UNIVERSITÀ DI PISA

PROCESS MINING PROJECT

ACADEMIC YEAR 2022-23

Massagli Lorenzo

Patimo Antonio

Buchignani Fabio

Fabijan Maria

1. Process Landscape ... 4

2. Assumptions ... 5

2.1 Parameters to configure in all the systems ... 5

3. BPMN Models ... 7

3.1 Configure sleep quality service (all) ... 7

3.2 Prepare sleep session (Maria Fabijan) ... 8

3.3 Generate sleep session set (Fabio Buchignani) ... 9

3.4 Develop sleep quality classifier (Antonio Patimo) ... 9

3.5 Classify sleep session (Lorenzo Massagli) ... 10

3.6 Check sleep quality classifier (Lorenzo Massagli) ... 10

4. Task Level Modeling ... 11

4.1 Salaries Proportion ... 11

4.2 Company Roles ... 11

4.3 Cognitive Effort values ... 11

4.4 Task level diagrams ... 12

4.4.1 Prepare Sleep Session (Maria Fabijan) ... 12
4.4.2 Generate Sleep Session Set (Fabio Buchignani) ... 21
4.4.3 Develop sleep quality classifier (Antonio Patimo) ... 29
4.4.4 Check Sleep Quality Classifier (Lorenzo Massagli) ... 37
4.4.5 Restart data collection (Lorenzo Massagli) ... 40

5. As-Is Simulation model ... 41

5.1 Assumptions ... 41

5.2 Percentage of gateways ... 42

5.3 Simulation Results ... 43

6. To be Model ... 45

6.1 To be task level diagrams ... 47

Evaluate the class distribution balancement ... 47
Configure segregation system parameters ... 48
Configure the Development system parameters ... 49

6.2 To be Simulation model ... 52

6.2.1 Assumptions ... 52
6.2.2 Percentage of tasks and gateways ... 53

6.3 Simulation results ... 54

7. Process Mining ... 56

7.1 Simulation parameters used ... 56

7.2 Transition map Disco ... 57

7.3 Transition map Apromore ... 57

7.4 Comparison ... 58

7.5 BPMN Mining from Original Logs **58**
7.5.1 ProM 58
7.5.2 Apromore 60
7.5.3. Conformance checking 60

7.6 Modified logs experiments **64**
7.6.1 Violation list 64
7.6.2 Fitness with modified logs and BPMN models mined on the original logs 65
7.6.3 Transition map disco from the modified logs 67
7.6.4 Transition map Apromore from the modified logs 67
7.6.5 BPMN Mining from modified logs - ProM 68
7.6.6 BPMN Mining from modified logs - Apromore 68
7.6.7 Conformance checking with the modified logs 68

7.7 Conclusions **70**

Clockify ***71***

1. Process Landscape

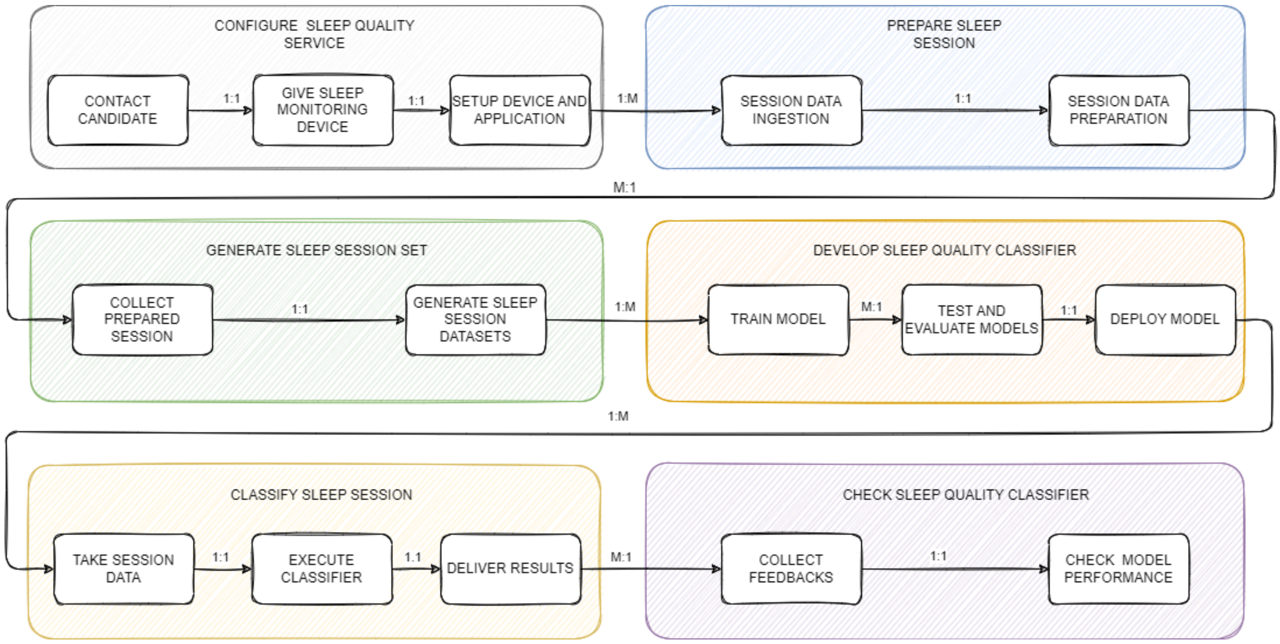


Figure 1: Process Landscape

2. Assumptions

In this section there are described all the assumption that has been done during the project.

1. Sleep session: Made of 4 records:
    - Label from expert: low, medium, high
    - Label from user: low, medium, high
    - Raw Values Smartwatch sensor: heart rate time series, wrist acceleration time series.
    - Values from the bed sensor: awaken time.
2. Preprocessed data / Features:
    a. max heart rate,
    b. min heart rate,
    c. avg heart rate,
    d. max wrist acceleration,
    e. avg wrist acceleration.
3. Parameters of customer to consider:
    - Age
    - Sleep regularity (0, 1)
    - List of illnesses
    - Physical Activity Level (low, medium, high)
    - Nutrition (low, medium, high)
    - Stress level (low, medium, high)

2.1 Parameters to configure in all the systems

1) Ingestion system:

- Address of preparation system: 1 given
- Number of records to synchronize: 1 given
- Threshold number of missing samples (heart rate value of the time series): 3 rule heart rates constant or not; if constant 10 samples, if not 5 values
- Number of sleep sessions to collect in execution and in monitoring modes: 1 given

2) Preparation system:

- Address of segregation system: 1 given
- Address of execution system: 1 given
- Execution mode: 1 given

3) Segregation system:

- Address of the development system and the messaging system: 1 given
- The “sufficient sleep sessions” threshold: 4 multiple criteria
- The threshold regarding the tolerance about class balancement: 1 given
- The percentage of the data that will form the training set: 1 given.

4) Development system

- Number of hidden layer interval: 4 multiple criteria
- Number of neuron interval: 4 multiple criteria
- Number of initial iterations: 1 given

- Threshold for the test error: 1 given
- Ip address of the execution system: 1 given

6) Monitoring system:

- Number of misclassified sessions: 1 given.
- Threshold number of pairs of labels: 1 given.

3. BPMN Models

In this section we will describe the different BPMN models of the process that we described in the process landscape.

3.1 Configure sleep quality service (all)

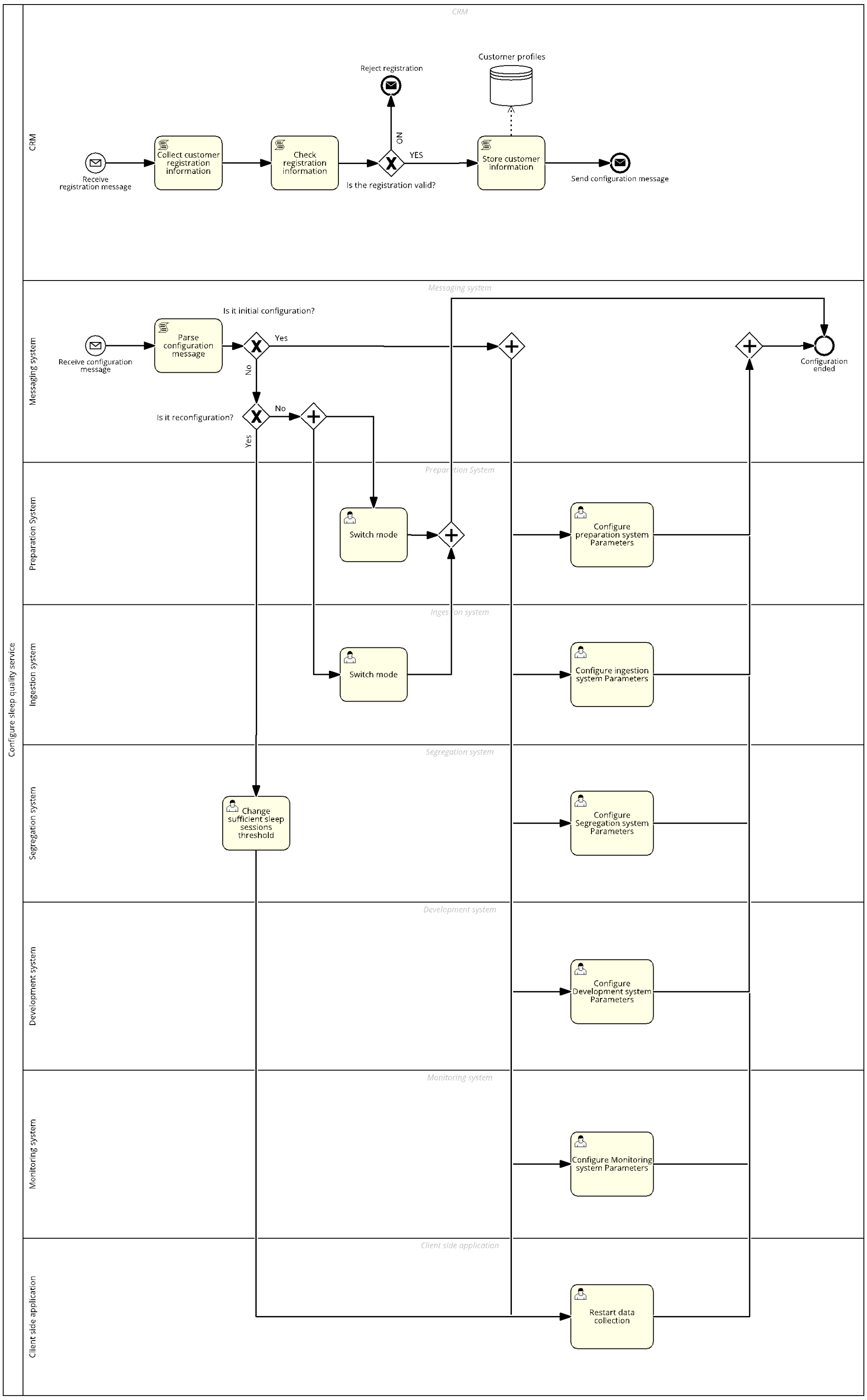


Figure 2: BPMN model of the "configure sleep quality service" process

For simplicity, we skip the configuration of the client-side when a customer register to the CRM.

3.2 Prepare sleep session (Maria Fabijan)

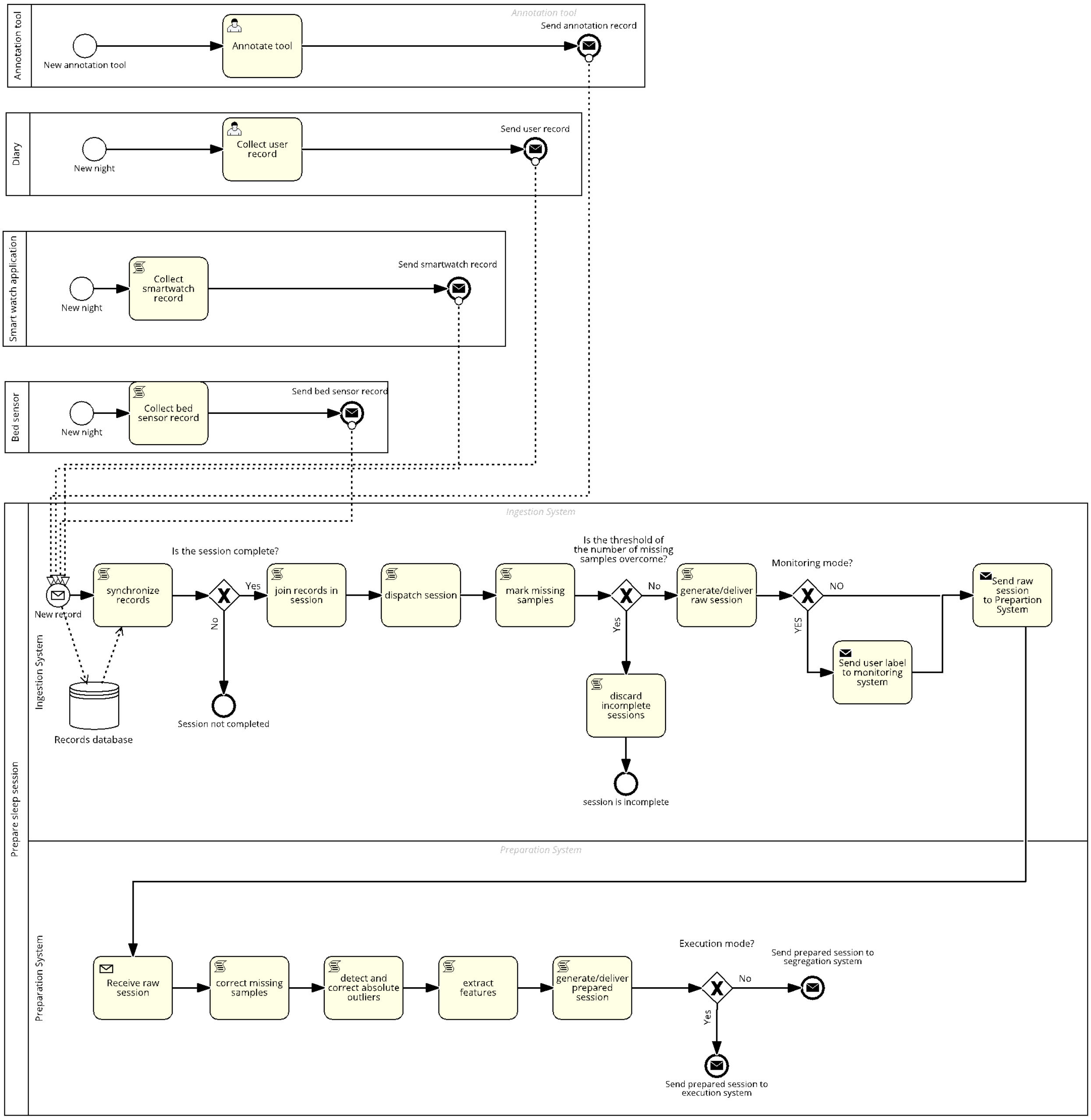


Figure 3: BPMN model of the "prepare sleep sessions" process

3.3 Generate sleep session set (Fabio Buchignani)

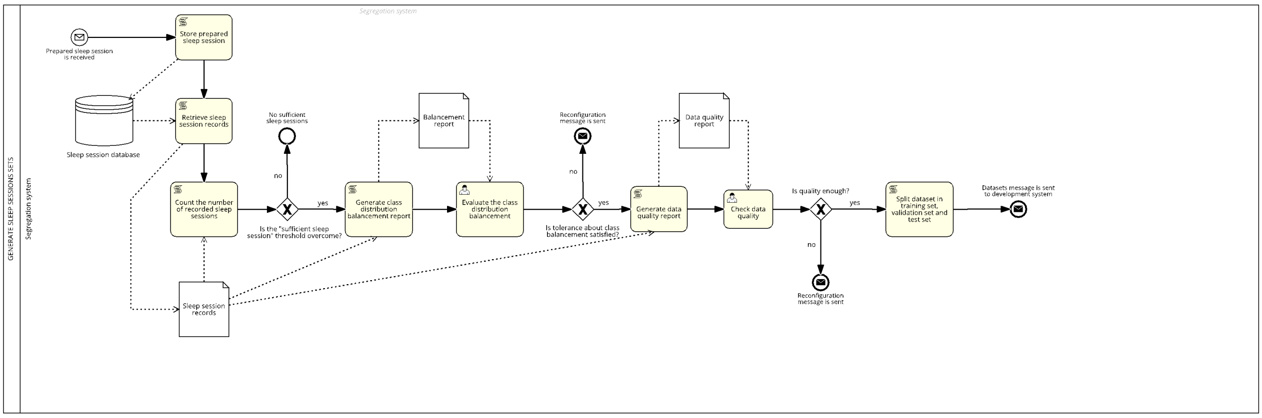


Figure 4: BPMN model of the "generate sleep session set" process

3.4 Develop sleep quality classifier (Antonio Patimo)

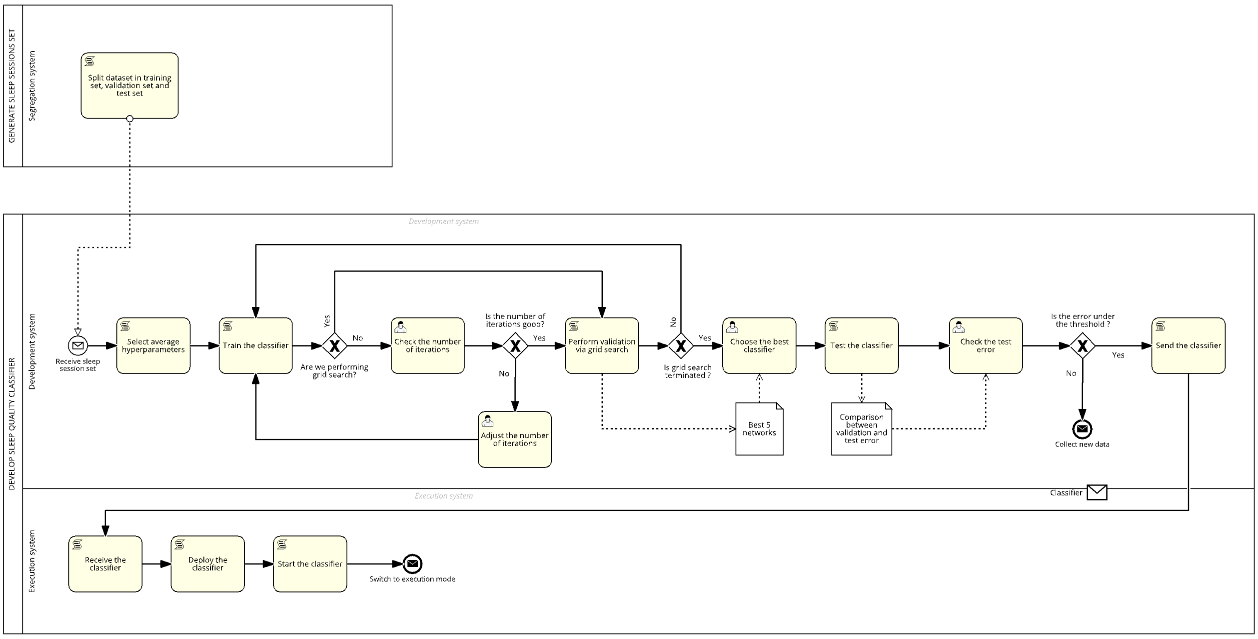


Figure 5: BPMN model of the "develop sleep quality classifier" process

3.5 Classify sleep session (Lorenzo Massagli)

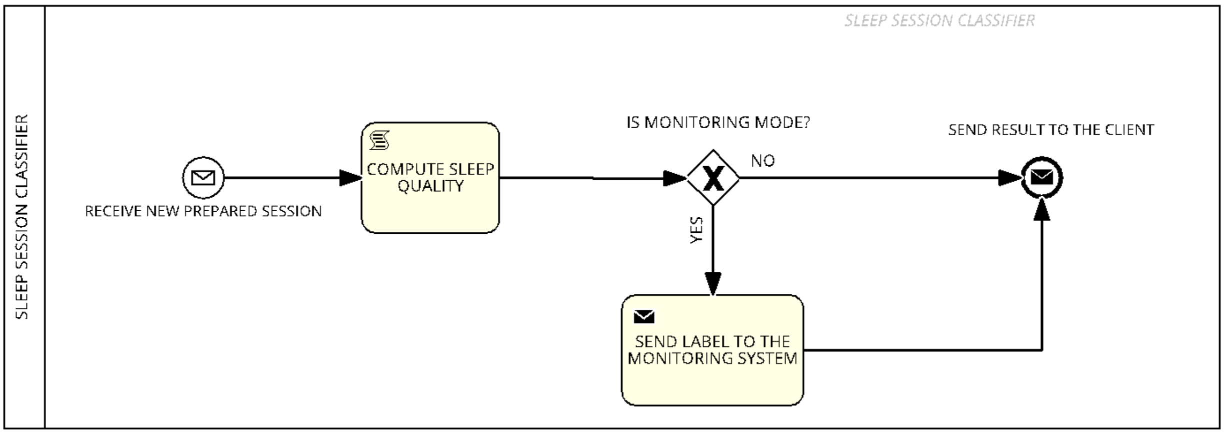


Figure 6: BPMN model of the "classify sleep session" process

3.6 Check sleep quality classifier (Lorenzo Massagli)

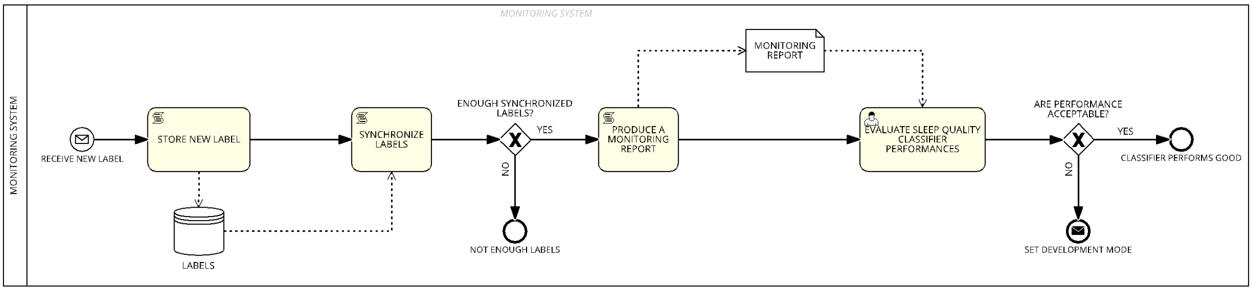


Figure 7: BPMN model of the "check sleep quality classifier" process

4. Task Level Modeling

In this section we will describe the task level of the BPMN processes.

4.1 Salaries Proportion

ACTOR	LINK	COST (YEARLY)	NORMALIZED COST
Data Engineer	https://www.glassdoor.com/Salaries/data-engineer-salary-SRCH_KO0,13.htm	112.715$	1.58
Data Analyst	https://www.glassdoor.com/Salaries/data-analyst-salary-SRCH_KO0,12.htm	71.228$	1
ML Engineer	https://www.glassdoor.com/Salaries/machine-learning-engineer-salary-SRCH_KO0,25.htm	131.347$	1.84
System Administrator	https://www.glassdoor.com/Salaries/systems-administrator-salary-SRCH_KO0,21.htm	80,025$	1.12

4.2 Company Roles

- Data Analyst: A Data Analyst is a professional who is responsible for collecting, analyzing, and interpreting large sets of data to inform business decisions. They use various tools and techniques to extract insights from data and present their findings to stakeholders in a clear and meaningful way. They also play a key role in identifying and defining new process improvement opportunities.
- Data Engineer: A Data Engineer is a professional responsible for the design, development, maintenance, and management of an organization's data infrastructure. They play a key role in helping data scientists and analysts' access and work with large, complex data sets. They work on tasks such as building and maintaining data pipelines, designing, and implementing data storage solutions, and developing and implementing data security measures.
- ML Engineer: A Machine Learning Engineer is a professional with expertise in designing, developing, and deploying machine learning models and systems. These individuals are responsible for applying their knowledge of computer science, statistics, and mathematics to build, implement and maintain machine learning algorithms.
- System Administrator: A system administrator is responsible for the installation, configuration, and ongoing maintenance of an organization's computer systems and servers.

