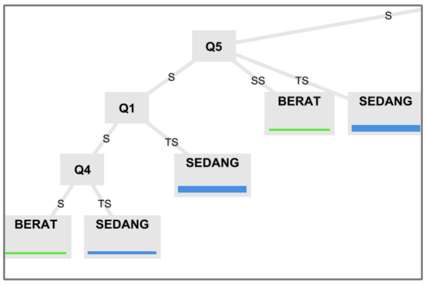


Figure 3. Part of the tree generated from the model

```
RINGAN=0}
|   |   |   Q2 = SS: BERAT {SEDANG=0, BERAT=1, RINGAN=0}
|   |   |   Q2 = TS: SEDANG {SEDANG=6, BERAT=0, RINGAN=0}
|   |   Q5 = SS: BERAT {SEDANG=0, BERAT=2, RINGAN=0}
|   |   Q5 = TS: SEDANG {SEDANG=7, BERAT=0, RINGAN=0}
|   Q3 = SS
|   |   Q6 = S
|   |   |   Q4 = S: BERAT {SEDANG=0, BERAT=4, RINGAN=0}
|   |   |   Q4 = SS: BERAT {SEDANG=0, BERAT=5, RINGAN=0}
|   |   |   Q4 = STS: SEDANG {SEDANG=1, BERAT=0,
RINGAN=0}
|   |   |   Q4 = TS: BERAT {SEDANG=0, BERAT=1, RINGAN=0}
|   |   Q6 = SS: BERAT {SEDANG=0, BERAT=2, RINGAN=0}
|   |   Q6 = TS: SEDANG {SEDANG=1, BERAT=0, RINGAN=0}
|   Q3 = STS: SEDANG {SEDANG=1, BERAT=0, RINGAN=0}
|   Q3 = TS: SEDANG {SEDANG=10, BERAT=0, RINGAN=0}
```

Figure 4. Part of the rules generated by tree

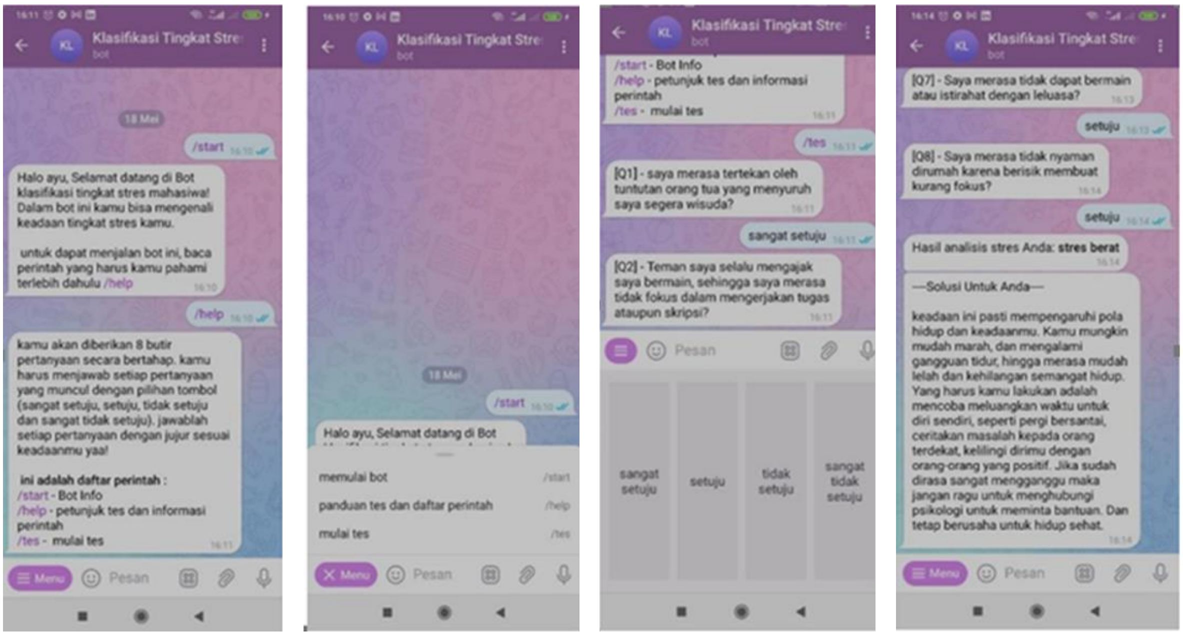
3.3. Developing the detection system for stress level

The coding process is carried out using the Google Apps Script tool. To access the Google Apps script, a search tool is needed, namely Google Chrome. Then, a Gmail account is created to access this Google Apps script. Before coding, to create the desired Telegram bot, the bot must be registered on Botfather on Telegram first to get tokens. This token is useful for the bot creation process, with this token the bot maker has access to change and create the bot as desired. Apart from getting tokens, BotFather can also set the menu, name, and username of the new bot that will be created.

