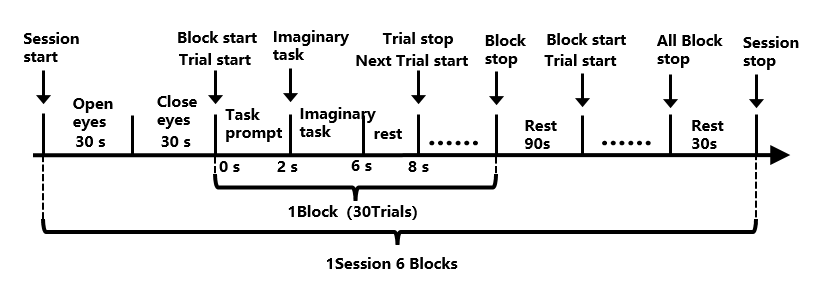
**Description of sports imagination games**

1. **Experimental paradigm**

Each group of experiments contains left hand motor imagination task data, right hand motor imagination task data, and foot motor imagination task data, as shown in Figure 1. This competition provides 3 blocks of data of each subject as training data, and 3 blocks of this subject as test data. Each block contains 30 trials (10 for left hand, right hand and feet). Each trial consists of a task prompt for 2s, a movement imagination for 4s, and a rest for 2s, as shown in Figure 1. The data used in this competition includes the data of normal people and some patients.

|  |  |  |
| --- | --- | --- |
| 左手 | 右手 | 双脚_纯白 |
| Left hand movement imagination prompts | Right hand movement imagination prompts | Foot movement imagination prompts |



**Figure 1** Experimental process

The experimental data was collected by the Boruikang 64-channel EEG acquisition device, the 65th lead was the trigger information, and the original sampling rate was 1000Hz. In the single Trial, only the 4s data between the beginning of the running imagination task and the end of the motor imagination task were retained, and the data of the task prompt and rest time were set to 0.

The test data in the frame was downsampled to 250Hz, and the training data provided to the participant was not downsampled or processed by other filtering.

The specific trigger definition is shown in Table 1, 2, and 3. The whole experimental paradigm includes Session start mark 250, Session end mark 251, Block start mark 242, and Block end mark 243.

Each trial contains ten triggers in the training data, and the results are submitted at mark 241. The data in the first paragraph starts from the left hand mark 11, right hand mark 21 and both feet mark 31; the data in the second paragraph starts from the left hand mark 12, right hand mark 22 and both feet mark 32; the data in the third paragraph starts from the left hand mark 13, right hand mark 23 and both feet mark 33. As shown in Figure 2.

Each trial contains four triggers in the test data: 200 at the beginning of the exercise imagination, 202 at the imagined 2s, 203 at the imagined 3s, and 241 at the score submission point. See Figure 3.

The rest is reserved for the system.

**Table 1** Definition of the overall experimental Trigger

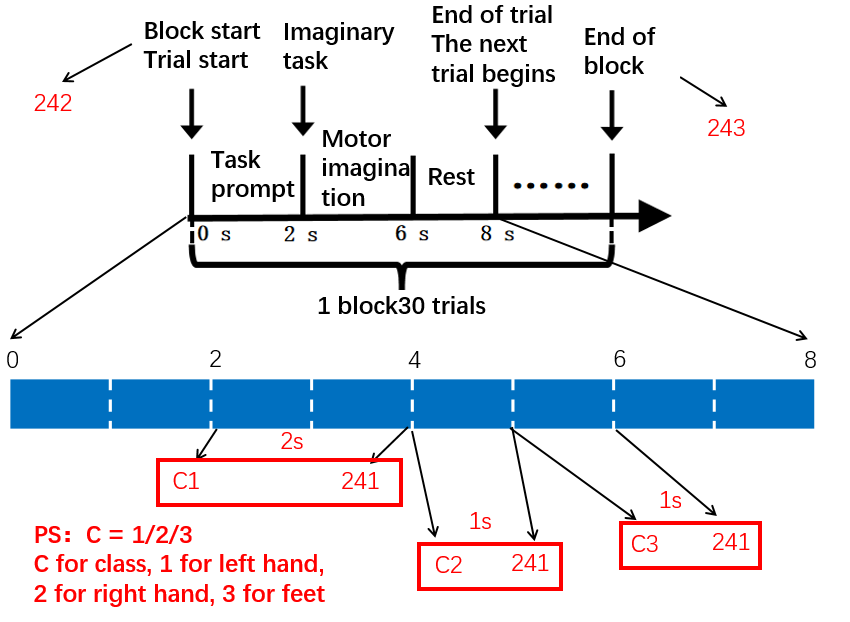
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Definition** | **Session start** | **Session stop** | **Block start** | **Block stop** |
| **Trigger number** | **250** | **251** | **242** | **243** |

