BRAIN-BASED WHEELCHAIR CONTROL

Project documentation 2024-25

Arduini Luca
Crispino Francesco Pio
Lucchesi Filippo
Speciale Martina

List of contents

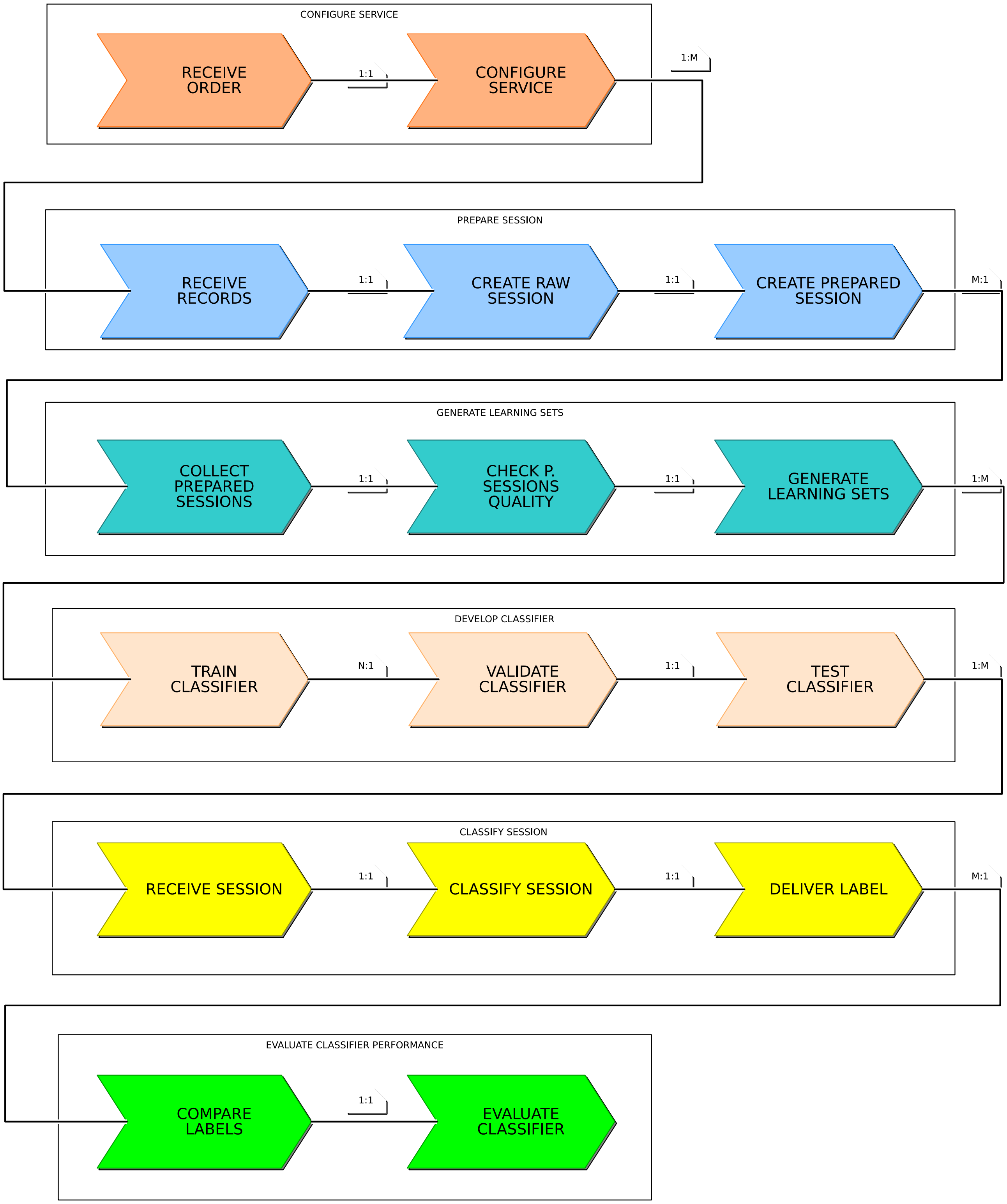
Introduction

Project carried out by the students: Luca Arduini, Francesco Pio Crispino, Filippo Lucchesi and Martina Speciale during the academic year 2024-25 for the 'Process Mining and Intelligence' class at the University of Pisa.

All the files you will see in this documentation are publicly available on a GitHub repository. You will find all the BPMN models, logs, and other resources mentioned in the documentation.

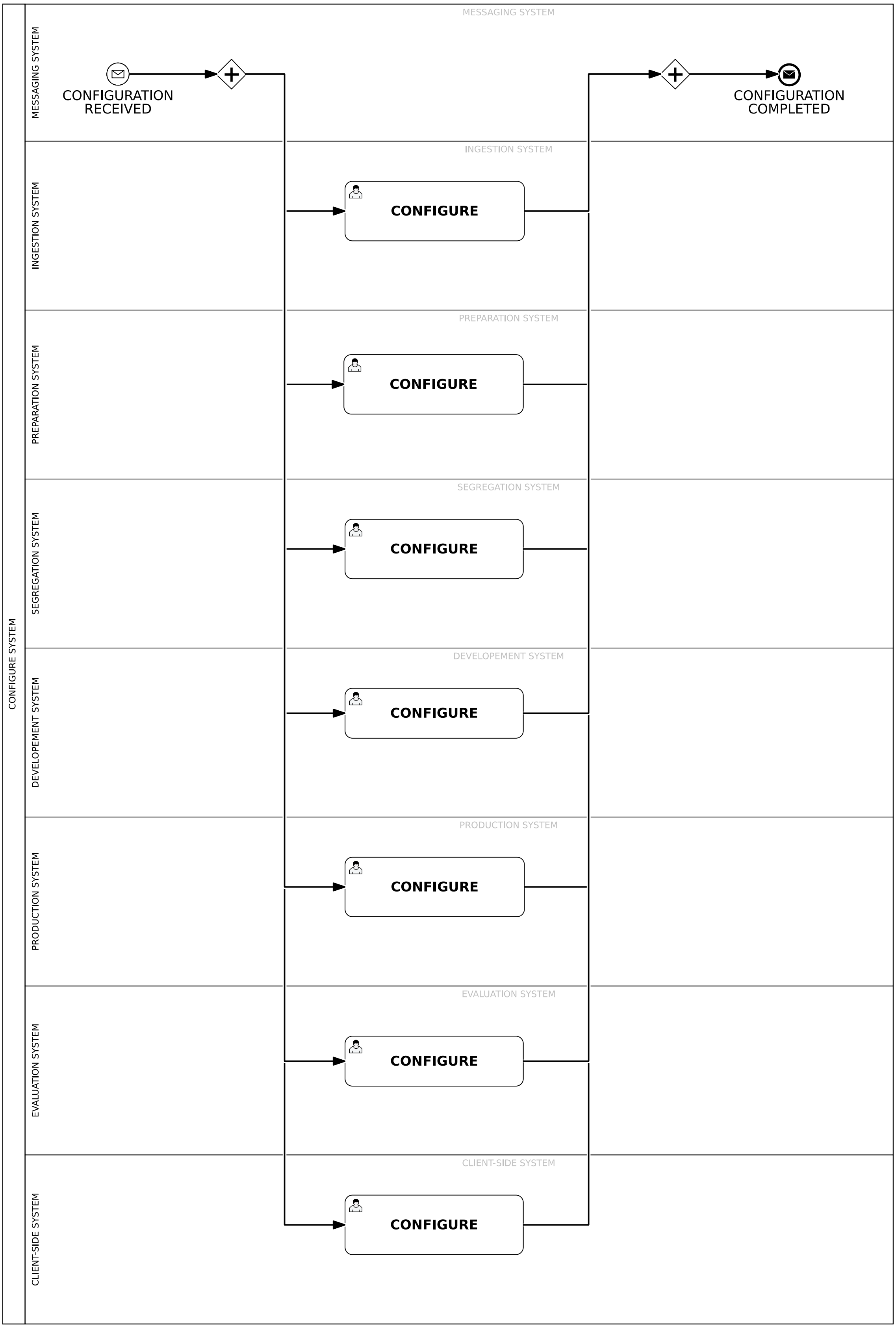
Resources available on GitHub at: `github.com/LucaArduini/BrainBasedWheelchairControl`