4.3 Cognitive Effort values

- Remember (1): The step can be carried out by remembering another occurrence of the same step;
- Understand (2): The step can be carried out by finding a value in a set of predefined categories;
- Apply (3): The step can be carried out by executing a predefined procedure, encoded by the company;
- Analyze (4): The step can be carried out by finding unknown categories.

4.4 Task level diagrams

4.4.1 Prepare Sleep Session (Maria Fabijan)

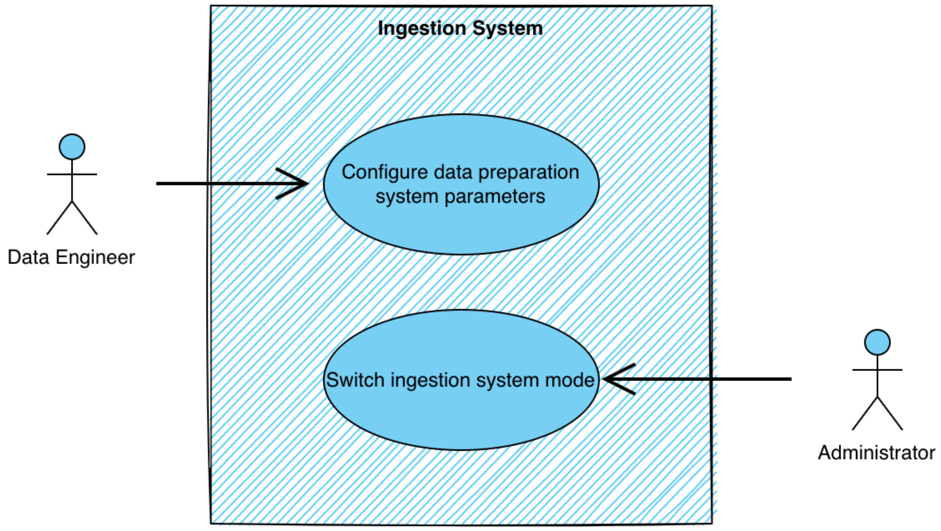


Figure 8: Use case diagram of the Ingestion System

Configure the Ingestion System

SUB-TASK	ACTOR	%	ACTION	COGNITIVE EFFORT	TOATAL COST
1.	**Data Engineer**		**The Data Engineer** clicks on button "Configure Ingestion System Parameters"	Remember (1)	1.58*1 = 1.58
2.	**System**		**The System** shows configuration ingestion system interface.		
3.	**Data Engineer**		**Data Engineer** click on "configure address of preparation system" button	Remember (1)	1.58*1 = 1.58
4.	**System**		**The System** shows tab to set address of preparation system		
5.	**Data Engineer**		**The Data Engineer** set up preparation system address	Remember (1)	1.58*1 = 1.58
6.	**Data Engineer**		**The Data Engineer** click on "save configuration" button	Remember (1)	1.58*1 = 1.58

7.	**System**		**The system** shows configuration ingestion system interface.		
8.	**Data Engineer**		**The Data Engineer** click on "Set up number of records to synchronize" button	Remember (1)	1.58*1 = 1.58
9.	**System**		**The System** shows tab to set up number of records to synchronize		
10.	**Data Engineer**		**The Data Engineer** set up number of records to synchronize	Remember (1)	1.58*1 = 1.58
11.	**Data Engineer**		**The Data Engineer** click on "save configuration" button	Remember (1)	1.58*1 = 1.58
12.	**System**		**The system** shows ingestion system interface		
13.	**Data Engineer**		**The Data Engineer** click on "configure threshold number of missing samples"	Remember (1)	1.58*1 = 1.58
14.	**System**		**The System** shows tab to configure threshold number of missing samples		
15.	**Data Engineer**		**The Data Engineer** checks if the heart rate is constant.	Apply (3)	1.58*3 = 4.74
		70%	**IF** the heart rate is costant		
15.1	**Data Engineer**		**The Data Engineer** set up the threshold number of missing samples to 10.	Remember (1)	0.7*1*1.58 = 1.106
		30%	**ELSE**		
15.2	**Data Engineer**		**The Data Engineer** set up the threshold number of missing samples to 5.	Remember (1)	0.3*1*1.58 = 0.474
16.	**Data Engineer**		**The Data Engineer** click on "save configuration" button	Remember (1)	1.58*1 = 1.58
17.	**System**		**The System** shows configuration ingestion system interface.		

18.	**Data Engineer**		**The Data Engineer** click on “configure number of sleep sessions to collect in execution and in monitoring modes”	Remember (1)	1.58*1 = 1.58
19.	**System**		**The System** shows tab to configure number of number of sleep sessions to collect in execution and in monitoring modes		
20.	**Data Engineer**		**The Data Engineer** set up number of sleep sessions to collect in execution and in monitoring modes	Remember (1)	1.58*1 = 1.58
21.	**Data Engineer**		**The Data Engineer** click on “save configuration” button	Remember (1)	1.58*1 = 1.58
22.	**System**		**The System** shows configuration ingestion system interface.		
23.	**Data Engineer**		**The Data Engineer** close the ingestion configuration panel.	Remember (1)	1.58*1 = 1.58
Total cost:					**26.86**

Cognitive effort description:

1. Remember (1): Interact with the user interface

3. Remember (1): Interact with the user interface

5. Remember (1): Interact with the user interface

6. Remember (1): Interact with the user interface

8. Remember (1): Interact with the user interface

9. Remember (1): Interact with the user interface

10. Remember (1): Interact with the user interface

11. Remember (1): Interact with the user interface

13. Remember (1): Interact with the user interface

15. Apply (3): **The Data Engineer** checks if the heart rate is constant

IF the heart rate is constant:

15.1. **The Data Engineer** set up the threshold number of missing samples to 10.

ELSE:

15.2. **The Data Engineer** set up the threshold number of missing samples to 5.

16. Remember (1): Interact with the user interface

18. Remember (1): Interact with the user interface

20. Remember (1): Interact with the user interface

21. Remember (1): Interact with the user interface

23. Remember (1): Interact with the user interface

Switch Ingestion System mode

SUB-TASK	%	ACTOR	ACTION	COGNITIVE EFFORT	TOATAL COST
1.		**Administrator**	**The Administrator** clicks on button "Configure Ingestion System Parameters"	Remember (1)	1.12*1 = 1.12
2.		**System**	**The System** shows configuration ingestion system interface.		
3.		**Administrator**	**The Administrator** click on "Switch mode" button	Remember (1)	1.12*1 = 1.12
4.		**System**	**The System** shows tab to switch mode		
5.		**Administrator**	**The Administrator** switch mode	Remember (1)	1.12*1 = 1.12
6.		**Administrator**	**The Administrator** change the number of records to synchronize.	Remember (1)	1.12*1 = 1.12
7.		**Administrator**	**The Administrator** click on "save	Remember (1)	1.12*1 = 1.12

			configuration" button		
8.		**System**	**The system** shows configuration ingestion system interface.		
9.		**Administrator**	**The Administrator** close the ingestion configuration panel.	Remember (1)	1.12*1 = 1.12
Total cost:					**6.72**

Cognitive effort description:

1. Remember (1): Interact with the user interface

3. Remember (1): Interact with the user interface

5. Remember (1): Interact with the user interface

6. Remember (1): Interact with the user interface

7. Remember (1): Interact with the user interface

9. Remember (1): Interact with the user interface

Configure the Preparation System

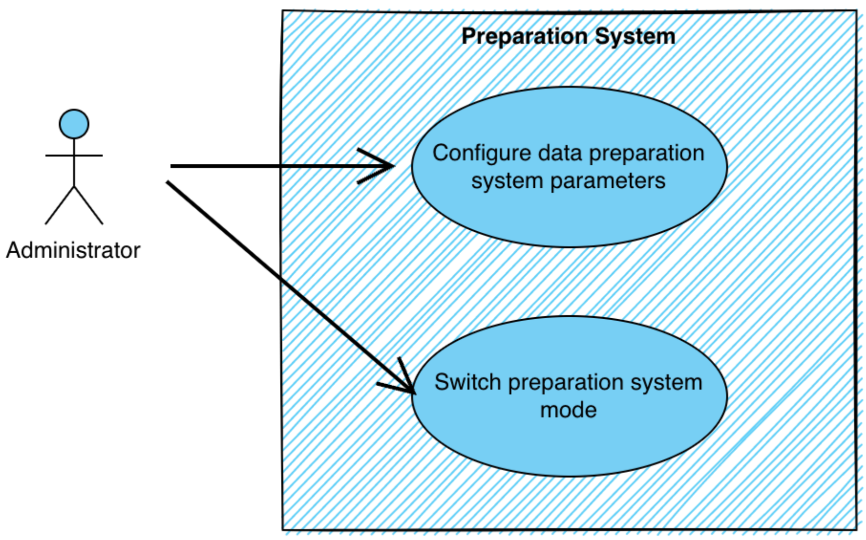


Figure 9: Use case diagram of the Preparation System

Configure the preparation system parameters

SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	TOATAL COST
1.	**Administrator**	**The Administrator** clicks on button "Configure Preparation System Parameters"	Remember (1)	1.12*1 = 1.12
2.	**System**	**The System** shows configuration ingestion system interface.		
3.	**Administrator**	**The Administrator** click on "configure address of segregation system" button	Remember (1)	1.12*1 = 1.12
4.	**System**	**The System** shows tab to set address of segregation system		
5.	**Administrator**	**The Administrator** set up segregation system address	Remember (1)	1.12*1 = 1.12
6.	**Administrator**	**The Administrator** click on "save configuration" button	Remember (1)	1.12*1 = 1.12
7.	**System**	**The system** shows configuration preparation system interface.		
8.	**Administrator**	**The Administrator** click on "configure address of execution system" button	Remember (1)	1.12*1 = 1.12
9.	**System**	**The System** shows tab to set address of execution system		
10.	**Administrator**	**The Administrator** set up execution system address	Remember (1)	1.12*1 = 1.12
11.	**Administrator**	**The Administrator** click on "save configuration" button	Remember (1)	1.12*1 = 1.12

12.	**System**	**The system** shows configuration preparation system interface.		
13.	**Administrator**	**The Administrator** click on "configure execution mode" button	Remember (1)	1.12*1 = 1.12
14.	**System**	**The System** shows tab to set execution mode		
15.	**Administrator**	**The Administrator** set up execution mode	Remember (1)	1.12*1 = 1.12
16.	**Administrator**	**The Administrator** click on "save configuration" button	Remember (1)	1.12*1 = 1.12
17.	**System**	**The System** shows configuration ingestion system interface.		
18.	**Administrator**	**The Administrator** close the ingestion configuration panel.	Remember (1)	1.12*1 = 1.12
Total cost:				**12.32**

Cognitive effort description:

1. Remember (1): Interact with the user interface

3. Remember (1): Interact with the user interface

5. Remember (1): Interact with the user interface

6. Remember (1): Interact with the user interface

8. Remember (1): Interact with the user interface

10. Remember (1): Interact with the user interface

11. Remember (1): Interact with the user interface

13. Remember (1): Interact with the user interface

15. Remember (1): Interact with the user interface

16. Remember (1): Interact with the user interface

18. Remember (1): Interact with the user interface

Switch Preparation System mode

SUB-TASK	%	ACTOR	ACTION	COGNITIVE EFFORT	TOATAL COST
1.		**Administrator**	**The Administrator** clicks on button "Configure Preparation System Parameters"	Remember (1)	1.12*1 = 1.12
2.		**System**	**The System** shows configuration ingestion system interface.		
3.		**Administrator**	**Administrator clicks on "Switch mode" button**	Remember (1)	1.12*1 = 1.12
4.		**System**	**The System** shows tab to switch mode		
5.		**Administrator**	**The Administrator** switch mode	Remember (1)	1.12*1 = 1.12
6.		**Administrator**	**The Administrator** click on "save configuration" button	Remember (1)	1.12*1 = 1.12
7.		**System**	**The system** shows configuration ingestion system interface.		
8.		**Administrator**	**The Administrator** close the ingestion configuration panel.	Remember (1)	1.12*1 = 1.12
Total cost:					5.6

1. Remember (1): Interact with the user interface

3. Remember (1): Interact with the user interface

5. Remember (1): Interact with the user interface

6. Remember (1): Interact with the user interface

8. Remember (1): Interact with the user interface

4.4.2 Generate Sleep Session Set (Fabio Buchignani)

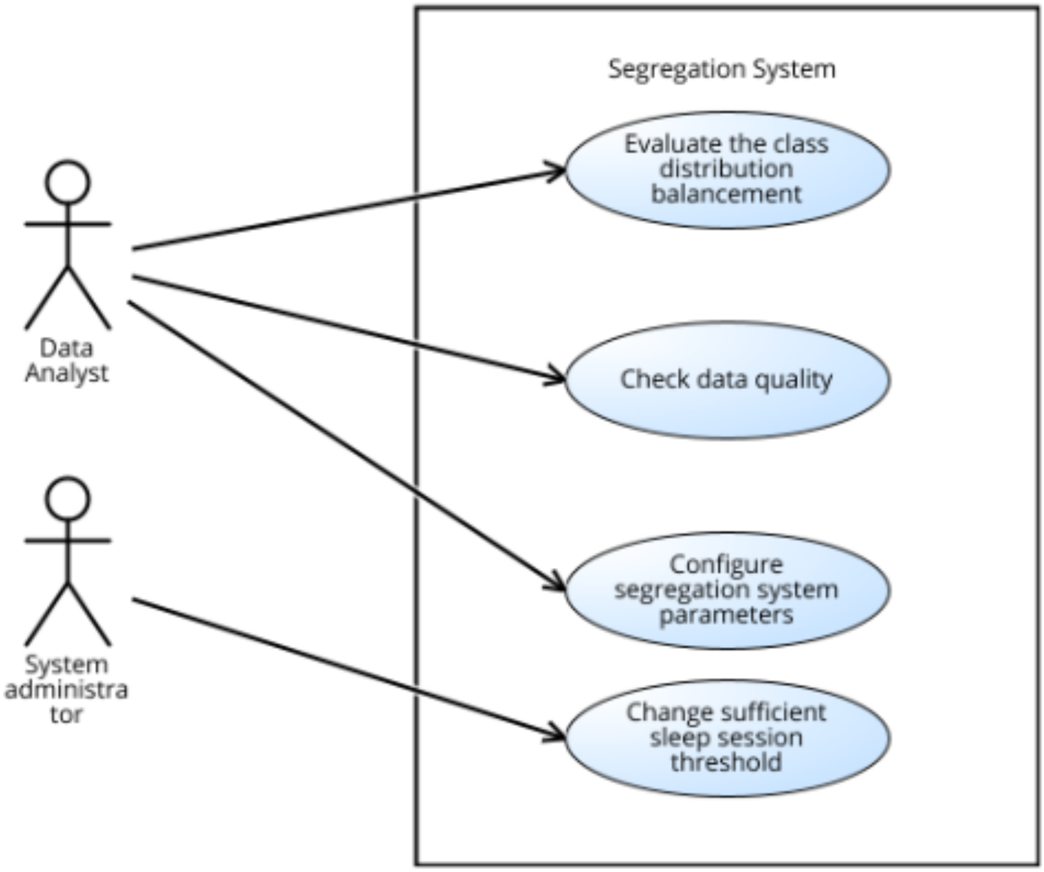


Figure 10: Use case diagram of the Segregation System

Evaluate the class distribution balancement

Description of the task: The data analyst must check whether the classes for sleep quality are balanced or not in the training data. The three classes are low sleep quality, medium sleep quality, and high sleep quality.

During the configuration, the threshold about data balancement is set (we assume this threshold is the maximum difference from the average number of samples, expressed in percentage on the total number of samples).

The system therefore is initialized with this threshold and can signal whether a class exceeds the tolerance or not, but the final decision is made by the human, that sets up the reconfiguration message accordingly, if needed.

Role	Subtask		%	Cognitive Effort	Cost
Data analyst	1. Clicks on balancement reports button			Remember (1)	1 * 1 * 1= 1
System	2. Shows balancement reports interface				
Data analyst	3. Selects the specific balancement report			Remember (1)	1 * 1 * 1= 1
System	4. Shows the specific balancement report				

Data Analyst	5. Investigates the balancement report			Understand (2)	1 * 2 * 1= 2
	6. IF classes are unbalanced		30%		
Data Analyst		6.1. Clicks on reconfigure button		Remember (1)	1 * 1 * 0.3 = 0.3
System		6.2. Shows reconfigure interface			
Data Analyst		6.3. Sets up reconfigure message		Analyze (4)	1 * 4 * 0.3 = 1.2
Data Analyst		6.4. Clicks on send button		Remember (1)	1 * 1 * 0.3 = 0.3
System		6.5. Sends the reconfigure message to the messaging system			
Data Analyst	7. Closes balancement report			Remember (1)	1 * 1 * 1 = 1
Data Analyst	8. Closes balancement reports interface			Remember (1)	1 * 1 * 1 = 1
	9. END				Total cost: **7.8**

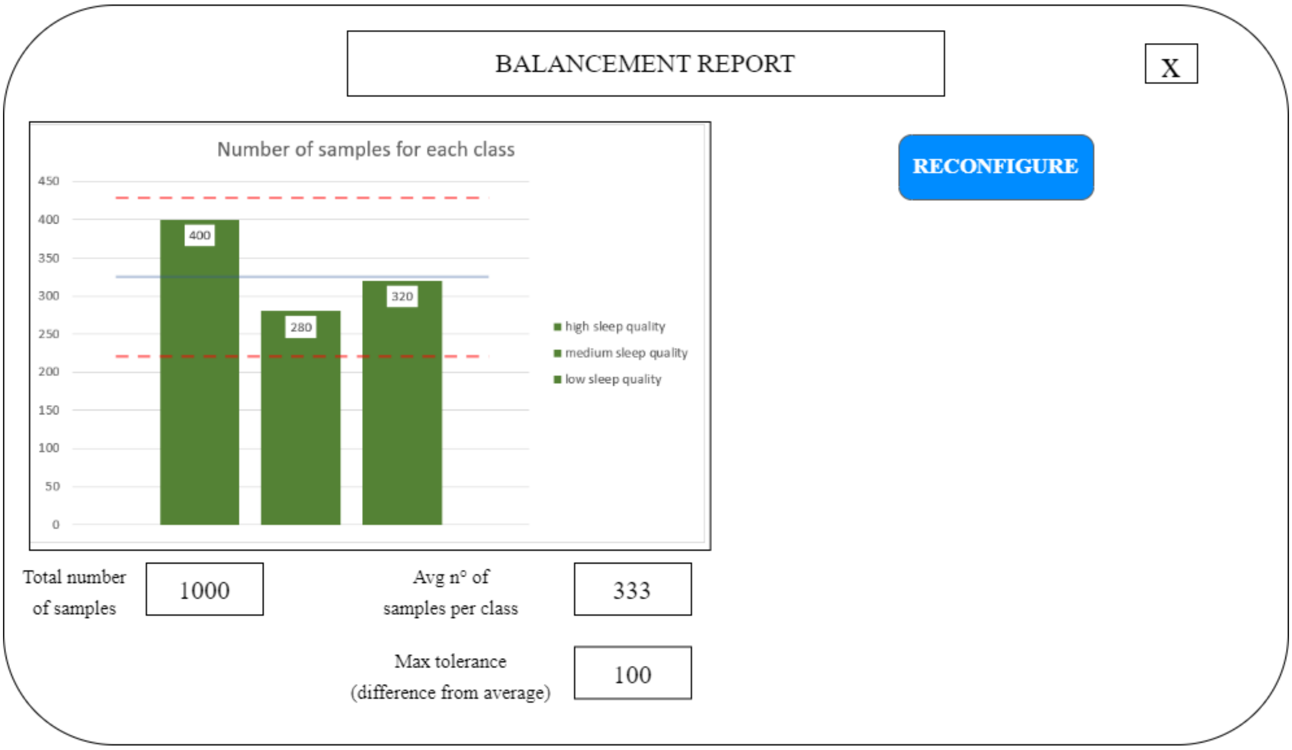


Figure 11: Balancement report

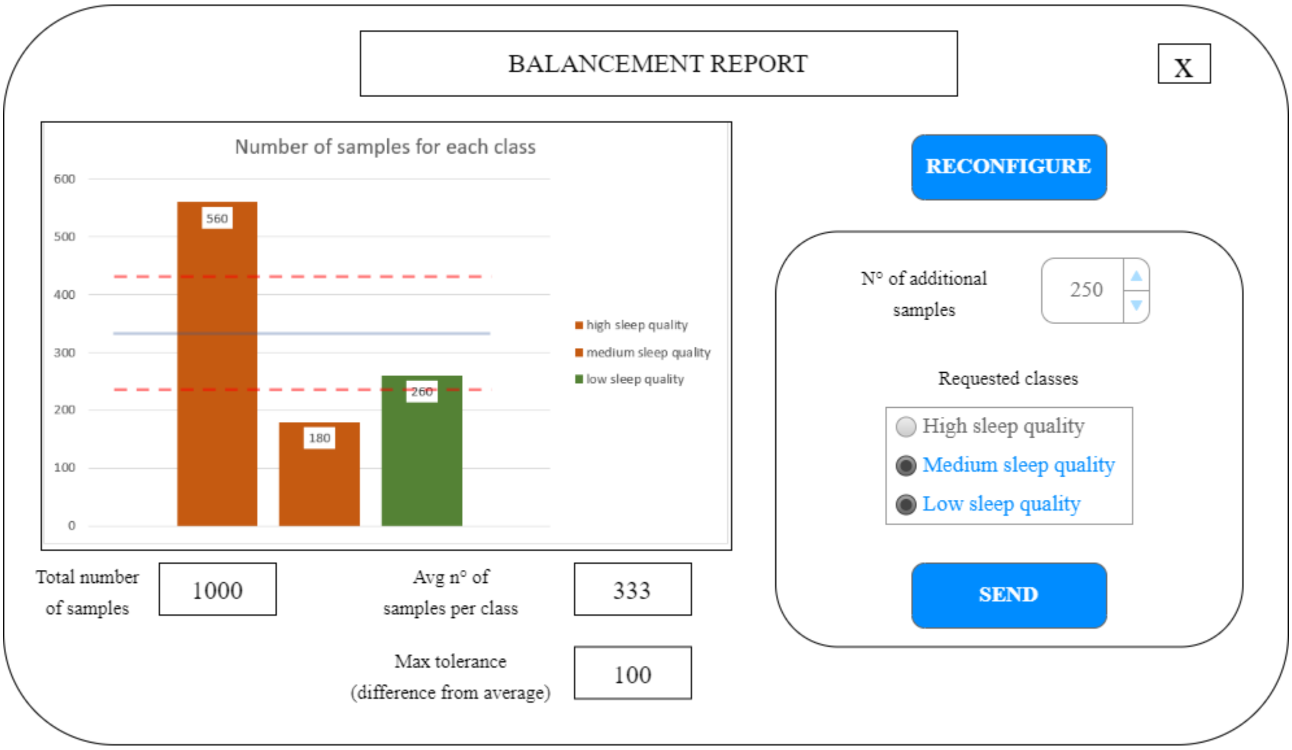


Figure 12: Balancement report reconfigure interface

Description of cognitive effort:

- 1.Remember(1): interact with user interface
- 3.Remember(1): interact with user interface
- 5.Understand(2): If the color of at least one bar in the histogram is orange, the system is signaling that reconfiguration is needed
- 6.1.Remember(1): interact with user interface
- 6.3.Analyze(4): The human configures the reconfiguration message depending on the specific shape of the graph and the information contained in the report. For example:
  - If the dataset is highly imbalanced, with one class over-represented and two classes under-represented with respect to the tolerance, then the number of additional samples is equal to the total number of samples multiplied by two and the requested classes are the under-represented classes.
  - If the dataset is quite imbalanced, with one class over-represented and two classes under-represented with respect to the tolerance, then the number of additional samples is equal to the total number of samples and the requested classes are the under-represented classes.
  - If the dataset is a bit imbalanced, with one class over-represented and two classes under-represented with respect to the tolerance, then the number of additional samples is equal to half of the total number of samples and the requested classes are the under-represented classes.
  - If the dataset is a highly imbalanced, with two classes over-represented and one class under-represented with respect to the tolerance, then the number of additional samples is equal to half of the total number of samples and the requested classes is the under-represented class.
  - Etc.
- 6.4.Remember(1): interact with user interface

- 7.Remember(1): interact with user interface
- 8.Remember(1): interact with user interface

Check data quality

Description of the task: the data analyst must check the report on data quality produced by the system. In particular the system gets the prepared sleep session and produces a radar diagram for representing the input coverage of the features (Max heart rate, Max wrist movement, Min heart rate, Avg heart rate, Avg wrist movement, awake time).