The coding process begins by creating a global token variable. To be able to connect the new bot with the created code (server), the 'set webhook' function needs to be created. To carry out this function, the server must be placed on HTTPS or hosting. With this hosting, users can access bots in real-time. In this research, the hosting used follows Google hosting, namely by inputting the Google Apps script URL address in the set webhook function, so that third parties no longer need hosting. This set webhook function is only executed once. Then the TelegramBot class is created to accommodate functions on Telegram. These functions include the request function is used to get data updates, the sendMessage function which is created as a method of sending messages to users, sendMassageKeyboard function which is used to send messages or commands via the keyboard. Another function used in implementing this Telegram bot is the doPost function for presenting questions. This function also creates a keyboard for user answers. Then, create a calculateResult function to accommodate all user answers and manage user answers with an analysis of existing rules.

The implementation results are shown in Figure 5. There are the main features of this Telegram bot as follows: i) features about the bot (start): this feature displays a welcome message and an explanation of the function of the Telegram bot for classifying stress levels for students. ii) Help feature as shown in Figure 5(a) this feature contains a test guide that users must read when starting the test. This feature also displays all the commands that apply to the telegram bot for classifying stress levels in students. Apart from being accessible via the help feature, all applicable commands can also be accessed via the menu as shown in Figure 5(b) to the left of the message writing column. iii) Test features as shown in Figure 5(c): this feature displays the test questions in stages, from question 1 to question 8. Users can answer using the buttons. There are 4 buttons provided, namely the strongly agree button, agree button, disagree button, and strongly disagree button. If the number of questions answered has reached 8 or is equal to the number of questions, the bot will display the results of the user's stress level and possible solutions to reduce the effects of user stress.

Relating the solution as shown in Figure 5(d), they are as follows: i) If a student experiences 'light stress', then the solution displayed on the display is "Your stress has no negative impact on the human body, your condition is considered normal. What should you do? what you should do is continue to live a healthy life, such as don't stay up late, eat healthy food, and always think positive". ii) If students experience 'moderate stress', then the solution displayed on the display is "This situation sometimes starts to disturb your situation. What you have to do is stay healthy, think positively, spend your time exercising because exercise can reduce the hormone cortisol (stress hormone), and spend its time for yourself, and always give yourself positive affirmations". iii) If students experience 'heavy stress', the solution displayed on the display is "'This situation certainly affects life patterns and your situation. You may get angry easily, and experience sleep disorders, making you feel tired easily and lose the will to live. What do you have to do? Is to try to make time for yourself, like if you want to relax, tell him your problems and people nearby, and surround yourself with people the positive. If you find it very annoying Don't hesitate to contact a psychologist or expert to ask for help. And keep trying to live Healthy".



(a) (b) (c) (d)

Figure 5. Display of implementation results: (a) help feature, (b) menu bar, (c) test feature, and (d) solution

3.4. Testing the telegram bot-based detection system

After the Telegram bot has been built, the system needs to be tested. This research uses black-box testing techniques. This testing is carried out internally, namely by the bot maker and five other users. The results of this test are presented in Table 6. Testing goes well, all functions work as they should. However, for the analysis results, two users did not find the results of their stress levels. This is because there are only 29 rules formed, so answers outside these rules produce the output "stress level results not found". However, this shows that the system is functioning well. Error messages can also be displayed and conveyed properly. After the testing process is complete and the system is running well, students can use it to detect their emotions anywhere and anytime. Therefore, this system can be used to detect students' stress levels early and students immediately get a solution so that the problem does not disturb and decline their academic performance.

Finally, the research results in the form of this detection system are very important because this system can be an innovative solution for managing student stress after the pandemic. The sudden transition from online learning back to face-to-face education has the potential to significantly impact students' mental health and academic performance. Thus, Telegram's bot-based stress level detection system that uses the ID3 algorithm is highly relevant because it offers an accessible, easy-to-use, and real-time method for students to monitor and manage their stress for their academic success.