**Table 2** Trigger definition of training data

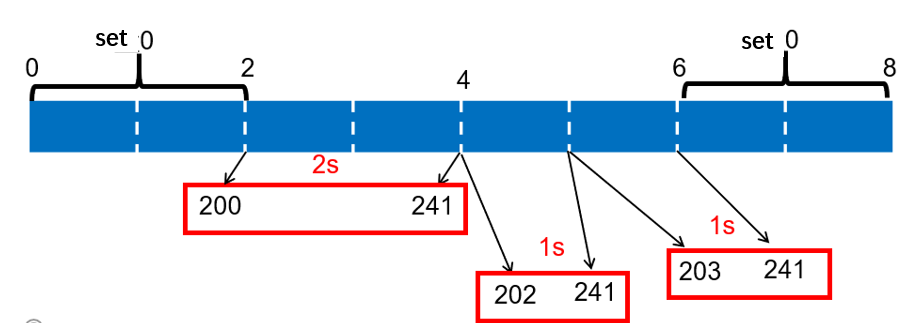
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Definition** | **Submission**  **point** | **Left hand**  **First paragraph** | **Right hand**  **First paragraph** | **feet**  **First paragraph** | **Left hand**  **The second paragraph** | **Right hand**  **The second paragraph** | **feet**  **The second paragraph** | **Left hand**  **The third paragraph** | **Right hand**  **The third paragraph** | **feet**  **The third paragraph** |
| **Trigger number** | **241** | **11** | **21** | **31** | **12** | **22** | **32** | **13** | **23** | **33** |

**Table 3** Test data Trigger definitions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Definition** | **Kinesthetic initiation** | **Imagined 2s** | **Imagined 3s** | **Score submission point** |
| **Trigger number** | **200** | **202** | **203** | **241** |



**Figure 2** Training data Trigger use case



**Figure 3** Test data Trigger use case

**Table 4** Lead numbers - Lead names

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lead number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Lead name | Fpz | Fp1 | Fp2 | AF3 | AF4 | AF7 | AF8 | FZ |
| Lead number | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Lead name | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 |
| Lead number | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| Lead name | FCz | FC1 | FC2 | FC3 | FC4 | FC5 | FC6 | FT7 |
| Lead number | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| Lead name | FT8 | Cz | C1 | C2 | C3 | C4 | C5 | C6 |
| Lead number | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| Lead name | T7 | T8 | CP1 | CP2 | CP3 | CP4 | CP5 | CP6 |
| Lead number | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| Lead name | TP7 | TP8 | Pz | P3 | P4 | P5 | P6 | P7 |
| Lead number | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 |
| Lead name | P8 | POz | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
| Lead number | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |
| Lead name | Oz | O1 | O2 | ECG | HEOR | HEOL | VEOU | VEOL |

The data stream is provided in analog online mode. Each time the data read method is called, a new packet is obtained containing 40ms of experimental EEG data (the last packet may be less than 40ms long), along with trigger information received during the recording of that packet. Within the same block, packets are sent in chronological order. If the test data contains multiple groups of block data, after one group of block data is sent, the next group of block data will be sent when the data reading method is called again. When all experimental data has been sent, the program termination flag finishedFlag is set to 1. After detecting that finishedFlag is 1, the running algorithm terminates the run() method. It should be noted that since the experimental data comes from real EEG signals, the length of the last packet in each block may not be a fixed value. Please pay special attention to this during algorithm development. In addition, data may not be collected for some leads in very few subjects, but this does not affect the final classification effect.

1. **Algorithm specification**

The competition algorithm calls the data reading method to obtain EEG data. Once the data reading method is called, the competition system will return a new packet, which can be cached and processed by the competition algorithm. When the algorithm considers that the received data is sufficient to meet the decision conditions, it needs to call the feedback method to report the recognition results to the competition system. The competition system calculated the effective data length of the algorithm according to the call times of the data reading method, and combined with the feedback accuracy rate, comprehensively calculated the average simulated information transmission rate.

The competition algorithm needs to meet the following constraints:

1. Constraints on the start and end of the trial:

In the process of detection and recognition of single trial data, the competition algorithm needs to start detection after receiving the trial trigger, and make feedback report before receiving the score submission trigger. Otherwise, the reported results will be judged as invalid recognition results.