The human, on the basis of experience, with the aid of the radar diagram and the information provided by the system on the characteristics of the user should judge the quality of the data and decide if it is enough.

If the data quality is not enough, the human will click the reject button, in this way the system will automatically prepare the default reconfigure message to be used when data quality is not enough and will send it to the messaging system. If the data quality is enough, the human will click the accept button, and the workflow will go on.

Role	Subtask		%	Cognitive Effort	Cost
Data analyst	1. Clicks on data quality reports button			Remember (1)	1 * 1 * 1 = 1
System	2. Shows data quality reports interface				
Data analyst	3. Selects the specific data quality report			Remember (1)	1 * 1 * 1 = 1
System	4. Shows the specific data quality report				
Data Analyst	5. Analyses the radar diagram			Analyse (4)	1 * 4 * 1 = 4
	6. IF data quality is enough		10%		
Data Analyst		6.1. Clicks on accept button		Remember (1)	1 * 1 * 0.1 = 0.1
	7. **ELSE**		90%		
Data Analyst		7.1. Click on reject button		Remember (1)	1 * 1 * 0.9 = 0.9
System		7.2. Sends the reconfigure message to the messaging system			
Data Analyst	8. Closes data quality report			Remember (1)	1 * 1 * 1 = 1
Data Analyst	9. Closes data quality reports interface			Remember (1)	1 * 1 * 1 = 1
	10. END				Total cost: **9.0**

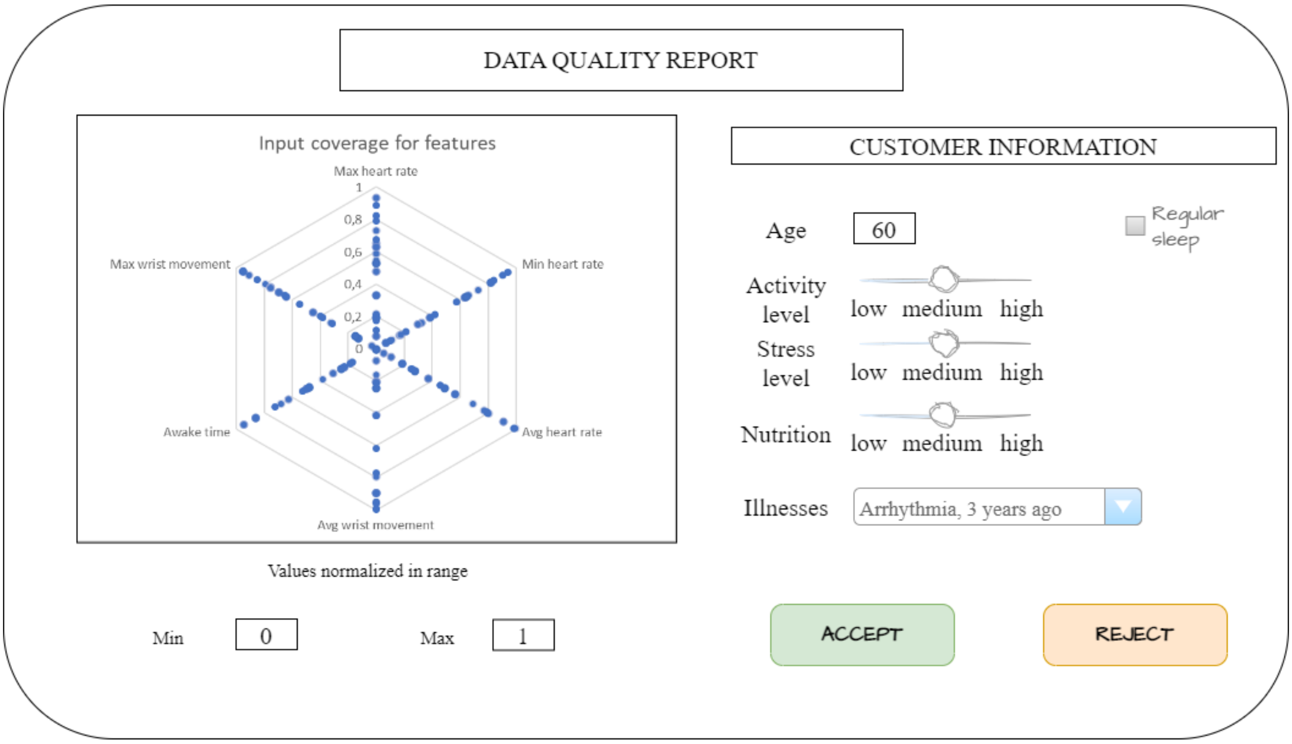


Figure 13: Data quality report interface

Description of cognitive effort:

- 1.Remember(1): interact with user interface
- 3.Remember(1): interact with user interface
- 5.Analyse(4): The data analyst needs to analyse the radar diagram and the customer information, deciding whether this level of input coverage is enough or not. In particular the data analyst should evaluate each feature of the radar diagram and assess if the coverage is uniformly distributed, if it is not, it still could be fine depending on the user age, user level of activity, level of wealth, current illnesses and so on. For example, depending on certain illnesses or age, the lack of values for max heart rate, min heart rate and avg heart rate could be acceptable.
- 6.1.Remember(1): interact with user interface
- 7.1.Remember(1): interact with user interface
- 8.Remember(1): interact with user interface
- 9.Remember(1): interact with user interface

Configure segregation system parameters

Description of the task: During the initial configuration, the segregation system needs to be configured with some parameter. These are:

- The IP addresses of the development system and the messaging system.
- The “sufficient sleep sessions” threshold, representing how many samples are to be taken in account before generating the sets, that depend on customer behavior and customer information.
- The threshold regarding the tolerance about class balancement, which is expressed in percentage of the total number of samples and represents the maximum difference from the average number of samples to consider a class balanced with respect to the others. This value is always the same.
- The percentage of the data that will form the training set, the one that will form the validation set and the one that will form the test set. This value is always the same.

Role	Subtask		%	Cognitive Effort	Cost
Data analyst	1. Clicks on configure segregation system			Remember (1)	1 * 1 * 1 = 1
System	2. Shows the segregation system configuration interface				
Data analyst	3. Selects the textbox representing the IP address of the messaging system			Remember (1)	1 * 1 * 1 = 1
Data analyst	4. Enters the IP address of the messaging system			Remember (1)	1 * 1 * 1 = 1
Data Analyst	5. Selects the textbox representing the IP address of the development system			Remember (1)	1 * 1 * 1 = 1
Data Analyst	6. Enters the IP address of the development system			Remember (1)	1 * 1 * 1 = 1
Data Analyst	7. Selects the textbox about the “sufficient sleep sessions” threshold value			Remember (1)	1 * 1 * 1 = 1
Data Analyst	8. Enters the “sufficient sleep sessions” threshold value			Analyse (4)	1 * 4 * 1 = 4
Data Analyst	9. Selects the textbox about the class balancement tolerance			Remember (1)	1 * 1 * 1 = 1
Data Analyst	10. Enters the class balancement tolerance value			Remember (1)	1 * 1 * 1 = 1
Data Analyst	11. Selects the panel about split percentage of each set			Remember (1)	1 * 1 * 1 = 1
Data Analyst	12. Enters the percentages of the data for forming each sleep session set			Remember (1)	1 * 1 * 1 = 1
Data Analyst	13. Closes segregation system configuration interface			Remember (1)	1 * 1 * 1 = 1
	14. END				Total cost: **15.0**

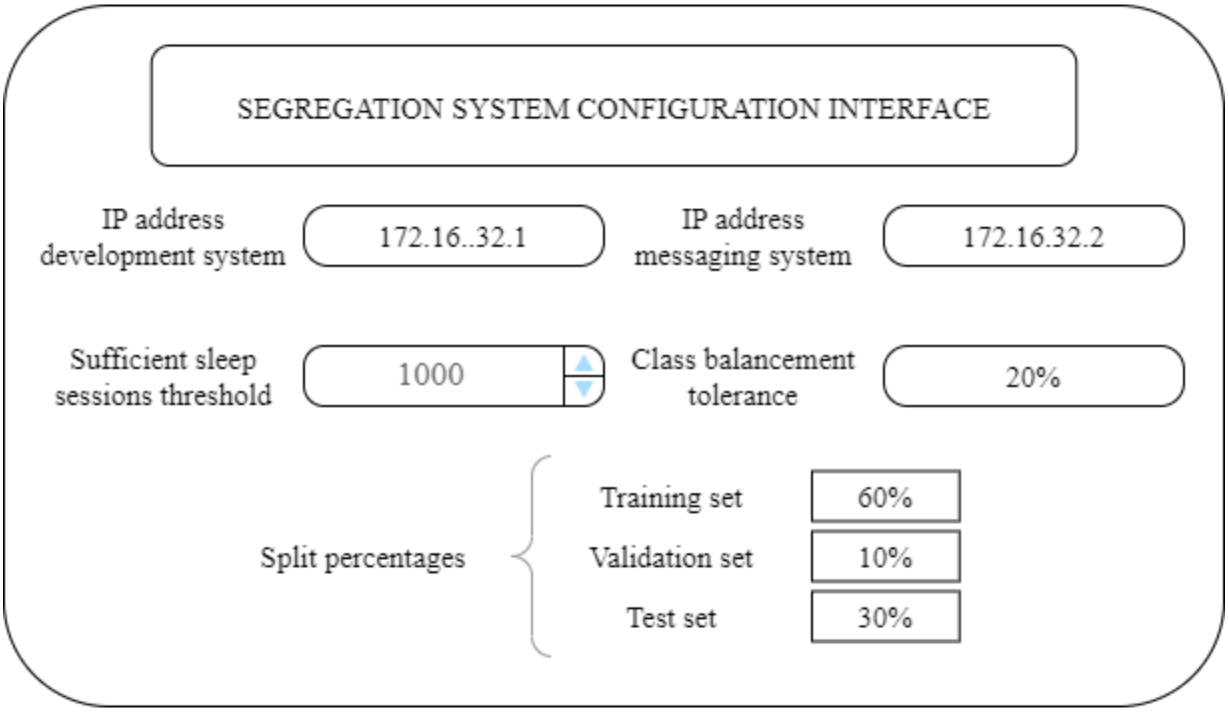


Figure 14: Segregation system configuration interface

Description of cognitive effort:

- 1.Remember(1): interact with user interface
- 3.Remember(1): interact with user interface
- 4.Remember(1): the IP address of the messaging system is always the same
- 5.Remember(1): interact with user interface
- 6.Remember(1): the IP address of the development system is always the same
- 7.Remember(1): interact with user interface
- 8.Analyse(4): The data analyst has to understand which value of the "sufficient sleep sessions" threshold could be the right one depending on the customer information and behavior. For example, if the person is an elderly, this value could be higher because in general he could have no fixed sleep schedule, so values could be more noisy. If the person instead is younger, works, does activity during the day and is in a good form, sleep sessions could be less noisy and this value could be decreased. Since the potential cases are a lot, experience from data analyst is needed to select the right value of this threshold
- 9.Remember(1): interact with user interface
- 10.Remember(1): The data analyst sets the threshold about class balancement always at the same level.
- 11.Remember(1): interact with user interface
- 12.Remember(1): The data analyst decides which percentages of the data to use for the training set, the test set and the validation set. These percentages are always the same.
- 13.Remember(1): interact with user interface

Change sufficient sleep sessions threshold

Description of the task: During reconfiguration, the segregation system must be informed on how many additional prepared sleep sessions are going to arrive, and the system administrator has to change the "sufficient sleep sessions" threshold accordingly.

Role	Subtask		%	Cognitive Effort	Cost
Administrator	1. Clicks on configure segregation system			Remember (1)	1 * 1 .12 = 1.12
System	2. Shows the segregation system configuration interface				
Administrator	3. Selects the textbox about the "sufficient sleep sessions" threshold value			Remember (1)	1 * 1 .12 = 1.12
Administrator	4. Set the value of sufficient sleep sessions threshold			Remember (1)	1 * 1 .12 = 1.12
Administrator	5. Closes segregation system configuration interface			Remember (1)	1 * 1 .12 = 1.12
	6. End				Total cost: 4.48

Description of cognitive effort:

- 1.Remember(1): interact with user interface
- 3.Remember(1): interact with user interface
- 4.Remember(1): The new value of the "sufficient sleep session threshold" is just the previous one summed up with the value in the configuration message. This value is visible from the user interface.
- 5.Remember(1): interact with user interface

4.4.3 Develop sleep quality classifier (Antonio Patimo)

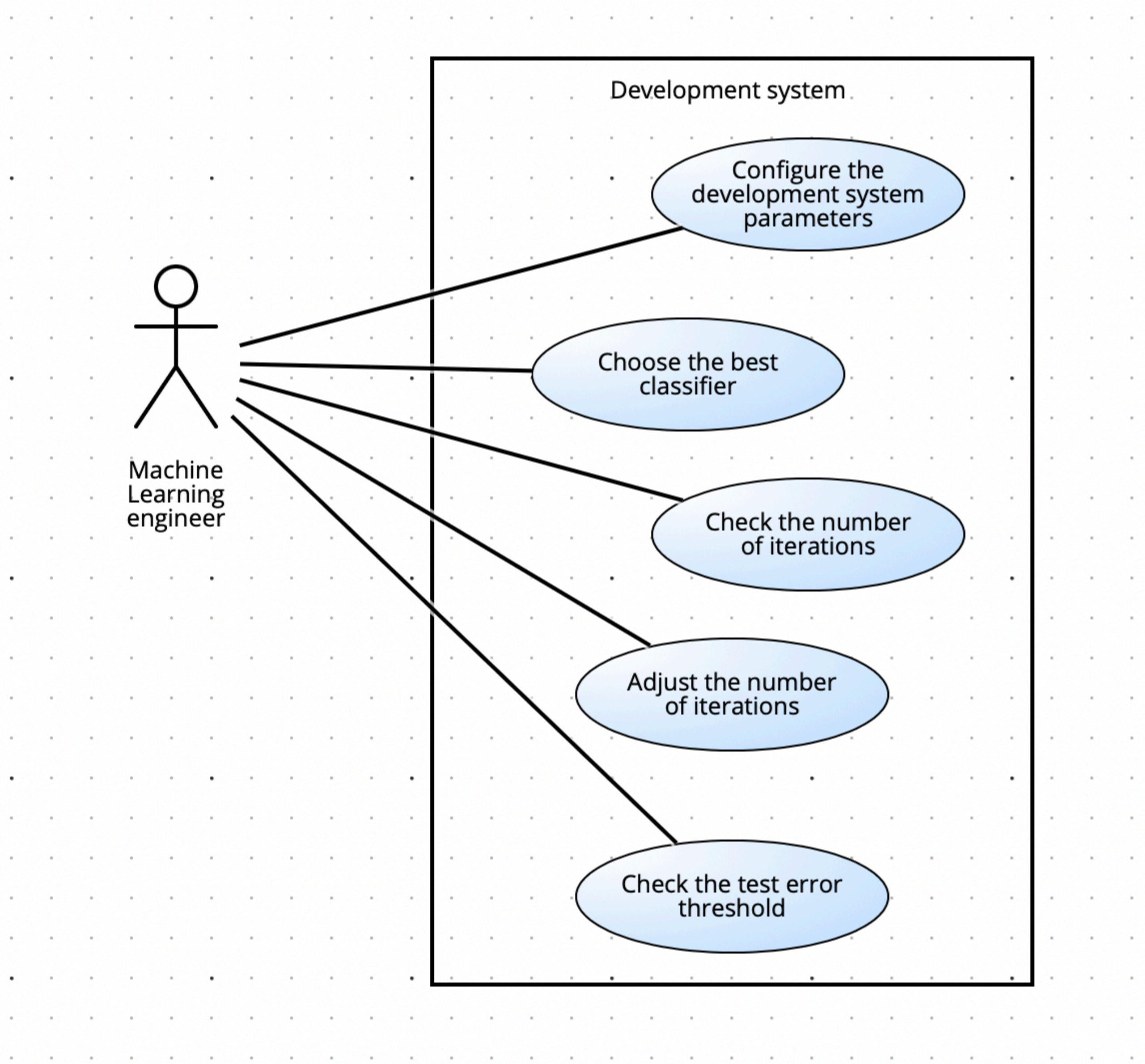


Figure 15: Use case diagram of the Development System

Configure the Development system parameters

Task	%	Role	Cognitive effort	Total cost
1. Open configuration interface		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
2. Shows configuration interface		System		

3. Click on the textbox to set the hidden layer interval		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
4. Set hidden layer interval		M.L. engineer	Analyze (4)	4 * 1.84 = 7.36
5. Click on the textbox to set the interval of the number of neurons		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
6. Set number of neurons interval		M.L. engineer	Analyze (4)	4 * 1.84 = 7.36
7. Click on the textbox to set the test error threshold		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
8. Set test error threshold		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
9. Click on the textbox to set the number of initial iterations		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
10. Set the initial number of iterations		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
11.Click on the textbox to set the ip address of the execution system		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
12. Set the ip address of the execution system		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
13. Save configuration		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
14. Close configuration interface		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
				Total Cost: 34.96

Cognitive effort description:

1. Remember (1): Interact with the user interface

3. Remember (1): Interact with the user interface

4. Analyze (4): The machine learning engineer must assess the correct interval of the hidden layers based on his knowledge and experience and based on multiple criteria of the user to evaluate the complexity needed for the classifier.

5. Remember (1): Interact with the user interface

6. Analyze (4): The machine learning engineer must assess the correct interval of the number of neurons based on his knowledge and experience and based on multiple criteria of the user to evaluate the complexity needed for the classifier.

7. Remember (1): Interact with the user interface

8. Remember (1): Se a given parameter

9. Remember (1): insert the given threshold

10. Remember (1): Set a given parameter

11. Remember (1): Interact with the user interface

12. Remember (1): Set a given parameter

13. Remember (1): Interact with the user interface

14. Remember (1): Interact with the user interface

Development system configuration interface:

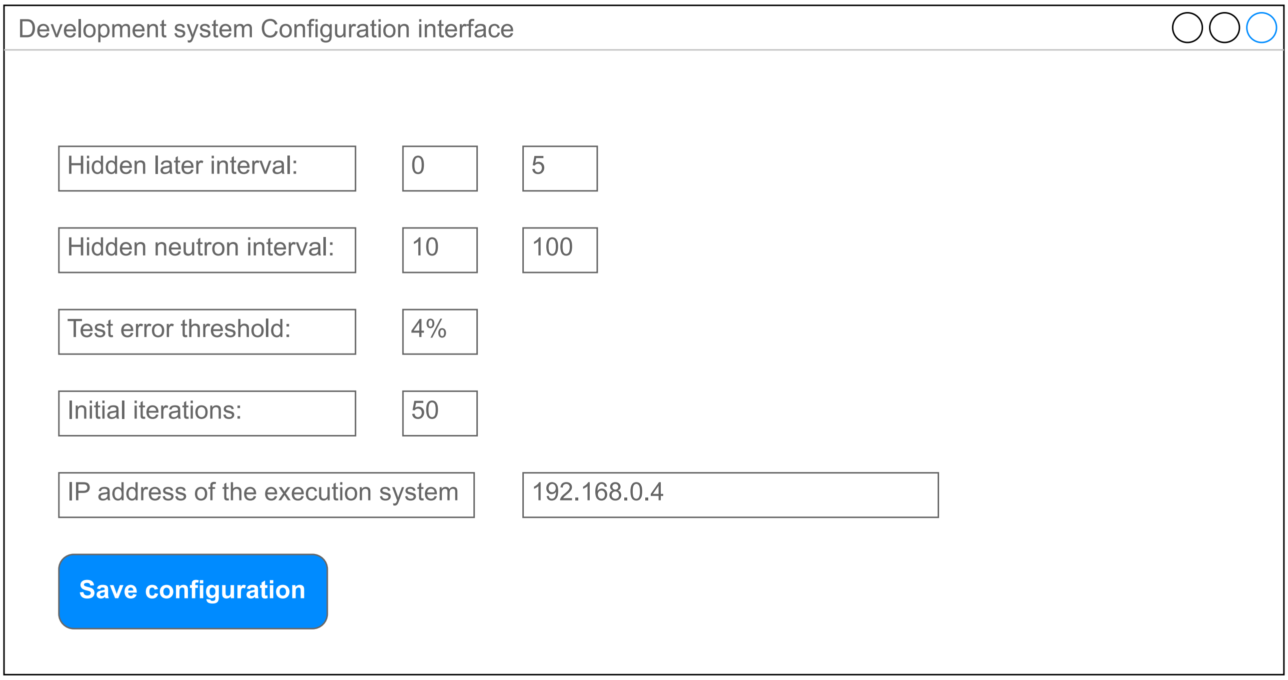


Figure 16: Development system configuration interface

Choose the best classifier

Task	%	Role	Cognitive effort	Total Cost
1. Open best 5 classifiers report		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
2. Shows a list of the best classifiers		System		
3. Choose the three classifiers with the least validation error		M.L. engineer	Apply (3)	3 * 1.84 = 5.52
4. Among these choose the two classifiers with		M.L. engineer	Apply (3)	3 * 1.84 = 5.52

the lower number of neurons				
5. Among these choose the classifier with least hidden layer. In case of draw choose the one with the lowest validation error.		M.L engineer	Apply (3)	3 * 1.84 = 5.52
6. Select the best network		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
7. Save the best network		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
8. Close report		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
				Total cost: 23.92

Cognitive effort description:

1. Remember (1): Interact with the user interface

3. Apply (3): Apply a rule to select the best classifier among the best 5 classifiers

4. Apply (3): Apply a rule to select the best classifier among the best 5 classifiers

5. Apply (3): Apply a rule to select the best classifier among the best 5 classifiers

6. Remember (1): Interact with the user interface

7. Remember (1): Interact with the user interface

8. Remember (1): Interact with the user interface

Best 5 classifiers report:

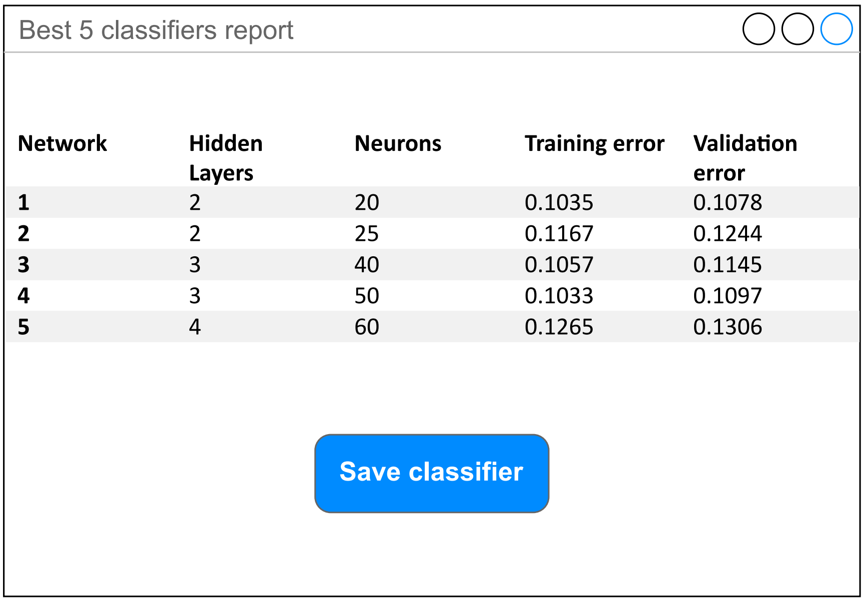


Figure 17: Best 5 classifiers report

Rules to apply:

1. Choose the three classifiers with the least validation error
2. Among these choose the two classifiers with the lower number of neurons
3. Among these choose the classifier with least hidden layer. In case of draw choose the one with the lowest validation error.