Process Landscape

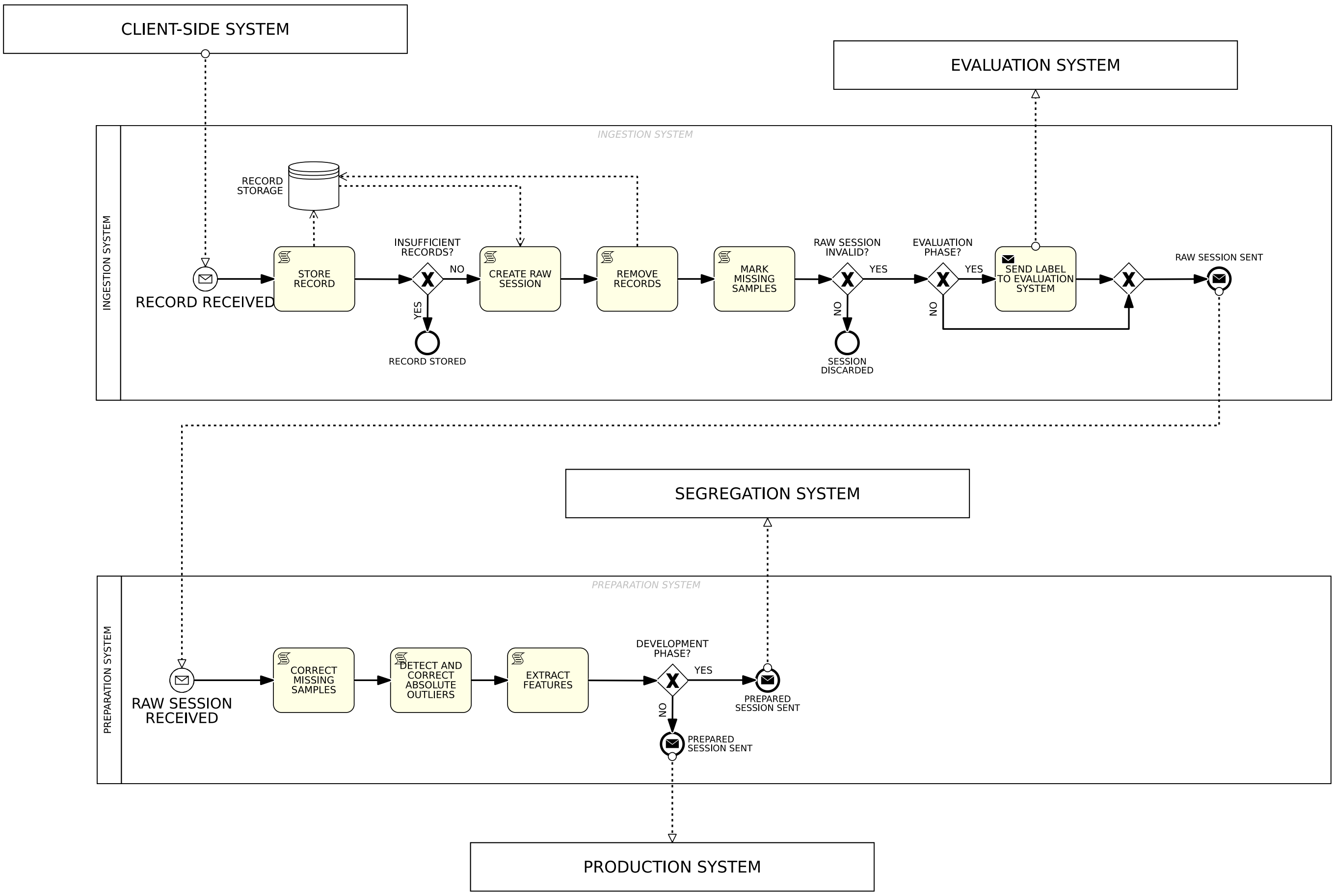


Service-Level BPMN Diagrams

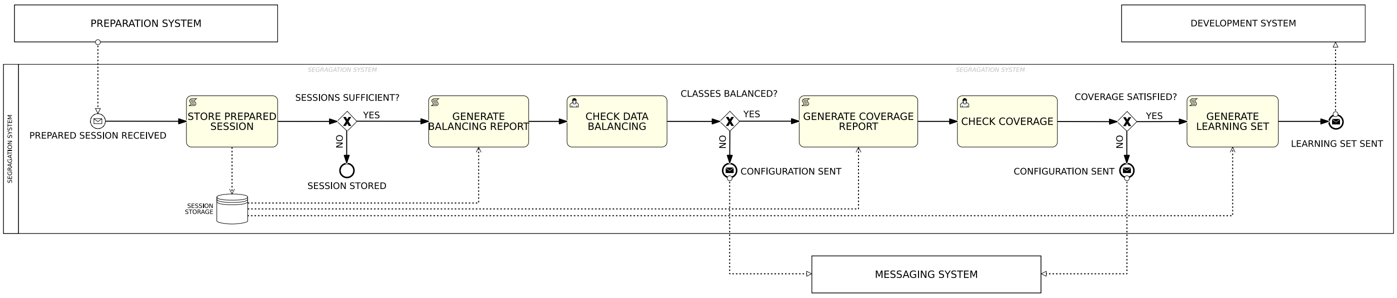
0. CONFIGURE SYSTEM process



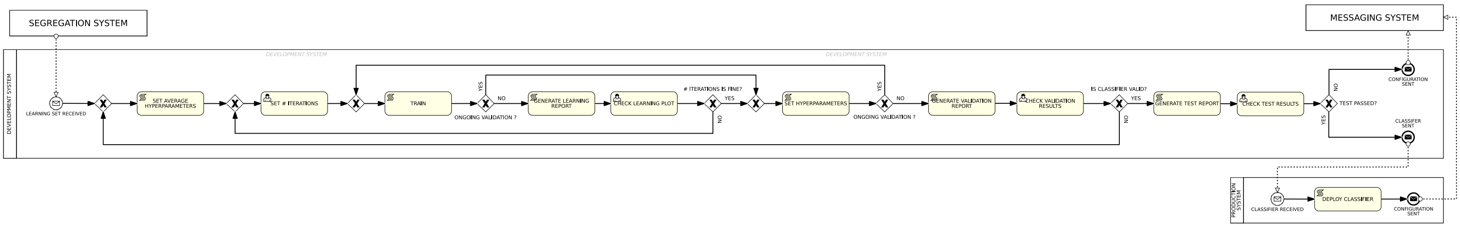
1. PREPARE SESSION process (Lucchesi)



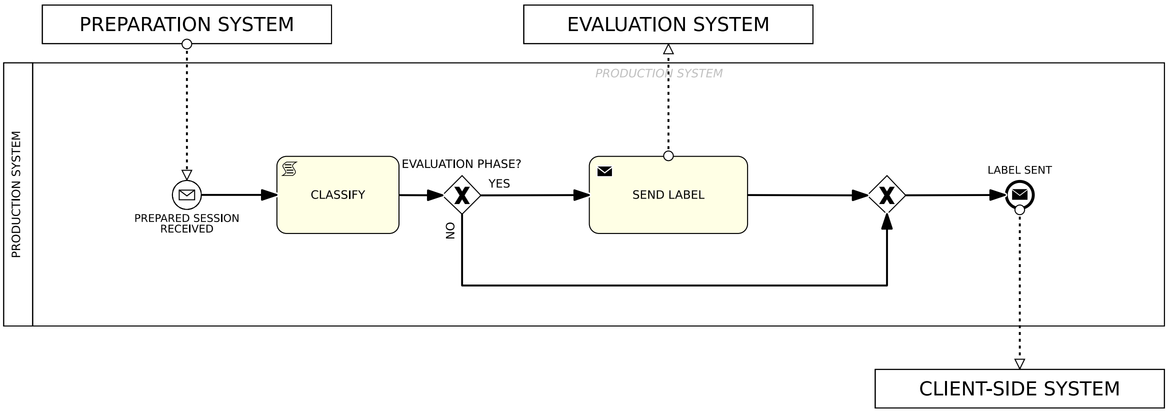
2. GENERATE LEARNING SETS process (Crispino)



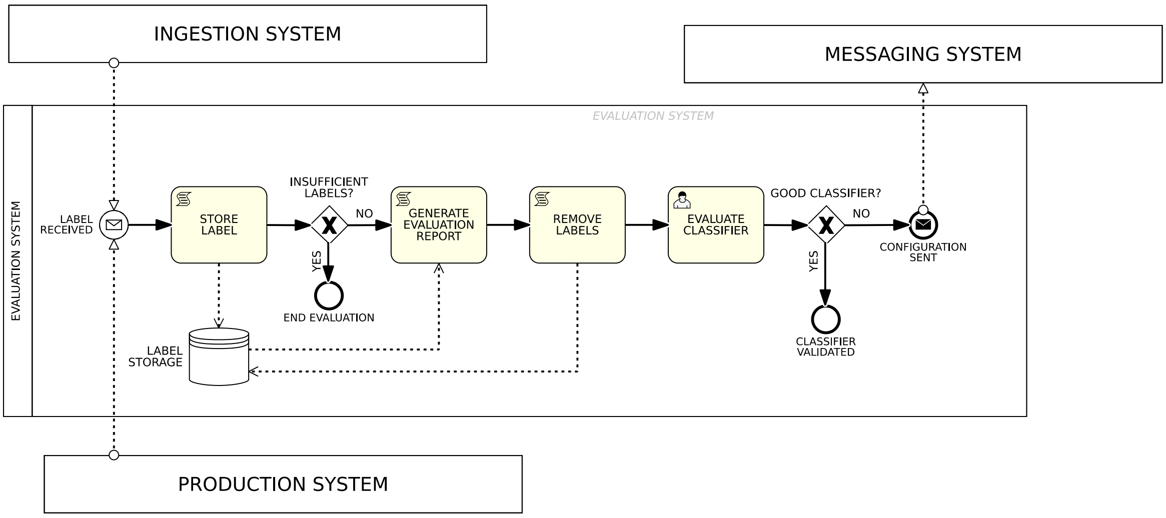
3. DEVELOP CLASSIFIER process (Arduini)



4. CLASSIFY SESSION process (Lucchesi)



5. EVALUATE CLASSIFIER PERFORMANCE process (Speciale)



Use Cases

Assumptions

1. IP addresses

The IP addresses are not arbitrarily chosen by the user filling out the configuration form for a specific system; the user simply inputs the address that has been provided to them.

This is because the factory operates across multiple IT systems located at different companies rather than in a single location. The network structure has been designed by a specialized technician, who subsequently assigned and communicated the IP addresses to the machines.

2. Professional Roles in the Factory

In the table below, we outline the professional roles present in our machine learning factory, focusing on the only two essential roles relevant to the tasks documented in this section.
However, we are fully aware that a factory of this kind requires additional professional roles that, for simplicity, have not been included here. For instance, we can consider various technicians, such as the network technician mentioned earlier, or scientists specialized in brain waves emitted by our brain, who are crucial for the proper configuration of the parameters of our helmet used to analyze the brain's electrical activity in our patients.

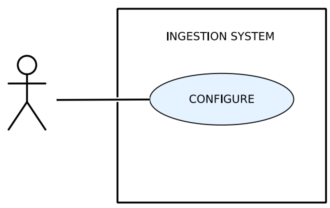
Actors

In this factory, the main professional roles are two:

JOB ROLE	DUTIES	ANNUAL SALARY	RATIO
Data Analyst	Primarily focuses on data analysis, providing insights and building reports to support business decisions. The role emphasizes preparing clean datasets, analyzing distributions, and delivering initial validations as input for models.	80,000 $	1
Machine Learning Engineer	Responsible for designing, developing, and training machine learning models. This role combines expertise in software development, data engineering, and machine learning, with a focus on scalability and operational efficiency.	140,000 $	1.75

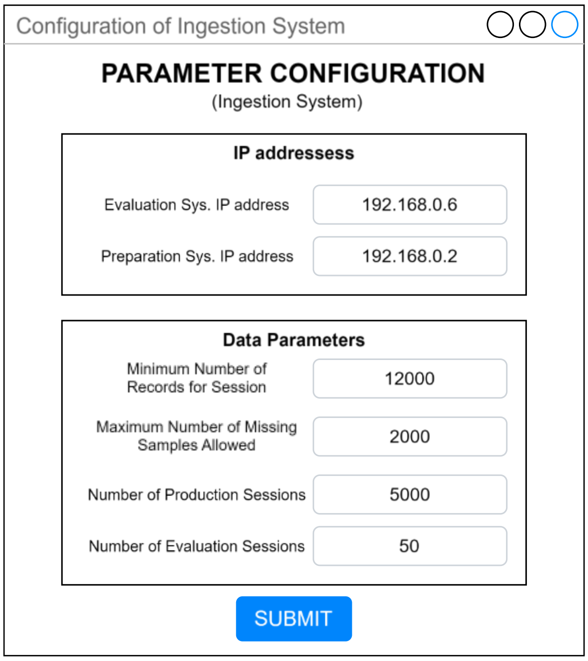
- The salaries listed in the table refer to average values for professionals with 5 years of experience.
- Furthermore, as previously mentioned, this table does not include all the professional roles present in our factory.