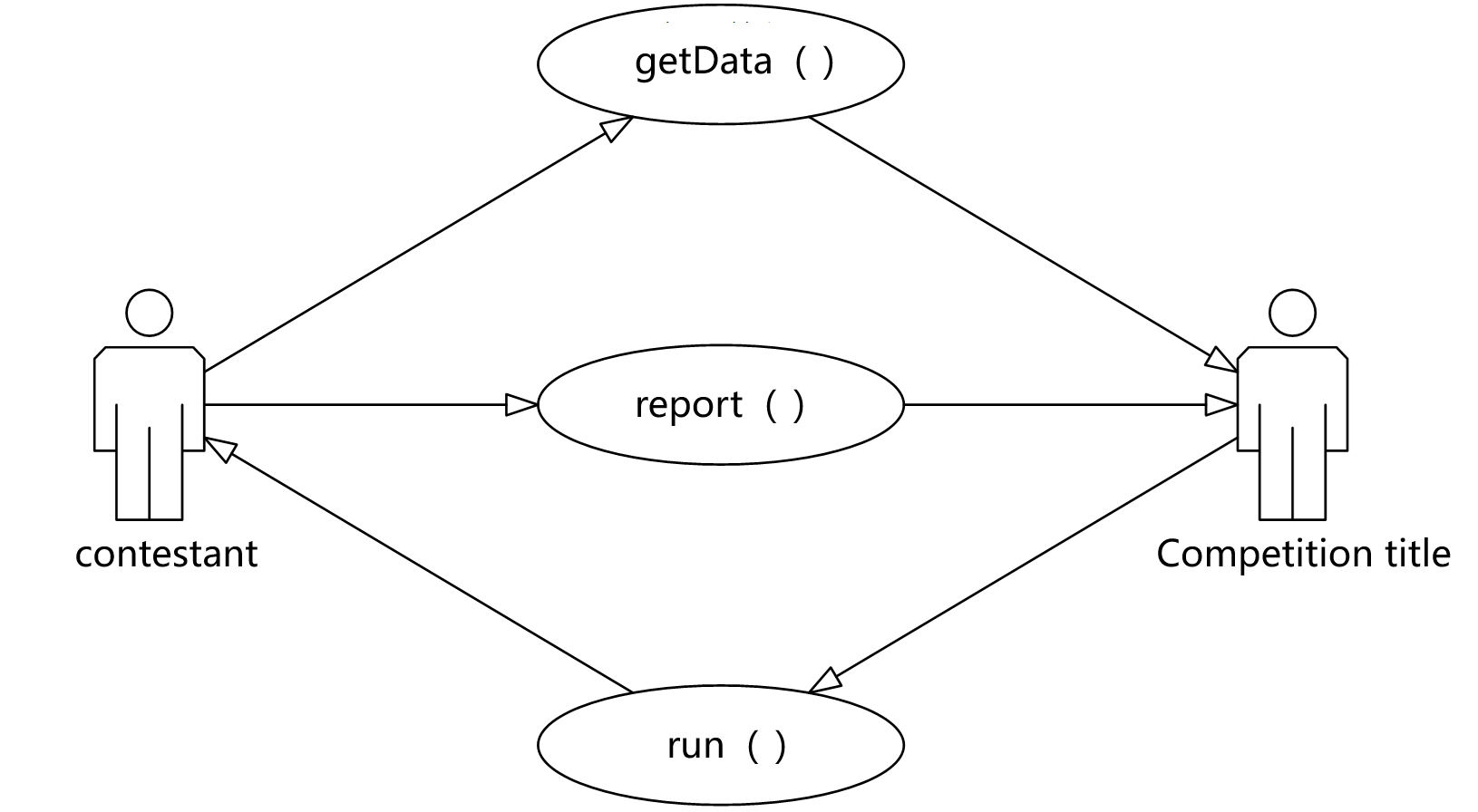
2. Constraints on the maximum data length of a single trial:

In this project, the maximum detection time for a single test should be less than 4 seconds. A maximum of 4 seconds (excluding 4 seconds) of EEG data will be collected from the imaginary trigger signal of the trial, otherwise the recognition result of the trial will be considered invalid and the result of the trial will be 0.