Check the number of iterations

Task	%	Role	Cognitive effort	Total Cost
1. Open training report		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
2. Shows the training report		System		
3. If the loss is flat for at least the half of the iterations	20%	M.L. engineer	Apply (3)	0.20 * 3 * 1.84 = 1.1
3.1. Adjust the number of iterations (reduce by one third) (external task)				
4. If the loss is not flat at the end of the iterations	20%	M.L. engineer	Apply (3)	0.20 * 3 * 1.84 = 1.1
4.1 Adjust the number of iterations (enlarge by one third) (external task)				
5. Else the number of iterations is good	60%	M.L. engineer	Remember (1)	0.60 * 1 * 1.84 = 1.1
6. Save current configuration		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
7. Close training report		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
				Total cost = 8.82

Cognitive effort description:

1. Remember (1): Interact with the user interface

3. Apply (3): Apply a rule

4. Apply (3): Apply rule

5. Remember (3): Else of a rule

6. Remember (1): Interact with the user interface

7. Remember (1): Interact with the user interface

Early training report:

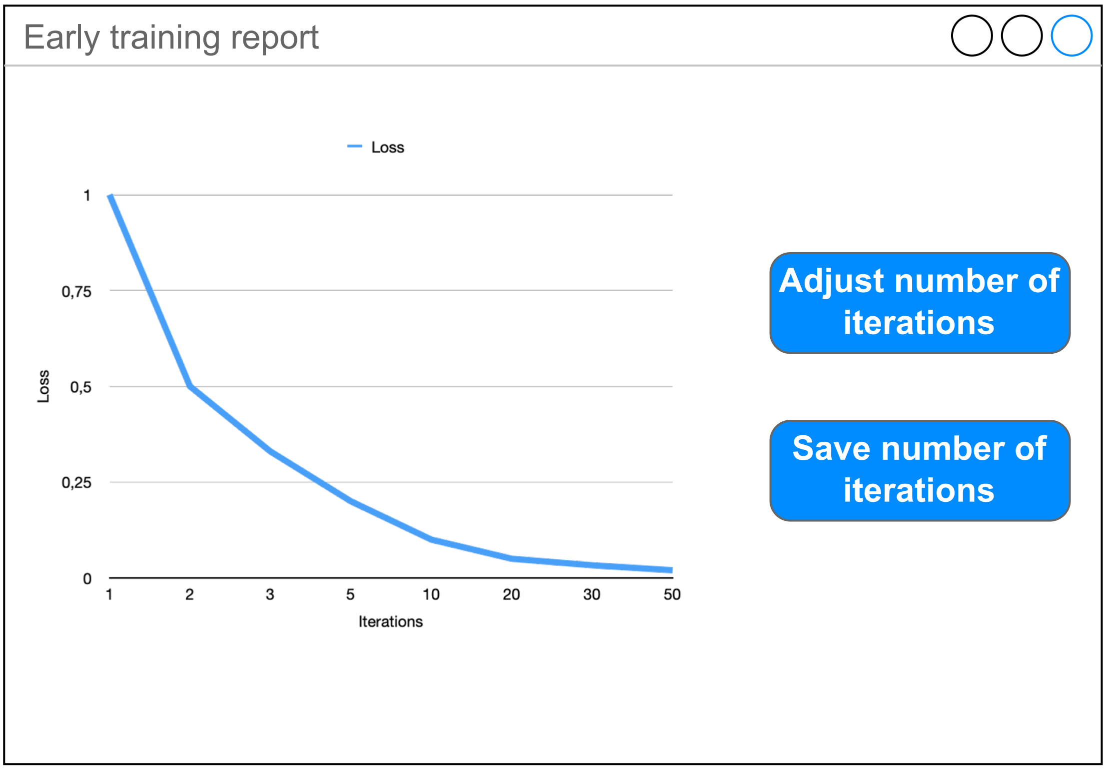


Figure 18: Early training report

Rules to apply:

1. If the loss is flat for at least the half of the iterations

    1.1. then reduce by one third the number of iterations to manage overfitting

2. if the loss is not flat at the end of the iterations

    2.1 then enlarge by one third the number of iterations

3. Else, the number of iterations is good

Adjust the number of iterations

Task	%	Role	Cognitive effort	Total cost
1. Open the number of iterations configuration interface		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
2. Shows the interface		System		
3. Adjust the number of iterations based on the passed parameter		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
4. Save the current configuration		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
5. Close the configuration interface		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
				Total cost = 7.36

Cognitive effort description:

1. Remember (1): Interact with the user interface

3. Remember (1): Interact with the user interface

4. Remember (1): Interact with the user interface

5. Remember (1): Interact with the user interface

Adjust the number of iterations interface:

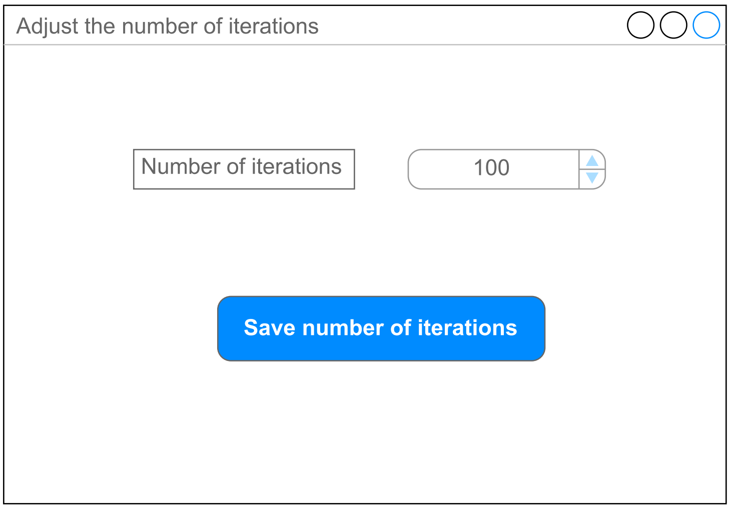


Figure 19: Adjust the number of iterations interface

Check the test error threshold

Task	%	Role	Cognitive effort	Total cost
1. Open test evaluation report		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
2. Show test report		System		
3. Understand if the error is under the threshold		M.L. engineer	Understand (2)	2 * 1.84 = 3.68
4. IF the test error is under the threshold save the classifier	90 %	M.L. engineer	Remember (1)	0.90 * 1 * 1.84 = 1.656
5. Else send “Collect new data” message	10 %	M.L. engineer	Remember (1)	0.10 * 1 * 1.84 = 0.184
6. Close test report		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
				Total cost: 9.2

Cognitive effort description:

1. Remember (1): Interact with the user interface

3. Understand (2): The machine learning engineer needs to understand if the test error is under the threshold

4. Remember (1): Remember the previous understanding if the test error is under the threshold

5. Remember (1): Interact with the user interface

Test evaluation report:

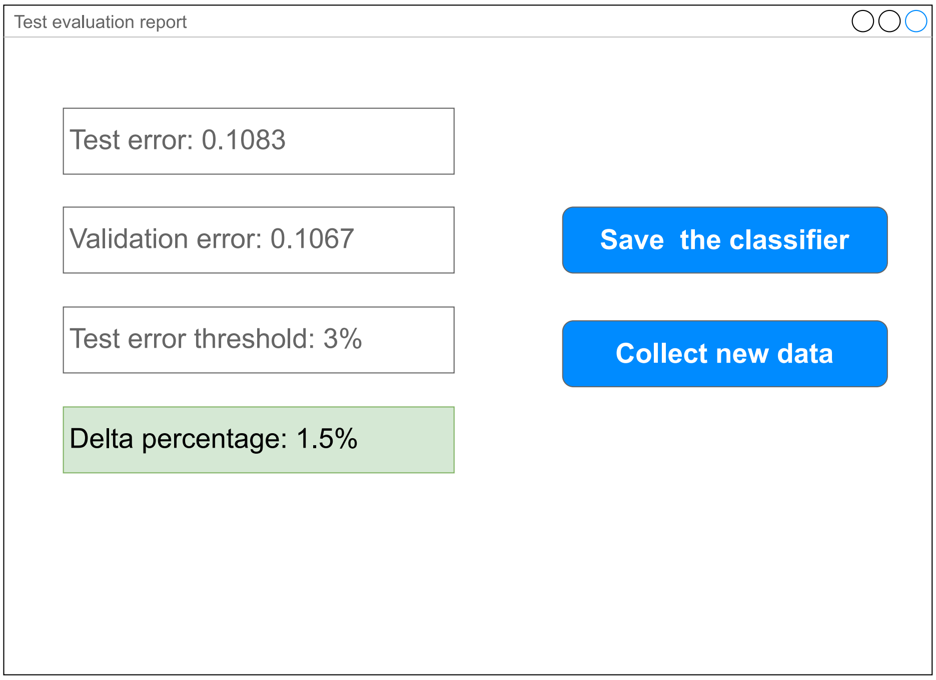


Figure 20: Test evaluation report interface

4.4.4 Check Sleep Quality Classifier (Lorenzo Massagli)

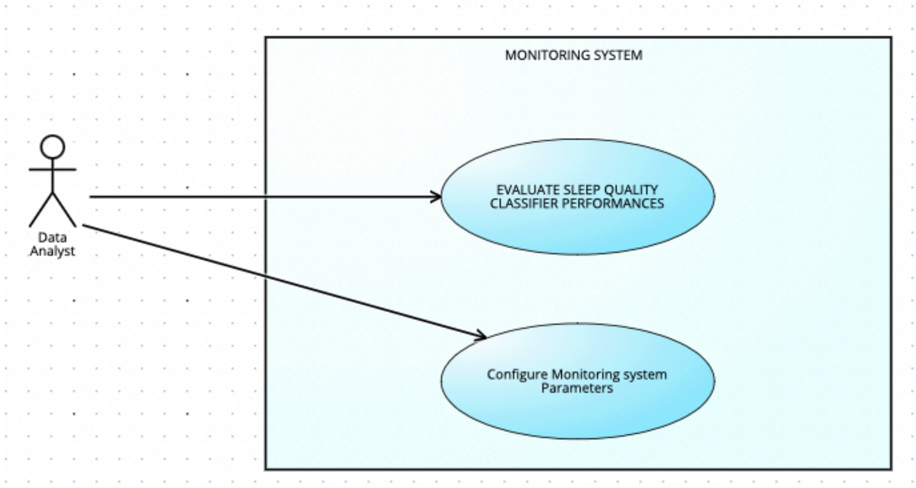


Figure 21: Use case diagram of monitoring system

Evaluate sleep quality classifier performances

Task	Subtask	%	Role	Cognitive Effort	Total cost
Evaluate sleep quality classifier performances	1. Access the monitoring interface		Data Analyst	Remember (1)	1 * 1 = 1
	2. Displays the Monitoring Report		System		
	3. Evaluates the number of misclassified items		Data Analyst	Understand (2)	1 * 2 = 2
	4. **IF** the number of misclassified labeled items is greater than the specified threshold	10			
	4.1. Sets Development mode		Data Analyst	Remember (1)	1 * 1 * 0.10 = 0.10
	4.2. Closes mode interface		Data Analyst	Remember (1)	1 * 1 * 0.10 = 0.10

	5. ELSE	90			
	5.1 Click “check Quality OK” button		Data Analyst	Remember (1)	1 * 1 * 0.90 = 0.90
	5.2. Closes monitoring interface		Data Analyst	Remember (1)	1 * 1 * 0.90 = 0.90
	6. END				Total cost = 5

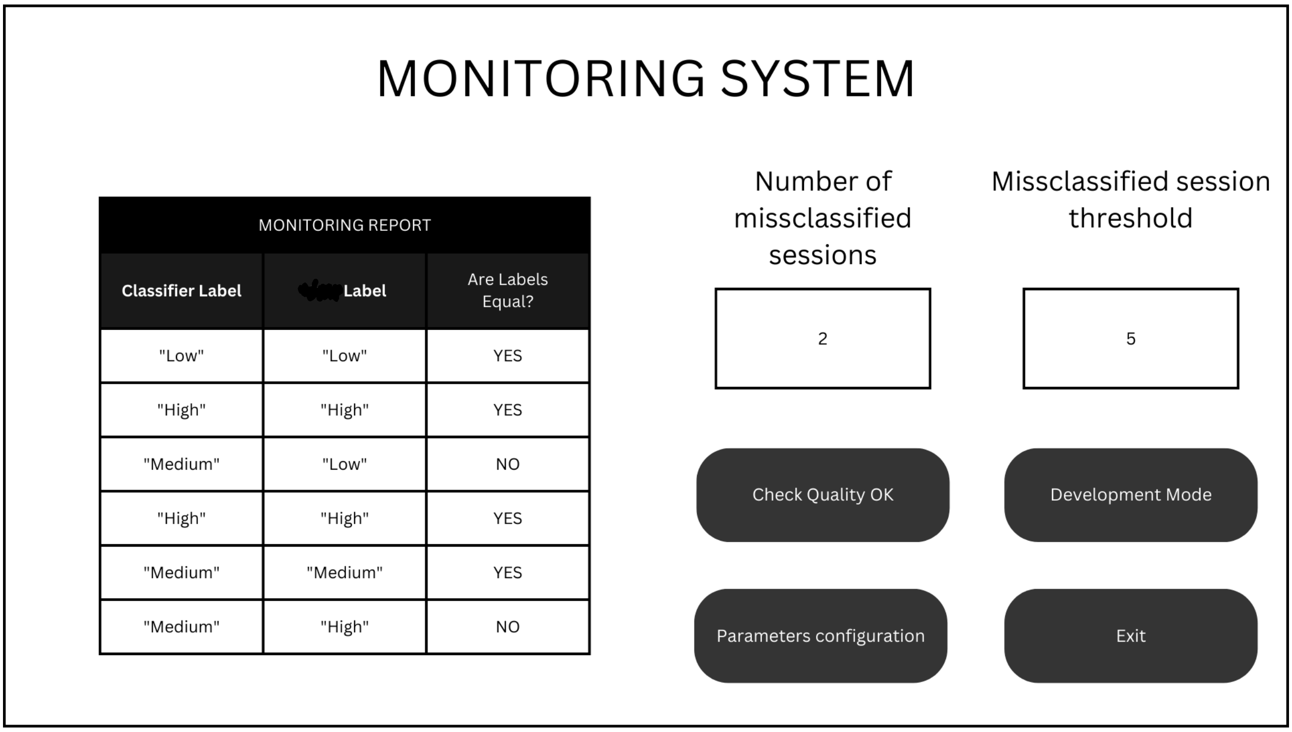


Figure 22: Monitoring system interface

Configure Monitoring System Parameters

Task	Subtask		%	Role	Cognitive Effort	Total cost
Configure Monitoring System Parameters	1. Access the monitoring interface			Data Analyst	Remember (1)	1 * 1 = 1
	2. Select the monitoring configuration interface			Data Analyst	Remember (1)	1 * 1 = 1

	3. Displays the monitoring configuration interface			System		
	4. Set number of session used for monitoring			Data Analyst	Remember (1)	1 * 1 = 1
	5. Set misclassified sessions threshold			Data Analyst	Remember (1)	1 * 1 = 1
	6. Save changes				Remember (1)	1 * 1 = 1
	7. Close monitoring configuration interface			Data Analyst	Remember (1)	1 * 1 = 1
	8. END					Total cost = 6

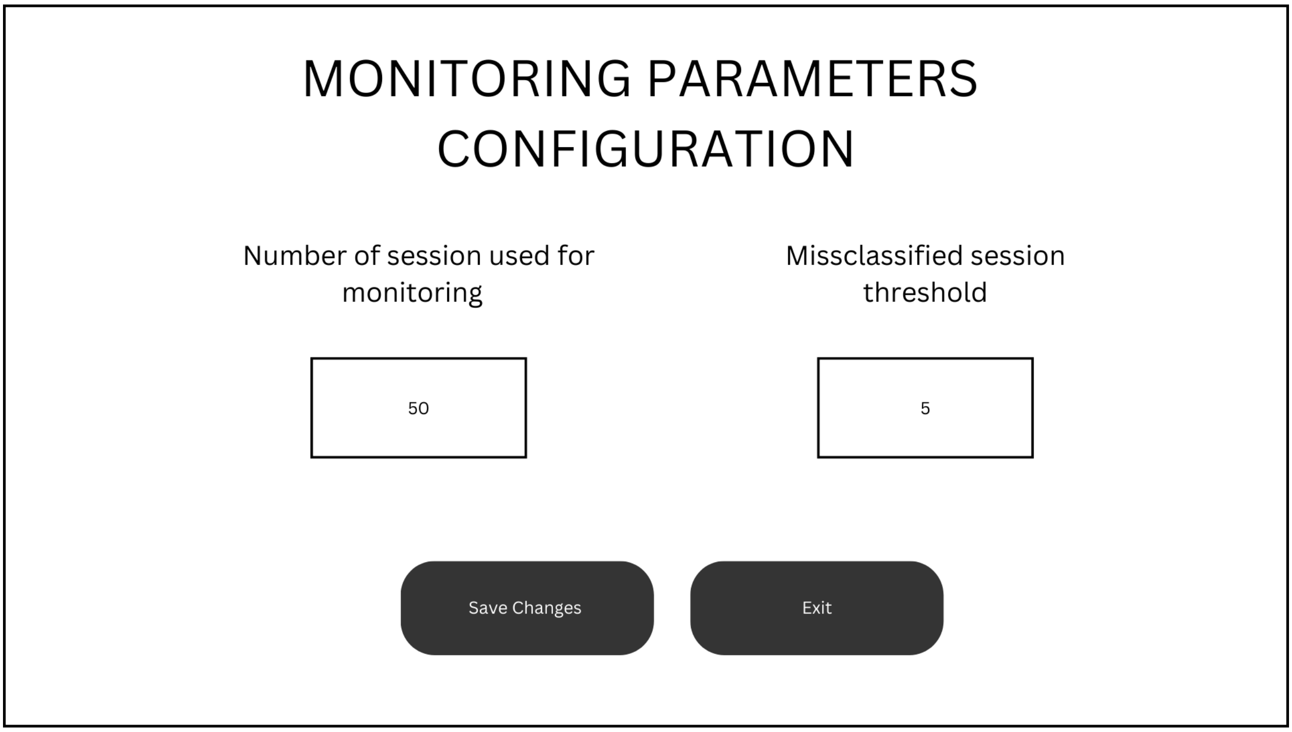


Figure 23: Monitoring parameters configuration

4.4.5 Restart data collection (Lorenzo Massagli)

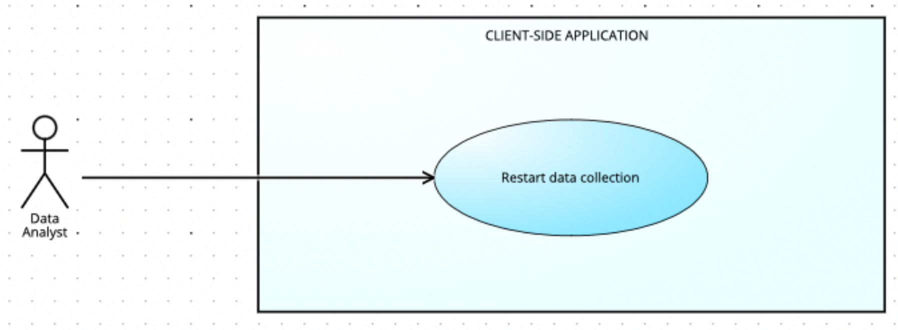


Figure 24: Use case of client-side application

Task	Subtask		%	Role	Cognitive Effort	Total cost
Restart data collection	1. Access the client-side application interface			Data Analyst	Remember (1)	1 * 1 = 1
	2. Select the client-side application configuration interface			Data Analyst	Remember (1)	1 * 1 = 1
	3. Displays the client-side application configuration interface			System		
	4. Activate the data collection			Data Analyst	Remember (1)	1 * 1 = 1
	5. Close client-side application configuration interface			Data Analyst	Remember (1)	1 * 1 = 1
	6. END					Total cost = 4

5. As-Is Simulation model

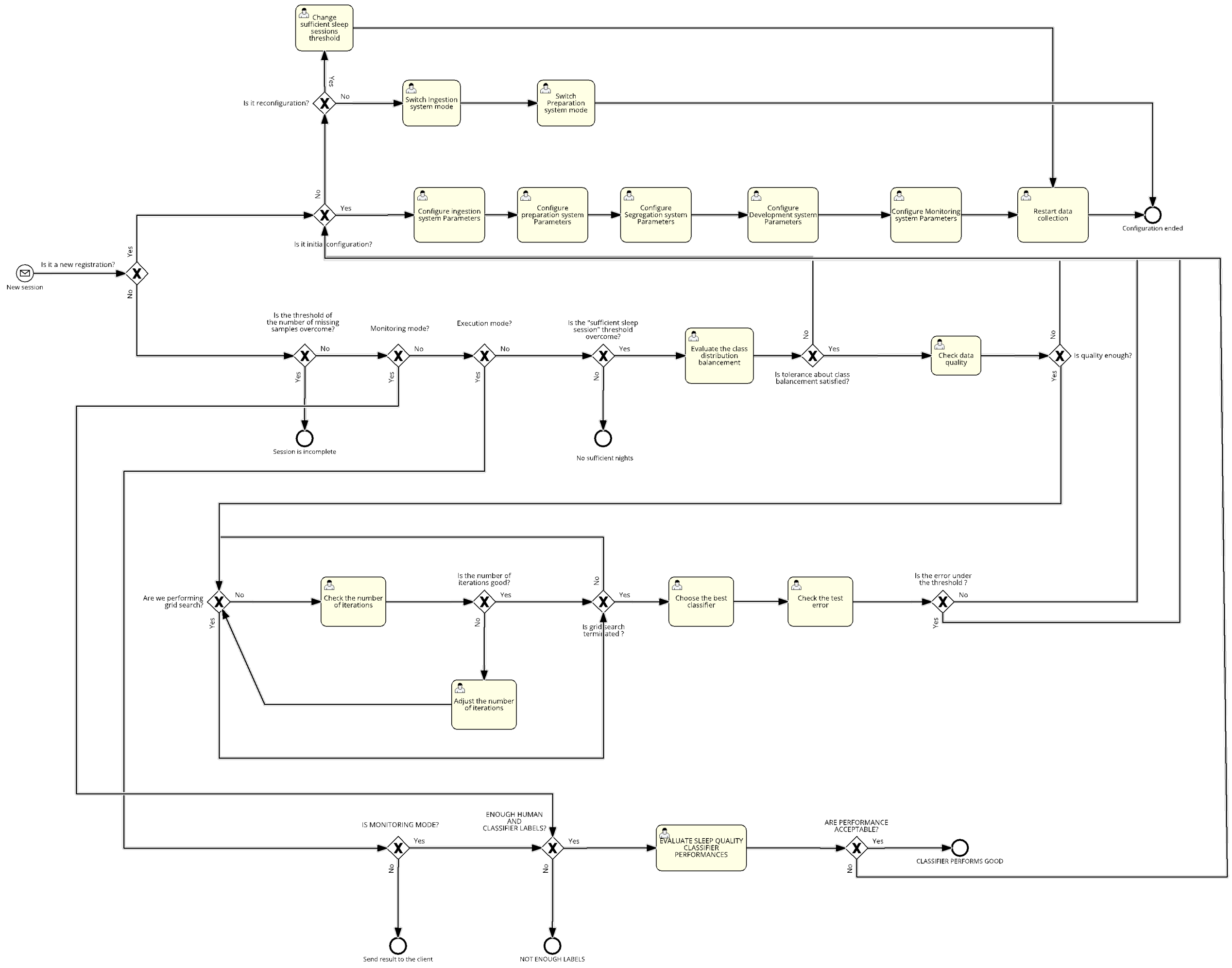


Figure 25: AS-IS simulation model

5.1 Assumptions

1. We consider as token for the simulation the arrival of the new sleep session
2. We considered to develop 10 classifiers
3. 100 sessions before generating the datasets for the development of each classifier
4. 700 sessions are for the execution for each classifier
5. 100 sessions are for monitoring for each classifier
6. 10 sessions discarded due to missing samples for classifier
7. 3 classifiers need class rebalance
8. 1 classifier has not enough data quality
9. We have 2 hyperparameters for the model (number of neurons and number of hidden layers) and we choose 2 values for every parameter so we perform grid search 4 times.
10. For 4 classifier the number of iterations is not good.
11. In 1 classifier the test error is over the threshold
12. 1 session over 100 monitor sessions is enough for producing the classifier monitoring report
13. In 1 classifier the performance is not acceptable.
14. 10 initial configurations.
15. 5 classifiers sends a reconfiguration message.
16. 11 configuration messages are for switching the operating mode.