Ingestion System



Task: Ingestion System Configuration (Lucchesi)

Mockup



Detailed Use Case

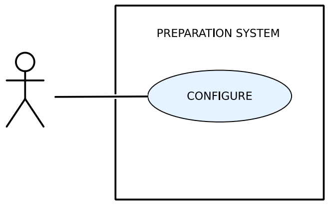
STEP	COST CALCULATION	SC
1) USER opens the configuration form for the Ingestion System	1x1x1	1
2) SYSTEM displays the configuration form		
3) FOR EACH IP address to set:	2	
3.1) USER looks at the IP address	2x(1+3)x1	8
3.2) USER enters the IP address	2x1x1	2
4) USER selects a value for the minimum number of records to collect for a session	1x(4x3)x1	12
5) USER selects a value for the maximum number of missing samples allowed	1x(4x3)x1	12
6) USER selects a value for the number of production sessions	1x(4x3)x1	12
7) USER selects a value for the number of evaluation sessions	1x(4x3)x1	12
8) USER selects SUBMIT	1x1x1	1
9) SYSTEM displays a confirmation		
10) USER closes the form	1x1x1	1
	HUMAN TASK COST	61

The cost shown in red (+3) indicates the IT technician's cost for assigning the IP address to the Evaluation System and the Preparation System.

The cost shown in green (x3) indicates that this is a multi-criteria choice. Specifically, the USER must consider: total number of samples, rate of lost samples in the client-side systems (helmet, environment, activity), rate of arrival of samples.

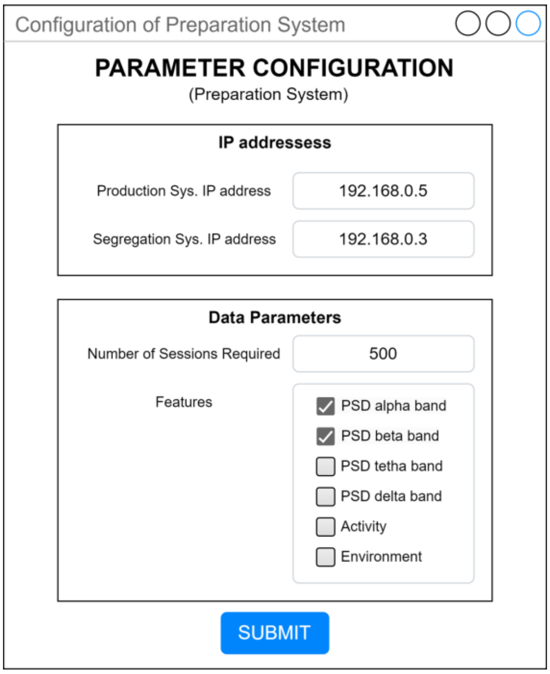
The cost shown in blue (x3) represents another multi-criteria choice. The USER must consider: age of the patient, illnesses/medical history of patient, preferred activities of the patient.

Preparation System



Task: Preparation System Configuration (Speciale)

Mockup



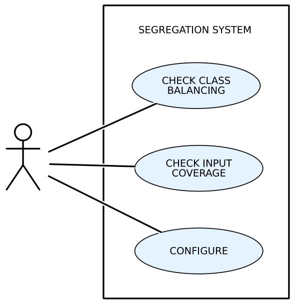
Detailed Use Case

Step	Description	Cost	SC
1	The USER opens the configure production system form	1x1x1	1
2	The SYSTEM shows the configuration form		
3	3) FOR EACH IP address to set:	2	
3.1	3.1) USER looks at the IP address	2x(1+3)x1	8
3.2	3.2) USER enters the IP address	2x1x1	2
4	The USER sets the Number of Sessions Required	1x(4x3)x1	12
5	FOR EACH feature	6	
5.1	The USER sets the feature to extract	6x(4x3)x1	72
6	The USER clicks on “Submit”	1x1x1	1
7	The SYSTEM shows a confirmation message		
8	The USER closes the form	1x1x1	1
		HUMAN TASK COST	97

The cost shown in red (+3) indicates the IT technician's cost for assigning the IP address to the Evaluation System and the Preparation System.

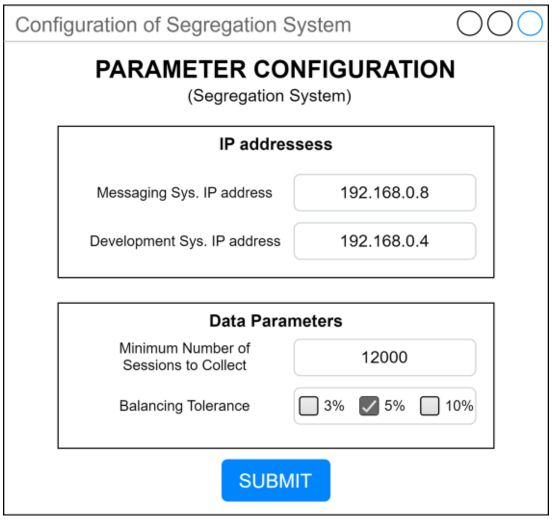
The cost shown in green (x3) indicates that this is a multi-criteria choice. Specifically, the USER must consider: patient's age, medical history/illnesses, previous experience with similar cases.

Segregation System



Task: Segregation System Configuration (Arduini)

Mockup



Detailed Use Case

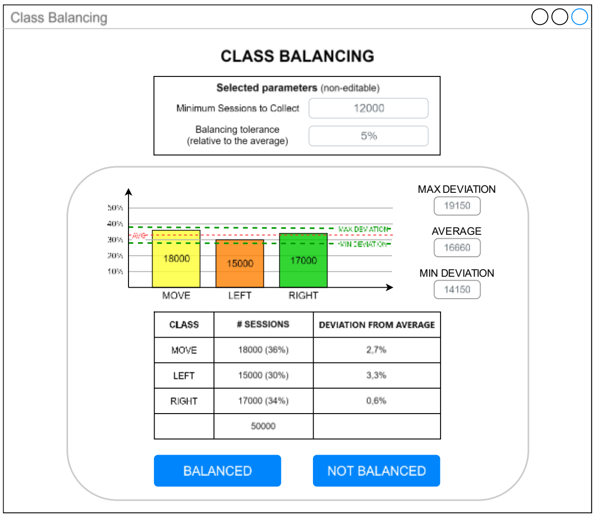
STEP	COST CALCULATION	SC
1) USER opens the configuration form for the Segregation System	1x1x1	1
2) SYSTEM displays the configuration form		
3) FOR EACH IP address to set:	2	
3.1) USER looks at the IP address	2x(1+3)x1	8
3.2) USER enters the IP address	2x1x1	2
4) USER selects a value for the minimum number of sessions to collect	1x(4x3)x1	12
5) USER selects a value for the Balancing Tolerance	1x(4x3)x1	12
6) USER selects SUBMIT	1x1x1	1
7) SYSTEM displays a confirmation		
8) USER closes the form	1x1x1	1
	HUMAN TASK COST	37

The cost shown in red (+3) indicates the IT technician's cost for assigning the IP address to the Messaging System and the Development System.

The cost shown in green (x3) indicates that this is a multi-criteria choice. Specifically, the USER must consider: general conventions, medical history/illnesses and patient's habitual activities.

Task: Check Data Balancing (Arduini)

Mockup



Detailed Use Case

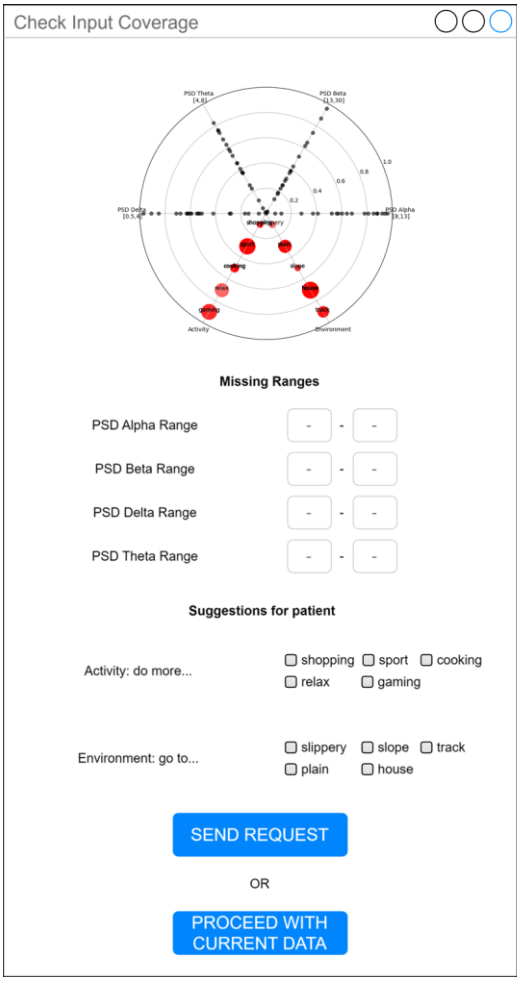
STEP	COST CALCULATION	SC
1) USER opens Class Balancing form	1x1x1	1
2) SYSTEM displays the report		
3) FOR EACH class	3	
3.1) USER checks the number of available sessions (by viewing the histogram)	3x3x1	9
3.2) USER ensures that the available sessions exceed the minimum number of sessions to collect	3x3x1	9
3.3) USER ensures that the available sessions exceed the minimum deviation	3x3x1	9
4.1) IF all classes meet the balance tolerance	0.8	
4.1.1) USER selects BALANCED	0.8x1x1	0,8
4.2) ELSE	0.2	
4.2.1) USER selects NOT BALANCED	0.2x1x1	0,2
4.2.2) SYSTEM displays an additional pop-up to request additional sessions		
4.2.3) FOR EACH class	3	
4.2.3.1) USER set the number of sessions to request	0.2x3x(4x3)x1	7,2
4.2.4) USER selects REQUEST	0.2x1x1	0,2
5) SYSTEM displays a confirmation		
6) USER closes Class Balancing form	1x1x1	1

HUMAN TASK COST	37,4

The cost shown in green (x3) indicates that this is a multi-criteria choice. Specifically, the USER must consider record loss, sample loss, and balance tolerance.