3. Constraint on algorithm termination:

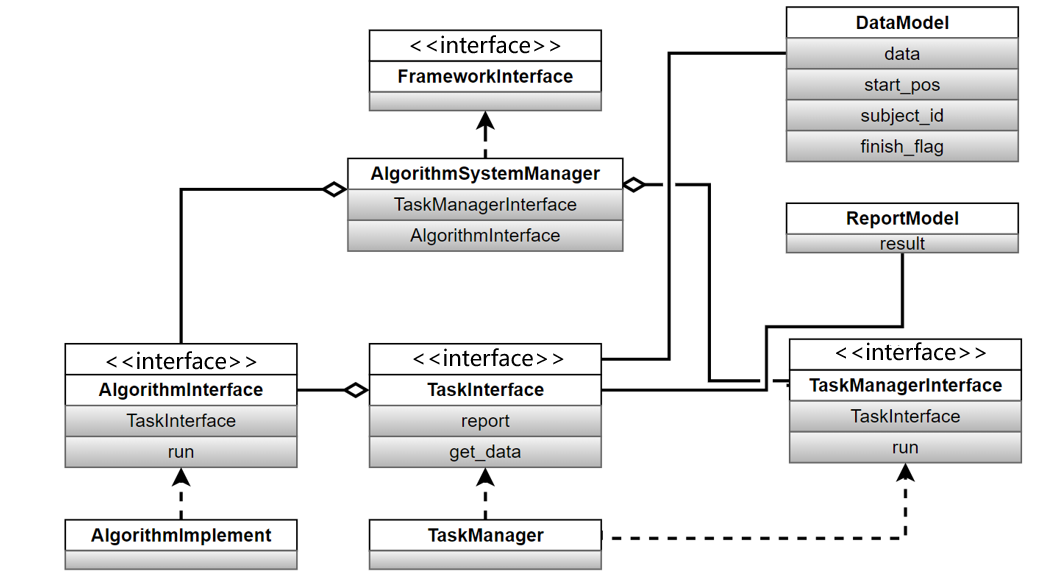
When Endflag = 1 is received in the data packet, it means that all experimental data has been sent, and the participating algorithm needs to stop processing and exit by itself.

1. **competition framework**
2. Contestant use case



**Figure 4** Contestant use case

1. The main frame of the system is shown in Figure 5.



**Figure 5** Main body framework of the system

(1) Framework interface

This interface is mainly responsible for the connection between the program and the external executive system. The implementation class for this framework interface is Algorithm interface and implements all the functions defined by the interface.

(2) Task interface

This interface is a competition question interface for participants, mainly responsible for data transmission and result report between the question and the competition algorithm. Participants can obtain race data through this interface and report the recognition results through this interface. Competition questions need to give a comprehensive score according to the number of times the competition data is obtained by the competition algorithm and the accuracy of the report results.

(3) Indicates the match interface

This interface is mainly responsible for the data filling, obtaining scores, clearing data and clearing report results of the competition questions. The framework realizes the call to the competition questions through this interface.

(4) Algorithm interface

Through this interface, the contest questions can be used to verify and calculate the contest algorithm. Contestants need to implement this interface. During execution, the algorithm needs to obtain data through the Task interface and report the results through the interface. At the same time, participants need to control the computational complexity of the algorithm, otherwise when the running time exceeds the predetermined length, the system will automatically terminate the calculation process, and the results obtained will be invalid.

1. Data model
2. DataModel Participant data model
3. data：float Type matrix, segmented data. For example, if 64 EEG data and 1 trigger signal are included and the data is segmented in 40ms ata sampling rate of 250Hz, the data obtained ata single time is 65\*10 points.
4. start\_pos: int Type scalar, the index position of the current segmentation data start time relative to the block data start time。
5. subeject\_id：int Type scalar, current data source subject number。
6. finish\_flag：bool Type scalar, test end flag. If the competition algorithm obtains True in the packet through data\_model.finish\_flag, it needs to exit the program running by itself.
7. Interface functions related to contestants

(1) TaskInterface

The interface is implemented by the developer, including data acquisition method and result feedback method. The implementation class of the interface is injected into the competing algorithm implementation class before the algorithm is run. In the process of algorithm execution, the interface can be called to obtain data and identify the results through the result feedback method report. According to the number of times the data acquisition method is called and the correctness of the result feedback, the question maker will give a comprehensive score.

1) def get\_data(self):

Input parameter：null

Output parameter：DataModel

Realization function：Obtain the experimental data of the next section

2) def report(self, report\_model):

Input parameter：report\_model

Output parameter：null

Realization function：Feedback recognition result

(2) AlgorithmInterface

Runners need to fill in the run function with the program run. During the algorithm execution, the DataModel data is obtained by TaskInterface get\_data and the ReportModel result is returned by the report method. When finish\_flag in the DataModel data obtained through get\_data is true, it means that data processing is complete and the function needs to exit the operation by itself.