5.2 Percentage of gateways

In this table are described the percentages of gateways derived from the assumptions:

Exclusive gateway	True %	False %	Computation
Is it a new registration?	0,11	99,89	1 (for the registration) / 900(develop + exec + monitoring) -> 0,11%
Is it initial configuration?	38,5	61,5	10 (initial configs) / 26 (15 (initial + reconfiguration) + 11 switch mode) -> 38,5%
Is it reconfiguration?	31,25	68,75	5 (reconfig msgs) / 16 (5 recon + 11 switch mode)-> 31,25 %
Is the threshold of the number of missing samples overcome?	1,1	98,9	10 (session discard) / 900 -> 1,1%
Monitoring mode?	11,1	88,9	100 / 900 -> 11,1 %
Execution mode?	87,5	12,5	700 (exec) / 800 (develop + exec) -> 87,5 %
Is the "sufficient sleep session threshold overcome"?	1	99	1 / 100 (develop) -> 1 %
Is tolerance about class balancement satisfied?	70	30	7 / 10 -> 70%
Is quality enough?	90	10	9 / 10 -> 90%
Are we performing grid search?	80	20	4 (iter. grid search) / 5 -> 80%
Is the number of Iterations good?	60	40	6/10 -> 60%
Is grid search terminated?	20	80	1/5 -> 20%
Is the error under the threshold?	90	10	9 / 10 -> 90%
Is monitoring mode?	12,5	87,5	100 (monitoring) / 800 (exec + monitoring) -> 12,5 %
Enough human and classifier labels?	1	99	1 / 100 (monitoring) -> 1%
Are performance acceptable?	90	10	9/10 -> 90%

5.3 Simulation Results

In the figures below are shown the result obtained from the simulation of the As-is model in Bimp.

	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	0 seconds	2.8 minutes	0.2 seconds
Process instance cycle times excluding off-timetable hours	0 seconds	2.8 minutes	0.2 seconds
Process instance costs	0 EUR	0 EUR	0 EUR

Activity Durations, Costs, Waiting times, Deviations from Thresholds

Name		Waiting time			Duration			Duration over threshold			Cost			Cost over threshold		
	Count	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Check data quality	9	0 s	0 s	0 s	8.3 s	9 s	9.5 s	0 s	0 s	0 s	0	0	0	0	0	0
Adjust the number of iterations	3	0 s	0 s	0 s	7.5 s	7.6 s	7.9 s	0 s	0 s	0 s	0	0	0	0	0	0
Change sufficient sleep sessions threshold	5	0 s	0 s	0 s	4 s	4.4 s	4.7 s	0 s	0 s	0 s	0	0	0	0	0	0
Check the number of iterations	6	0 s	0 s	0 s	8 s	8.7 s	9.3 s	0 s	0 s	0 s	0	0	0	0	0	0
Check the test error	9	0 s	0 s	0 s	8.4 s	9 s	10.1 s	0 s	0 s	0 s	0	0	0	0	0	0
Choose the best classifier	9	0 s	0 s	0 s	22.9 s	24.9 s	26.3 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure Development system Parameters	12	0 s	0 s	0 s	31.6 s	34.6 s	37.9 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure Monitoring system Parameters	12	0 s	0 s	0 s	5.5 s	6 s	6.5 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure Segregation system Parameters	12	0 s	0 s	0 s	13.9 s	15.3 s	16.3 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure ingestion system Parameters	12	0 s	0 s	0 s	25.4 s	27.9 s	29.5 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure preparation system Parameters	12	0 s	0 s	0 s	11.2 s	12.5 s	13.4 s	0 s	0 s	0 s	0	0	0	0	0	0
EVALUATE SLEEP QUALITY CLASSIFIER PERFORMANCES	27	0 s	0 s	0 s	4.5 s	5 s	5.5 s	0 s	0 s	0 s	0	0	0	0	0	0
Evaluate the class distribution balancement	9	0 s	0 s	0 s	7.5 s	8 s	8.5 s	0 s	0 s	0 s	0	0	0	0	0	0
Restart data collection	17	0 s	0 s	0 s	3.6 s	4 s	4.3 s	0 s	0 s	0 s	0	0	0	0	0	0
Switch Preparation system mode	11	0 s	0 s	0 s	5.1 s	5.6 s	6.1 s	0 s	0 s	0 s	0	0	0	0	0	0
Switch Ingestion system mode	11	0 s	0 s	0 s	6.3 s	6.7 s	7 s	0 s	0 s	0 s	0	0	0	0	0	0

© 2020 QBP Simulator

This website uses cookies to enhance your site experience and for analytical purposes. By proceeding on this website you are OK with this. Close

Figure 26: AS-IS simulation results

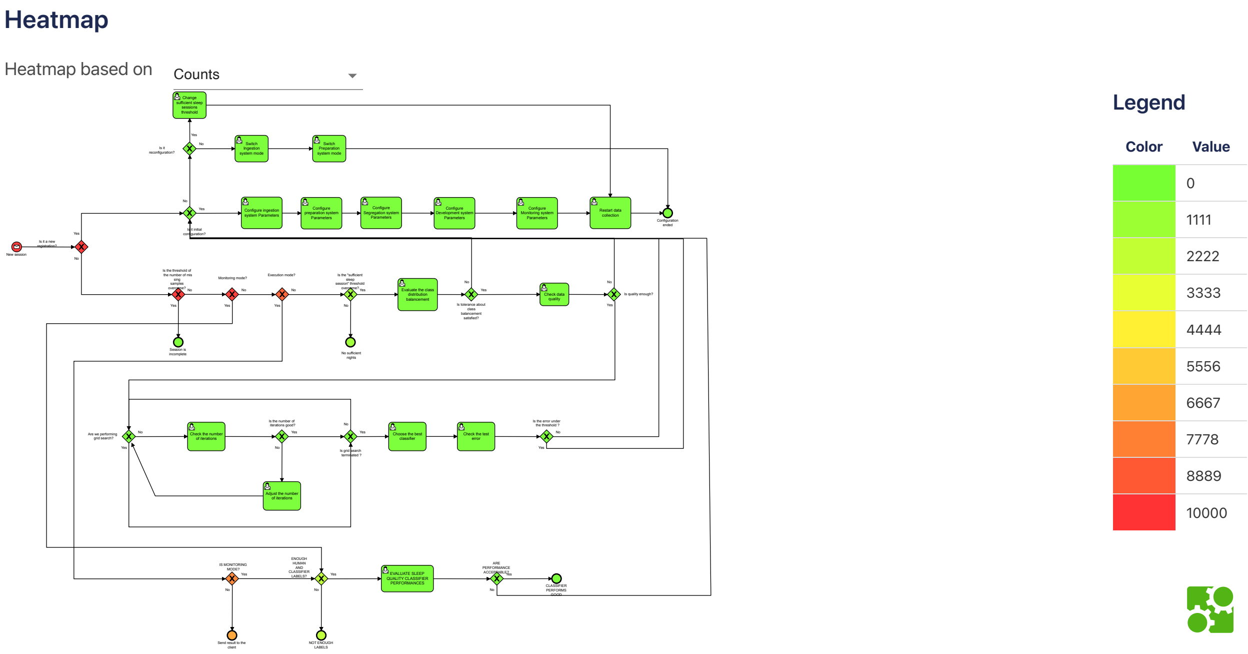


Figure 27: AS-IS simulation counts heatmap

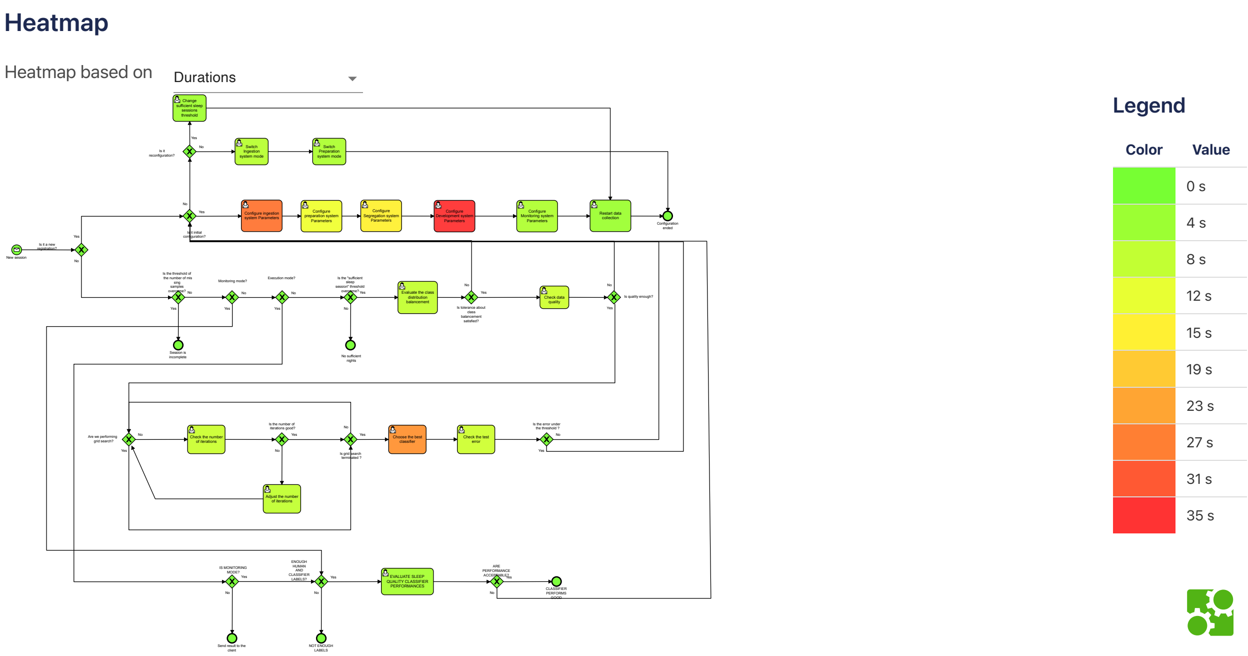


Figure 28: AS-IS simulation durations heatmap

We can observe from the counts heatmap that most sessions tokens arrive at the following 5 gateways:

- “Is it a new registration?”
- “Is the threshold of the number of missing samples overcome?”
- “Monitoring mode?”
- “Execution mode?”
- “Is monitoring mode”

This happens because most of the tokens are related to execution mode and just few of them will go on to development. The rest of the tasks related to the development of the classifier have just a low number of tokens.

About the duration heatmap, many of the most expensive tasks are related to configuration in general. During development the most expensive task is “choose the best classifier”, this is because we need to apply some rules to reason about what is the best classifier.

6. To be Model

In this section we will describe the different improvements that we have done to the AS-IS model and the different results that we obtained with respect to the simulation.

1) First improvement: Balance classes gathering data from similar customers

In case of unbalanced classes during the generate sleep session set phase, we can gather data from similar customers assuming we have a clustering of customers based on their features. This improvement avoids us the reconfigure the system to gather new data. The new Bpmn model of segregation system is shown below.

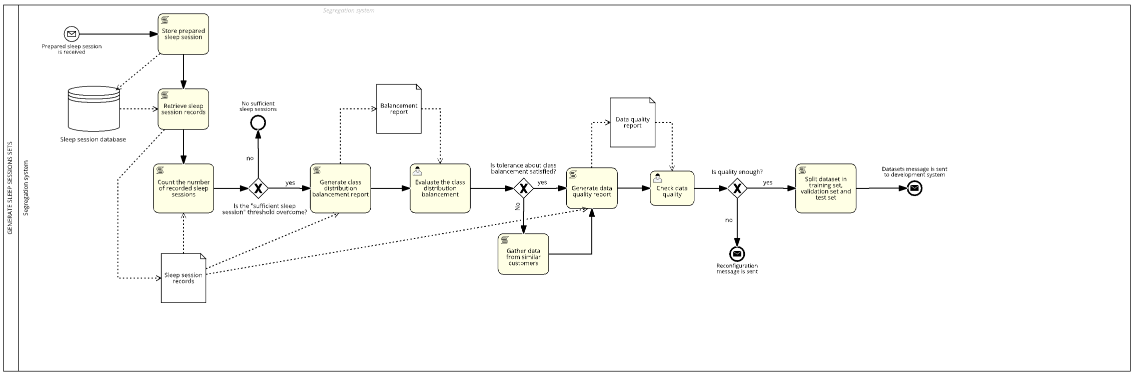


Figure 29: Generate sleep session set TO-BE

2) Second improvement: Select hyperparameters from similar customers
We decided to take the hyperparameters for the training of the classifier from already trained classifiers of other similar customers assuming we have a clustering of customers based on their features. This operation avoids performing grid search to select the best hyperparameters and to select the best classification model. If we have not a cluster of similar customers and so relative hyperparameters, we perform the same operations of the As is model, performing grid search. The new BPMN diagram of the development of the classifier is show in the figure below.

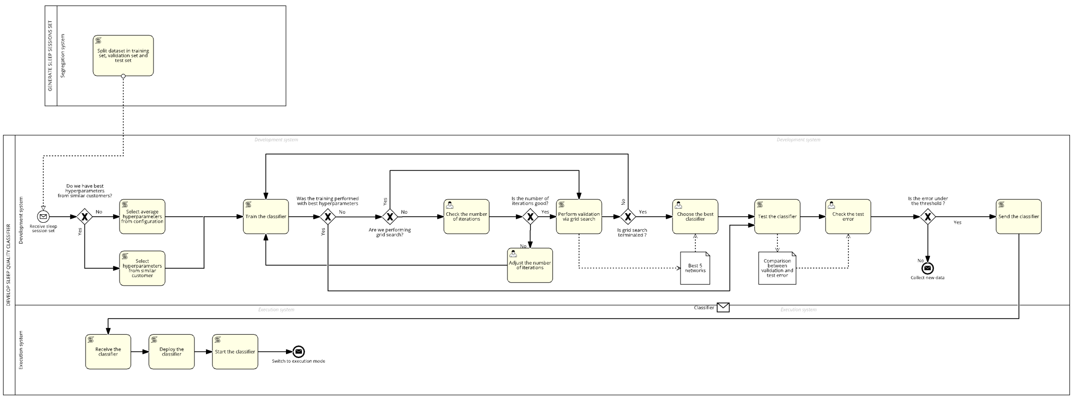


Figure 30: Develop sleep quality classifier TO-BE

3) Third improvement: Select parameters configuration from similar customers

We decided to take the configuration parameters from similar customers assuming we have a clustering of customers based on their features. In this way we can reduce the cognitive effort of some configuration task reducing the total cost of the task. The new Bpmn diagram of the configuration phase is shown below.

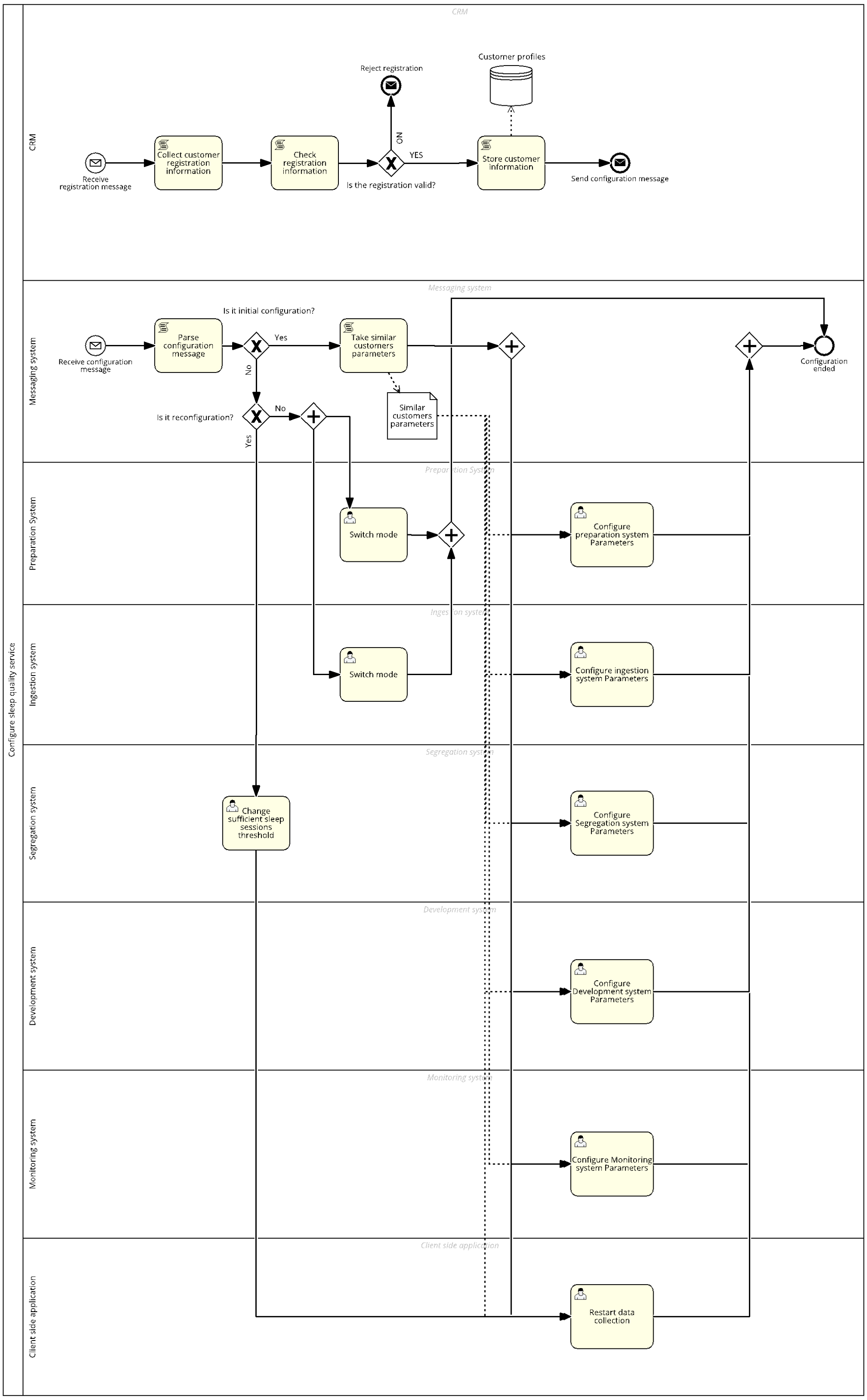


Figure 31: Configure sleep quality service TO-BE

6.1 To be task level diagrams

In this section we describe how the task level has been modified after the improvements decided for the TO-BE model.

Evaluate the class distribution balancement

Description of the task: The data analyst has to check whether the classes for sleep quality are balanced or not in the training data. The three classes are low sleep quality, medium sleep quality, and high sleep quality.

During the configuration, the threshold about data balancement is set (we assume this threshold is the maximum difference from the average number of samples, expressed in percentage on the total number of samples).

The system therefore is initialized with this threshold and can signal whether a class exceeds the tolerance or not, but the final decision is made by the human, that asks the system to balance classes gathering the needed sleep sessions from other similar customers.

Role	Subtask		%	Cognitive Effort	Cost
Data analyst	1. Clicks on balancement reports button			Remember (1)	1 * 1 * 1= 1
System	2. Shows balancement reports interface				
Data analyst	3. Selects the specific balancement report			Remember (1)	1 * 1 * 1= 1
System	4. Shows the specific balancement report				
Data Analyst	5. Investigates the balancement report			Understand (2)	1 * 2 * 1= 2
	6. IF classes are unbalanced		30%		
Data Analyst		7. Clicks on rebalance button		Remember (1)	1 * 1 * 0.3 = 0.3
System		8. The system finds data from other similar customers and rebalance the classes			
Data Analyst	9. Closes balancement report			Remember (1)	1 * 1 * 1 = 1
Data Analyst	10. Closes balancement reports interface			Remember (1)	1 * 1 * 1 = 1
					Total cost: **6.3**

Description of cognitive effort:

- 1.Remember(1): interact with user interface
- 3.Remember(1): interact with user interface
- 5.Understand(2): If the color of at least one bar in the histogram is orange, the system is signaling that reconfiguration is needed

- 7.Remember(1): interact with user interface
- 9.Remember(1): interact with user interface
- 10.Remember(1): interact with user interface

Configure segregation system parameters

Description of the task: During the initial configuration, the segregation system needs to be configured with some parameter. These are:

- The IP addresses of the development system and the messaging system.
- The “sufficient sleep sessions” threshold, representing how many samples are to be taken in account before generating the sets.
- The threshold regarding the tolerance about class balancement, which is expressed in percentage of the total number of samples and representents the maximum difference from the average number of samples to consider a class balanced with respect to the others.
- The percentage of the data that will form the training set, the one that will form the validation set and the one that will form the test set.

The system provides the data analyst with configuration parameters from previous similar customers, so these values are all already present and the human has only to copy them to the configuration interface.