Task: Check coverage (Lucchesi)

Mockup



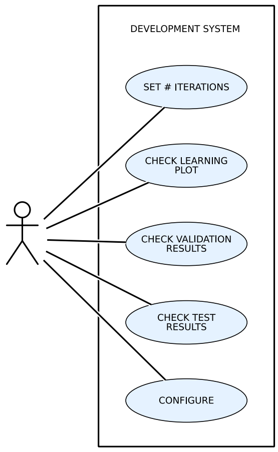
Detailed Use Case

STEP	COST CALCULATION	SC
1) USER opens ‘Check Input Coverage’ Form	1x1x1	1
2) SYSTEM shows the report		
3) FOR each feature	6	
3.1) USER observes distribution of values of feature	6x(4x3)x1	72
3.2) IF some values are not distributed as required	0.66	
3.2.1) USER inserts which ranges of values are missing for continuous attributes (PSD)	(4*0.66)x3x1	7,92
3.2.2) USER inserts suggestions of activity and environment to the patient	(2*0.66)x3x1	3,96
3.2.3) USER selects REQUEST	1x1x1	1
4) ELSE	0.33	

4.1) USER selects PROCEED WITH CURRENT DATA	0.33x1x1	0,33
5)SYSTEM shows a confirmation		
6)USER closes 'Check Input Coverage' form	1x1x1	1
	HUMAN TASK COST	87,21

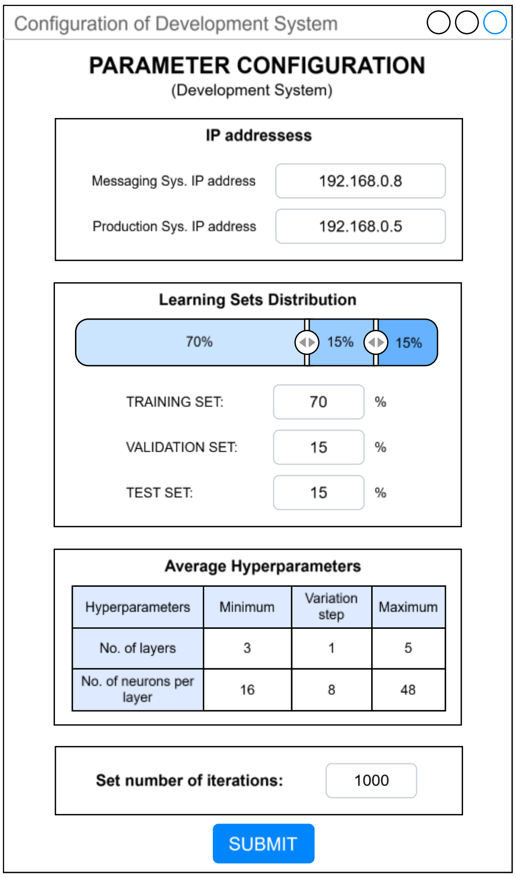
The cost shown in green (x3) indicates that this is a multi-criteria choice. These multi-criteria are: age of the patient, illnesses/medical history of patient, preferred activities of the patient.

Development System



Task: Development System configuration (Crispino)

Mockup



Detailed Use Case

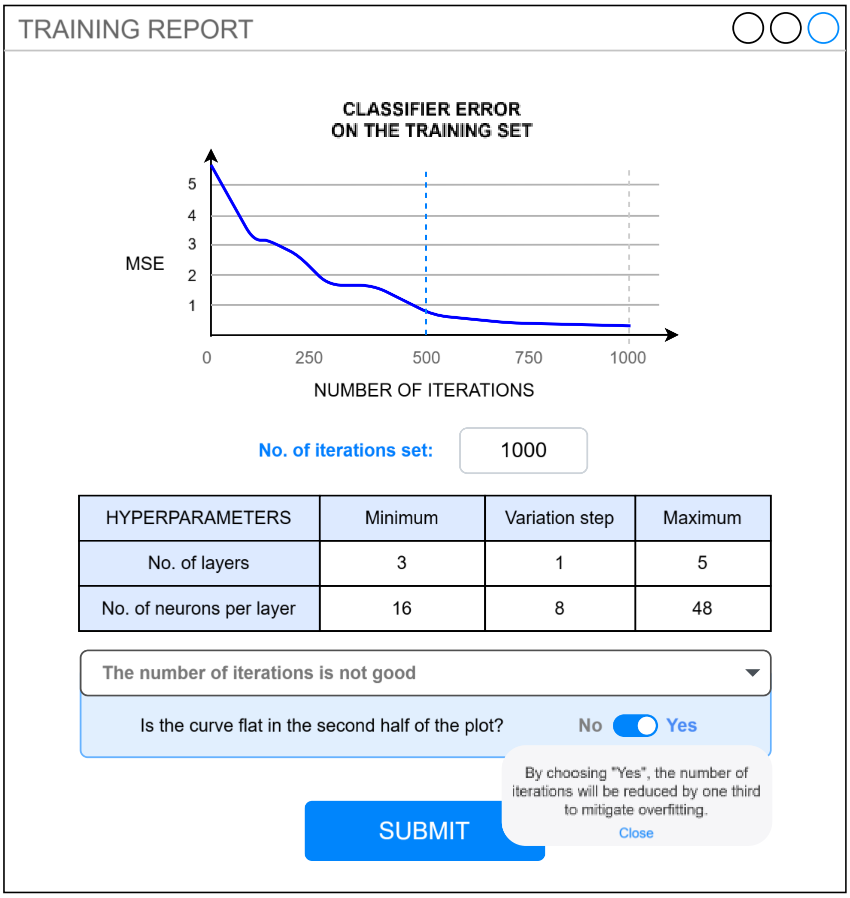
STEP	COST CALCULATION	SC
1) USER opens the configuration form for the Development System	1x1x1.75	1,75
2) SYSTEM shows the configuration form		
3) FOR EACH IP address to set:	2	
3.1) USER looks at the IP address	2x1x1.75	3,5
3.2) USER enters the IP address	2x1x1.75	3,5
4) USER enters the size of training set	1x4 x3 x1.75	21
5) USER enters the size of validation set	1x4 x3 x1.75	21
6) USER enters the size of test set	1x1x1.75	1,75
7) FOR EACH hyperparameter of the table	6	
7.1) USER sets the hyperparameter	6x4 x3 x1.75	126
8) USER chooses the number of iterations to execute	1x4 x3 x1.75	21
9) USER selects SUBMIT	1x1x1.75	1,75
10) SYSTEM shows a confirmation		
11) USER closes the configuration form	1x1x1.75	1,75
	HUMAN TASK COST	203

The cost shown in green (x3) indicates that this is a multi-criteria choice:

- at elements 4), 5) and 6) the USER must consider: general conventions, problem domain and number of sessions available. The choice of one of the three has cost 1 since it is automatically resized based on the previous two choices;
- at element 7.1), instead, the USER must consider: conventions based on domain knowledge, available computational power and previous experience with similar cases.
- at element 8), instead, the USER must consider: number of sessions in the training set, architecture of the prototype neural networks and previous experience with similar cases.

Task: Check learning plot (Crispino)

Mockup



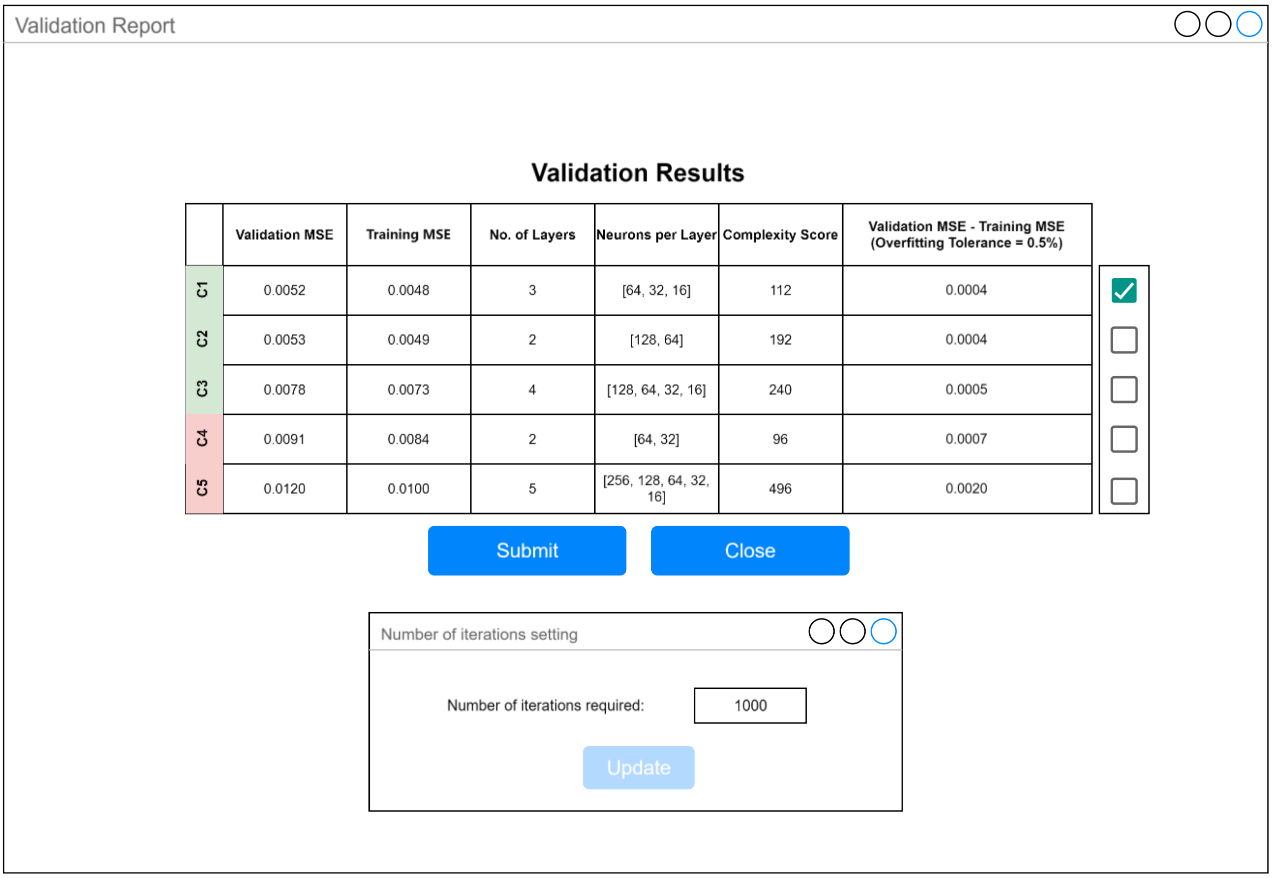
Detailed Use Case

STEP	COST CALCULATION	SC
1) USER opens the training report form	1x1x1.75	1,75
2) SYSTEM shows the form		
3) USER checks the trend of the MSE plot	1x3x1.75	5,25
4) IF based on the trend, the number of iterations is not good	0.3	
4.1) IF the learning plot is flat in the second half of the plot	0.5	
4.1.1) USER opens the dropdown menu	0.15x1x1.75	0,2625
4.1.2) USER selects “YES” using the switch	0.15x1x1.75	0,2625
4.1.3) SYSTEM reduces the number of iterations by one third		
4.2) ELSE	0.5	
4.2.1) USER opens the dropdown menu	0.15x1x1.75	0,2625
4.2.2) USER selects “NO” using the switch	0.15x1x1.75	0,2625
4.2.3) SYSTEM increments the number of iterations by one third		
5) ELSE	0.7	
5.1) USER does not open the dropdown menu, number of iterations is good	0.7x1x1.75	1,225
6) USER selects SUBMIT	1x1x1.75	1,75

7) SYSTEM shows a confirmation		
8) USER closes the training report form	1x1x1.75	1,75
	HUMAN TASK COST	11,025