Table 6. Result of system testing

No.	Success indicators	Yes	No
1	The system can respond to the start command	✓	
2	The system can respond to help commands	✓	
3	The system can respond to test commands	✓	
4	The system can display test questions	✓	
5	The system can receive test answers	✓	
6	The system can analyze test answers and display stress level results to the user	✓	
7	The system can display solutions related to user stress levels	✓	

4. CONCLUSION

Tree algorithms are quite effective in detecting stress levels in students. Specifically for our research, ID3 achieves the highest performance in stress level modeling. Then, the best model can be implemented in a telegram bot-based system. This implementation is useful to students because they can interact easily with the stress level detection system. With the feature of a solution in the system, the student's stress problem can be solved as soon as possible. So, the existence of this detection system can prevent the decline of students' academic performance. In future work, collecting more data from diverse student populations needs to be carried out to improve the accuracy and generalization ability of the stress detection model. In addition, the exploration of ML algorithms and other hybrid models may be used to improve the performance of stress detection systems.

ACKNOWLEDGEMENTS

We want to thank the Ministry of Education, Culture, Research, and Technology for funding through fundamental research grants from the Directorate of Research, Technology and Community Service.

REFERENCES

[1] N. F. Shaafi, Z. A. Zulkipli, N. A. N. M. Azam, and M. M. M. Yusof, “Student’s acceptance of e-learning during the COVID-19 pandemic,” *International Journal of Evaluation and Research in Education (IJERE)*, vol. 12, no. 4, pp. 2207–2214, Dec. 2023, doi: 10.11591/IJERE.V12I4.25404.

[2] I. S. Saputri and E. Etlidawati, “The description of stress level and learning saturation during online learning of nursing students,” *Proceedings Series on Health & Medical Sciences*, vol. 2, 2022, doi: 10.30595/pshms.v2i.232.

[3] S. Gundogan, “The relationship of COVID-19 student stress with school burnout, depression and subjective well-being: adaptation of the COVID-19 student stress scale into Turkish,” *Asia-Pacific Education Researcher*, vol. 32, no. 2, 2023, doi: 10.1007/s40299-021-00641-2.

[4] D. A. AlAteeq, S. Aljhani, and D. AlEesa, “Perceived stress among students in virtual classrooms during the COVID-19 outbreak in KSA,” *Journal of Taibah University Medical Sciences*, vol. 15, no. 5, 2020, doi: 10.1016/j.jtumed.2020.07.004.

[5] G. F. Friskawati, “Students’s academic level stress during online learning in physical education,” *JUARA : Jurnal Olahraga*, vol. 6, no. 1, 2021, doi: 10.33222/juara.v6i1.1164.

[6] S. Ningsih, H. Yandri, N. Sasferi, and D. Juliawati, “An analysis of Junior High School Students’ learning stress levels during the COVID-19 outbreak: review of gender differences,” *Psychocentrum Review*, vol. 2, no. 2, 2020, doi: 10.26539/pcr.22321.

[7] H. Azmiyyah and A. Lianawati, “Profile of self-adjustment and level of academic stress during online learning in higher education,” *ProGCouns: Journal of Professionals in Guidance and Counseling*, vol. 2, no. 2, 2021, doi: 10.21831/progcouns.v2i2.42554.

[8] D. Lovin and D. Bernardeau-Moreau, “Stress among students and difficulty with time management: a study at the University of Galați in Romania,” *Social Sciences*, vol. 11, no. 12, 2022, doi: 10.3390/socsci11120538.

[9] M. J. Hossain, F. Ahmmed, S. M. A. Rahman, S. Sanam, T. Bin Emran, and S. Mitra, “Impact of online education on fear of academic delay and psychological distress among university students following one year of COVID-19 outbreak in Bangladesh,” *Heliyon*, vol. 7, no. 6, p. e07388, Jun. 2021, doi: 10.1016/J.HELIYON.2021.E07388.

[10] P. Siirtola and J. Röning, “Comparison of regression and classification models for user-independent and personal stress detection,” *Sensors (Switzerland)*, vol. 20, no. 16. 2020, doi: 10.3390/s20164402.

[11] S. Ishaque, N. Khan, and S. Krishnan, “Physiological signal analysis and stress classification from VR simulations using decision tree methods,” *Bioengineering*, vol. 10, no. 7, 2023, doi: 10.3390/bioengineering10070766.

[12] J. L. Delgado-Gallegos *et al.*, “Application of C5.0 algorithm for the assessment of perceived stress in healthcare professionals attending COVID-19,” *Brain Sciences*, vol. 13, no. 3, 2023, doi: 10.3390/brainsci13030513.