Role	Subtask		%	Cognitive Effort	Cost
Data analyst	1. Clicks on configure segregation system			Remember (1)	1 * 1 * 1 = 1
System	2. Shows the segregation system configuration interface				
Data analyst	3. Selects the textbox representing the IP address of the messaging system			Remember (1)	1 * 1 * 1 = 1
Data analyst	4. Enters the IP address of the messaging system			Remember (1)	1 * 1 * 1 = 1
Data Analyst	5. Selects the textbox representing the IP address of the development system			Remember (1)	1 * 1 * 1 = 1
Data Analyst	6. Enters the IP address of the development system			Remember (1)	1 * 1 * 1 = 1
Data Analyst	7. Selects the textbox about the “sufficient sleep sessions” threshold value			Remember (1)	1 * 1 * 1 = 1
Data Analyst	8. Enters the “sufficient sleep sessions” threshold value			Remember (1)	1 * 1 * 1 = 1

Data Analyst	9. Selects the textbox about the class balancement tolerance			Remember (1)	1 * 1 * 1 = 1
Data Analyst	10. Enters the class balancement tolerance value			Remember (1)	1 * 1 * 1 = 1
Data Analyst	11. Selects the panel about split percentage of each set			Remember (1)	1 * 1 * 1 = 1
Data Analyst	12. Enters the percentages of the data for forming each sleep session set			Remember (1)	1 * 1 * 1 = 1
Data Analyst	13. Closes segregation system configuration interface			Remember (1)	1 * 1 * 1 = 1
	14. END				Total cost: **12.0**

Description of cognitive effort:

- 1.Remember(1): interact with user interface
- 3.Remember(1): interact with user interface
- 4.Remember(1): the IP address of the messaging system is always the same
- 5.Remember(1): interact with user interface
- 5.Remember(1): the IP address of the development system is always the same
- 6.Remember(1): interact with user interface
- 7.Remember(1): interact with user interface
- 8.Remember(1): This value is provided by the system on the basis of previous similar customer configuration parameters
- 9.Remember(1): interact with user interface
- 10.Remember(1): The data analyst sets the threshold about class balancement always at the same level.
- 11.Remember(1): interact with user interface
- 12.Remember(1): The data analyst decides which percentages of the data to use for the training set, the test set and the validation set. These percentages are always the same.
- 13.Remember(1): interact with user interface

Configure the Development system parameters

Task	%	Role	Cognitive effort	Total cost
1. Open configuration interface		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
2. Shows configuration interface		System		

3. Click on the textbox to set the hidden layer interval		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
4. Set hidden layer interval		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
5. Click on the textbox to set the interval of the number of neurons		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
6. Set number of neurons interval		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
7. Click on the textbox to set the test error threshold		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
8. Set test error threshold		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
9. Click on the textbox to set the number of initial iterations		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
10. Set the initial number of iterations		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
11.Click on the textbox to set the ip address of the execution system		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
12. Set the ip address of the execution system		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
13. Save configuration		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
14. Close configuration interface		M.L. engineer	Remember (1)	1 * 1.84 = 1.84
				Total Cost: 23.92

Cognitive effort description:

1. Remember (1): Interact with the user interface

3. Remember (1): Interact with the user interface

4. Analyze (4): The machine learning engineer must assess the correct interval of the hidden layers based on his knowledge and experience and based on multiple criteria of the user to evaluate the complexity needed for the classifier.

5. Remember (1): Interact with the user interface

6. Analyze (4): The machine learning engineer must assess the correct interval of the number of neurons based on his knowledge and experience and based on multiple criteria of the user to evaluate the complexity needed for the classifier.

7. Remember (1): Interact with the user interface

8. Remember (1): Se a given parameter

9. Remember (1): insert the given threshold

10. Remember (1): Set a given parameter

11. Remember (1): Interact with the user interface

12. Remember (1): Set a given parameter

13. Remember (1): Interact with the user interface

14. Remember (1): Interact with the user interface

6.2 To be Simulation model

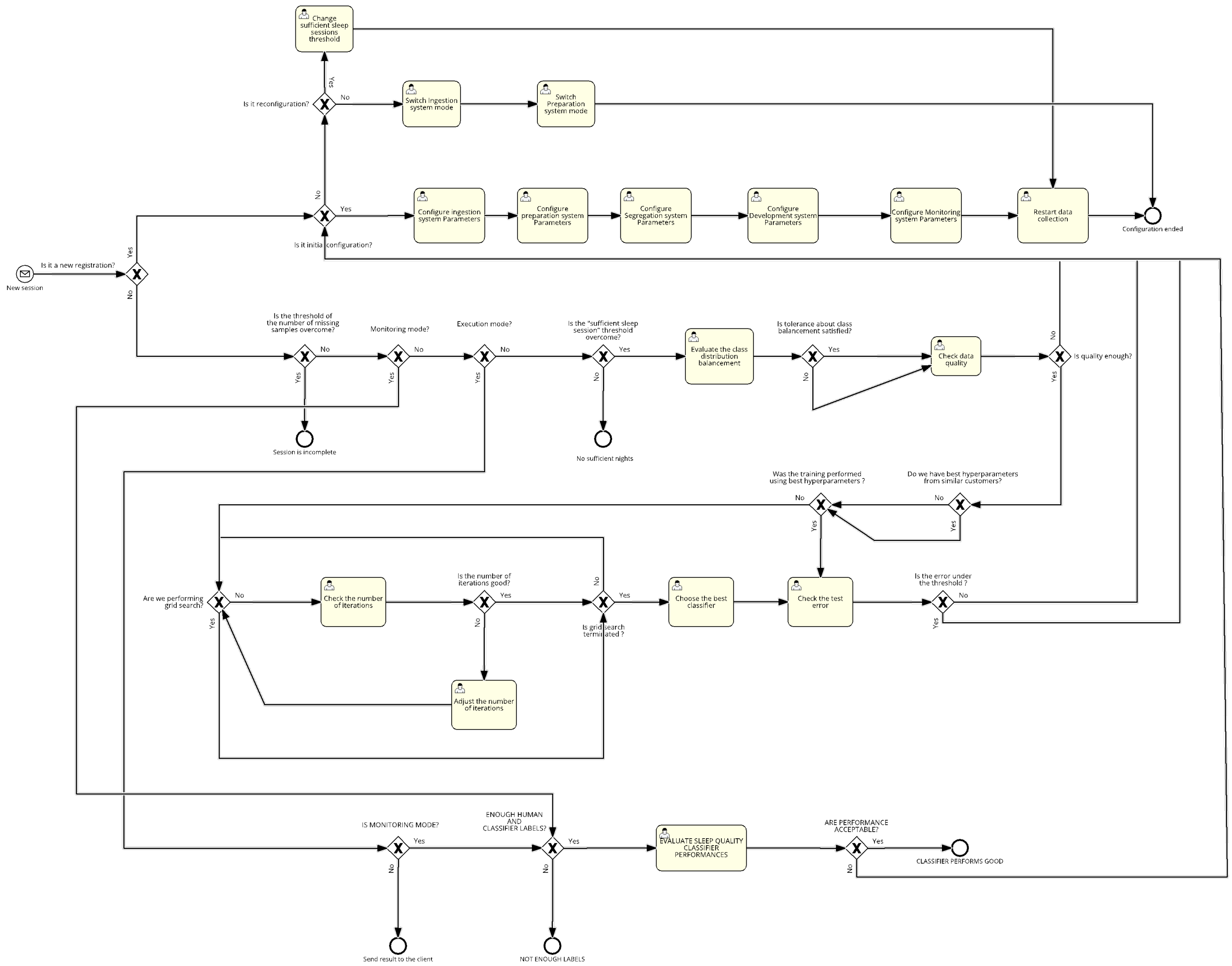


Figure 32: TO-BE simulation model

6.2.1 Assumptions

1. We consider as token for the simulation the arrival of the new sleep session
2. In monitoring system we assume that – We are not convinced.
3. 10 classifiers
4. 100 sessions before generating the datasets for each classifier
5. 700 sessions in execution for each classifier
6. 100 sessions are for monitoring for each classifier
7. 10 sessions discarded due to missing samples for classifier
8. 3 classifiers need class rebalance (3 sessions)
9. 1 classifier quality not enough
10. We have 2 hyperparameters for the model (number of neurons and number of hidden layers) and we choose 2 values for every parameter we perform grid search 4 times.
11. For 4 classifier the number of iterations is not good.
12. 4 times over 5 the grid search is not terminated.
13. In 1 classifier the test error is over the threshold
14. 1 session over 100 monitor sessions is enough for producing the classifier monitoring report
15. In 1 classifier the performance is not acceptable.
16. 10 initial configurations.
17. 5 classifier sends a reconfiguration message.

18. 11 configuration messages are for switching the operating mode.
19. 6 classifier uses best hyperparameters taken from similar customers

6.2.2 Percentage of tasks and gateways

Exclusive gateway	True %	False %	Computation
Is it a new registration?	0,11	99,89	1 (for the registration) / 900(develop + exec + monitoring) -> 0,11%
Is it initial configuration?	43,5	56,5	10 (initial configs) / 23 (12 (initial + reconfiguration) + 11 switch mode) -> 43,5%
Is it reconfiguration?	38,5	61,5	5 (reconfig msgs) / 13 (2 recon + 11 switch mode)-> 38,5 %
Is the threshold of the number of missing samples overcome?	1,1	98,9	10 (session discard) / 900 -> 1,1%
Monitoring mode?	11,1	88,9	100 / 900 -> 11,1 %
Execution mode?	87,5	12,5	700 (exec) / 800 (develop + exec) -> 87,5 %
Is the “sufficient sleep session threshold overcome”?	1	99	1 / 100 (develop) -> 1 %
Is tolerance about class balancement satisfied?	70	30	7 / 10 -> 70%
Is quality enough?	90	10	9 / 10 -> 90%
Are we performing grid search?	80	20	4 (iter. Grid search) / 5 -> 80%
Is the number of Iterations good?	60	40	6/10 -> 60%
Is grid search terminated?	20	80	1/5 -> 20%
Is the error under the threshold?	90	10	9 / 10 -> 90%
Is monitoring mode?	12,5	87,5	100 (monitoring) / 800 (exec + monitoring) -> 12,5 %
Enough human and classifier labels?	1	99	1/ 100 (monitoring) -> 1%
Are performance acceptable?	90	10	9/10 -> 90%
Do we have best hyperparameters from similar customers?	60	40	6 (use best hyperparams) / 10 -> 60%
Was the training performed using best hyperparameters?	60	40	5 (use best hyperparams) / 10 -> 60%

6.3 Simulation results

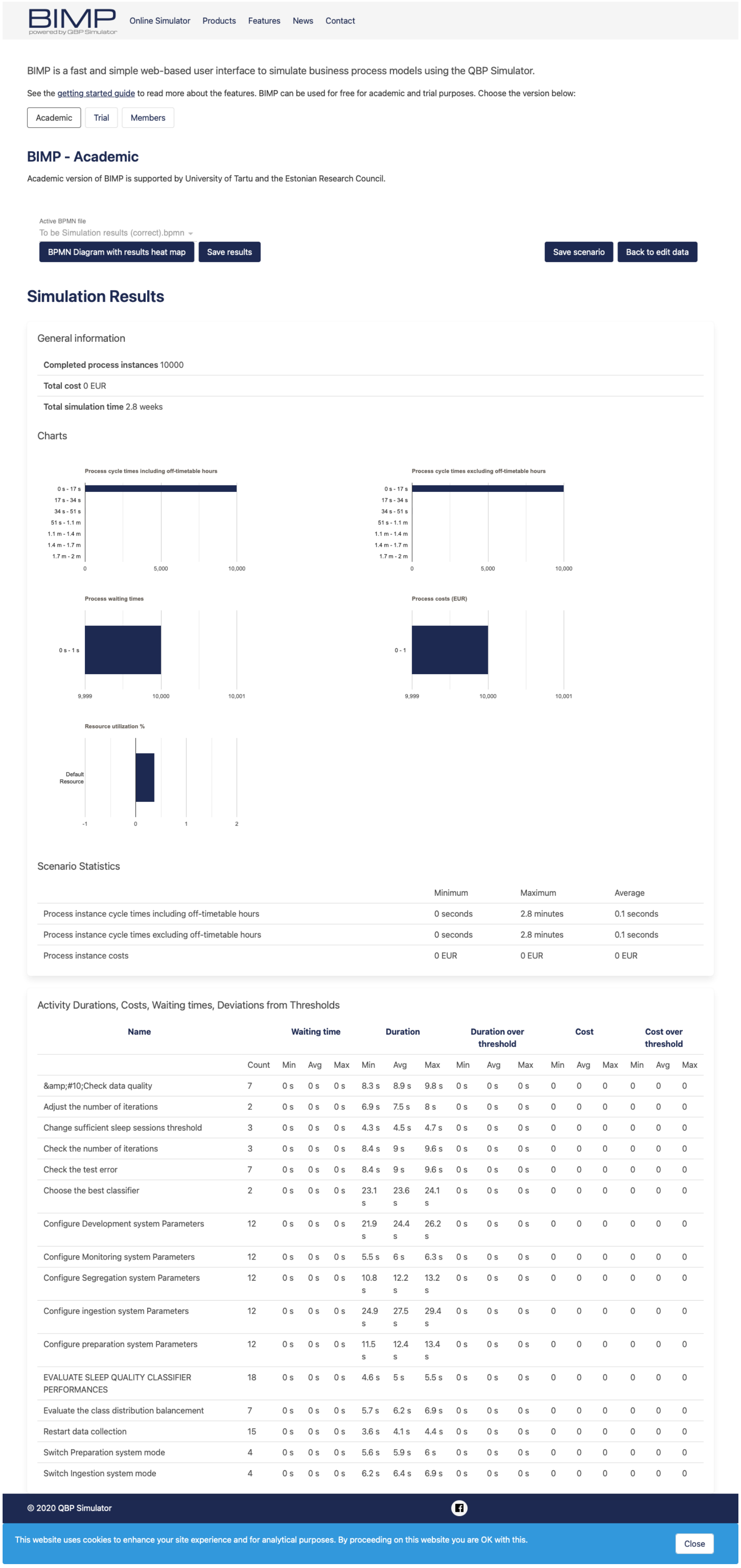
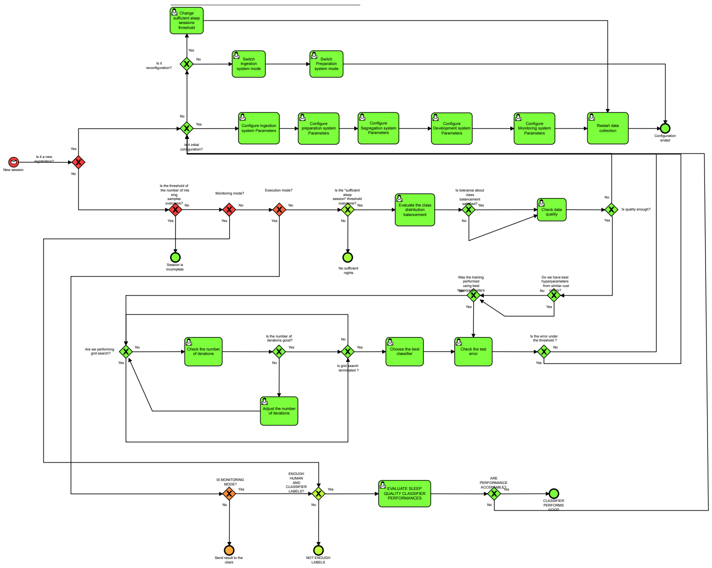


Figure 33: TO-BE simulation results

Heatmap

Heatmap based on Counts



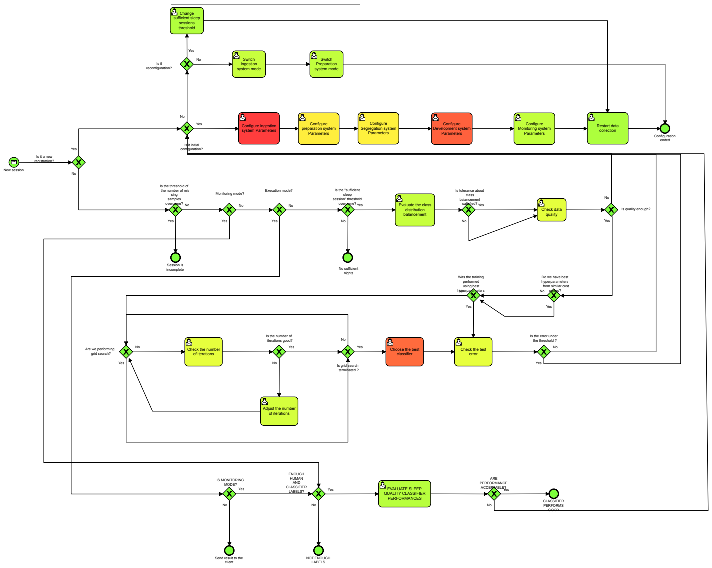
Legend

Color	Value
	0
	1111
	2222
	3333
	4444
	5556
	6667
	7778
	8889
	10000

Figure 34: TO-BE simulation counts heatmap

Heatmap

Heatmap based on Durations



Legend

Color	Value
	0 s
	3 s
	6 s
	9 s
	12 s
	15 s
	18 s
	21 s
	24 s
	27 s

Figure 35: TO-BE simulation durations heatmap

We can see that from what is reported on the BIMP simulation results for the TO BE model the average process instance cycle times has decreased from 0.2 seconds to 0.1 seconds, so it means that we were able to improve the workflow with the modifications proposed above.

7. Process Mining

7.1 Simulation parameters used

We simulated the as is model with BIMP assigning a default 1 eur cost and 1 sec duration to each task, 50% probability to each gateway and 1000 input tokens. Then we exported the generated MXML log file, and we imported it into Disco generating a transition map. Then we exported this file in the csv format and we loaded it into ProM. Furthermore, we converted this csv file using the specific plugin "Convert CSV to XES" and we used this Xes file to generate the transition map in Apromore and the BPMN model in Apromore and ProM.

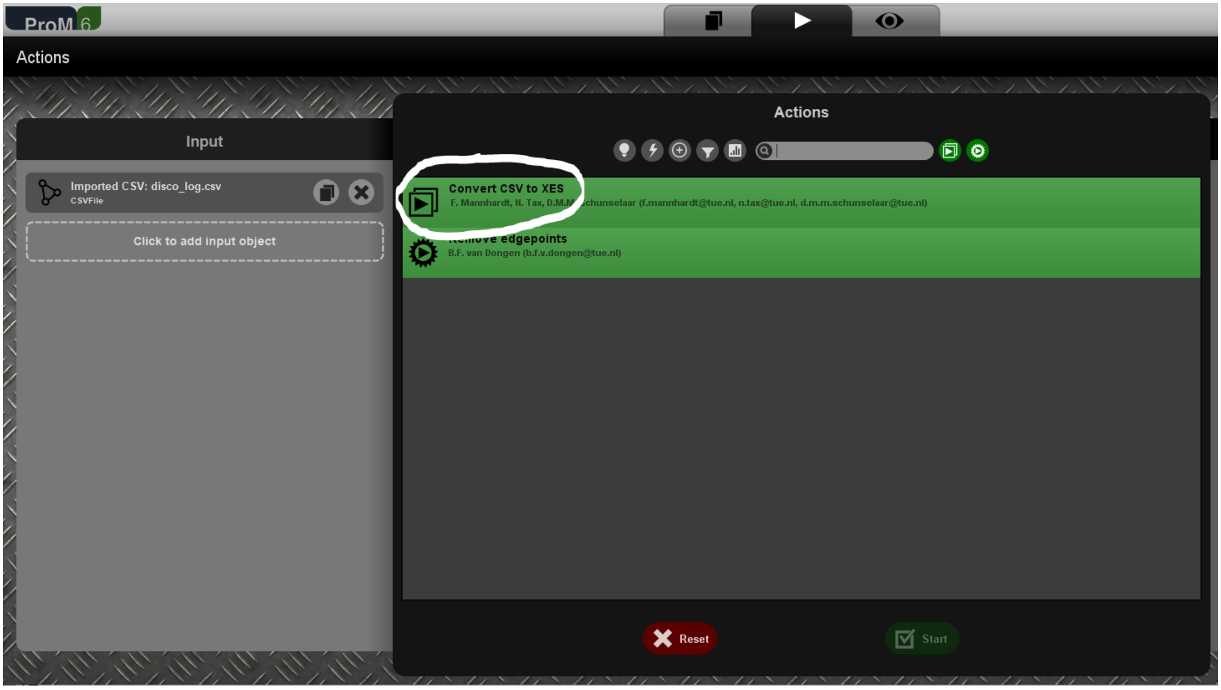


Figure 36: CSV to XES convertion

7.2 Transition map Disco

This is the transition map we obtained using Disco. It reflects the workflow of our company but not the workloads because for this simulation we used a value of 50% for every exclusive gateway and this modifies a lot the normal behavior of the company. The end task at the end of the transition map was added by Disco in an automatic way. Because we used the csv exported from disco to generate the XES file that we used in ProM and Apromore, we find this end task also in the following diagrams.

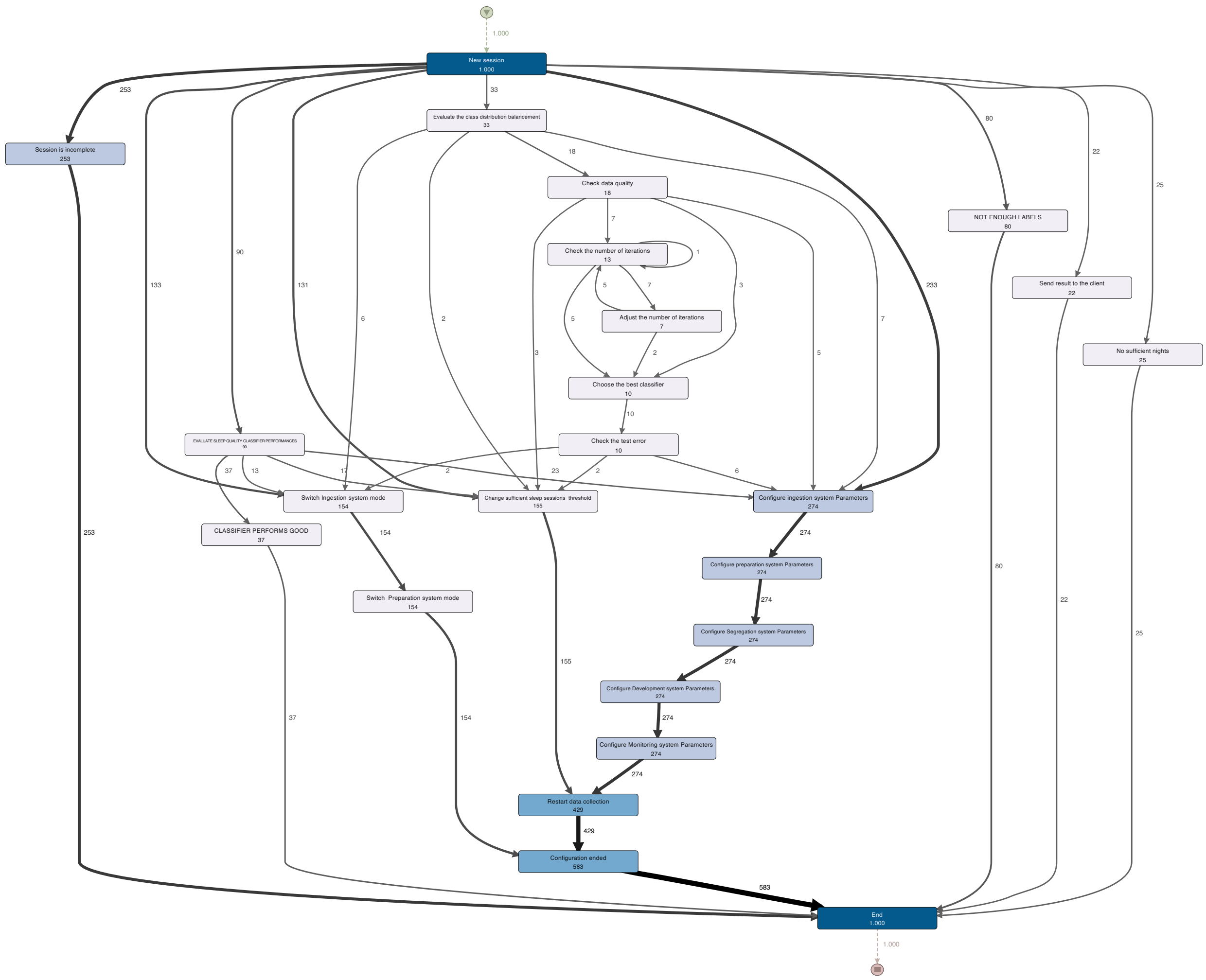


Figure 37: Transition map Disco original logs

7.3 Transition map Apromore

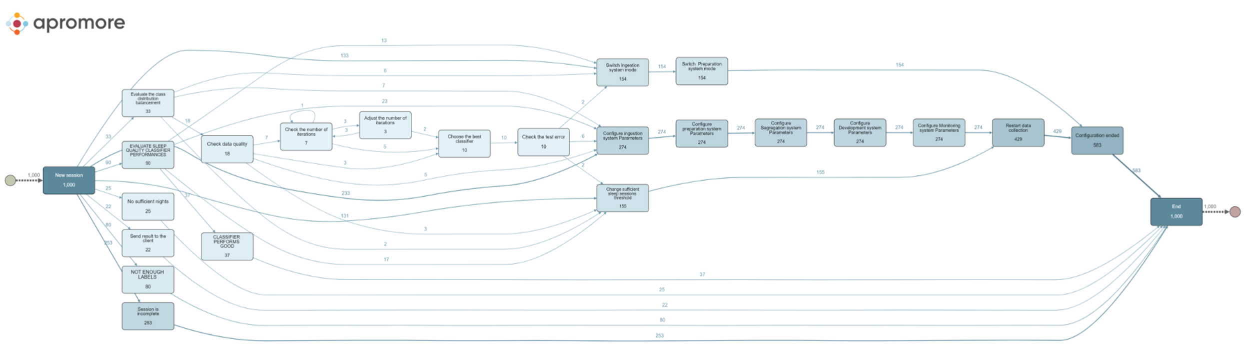


Figure 38: Transition map Apromore original logs

This is the transition map generated from the XES file in Apromore. As said before we used the csv file exported from Disco to generate the XES file so this file contained also the end task created automatically from Disco.