Task: Check Validation Results (Speciale)

Mockup



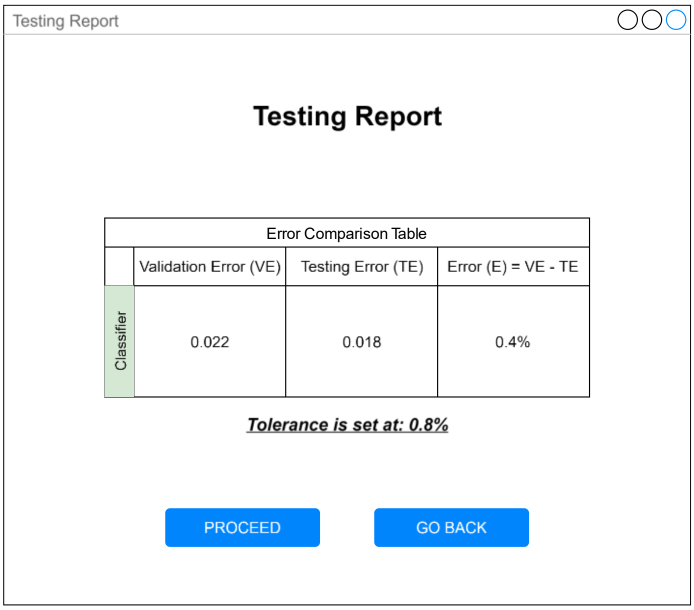
Detailed Use Case

Step	Description	Cost	SC
1	The USER opens the validation result form	1x1x1.75	1,75
2	The SYSTEM displays the report: • Green for classifiers with (Validation MSE – Training MSE) ≤ tolerance • Red for classifiers with (Validation MSE – Training MSE) > tolerance		
3	The USER checks the color of the first classifier	1x2x1.75	3,5
4	IF the color is red	0.05	
4.1	The SYSTEM shows a warning message: "No selectable classifier. Check parameters." since no classifier meets the tolerance threshold. The “Number of iterations setting” tab opens, and the “Update” button is enabled once the new value is inserted		
4.2	The USER updates the Number of iterations required	0.05x(4x3)x1.75	1,05
4.3	The USER clicks on the UPDATE button	0.05x1x1.75	0,0895
5	ELSE (if color is green)	0.95	

5.1	The USER selects the first classifier	0.95x1x1.75	1,6625
5.2	FOR EACH classifier starting from the second		
5.2.1	The USER checks the color of the i-th classifier in the first column of the validation report (which shows the classifier ID)	4x2x1.75	14
5.2.2	IF the i-th classifier is green	0.5	
5.2.2.1	The USER reads the Validation MSE of the i-th classifier	0.475x3x1.75	2,49375
5.2.2.2	The USER reads the Validation MSE of the selected classifier	0.475x3x1.75	2,49375
5.2.2.3	The USER compares the Validation MSE of the i-th classifier with the selected classifier's Validation MSE:	0.475x3x1.75	2,49375
5.2.2.4	IF the two values are of the same order of magnitude	0.2	
5.2.2.4.1	The USER reads the Complexity Score (**Σ** (number of neurons in the current layer * number of neurons in the next layer) + biases) of the selected classifier	0.095x3x1.75	0,49875
5.2.2.4.2	The USER reads the Complexity Score of the i-th classifier	0.095x3x1.75	0,49875
5.2.2.4.3	The USER compares the two values and selects the classifier with the lower value (less complex classifier)	0.095x3x1.75	0,49875
5.2.2.5	ELSE (if the Validation MSE values are significantly different)	**0.8**	
5.2.2.5.1	The USER selects the classifier with the minimum Validation MSE	0.38x3x1.75	1,995
5.2.3	ELSE (the i-th classifier is red)	**0.5**	
5.2.3.1	The USER clicks SUBMIT to confirm the selection, since there is no more classifier that needs to be checked	0.475x3x1.75	2,49375
6	The SYSTEM shows a confirmation message: "Classifier selected successfully."		
7	The USER closes the validation results form	1x1x1.75	1,75
		HUMAN TASK COST	37,26825

Task: Check Testing Report (Lucchesi)

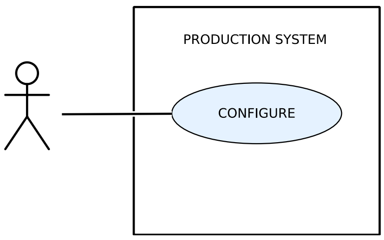
Mockup



Detailed Use Case

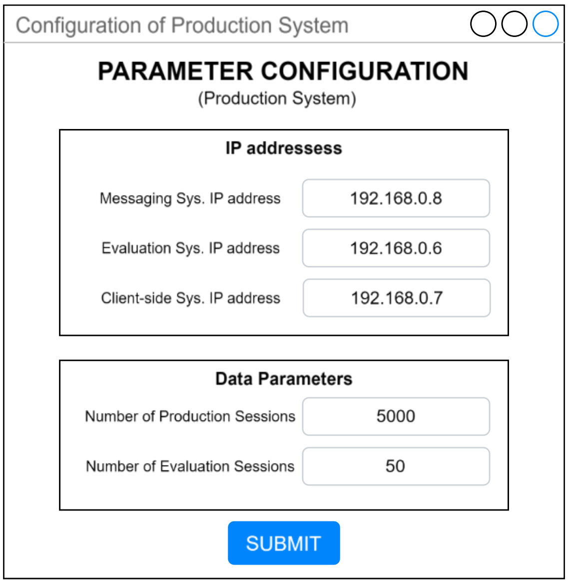
STEP	COST CALCULATION	SC
1) USER opens 'Testing Report' Form	1x1x1,75	1,75
2) SYSTEM shows the report		
3) USER compares error E with tolerance	1x3x1,75	5,25
4) IF E ≤ tolerance	0.99	
4.1) USER selects PROCEED	0.99x1x1,75	1,7325
4.2) ELSE	0.01	
4.2.1) USER selects GO BACK	0.01x1x1,75	0,0175
5) SYSTEM shows a confirmation box		
6) USER closes 'Testing Report' form	1x1x1,75	1,75
	HUMAN TASK COST	10,5

Production System



Task: Production System Configuration (Arduini)

Mockup



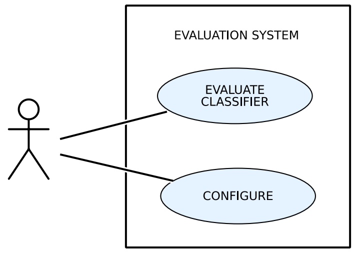
Detailed Use Case

STEP	COST CALCULATION	SC
1) USER opens the configuration form for the Production System	1x1x1	1
2) SYSTEM displays the configuration form		
3) FOR EACH IP address to set:	3	
3.1) IF setting the IP address of the Client-side system	0.33	
3.1.1) USER looks at the Client-side IP address	0.33x3x(1+3)x1	3,96
3.2) ELSE	0.67	
3.2.1) USER looks at the IP address	0.67x3x1x1	2,01
3.3) USER enters the IP address	3x1x1	3
4) USER enters the number of production sessions	1x(4x3)x1	12
5) USER enters the number of evaluation sessions	1x(4x3)x1	12
6) USER selects SUBMIT	1x1x1	1
7) SYSTEM displays a confirmation		
8) USER closes the form	1x1x1	1
	HUMAN TASK COST	35,97

The cost shown in red (+3) indicates the IT technician's cost for assigning the IP address to the Client-side System. <u>The costs for the other two addresses are accounted for in other tables</u>.

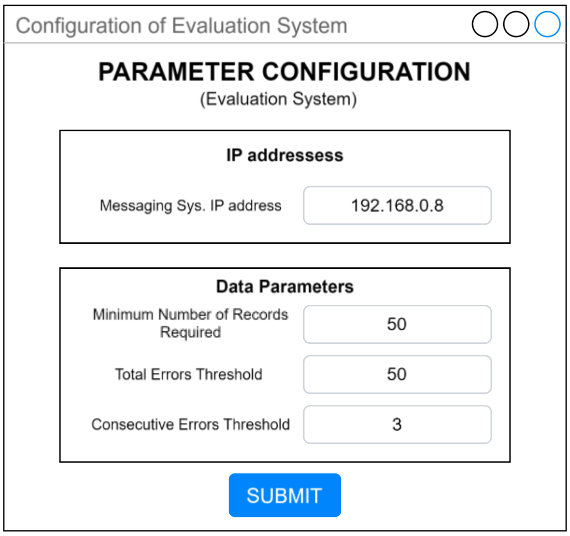
The cost shown in green (x3) represents a multi-criteria choice. In this case, the USER must consider: the patient's age, medical history/illnesses, and the patient's habitual activities.

Evaluation System



Task: Evaluation System Configuration (Speciale)

Mockup



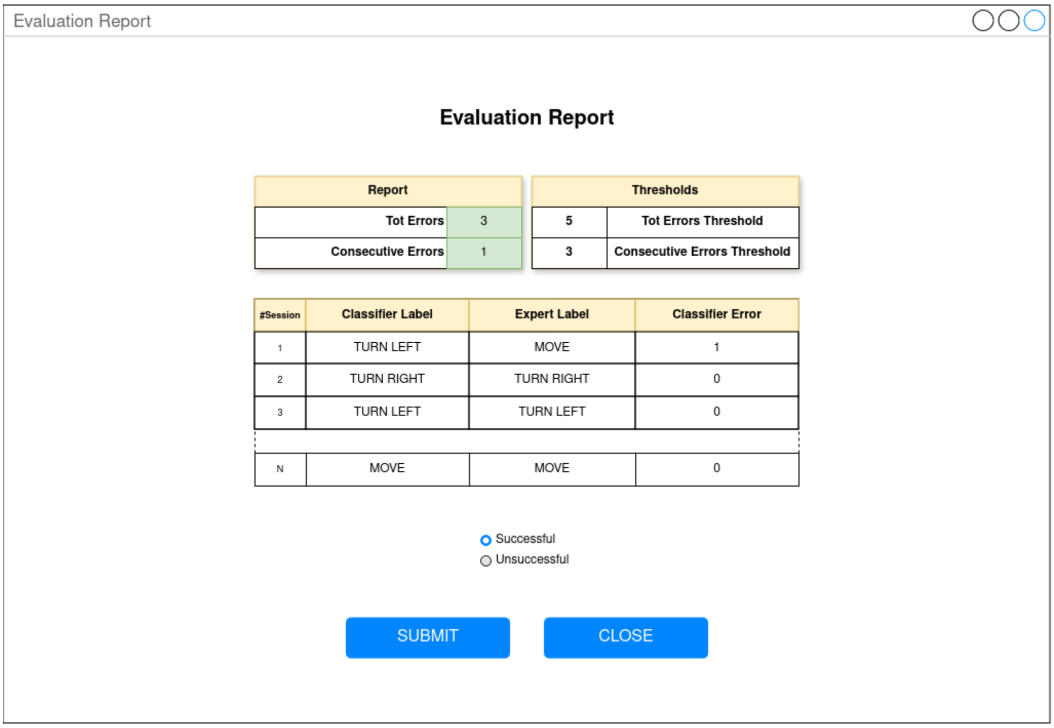
Detailed Use Case

Step	Description	Cost	SC
1	The USER opens the parameter configuration form	1x1x1	1
2	The SYSTEM shows the form		
3	The USER sets the Evaluation System IP Address	1x1x1	1
4	The USER sets the Ingestion System IP Address	1x1x1	1
5	The USER sets the Production System IP Address	1x1x1	1
6	The USER sets the Messaging System IP Address	1x1x1	1
7	The USER sets the Minimum Number of Records required to perform an evaluation	1x(4x3)x1	12
8	The USER sets the Total Number of Errors allowed for a Classifier to be considered "Successful"	1x(4x3)x1	12
9	The USER sets the Total Number of Consecutive Errors allowed for a Classifier to be considered "Successful"	1x(4x3)x1	12
10	The SYSTEM shows a confirmation message		
11	The USER closes the configuration form	1x1x1	1
		HUMAN TASK COST	42

The cost shown in green (x3) represents another multi-criteria choice. The USER must consider: age of the patient, illnesses/medical history of patient, preferred activities of the patient.