[13] T. Iqbal, A. Elahi, W. Wijns, and A. Shahzad, “Exploring unsupervised machine learning classification methods for physiological stress detection,” *Frontiers in Medical Technology*, vol. 4, 2022, doi: 10.3389/fmedt.2022.782756.

[14] Y. S. Can, N. Chalabianloo, D. Ekiz, J. Fernandez-Alvarez, G. Riva, and C. Ersoy, “Personal stress-level clustering and decision-level smoothing to enhance the performance of ambulatory stress detection with smartwatches,” *IEEE Access*, vol. 8, pp. 38146–38163, 2020, doi: 10.1109/ACCESS.2020.2975351.

[15] L. von Keyserlingk, K. Yamaguchi-Pedroza, R. Arum, and J. S. Eccles, “Stress of university students before and after campus closure in response to COVID-19,” *Journal of Community Psychology*, vol. 50, no. 1, 2022, doi: 10.1002/jcop.22561.

[16] D. Saleh, N. Camart, and L. Romo, “Predictors of stress in college students,” *Frontiers in Psychology*, vol. 8, no. JAN. 2017, doi: 10.3389/fpsyg.2017.00019.

[17] K. Pabreja, A. Singh, R. Singh, R. Agnihotri, S. Kaushik, and T. Malhotra, “Prediction of stress level on indian working professionals using machine learning,” *International Journal of Human Capital and Information Technology Professionals*, vol. 13, no. 1, 2022, doi: 10.4018/IJHCITP.297077.

[18] J V Gorabal, Y. S. Vipul, Y. S. Bharadwaj, M. Anas, andM. N. Ameem, "Predictive analysis of student stress level using machine learning," *International Journal of Advanced Research in Science, Communication and Technology*, 2023, doi: 10.48175/ijarsct-10387.
[19] P. Melillo, C. Formisano, U. Bracale, and L. Pecchia, "Classification tree for real-life stress detection using linear heart rate variability analysis. Case study: Students under stress due to university examination," in *IFMBE Proceedings*, 2013, vol. 39 IFMBE, doi: 10.1007/978-3-642-29305-4_126.
[20] N. Byshevets *et al.*, "Prediction of stress-related conditions in students and their prevention through health-enhancing recreational physical activity," *Journal of Physical Education and Sport*, vol. 23, no. 4, 2023, doi: 10.7752/jpes.2023.04117.
[21] S. Park, J. Jeon, H. Ahn, Y. Yang, and W. So, "Big data analysis of the key attributes related to stress and mental health in korean taekwondo student athletes," *Sustainability (Switzerland)*, vol. 14, no. 1, 2022, doi: 10.3390/su14010477.
[22] S. Hwang and Y. Choi, "Data mining in the exploration of stressors among NCAA student athletes," *Psychological Reports*, vol. 119, no. 3, 2016, doi: 10.1177/0033294116674776.
[23] J. R. Polo, C. C. Riveros, W. A. Diaz, A. C. Cansaya, and M. R. Anticona, "Characterization of the level of stress of engineering students using data mining tools," in *Proceedings of the LACCEI international Multi-conference for Engineering, Education and Technology*, 2021, vol. 2021-July.
[24] R. A. Ramírez-Valenzuela, R. Monroy, O. Loyola-González, F. Godínez, and A. Soberanes-Martín, "A one-class-classification approach to create a stresslevel curve plotter through wearable measurements and behavioral patterns," *International Journal on Interactive Design and Manufacturing*, vol. 15, no. 2–3, 2021, doi: 10.1007/s12008-021-00765-1.
[25] F. de J. Núñez Cárdenas, E. Aguirre Hernández, A. E. Guerrero Zenil, and A. M. Felipe Redondo, "Application method of data mining using the k means algorithm for the determination of stress level in high school students using the beck depression inventory," *Ciencia Huasteca Boletín Científico de la Escuela Superior de Huejutla*, vol. 8, no. 15, 2020, doi: 10.29057/esh.v8i15.5078.
[26] S. Chen, "A study on psychological stress assessment of higher educational institution's students based on computer data mining technology," *Frontiers in Psychology*, vol. 13, 2023, doi: 10.3389/fpsyg.2022.1054907.
[27] C. R. Putri Amalia, "Expert system to diagnose the level of learning stress in high school students using the forward chaining method," *International Journal Education and Computer Studies (IJECS)*, vol. 1, no. 1, 2021, doi: 10.35870/ijecs.v1i1.329.
[28] S. Sufaidah and M. Al Faqih, "UNWAHA cooperative digital marketing automation based on BOT engine," *NEWTON: Networking and Information Technology*, vol. 1, no. 3, 2022, doi: 10.32764/newton.v1i3.1370.
[29] M. F. Mustapha, A. N. I. Zulkifli, O. Kairan, N. N. S. M. Zizi, N. N. Yahya, and N. M. Mohamad, "The prediction of student's academic performance using RapidMiner," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 32, no. 1, 2023, doi: 10.11591/ijeecs.v32.i1.pp363-371.
[30] V. Matzavela and E. Alepis, "Decision tree learning through a predictive model for student academic performance in intelligent M-Learning environments," *Computers and Education: Artificial Intelligence*, vol. 2, 2021, doi: 10.1016/j.caeai.2021.100035.
[31] E. A. Kavats and A. A. Kostenko, "Analysis of connection methods of telegram robots with server part," *System technologies*, vol. 3, no. 122, 2019, doi: 10.34185/1562-9945-3-122-2019-03.
[32] Apriela Trirahma, "Telegram bot as a data collection tool for progress reports in area mapping progress monitoring system," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 6, 2021, doi: 10.29207/resti.v5i6.3654.
[33] A. D. Djoelianto, I. A. Kautsar, and M. A. Rosid, "Development of web service and telegram bot for location-based health service information system," *Procedia of Engineering and Life Science*, vol. 2, no. 2, 2022, doi: 10.21070/pels.v2i2.1280.