7.4 Comparison

The disparity between the Transition Maps derived from Disco and Apromore lies in the way task access is depicted. While Disco displays the frequency of task access, Apromore exhibits the number of tokens that access a task. Additionally, Apromore does not emphasize the edges that carries more tokens as Disco does.

By combining these two sources, a more comprehensive understanding of the simulation can be obtained. For instance, Apromore takes in consideration only the first time the tokens access the tasks "Check the number of iterations" and "Adjust the number of iterations" while Disco provides insight into how many times these tokens circulate between these tasks.

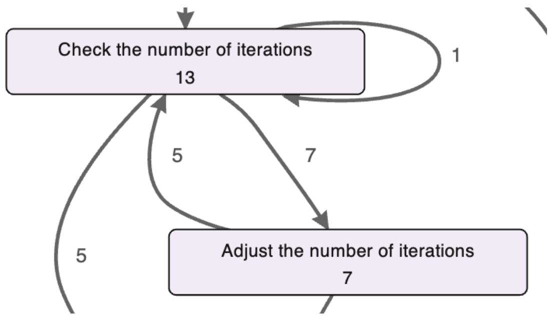


Figure 39: Disco cycle

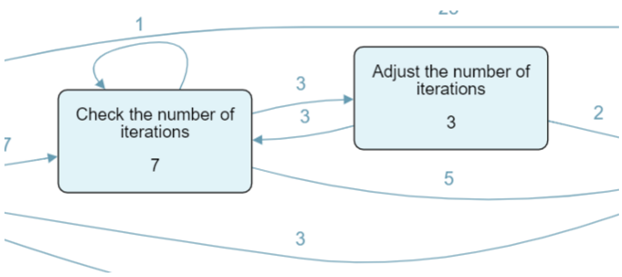


Figure 40: Apromore cycle

7.5 BPMN Mining from Original Logs

In this section we will describe how we used the tools ProM and Apromore to mine the BPMN model from the original logs.

7.5.1 ProM

The first step is to load the XES file.

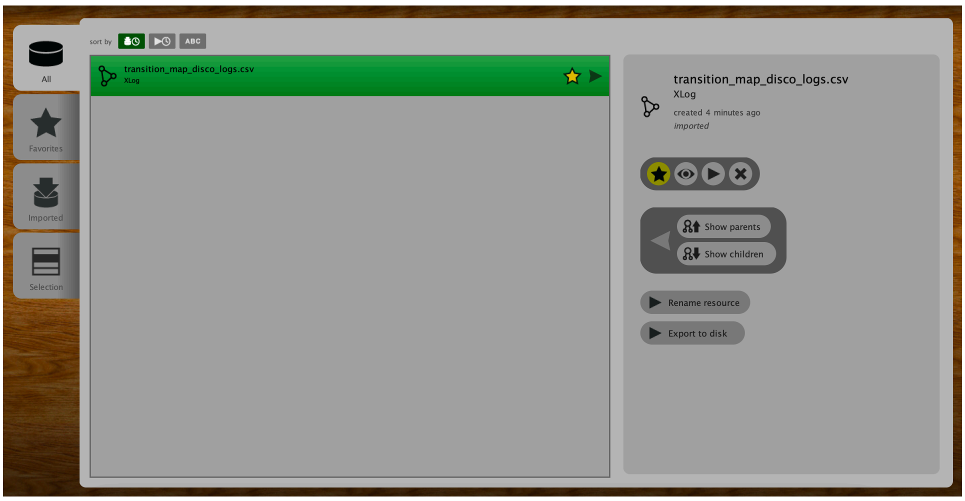


Figure 41: Load XES file in ProM

Then we must select the correct plugin to do the process mining task in ProM which is the "BPMN miner" plugin. As process miner algorithm we selected the inductive miner which uses a heuristics-based approach to analyze the data contained in the event log and extract the underlying process model.

Figure 42: Select BPMN Miner Plugin on ProM

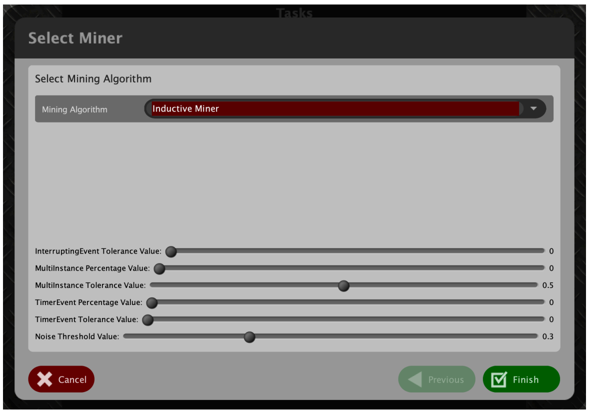
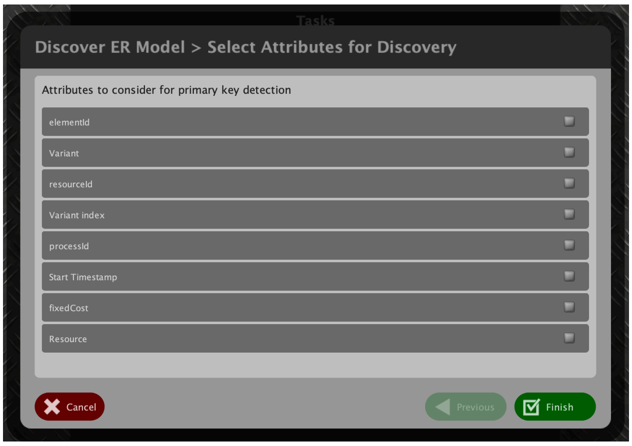


Figure 43: Select the Miner Algorithm on ProM

As settings for the algorithm:

- All the attributes have been selected to be considered for primary key detection
- Set the noise to 0 to guarantee maximum fitness



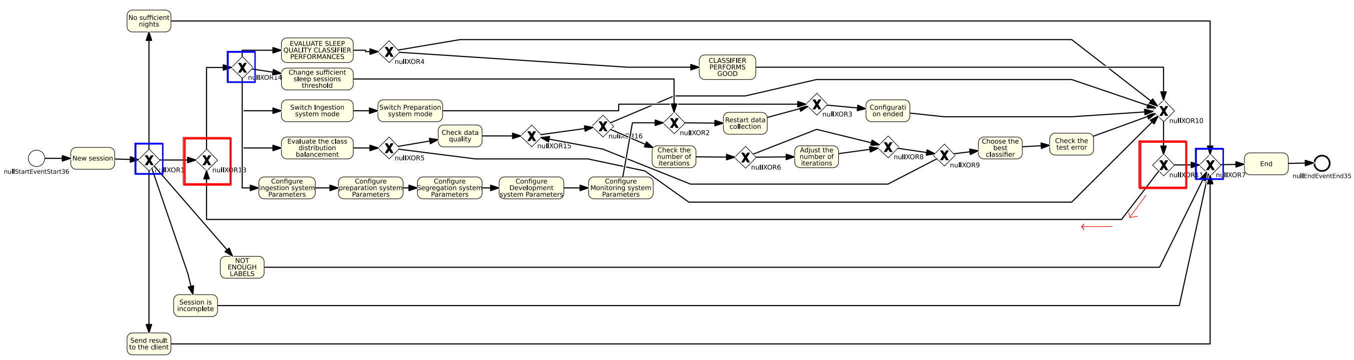


Figure 44: Mined BPMN from original logs with ProM

Activities 24, Gateways 15, Sequence Flows 49

In the figure 44 is shown the BPMN model obtain from ProM exported in BPMN format and loaded in Signavio. One of the main differences with the original simulation diagram is that here we can have multiple outflows from one single exclusive gateway instead in the original diagram from every exclusive gateway you could have only two paths.

7.5.2 Apromore

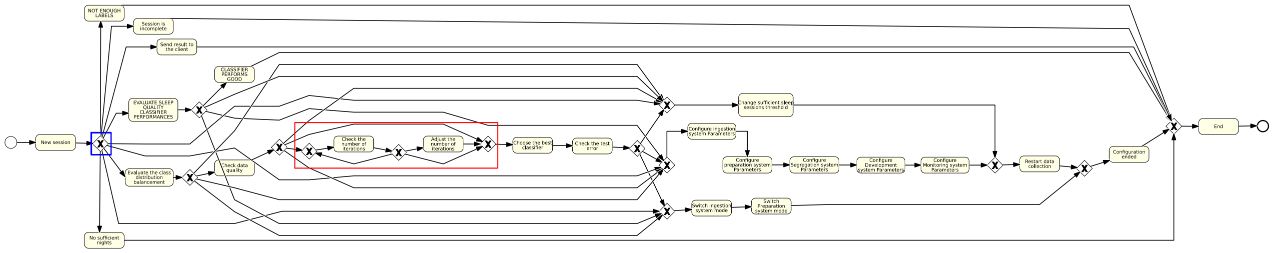


Figure 45: Mined BPMN from original logs with Apromore

Activities 24, Gateways 14, Sequence Flows 58

In figure 45 is shown the BPMN diagram we obtained from Apromore exporting the file in the BPMN format and loading it into Signavio.

One difference between the two models is that in the ProM model you can go back to the task “Check the number of iterations” after performing the task “Adjust the number of iterations” as it is in the original As-is model. In this model instead it is not possible and after adjusting the number of iterations you must go on choosing the best classifier.

Another difference is that in the ProM model you can go back from the XOR 11 gateway to the XOR 1 gateway. This is not allowed in the original BPMN. Instead in the Apromore model this sequence flow is not present.

Another difference we can observe between the two diagrams is how the gateways at the beginning are organized. In the one from ProM there is one gateway more that is adding a visual complexity to the diagram.

7.5.3. Conformance checking

In this section we will discuss about the metrics utilized to check the original logs against the mined model produced with proM and Apromore.

The metrics used are the following:

1) **Fitness**: Measures how much the observed behavior in the log is captured by the process model. It is computed as:

$$\boldsymbol{fitness} = \frac{1}{2}\left(1 - \frac{m}{c}\right) + \frac{1}{2}\left(1 - \frac{r}{p}\right)$$

Where:

- m is the number of missing unproduced output tokens

- c is the number of input tokens correctly consumed
- r is the number of input tokens remaining unconsumed
- p is the number of output tokens correctly produced

2) **Simplicity**: Measures how much a model is simple. The simplest model obtains better results for this metric.

$$\boldsymbol{Simplicity} = \#gateways + \#sequence\ flows + \#activitites$$

Where:

- $\mathrm{gateways}$ is the number of gateways that are present in the mined BPMN model
- $\mathrm{sequence\ flow}$ is the number of edges that are present in the mined BPMN model
- $\mathrm{activities}$ at the denominator is the number of activities that are present in the mined BPMN model

3) **Precision**: Measures how much the mined model behaves different from the original event logs (doesn't allow for too much different behavior with respect to the log).

$$\boldsymbol{Precision}(L, M) = \frac{1}{|\varepsilon|}\sum_{e \epsilon \varepsilon} \frac{|en_L(e)|}{|en_M(e)|}$$

Where:

- ε is the collection of unique events in a context of the log
- en_M represents the enabled activities in the mined model
- en_L represents the observed activities executed in a similar context in the logs

4) **Generalization**: Measures how much the model overfits the original event logs.

In order to evaluate those metrics, we have used the ProM tool. The steps that we have done are the following:

1) Load the BPMN models (ProM and Apromore models) and the original log to ProM

2) Convert them to PetriNet using the ProM plugin called "Convert BPMN to Petrinet"

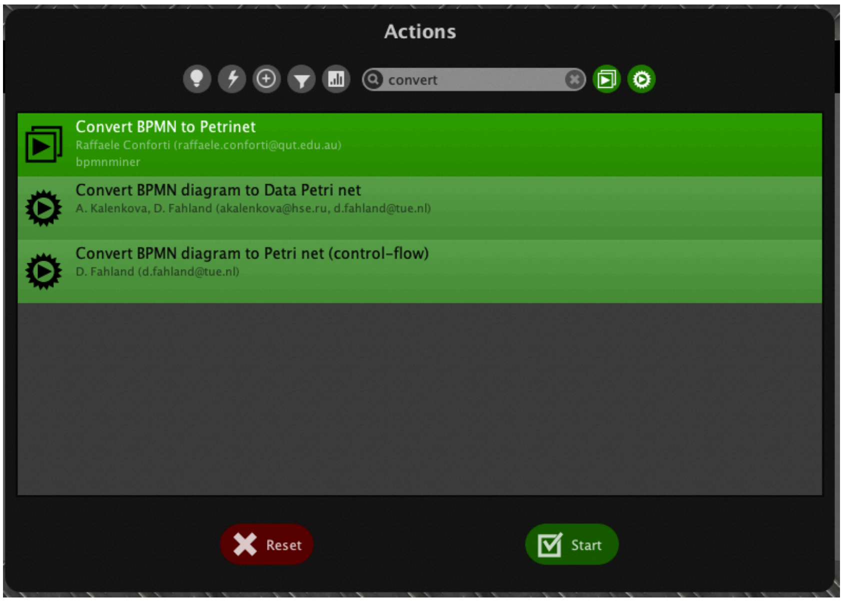


Figure 46: Convert BPMN to PetriNet plugin on ProM

3) Use the Petrinet file generated from the BPMN and the original logs with the plugin called “Replay a Log on Petri Net for Conformance Analysis” to generate the replay results (which contains the fitness metric). As a validation we computed the fitness metric also between the original MXML log and the original As-is BPMN model and we obtained a Fitness metric of 1. This is correct because all the cases in the original log can be replayed on the original BPMN diagram.

Figure 47: Replay a log on PetriNet for Conformance Analysis

4) To get the precision and generalization metrics, we use the Petrinet file, the original logs and the replay results with the plugin called “Measure Precision/Generalization”.

Figure 48: Compute Precision and Generalization

5) Compute the simplicity with the formula described above.

The results that we obtained by doing the steps described above are the following:

Mined Model	Fitness	Simplicity	Precision	Generalization
ProM	0.999	88	0.667	0.953
Apromore	0.999	96	0.891	0.938

ProM results **Apromore Results**

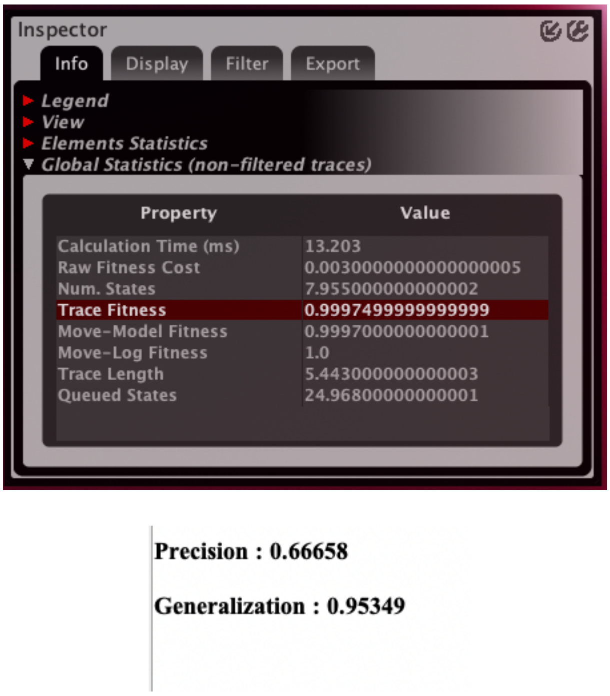


Figure 49: ProM results

Figure 50: Apromore Results

As we can see from the table, for both models we obtained a fitness value near to 1 that means that the log can be successfully replayed on the mined models. We didn't obtain a fitness value of 1 for the ProM model because in the log there are 2 cases where the "Check the number of iterations" is skipped and this is not allowed by the mined model, but it is possible in the As-is model with specific percentages.

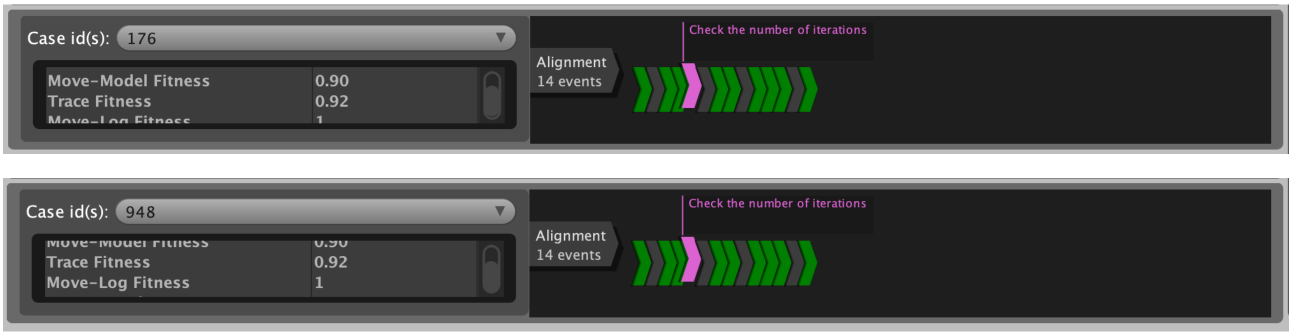


Figure 51: Log Alignment Violations ProM

Instead, we didn't obtain a fitness value of 1 for the Apromore model because in 3 cases we had a loop in the log that is possible in the As-is model but that is not allowed in the Mined model from Apromore. Indeed, in the Apromore model after adjusting the number of iterations you can't go back to check the number of iterations and you must go one choosing the best classifier. This operation instead is allowed in the original model and in the mined model in ProM.

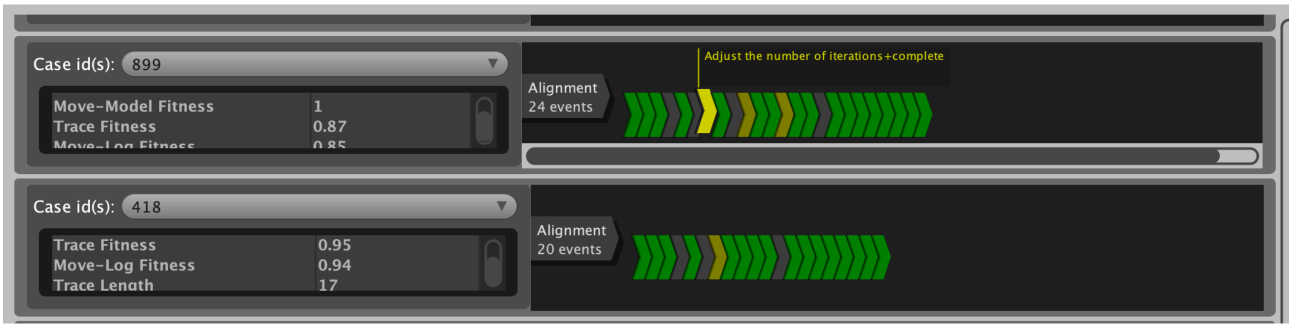


Figure 52: Log Alignment Violations Apromore

In terms of simplicity, we obtained a lower value for the ProM model. This is because the number of activities is the same in both models, there is one gateway less in the Apromore model but in the ProM model there are 7 sequence flow less, so this leads to a better simplicity of the model.

For what concern precision, we obtained a higher value for the Apromore mined model respect to the ProM mined model, that means that Apromore mined a model that do not allow behaviors very different from what we have seen in the original logs with respect to ProM. Indeed, in the ProM model, as discussed in the comparison of the two models, there is sequence flow from the XOR 11 gateway at the end of the model, to the XOR 1 gateway at the beginning of the model. This sequence flow allows multiple different behaviors with respect to the log. We didn't obtain a value of 1 for the Precision of the Apromore model because of what we discussed in the fitness metric, that is that this model doesn't allow a loop between the two tasks "Check the number of iterations" and "Adjust the number of iterations" meanwhile it is allowed in the As-is model and so it is present in the log. This means that the Apromore model doesn't overfit the log.

The Generalization metric is high for the two models, meaning that the model does not restrict to the behavior of the log. The Apromore mined model obtained a lower generalization and we could expect that because it had a higher precision with respect to the ProM model.

7.6 Modified logs experiments

In this section we will analyze three possible violations that can be done to our AS-IS model and see how the conformance checking metrics changes.

7.6.1 Violation list

The list of violations considered are the following:

- "Restart data collection": Under the assumption that we have similar customers that can have similar data, we can skip this activity taking the data directly from them instead of collecting new data from the customer.
    - Case ids deleted for this violation: 299, 180, 399.
- "Change sufficient sleep session threshold": With the same assumption described above, we can also remove this activity which is related to the "Restart data collection" one. We decided to remove the same case ids as they are related.
    - Case ids deleted for this violation: 299, 180, 399.
- "Choose the best classifier": Under the assumption that we take hyperparameters from similar customers, we don't perform grid search and we have only one classifier. So, we don't need to choose the best one.
    - Case ids deleted for this violation: 485, 7, 961

7.6.2 Fitness with modified logs and BPMN models mined on the original logs

We used the modified logs (with violations) with the original logs mined models to check the fitness.

ProM

Figure 53: ProM results modified logs and original logs mined models

Apromore

Figure 54: Apromore results modified logs and original logs mined models

ProM log alignments:

- Case 180, 299, 399: In this case we have a lower trace fitness because in the model we don't have the direct path from "new session" to "configuration ended", so it chooses the "not enough labels" path that doesn't contain the "configuration ended" activity. So, it skips the "not enough labels" activity because it's not present in the logs, but it doesn't do the configuration ended activity (that is present in the log) because it is not present in the model's path.

Figure 55: logs alignments of case ids 180, 299,399

- Case 485, 7: In this case we have a lower trace fitness because in the logs we don't perform the "choose the best classifier" activity that we must perform in the model. The only difference of those two cases is the time when we should have performed that task.