Task: Evaluate Classifier (Speciale)

Mockup



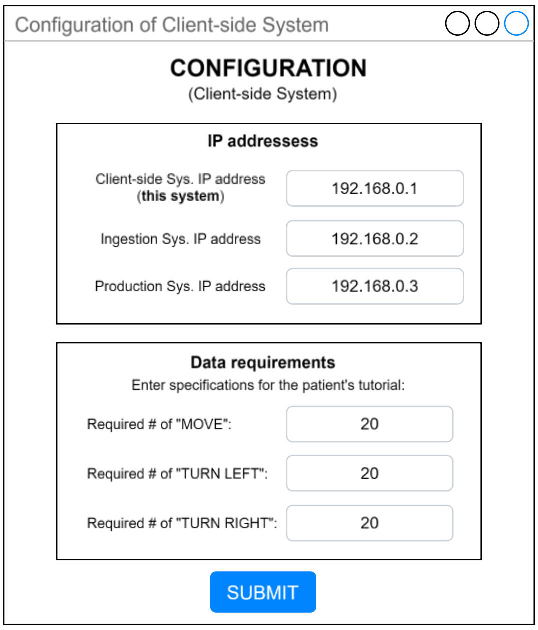
Detailed Use Case

STEP	DESCRIPTION	COST	SC
1	The USER opens the evaluate classifier form	1x1x1,75	1,75
2	The SYSTEM displays the report of the classifier. For each specific session, the USER can observe the errors (misclassifications) made by the classifier		
3	The USER reads the value of total error (*tot_error*)	1x3x1.75	5,25
4	The USER reads the value of total consecutive errors (*consecutive_error*)	1x3x1,75	5,25
5	The USER compares *tot_error* with *tot_error_threshold*	1x3x1,75	5,25
6	The USER compares *tot_consecutive_error* with *consecutive_error_threshold*	1x3x1,75	5,25
7	IF *tot_error* < *tot_error_threshold*	0.9	
7.1	IF *consecutive_error* < *consecutive_error_threshold*	0.9	
7.1.1	The USER selects SUCCESSFUL	0.81	0,81
8	ELSE	0.1	
8.1	The SYSTEM shows a warning message: "The number of errors exceeds the allowed threshold. It is recommended to review the classifier."		
8.2	The USER selects UNSUCCESSFUL	0.1x1x1,75	0,175
9	The USER selects SUBMIT, which is enabled only after a selection.	1x1x1,75	1,75
10	The SYSTEM shows a confirmation message: "Classifier evaluated successfully: [Successful/Unsuccessful]. You can proceed."		
11	The USER closes the evaluate classifier form	1x1x1,75	1,75
		HUMAN TASK COST	27,235

Client-Side System

Task: Client-Side System Configuration (Crispino)

Mockup



Detailed Use Case

STEP	COST CALCULATION	SC
1) USER opens the configuration form for the Client-side System	1x1x1	1
2) SYSTEM displays the configuration form		
3) FOR EACH IP address to set:	1	
3.1) USER looks at the IP address	1x(1+3)x1	4
3.2) USER enters the IP address	1x1x1	1
4) FOR EACH wheelchair possible movement (move, turn left, turn right)	3	
4.1) USER enters the number of tutorial activities required	3x(4x3)x1	36
5) USER selects SUBMIT	1x1x1	1
6) SYSTEM shows a confirmation		
7) USER closes the configuration form	1x1x1	1
	HUMAN TASK COST	44

The cost shown in red (+3) indicates the IT technician's cost for assigning the IP address to the Ingestion System.

The cost shown in green (x3) indicates that this is a multi-criteria choice. These multi-criteria past and present patient's health conditions, activities the patient enjoys most and environments they visit most.

Data Models

Configure Systems

CONFIGURE INGESTION SYSTEM
PreparationSysIP: String
EvaluationSysIP: String
MinRecordsForSession: unsigned integer
MaxMissingSamplesAllowed: unsigned integer
NumberOfEvaluationSessions: unsigned integer
NumberOfProductionSessions: unsigned integer

CONFIGURE DEVELOPMENT SYSTEM
MessagingSystemIP: String
ProductoinSystemIP: String
NumberOfIterations: Unsigned Integer
TrainingSetSize: Double
ValidationSetSize: Double
TestSetSize: Double
MinNeuronsPerLayer: Unsigned Integer
StepNeuronsPerLayer: Unsigned Integer
MiaxNeuronsPerLayer: Unsigned Integer
MinLayers: Unsigned Integer
StepLayers: Unsigned Integer
MaxLayers: Unsigned Integer

CONFIGURE PREPARATION SYSTEM
IsDevelopmentPhase: Boolean
IPsegregationSystem: String
IPproductionSystem: String
Features: Array<String>
RecordsRequired: Integer

CONFIGURE PRODUCTION SYSTEM
IsDevelopmentPhase: Boolean
Records: Array<String>
IPsegregationSystem: String
IPpreparationSystem: String
IsDevelopmentPhase: Boolean
Features: Array<String>
PSDbandsLowerBound: Double[4]
PSDbandsUpperBound: Double[4]
IPproductionSystem: String

CONFIGURE SEGRAGATION SYSTEM
Records: Array<String>
IPmessagingSystem: String
IPdevelopmentSystem: String
IPpreparationSystem: String
SessionsToCollect: Integer
BalancingTolerance: Double

CONFIGURE EVALUATION SYSTEM
PatientsInfo: Array<String>
IPmessagingSystem: String
IPdevelopmentSystem: String
IPproductionSystem: String
NumSessionEvaluation: Integer
TotalErrorsThreshold: Double
ConsecutiveErrorsThreshold: Double

CONFIGURE CLIENT-SIDE SYSTEM
IngestionSystemIP: String
ProductoinSystemIP: String
ActivitiesList: ArrayList<String>
EnvironmentsList: ArrayList<String>

Prepare Session (Lucchesi)

BRAIN RECORD
UUID: String
Data: EEGData

ENVIRONMENT RECORD
UUID: String
EnvironmentDescription: String

ACTIVITY RECORD
UUID: String
ActivityDescription: String

STORE RECORD PARAMETERS
MinRecordsForSession: Unsigned nt

RAW SESSION
UUID: String
BrainRecord: ArrayList<BrainRecord>
EnvironmentRecord: ArrayList<EnvironmentRecord>
ActivityRecord: ArrayList<ActivityRecord>
Label: String {optional}

MISSING SAMPLES PARAMETERS
MaxMissingSamplesAllowed: unsigned int

LABELS FOR EVALUATION SYSTEM
Labels: ArrayList<Enum (Move, TurnLeft, TurnRight>

FEATURES
SelectedFeatures: ArrayList<String>

PREPARED SESSION
UUID: Unsigned Integer
PSDAlpha: Double
PSDBeta: Double
PSDDelta: Double
PSDTheta: Double
Activity: Enum (shopping, sport, cooking, relax, gaming)
Environment: Enum ((slippery, plain, slope, house, track)
Label: Enum (Move, TurnLeft, TurnRight)

Generate Learning Sets (Crispino)

PREPARED SESSION
UUID: Unsigned Integer
PSDAlpha: Double
PSDBeta: Double
PSDDelta: Double
PSDTheta: Double
Activity: Enum (shopping, sport, cooking, relax, gaming)
Environment: Enum ((slippery, plain, slope, house, track)
Label: Enum (Move, TurnLeft, TurnRight)

SESSIONS STORE
MinSessions: Unsigned Integer

BALANCE REPORT
MinSessions: Unsigned Integer
BalancingTolerance: Double

UNSATISFIED BALANCE MESSAGE
RequestedMove: Unsigned Integer
RequestedTurnLeft: Unsigned Integer
RequestedTurnRight: Unsigned Integer

UNSATISFIED COVERAGE MESSAGE
Activity: Enum (shopping, sport, cooking, relax, gaming)
Environment: Enum ((slippery, plain, slope, house, track)
RequestedPSDAlpha: Unsigned Integer
RequestedPSDBeta: Unsigned Integer
RequestedPSDDelta: Unsigned Integer
RequestedPSDTheta: Unsigned Integer
RequestedActivity: ArrayList<Unsigned Integer>
RequestedEnvironment: ArrayList<Unsigned Integer>

LEARNING SET
UUID: Unsigned Integer
PSDAlpha: Double
PSDBeta: Double
PSDDelta: Double
PSDTheta: Double
Activity: Enum (shopping, sport, cooking, relax, gaming)
Environment: Enum ((slippery, plain, slope, house, track)
Label: Enum (Move, TurnLeft, TurnRight)

Develop Classifier (Arduini)

LEARNING SET
UUID: Unsigned Integer
PSDAlpha: Double
PSDBeta: Double
PSDDelta: Double
PSDTheta: Double
Activity: Enum (shopping, sport, cooking, relax, gaming)
Environment: Enum ((slippery, plain, slope, house, track)
Label: Enum (Move, TurnLeft, TurnRight)

HYPERPARAMETERS
MinLayers: Unsigned integer
StepLayers: Unsigned Integer
MaxLayers: Unsigned Integer
MinNeuronsPerLayer: Unsigned Integer
StepNeuronsPerLayer: Unsigned integer
MiaxNeuronsPerLayer: Unsigned Integer
NumberOfIterations: Unsigned Integer

LEARNING REPORT
LearningPlot: Map<Double, Double>

VALIDATION REPORT
OverfittingTolerance: Double
BestFiveModels: ArrayList<ModelValidationInformation>
IsClassifierValid: ArrayList<Boolean>