1) def run(self):

Input parameter：null

Output parameter：null

Realization function：Algorithm analysis procedure

Submission format

This program is written in python language, you need to submit a file with the extension.pyc based on python 3.8.

1. Submit the sample

Refer to the companion code.

Contestants can complete the Algorithm by modifying the code in the Algorithm folder. To avoid unknown errors, do not add folders in the home directory. When done, repackage the program (including AlgorithmImplement folder and config.toml) -> Grouping -> Specific grouping -> Cell -> Define cell -> Upload Package -> Submit to Match -> Select Match -> Deploy --> Finish the race.

After the deployment is complete, view the race results in the leaderboard specific to the race;

Note that the protection scorer fully covers the entrant's code (except for AlgorithmImplement directory and config.toml) when submitting to the contest --> deploy, in order to prevent the entrant from cheating by modifying the code framework. The actual starting is the scoring program + AlgorithmImplement directory of the competitor, and the rest of the running supporting code is the server built-in program (including main.py and other files. The server built-in scoring program is basically the same as the program framework in the example, but includes scoring function and reading server match data function).

1. Scoring method

This system takes the average information transmission rate as the scoring standard:



Among them, T\_i represents the length of time window T­­\_1=2, T\_2=3, T\_3=4, M represents the number of targets, and P\_i represents the recognition accuracy rate. The unit of ITR is bits/min. The results of each trial shall be submitted for three times. The ITR of three trials shall be calculated respectively and taken as the average result of one Session.

**Special need to point out：**

1. **Valid submission：**
2. **The length of the data window obtained in the frame when the result is first submitted is less than 2s (including 2s)；**
3. **The length of the data window obtained in the frame during the second submission of the result is less than 3s (including 3s)；**
4. **The length of the data window obtained in the frame when submitting the results for the third time is less than 4s (including 4s).If the conditions are not met, the submission is deemed invalid.**

**When submitting results for the third time, the length of data window obtained in the frame is less than 4s (including 4s).**

**If the conditions are not met, the submission is deemed invalid.**

**The results of effective submission will be correct rate statistics, and the results of invalid submission will be forced to be wrong.**

**Calculation result:**

1. **The statistical accuracy rate of all the results submitted for the first time is summarized, and ITR is calculated according to T\_1=2；**
2. **The statistical accuracy rate of all the results of the second submission is summarized, and ITR is calculated according to T\_2=3；**
3. **The statistical accuracy rate of all the results of the third submission is summarized, and ITR is calculated according to T\_3=4.The final score is given by averaging the three ITRs。**
4. **When the accuracy P\_i is less than 1/M, the ITR corresponding to the last commit is forced to be 0.**
5. Performance evaluation methods

The competition algorithm obtains the new packet through the data reading method. When the resulting packet contains trigger signals, the scoring system will automatically start recording the length of EEG signals used in the algorithm recognition process until the feedback method is invoked. The length of the EEG data obtained from the start of the trigger until the feedback method is invoked is used as the simulated trial duration for that trial. The average accuracy will be calculated according to the consistency of the algorithm feedback results with the real stimulus.

It should be noted that every data packet containing trigger in this competition project is still regarded as the data of the previous attempt. The new try data is calculated from the next packet containing the trigger packet. Therefore, the competition algorithm cannot feed back immediately when it obtains the data packet containing trigger signal, but can feed back only after it obtains the next data packet at the earliest.

1. Handle the result feedback exception

1) Repeat report

Within one trial time, the algorithm can feedback the results for a maximum of three times (2s,3s,4s), and if it exceeds that, it will only be recorded according to the time and results of the previous three feedbacks.

2)No result feedback

If there is no feedback from the algorithm within one trial time, the verdict will be recorded as a misjudgment.

3) Result feedback timeout

When the result is fed back, it is calculated that the entry algorithm has obtained EEG data beyond the current limit time since the trial trigger, and the judgment result will be recorded as a misjudgment.

4)Algorithm execution timeout

In order to meet the real-time processing requirements of brain-computer interface system, this project also has certain requirements on the computational complexity of the competition algorithm. This competition will determine a calculation time according to the amount of competition data. If the algorithm complexity is too high and the system runs out of time, the result of the algorithm will be considered invalid.