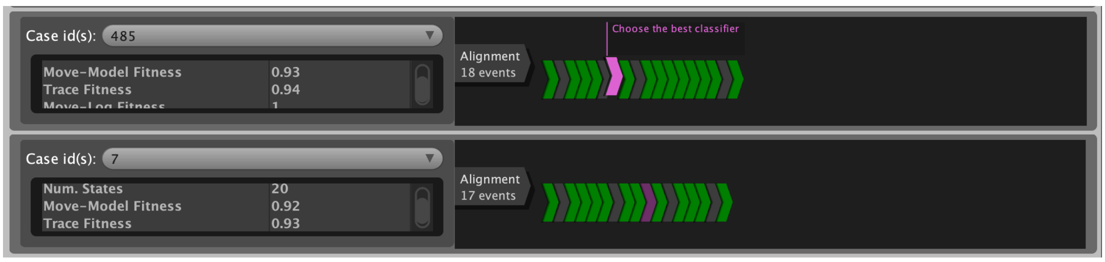


Figure 56: logs alignments of case ids 485, 7

- Case 961: In this case we had already a violation before modifying the logs but removing the activity “choose the best classifier” the model chose a different path then before, where the activity “Check the test error” that is present in the logs is not present in the model.

Figure 57: logs alignments of case ids 961

Apromore log alignments:

- Case 180, 299, 399: Same as ProM.
- Case 485: Same as ProM.
- Case 7: In this case, we have two violations, the one that we described before in the “conformance checking” section that was about “adjust the number of iterations” activity, and the second violation about the skip of “Choose the best classifier”.

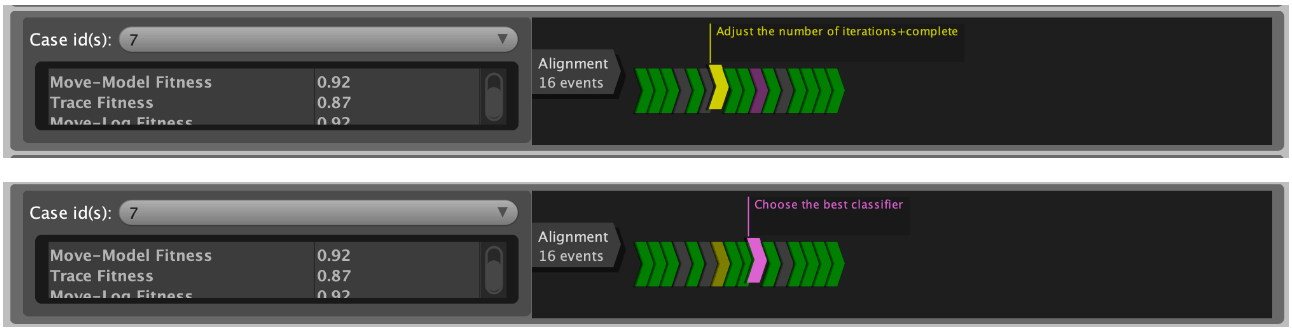


Figure 58: logs alignments of case id 7

- Case 961: In this case, respect to the ProM, we have not the violation about the activity “Check the test error” because it is performed in the path followed by the Apromore’s model. We have the expected violation on the activity “Choose the best classifier” that we skip in the logs respect to the model’s path.

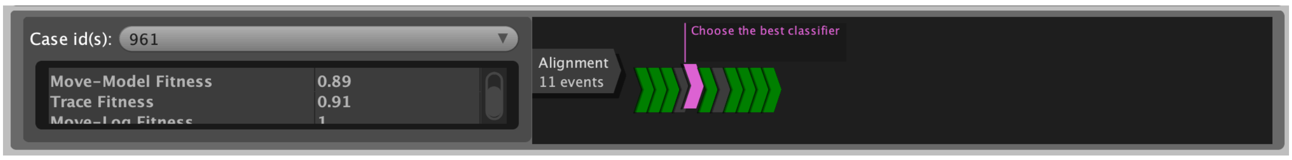


Figure 59: logs alignments of case ids 961

7.6.3 Transition map disco from the modified logs

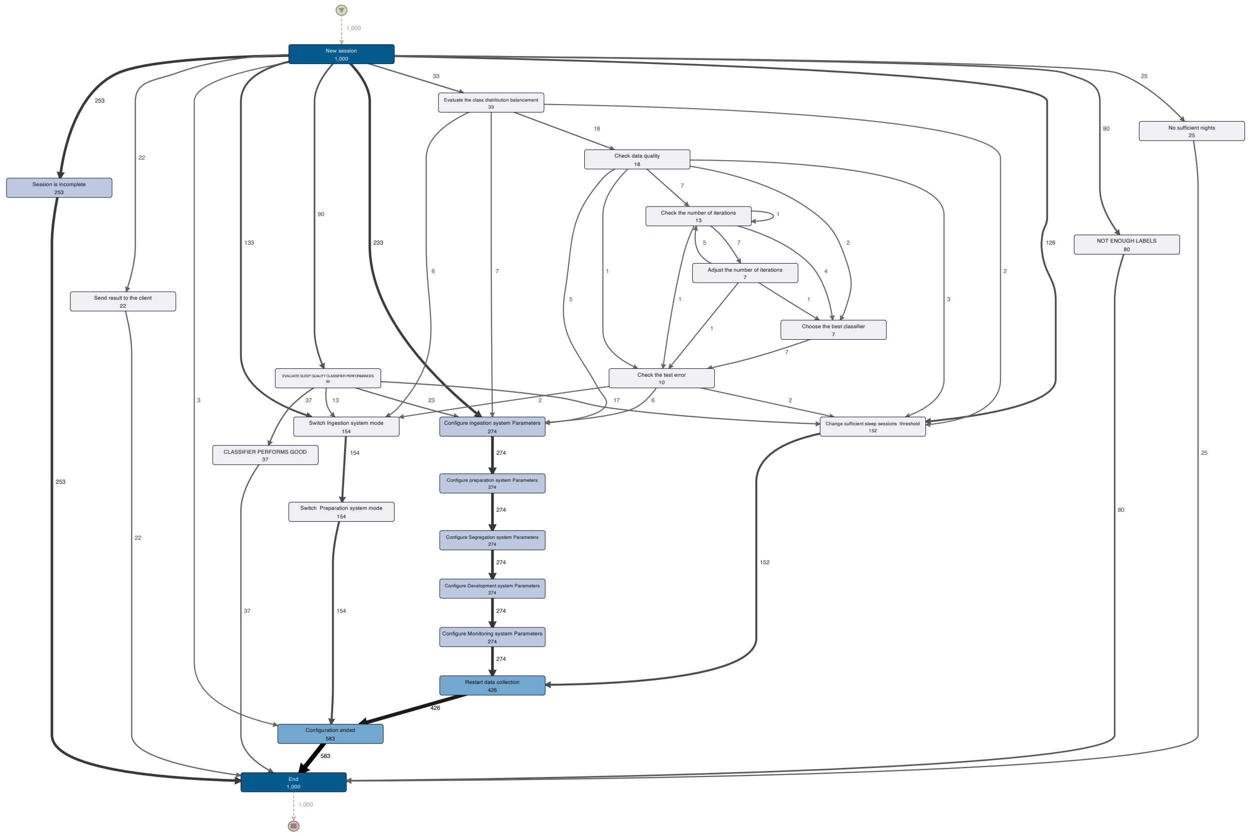


Figure 60: Transition map Disco from the modified logs

Transition map differences:

1. We have 1 new sequence flow that goes from “New Session” to “Configuration Ended” with 3 tokens.
2. We have 3 new sequence flow that goes from “Check the number of iterations”, “Adjust the number of iterations” and “Check data quality” to “Check the test error”.

7.6.4 Transition map Apromore from the modified logs

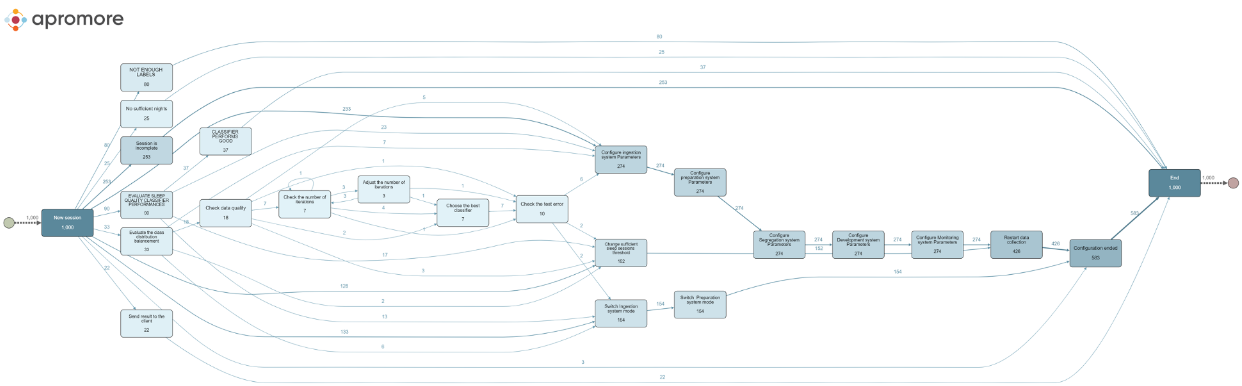


Figure 61: Transition map Apromore from the modified logs

The differences that we got on the Apromore’s transition map are the same that we described for Disco’s transition map.

7.6.5 BPMN Mining from modified logs - ProM

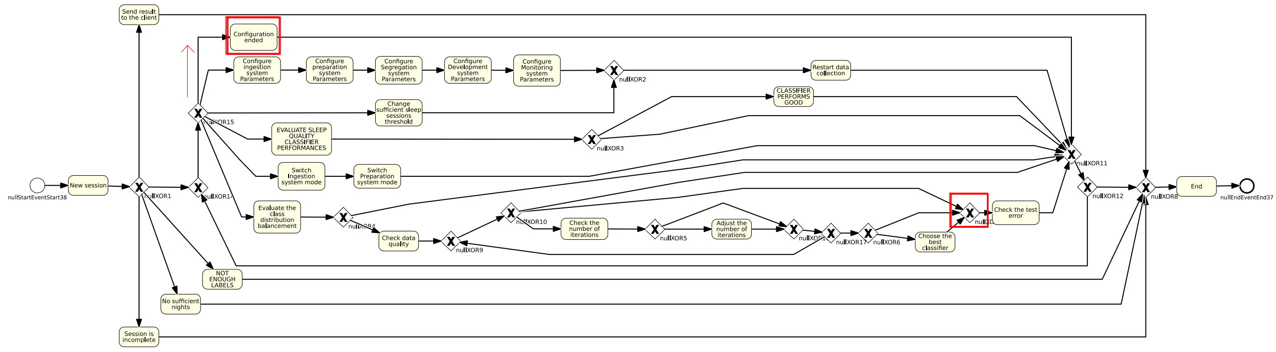


Figure 62: BPMN mined from modified logs with ProM

On this mined model from modified logs we have two differences respect to the ProM model mined on the original logs:

1. We have the direct path between the activities "New Session" and "Configuration ended"
2. We can perform the activity "Check the test error" after the following activities:
   a. Check data quality
   b. Check the number of iterations
   c. Adjust the number of iterations
   d. Choose the best classifier

   Before we could perform this activity only after the activity "Choose the best classifier".

Activities 24, Gateways 16, Sequence Flows 57

7.6.6 BPMN Mining from modified logs - Apromore

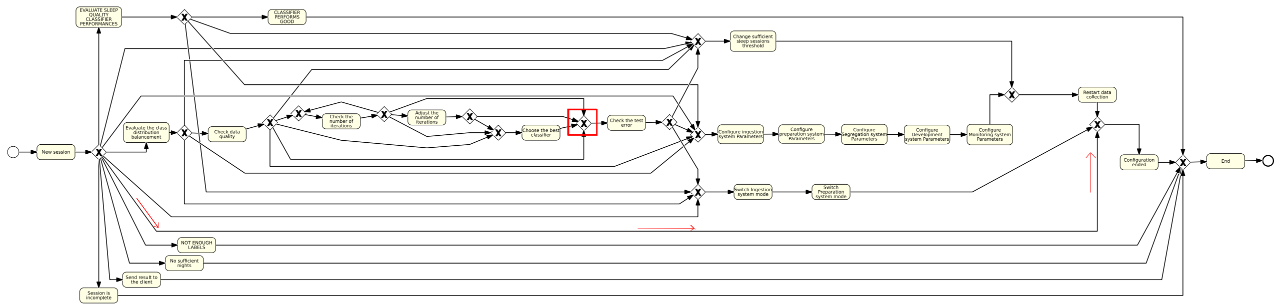


Figure 63: BPMN mined from modified logs with Apromore

The changes that we have in Apromore model are the same that we highlighted in ProM as we expected using the same modified logs.

Activities 24, Gateways 16, Sequence Flows 64

7.6.7 Conformance checking with the modified logs

We used the modified logs (with violations) with the models mined from them to perform the conformance checking metrics.

Mined Model	Fitness	Simplicity	Precision	Generalization
ProM	0.999	97	0.550	0.954
Apromore	0.999	104	0.780	0.939

ProM

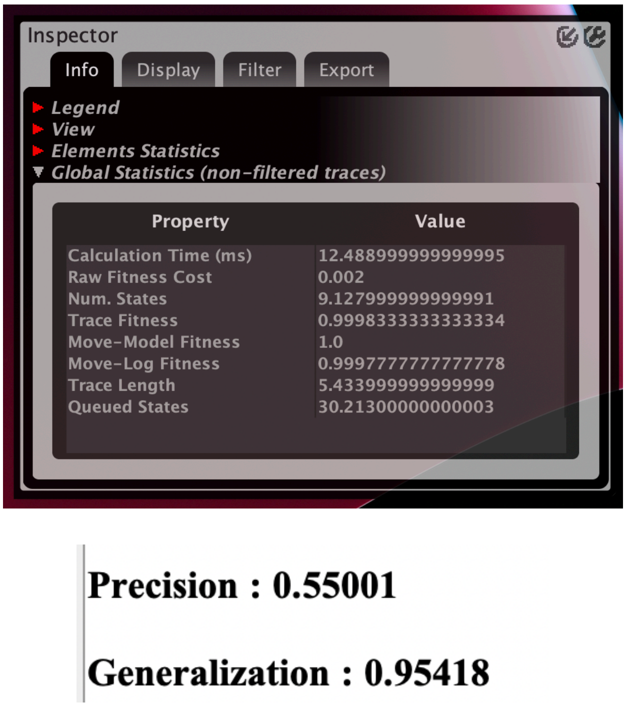


Figure 64: ProM results

Apromore

Figure 65: Apromore Results

Respect to the results that we have obtained on the original logs, we have a higher complexity (higher simplicity) and a lower precision since the BPMN models has more paths respect to the previous ones, so they permit extra behaviour. The added paths are the ones described in the section where we describe the differences between the ProM and Apromore mined models from the modified log versus the models mined from the original log.

The fitness, as expected, is improved with respect to when we were using the BPMN mined from the original logs. Both models obtained a fitness value near to 1 that means that the log can be successfully replayed on the mined models. We didn't obtain a fitness value of 1 for the ProM model because in the log there are 2 cases where the "Choose the best classifier" activity is present in the log but it is not present in the path chose by the mined model.

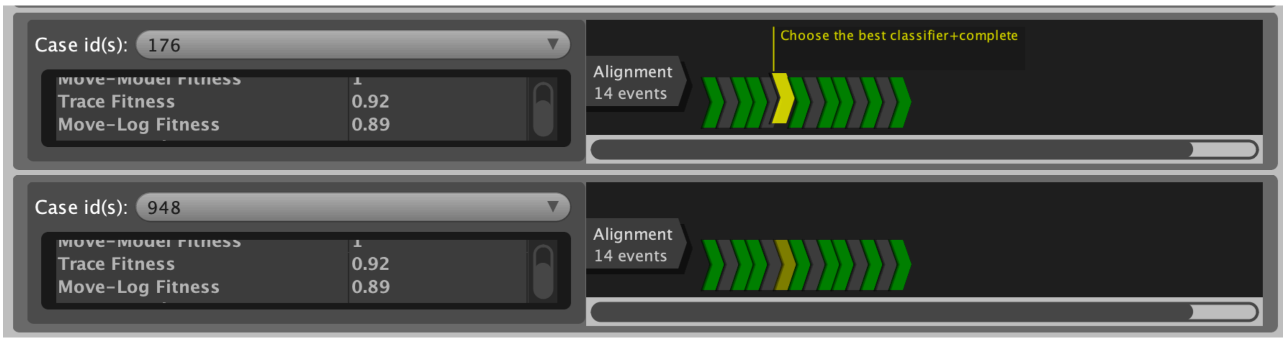


Figure 66: logs alignments of case id 176, 948

Instead, we didn't obtain a fitness value of 1 for the Apromore model for the same reason that we described for the model mined from the original logs. It is because in 3 cases we had a loop in the log that is possible in the As-is model but that is not allowed in the Mined model from Apromore. Indeed, in the Apromore model after adjusting the number of iterations you can't go back to check the number of iterations and you must go one choosing the best classifier. This operation instead is allowed in the original model and in the mined model in ProM.

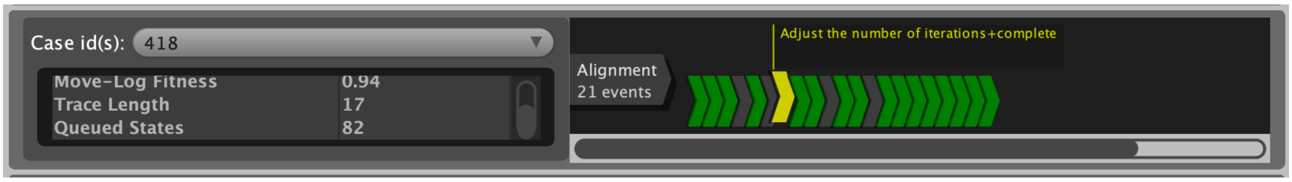


Figure 67: logs alignments of case id 418

7.7 Conclusions

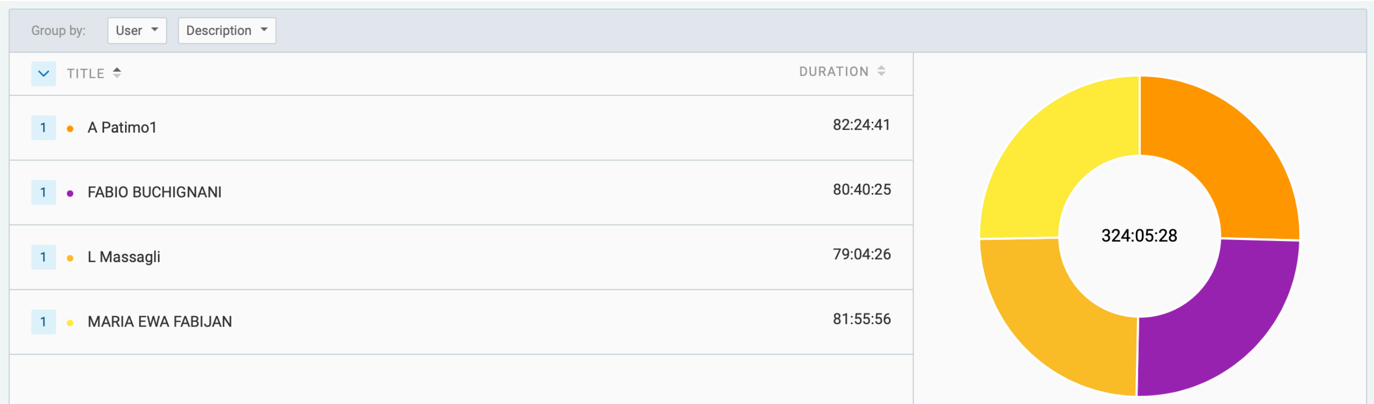
The results that we obtained are the following:

Model	Mined on	Evaluated on	Fitness	Simplicity	Precision	Generalization
ProM	Original Logs	Original Logs	0.999	88	0.667	0.953
Apromore	Original Logs	Original Logs	0.999	96	0.891	0.938
ProM	Modified Logs	Modified Logs	0.999	97	0.550	0.954
Apromore	Modified Logs	Modified Logs	0.999	104	0.780	0.939

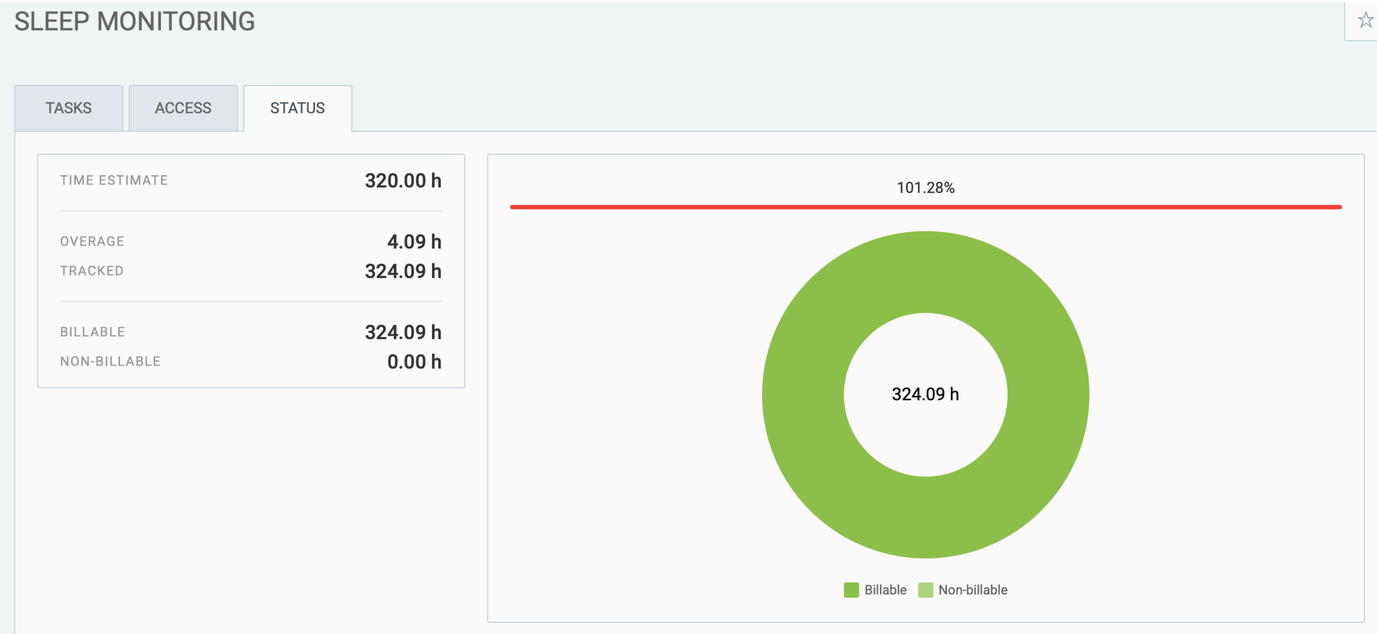
In this table we can see the 4 qualities calculated for the Original Logs and Modified Logs. For what concern the original logs, we got a better model from Apromore because it had a higher precision with respect to the ProM model without losing so much in generalization and complexity. Also, for what concern the modified logs we got that the better model is the Apromore's one with the same conclusions.

We got a lower precision for the ProM's models because we have a gateway in the last part of the model that is linked with a gateway in the first part of the model allowing a loop and a lot of extra behaviour.

Clockify



Group by: User Description
TITLE
DURATION
1 A Patimo1 82:24:41
1 FABIO BUCHIGNANI 80:40:25
1 L Massagli 79:04:26
1 MARIA EWA FABIJAN 81:55:56
324:05:28



SLEEP MONITORING
TASKS
ACCESS
STATUS
TIME ESTIMATE 320.00 h
OVERAGE 4.09 h
TRACKED 324.09 h
BILLABLE 324.09 h
NON-BILLABLE 0.00 h
101.28%
324.09 h
Billable
Non-billable