MODEL VALIDATION INFORMATION
Layers: Unsigned Integer
NeuronsPerLayer: Unsigned Integer
TrainingMSE: Double
ValidationMSE: Double
OverfittingTolerance: Double
IsClassifierValid: Boolean

TEST REPORT
Layers: Unsigned Integer
NeuronsPerLayer: Unsigned Integer
TrainingMSE: Double
ValidationMSE: Double
OverfittingTolerance: Double
IsClassifierValid: Boolean

MODEL
Layers: Unsigned Integer
NeuronsPerLayer: Unsigned Integer
Weights: ArrayList<Double>

Classify Session (Speciale)

PREPARED SESSION
UUID: Unsigned Integer
PSDAlpha: Double
PSDBeta: Double
PSDDelta: Double
PSDTheta: Double
Activity: Enum (shopping, sport, cooking, relax, gaming)
Environment: Enum ((slippery, plain, slope, house, track)

CLASSIFICATION
IsEvaluationPhase: Boolean
Label: Enum (Move, TurnLeft, TurnRight)

Evaluate Classifier (Speciale)

LABEL RETENTION PARAMS
MinRecordThreshold: Unsigned Integer
LabelsToRemove: Array<Integer>
RetentionPolicy: Enum

EVAL REPORT RESULTS
IsCorrect: Boolean
ErrorCount: Unsigned Integer
ConsecutiveErrorCount: Unsigned Integer
PerformanceScore: Float
Timestamp: DateTime

LABEL COMPARISON BUFFER
BufferID: Unsigned Integer
TargetLabel: Enum (Move, TurnLeft, TurnRight)
PredictedLabel: Enum (Move, TurnLeft, TurnRight)
ClassifierIdentifier: Unsigned Integer
ComparisonTimestamp: DateTime
ConfidenceScore: Float
ErrorFlag: Boolean

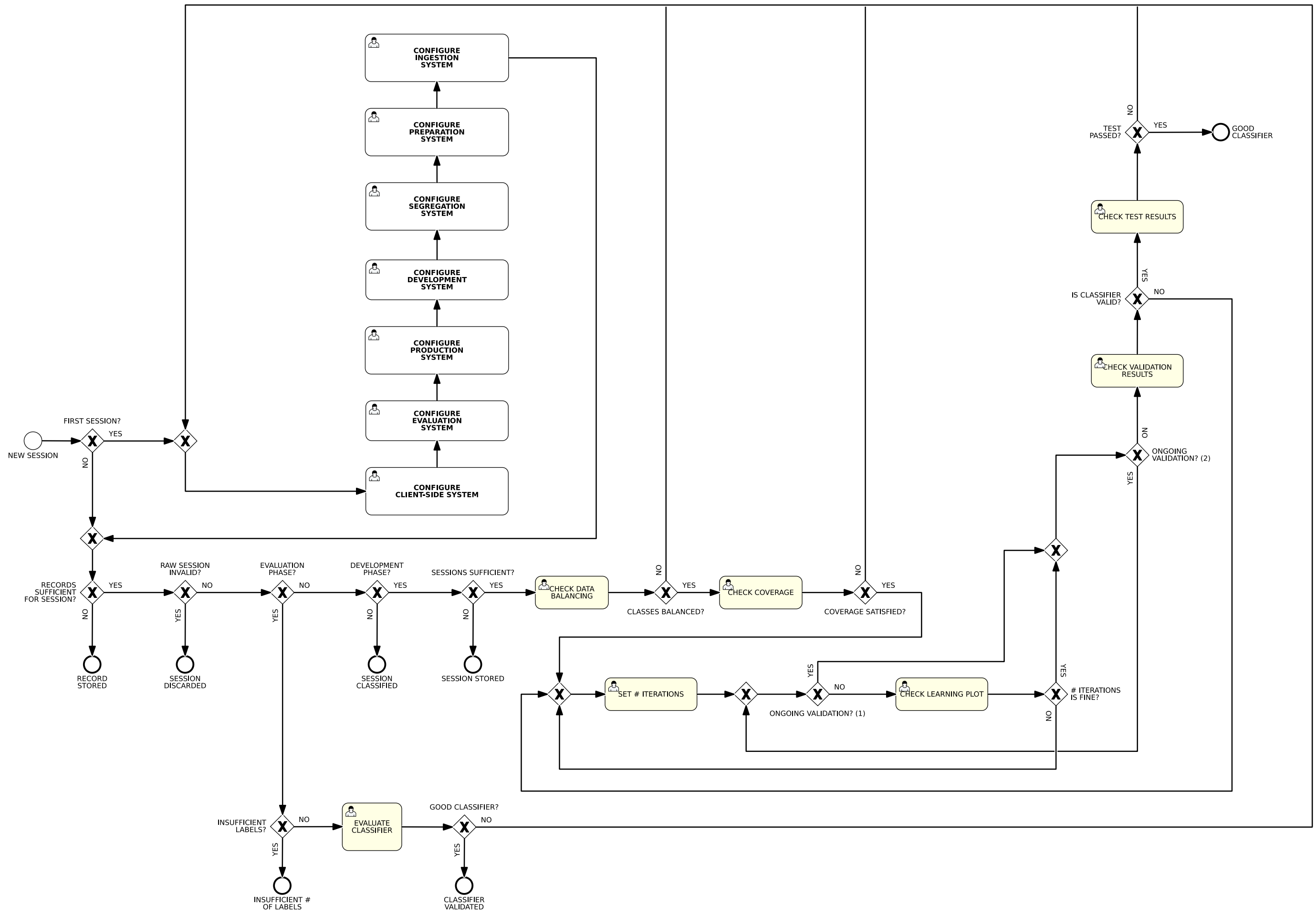
EVAL REPORT PARAMS
TotalErrorThreshold: Unsigned Integer
ConsecutiveErrorThreshold: Unsigned Integer
ClassifierIdentifier: Unsigned Integer
GeneralizationTolerance: Float
DatasetSize: Unsigned Integer

PREPARED SESSION
UUID: Unsigned Integer
PSDAlpha: Double
PSDBeta: Double
PSDDelta: Double
PSDTheta: Double
Activity: Enum (shopping, sport, cooking, relax, gaming)
Environment: Enum ((slippery, plain, slope, house, track)
Label: Enum (Move, TurnLeft, TurnRight)

ERROR ANALYSIS
ErrorType: Enum (Misclassification, Ambiguity, DataError)
ErrorDescription: String
ImpactScore: Float
TimeStamp: DateTime

AS-IS simulation (Lucchesi, Speciale)

Collapsed Workflow



Token Meaning

Although in the BPMN models created for the various internal factory processes, the first event is the receipt of records, it was decided to assign the tokens (used during the simulation phase) the meaning of 'session' rather than 'record'. This choice was made because BIMP has a maximum of 10k tokens, and using 'records' as their meaning would have been insufficient to adequately simulate the entire workflow. In fact, it would have been equivalent to working with only a quarter of the tokens.

By taking this approach, while also preserving all the gateways in the individual BPMN diagrams, we had to adjust the percentages of the initial gateways in the diagrams accordingly.

Since the goal is to simulate the development of 5 classifiers, and BIMP only allows 10k tokens at a time for simulation, we were forced to use a reduced number of sessions (and therefore tokens) for the simulation. With a maximum of 2000 sessions available per classifier, we allocated them as follows: 500 for development, 50 for evaluation and 1450 for production.

Gateway percentage table

EXCLUSIVE GATEWAY	TRUE (%)	FALSE (%)	COMMENT
FIRST SESSION?	0.05	99.95	We have in total 500 sessions for development and 50 for evaluation for each classifier. We have 1450 sessions for production. $$(500 + 50 + 1450) \times 5 = 10000$$ $$\text{Probability of first session} = \frac{5}{10000} \times 100 = 0.05$$
RECORDS SUFFICIENT FOR SESSION?	100	0	Since we are considering sessions as tokens, we assume that a session always has enough records to proceed. If instead we consider records, the number of 'final record' of development and evaluation sessions is 550/4. The number of 'final instances' of production sessions is (1450*5)/3. $$\frac{\frac{550}{4} + \frac{(1450 \times 5)}{3}}{10000} \times 100 = 25.5$$
RAW SESSION INVALID?	5	95	We assume that only 5% of the time the session is invalid due to having too many missing records
EVALUATION PHASE?	2.5	97.5	$$\frac{50 \times 5}{10000} \times 100 = 2.5$$
DEVELOPMENT PHASE?	25	75	$$\frac{500 \times 5}{10000} \times 100 = 25$$
SESSIONS SUFFICIENT?	0.2	99.8	$$\frac{1}{500} \times 100 = 0.2$$
CLASSES BALANCED?	80	20	Assume that the patients follow specific tutorials to help them deliver balanced data
COVERAGE SATISFIED?	33	66	Assume that 2 out of 3 iterations will not have sufficient coverage
ONGOING VALIDATION? (1)	96 // 75	4 // 52	Assume we are doing 25 iterations of validations // Assume we check 9 possible classifiers for our 5 patients. Also, 2/3 times we check the learning plot $$\frac{8 \times 5 + 1 \times 5}{5 \times 9 + 3 \times 5} = \frac{45}{60} = 75\%$$
# ITERATIONS IS FINE?	50	50	Assume that 50% of the time the number of iterations is correct (because the number of iterations is difficult to evaluate a priori)
ONGOING VALIDATION? (2)	96	4	Assume we are doing 25 iterations of validations
IS CLASSIFIER VALID?	95	5	Assume only 5% of the time the classifier is not valid
TEST PASSED?	99	1	Assume only 1% of the time the test is not passed
INSUFFICIENT LABELS?	98	2	We have 50 evaluation sessions, so 1/50 times we have sufficient labels
GOOD CLASSIFIER?	86	14	Assume that 6/7 times the classifier is considered good