BIOGRAPHIES OF AUTHORS

Yuni Yamasari is Associate Professor at Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia. She received the bachelor degree, the master's degree in Computer Science and the Dr. degree in Electrical Engineering from Institut Teknologi Sepuluh Nopember (ITS), Indonesia. She received scholarships, including *Peningkatan Kualitas Publikasi Internasional* (PKPI)/the Sandwich Like to conduct research in Japan. Her current research interests include artificial intelligence, data mining, and machine learning. She can be contacted at email: yuniyamasari@unesa.ac.id.

Anita Qoiriah received the bachelor degree and the master's degree in Computer Science, Institut Teknologi Sepuluh Nopember (ITS), Indonesia. She is Associate Professor at Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia. Her research interest areas are artificial intelligence, natural language processing, and machine learning. She can be contacted at email: anitaqoiriah@unesa.ac.id.

Naim Rochmawati completed her undergraduate studies in computer science at Institut Teknologi Sepuluh Nopember (ITS), Indonesia, and earned her master’s degree in computer science from Institut Teknologi Bandung in Bandung, Indonesia. Currently serving as an Assistant Professor at the Department of Informatics, Universitas Negeri Surabaya in Surabaya, Indonesia, her research interests lie in big data, artificial intelligence, and machine learning. She can be contacted at email: naimrochmawati@unesa.ac.id.

Aditya Prapanca earned his bachelor's degree in Electrical Engineering and his master's degree in Computer Science from Institut Teknologi Sepuluh Nopember in Indonesia. Currently holding the position of Associate Professor at the Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia. His research focuses on computer networks, computer security. He can be contacted at email: adityaprapanca@unesa.ac.id.

Agus Prihanto received the bachelor degree in Electrical Engineering and the master’s degree in Computer Science, Institut Teknologi Sepuluh Nopember, Indonesia. He is Assistant Professor at Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia. His research interest areas are computer network, computer security, virtualization, and cloud computing. He can be contacted at email: agusprihanto@unesa.ac.id.

I Made Suartana attained his bachelor's and master's degrees in computer science from Institut Teknologi Sepuluh Nopember in Indonesia. Currently serving as an Assistant Professor at the Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia. His academic pursuits revolve around computer network, multimedia network, and network security. He can be contacted at email: madesuartana@unesa.ac.id.

Tohari Ahmad received the bachelor degree in computer science from Institut Teknologi Sepuluh Nopember (ITS), Indonesia, the master’s degree in Information Technology from Monash University, Australia, and the Ph.D. degree in Computer Science from RMIT University, Australia. He was a consultant for some international companies. In 2003, he moved to ITS, where he is currently a Professor. His current research interests include network security, information security, data hiding, and computer networks. His awards and honors include the Hitachi Research Fellowship and JICA Research Program to conduct research in Japan. He is a reviewer of several journals. He can be contacted at email: tohari@if.its.ac.id.