AS-IS: stats

Scenario Statistics

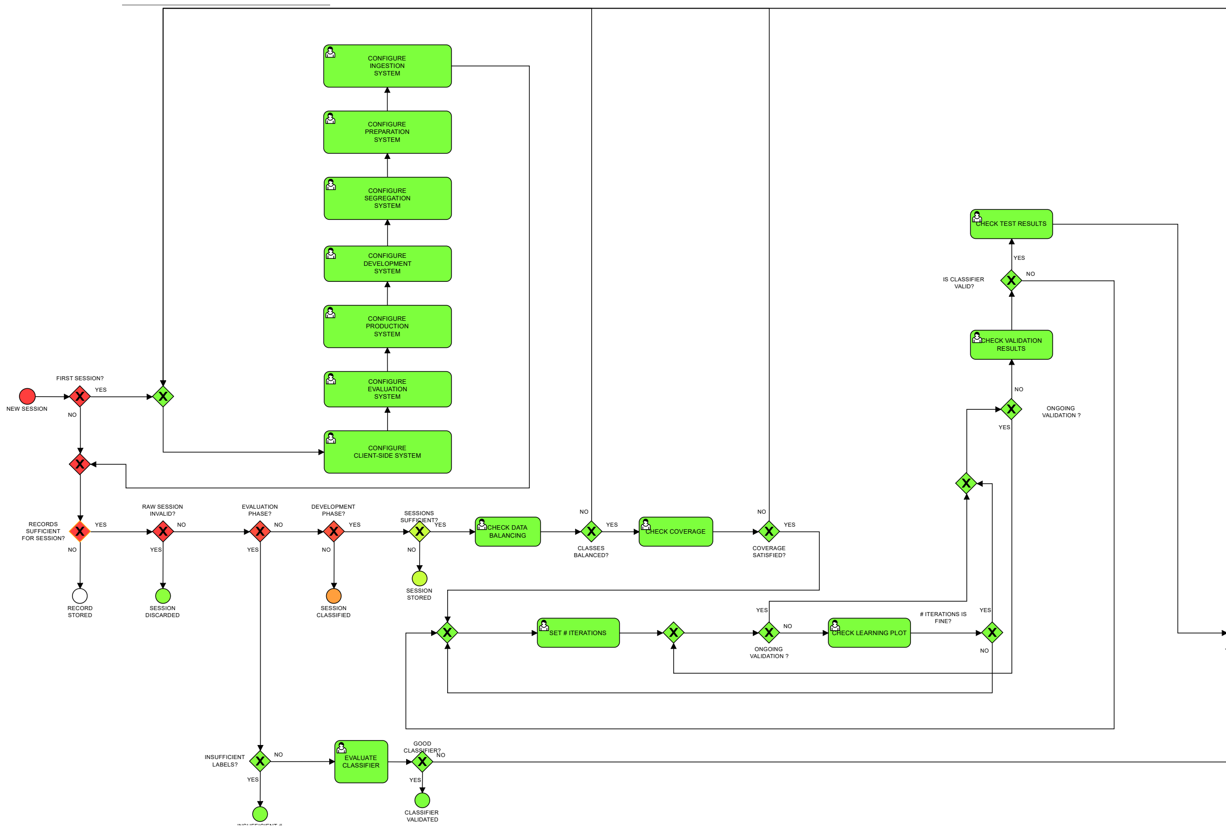
	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	0 seconds	10.4 minutes	0.8 seconds
Process instance cycle times excluding off-timetable hours	0 seconds	10.4 minutes	0.8 seconds
Process instance costs	0 EUR	0 EUR	0 EUR

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 COVERAGE	10	0 s	0 s	0 s	1.4 m	1.5 m	1.5 m	0 s	0 s	0 s	0	0	0	0	0	0
CHECK DATA BALANCING	14	0 s	0 s	0 s	35.5 s	37.5 s	39.2 s	0 s	0 s	0 s	0	0	0	0	0	0
CHECK LEARNING PLOT	26	0 s	0 s	0 s	10.5 s	11.1 s	11.5 s	0 s	0 s	0 s	0	0	0	0	0	0
CHECK TEST RESULTS	6	0 s	0 s	0 s	10.1 s	10.5 s	10.8 s	0 s	0 s	0 s	0	0	0	0	0	0
CHECK VALIDATION RESULTS	6	0 s	0 s	0 s	37.5 s	38.3 s	39.5 s	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
CLIENT-SIDE SYSTEM	11	0 s	0 s	0 s	19.3 s	20.1 s	20.9 s	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
DEVELOPMENT
SYSTEM	11	0 s	0 s	0 s	3.2 m	3.3 m	3.5 m	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
EVALUATION
SYSTEM	11	0 s	0 s	0 s	40.5 s	42.1 s	43.3 s	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
INGESTION
SYSTEM	11	0 s	0 s	0 s	59.3 s	1 m	1.1 m	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
PREPARATION
SYSTEM	11	0 s	0 s	0 s	1.5 m	1.6 m	1.7 m	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
PRODUCTION
SYSTEM	11	0 s	0 s	0 s	34.2 s	35.3 s	36.6 s	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
SEGREGATION
SYSTEM	11	0 s	0 s	0 s	35.2 s	37 s	38.7 s	0 s	0 s	0 s	0	0	0	0	0	0
EVALUATE
CLASSIFIER	5	0 s	0 s	0 s	27.3 s	27.8 s	28.3 s	0 s	0 s	0 s	0	0	0	0	0	0
SET # ITERATIONS	19	0 s	0 s	0 s	20 s	21.1 s	22 s	0 s	0 s	0 s	0	0	0	0	0	0

Heatmap

Heatmap based on Counts

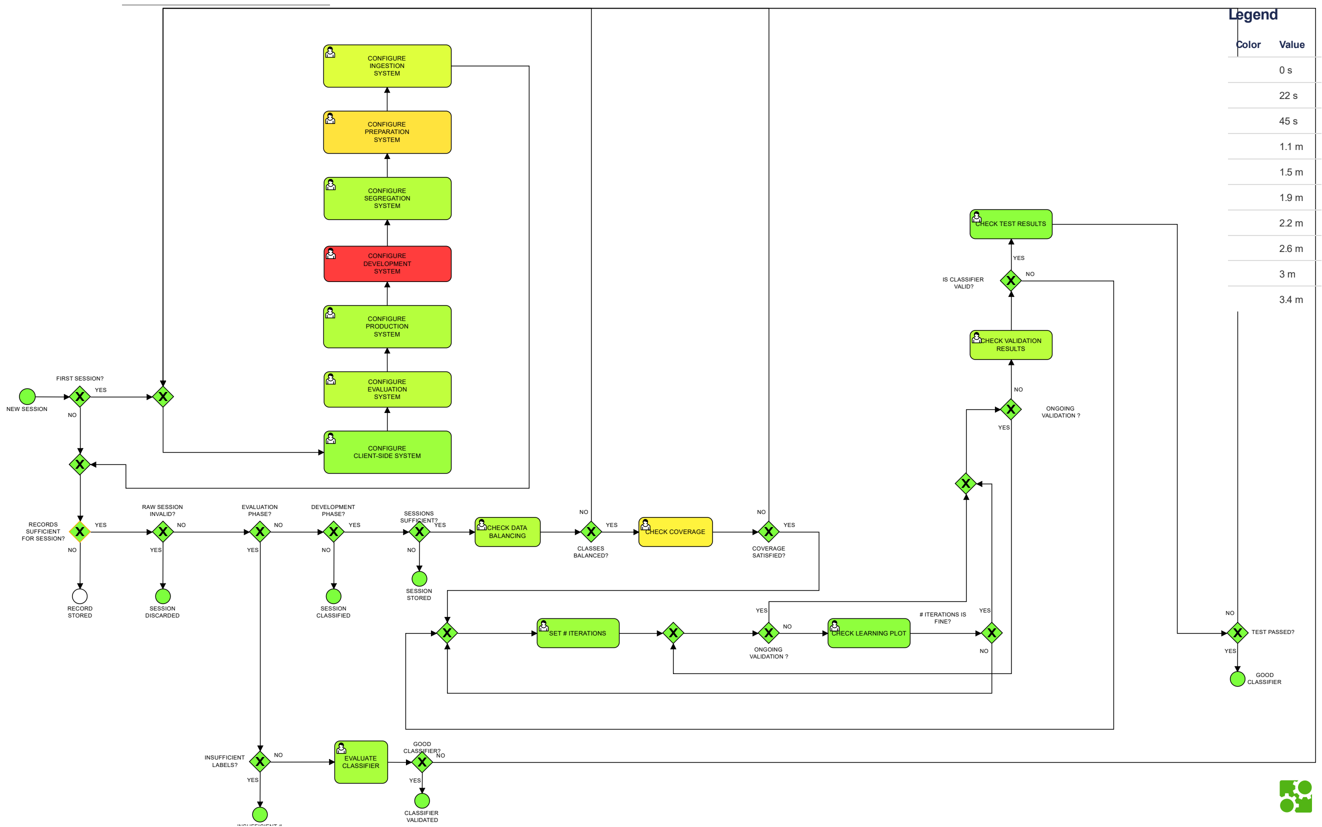


Legend

Color	Value
	0
	1112
	2224
	3336
	4448
	5561
	6673
	7785
	8897
	10009

Heatmap

Heatmap based on Durations



TO-BE simulation (Arduini, Crispino)

In this section, we will analyze the various improvements made to the AS-IS model and how these enhance the performance of our factory. Specifically, based on certain fixed assumptions, we streamlined some processes along the production line.
We will then examine the results achieved with the updated model, comparing them to the original AS-IS model.

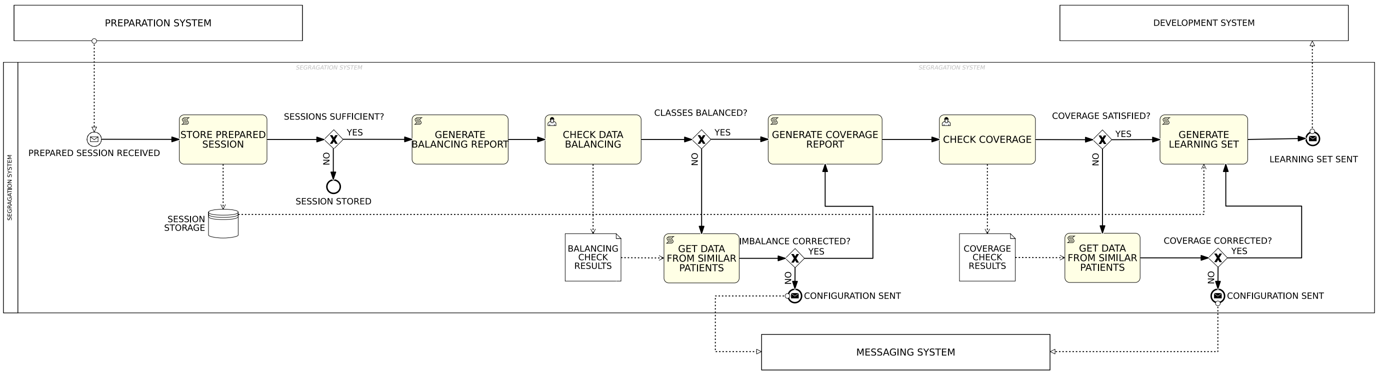
The **assumption** underlying the three improvements we'll discuss is that our clients fill out a form collecting basic information such as age, medical history/illnesses and habitual activities. This information is used to create a profile for each client. These profiles are then processed by a research center, which uses data mining techniques to perform clustering. The goal is to identify groups of clients with similar characteristics.

The results of this clustering are later utilized in our factory to leverage pre-existing information about a specific client category when developing a new classifier for a new client belonging to that category. Specifically, for each client cluster, <u>session data</u>, <u>"best" hyperparameters</u> and the <u>number of iterations</u> is retained.

Improvements

Handoff level improvement – Data retrieved from similar patients (Arduini)

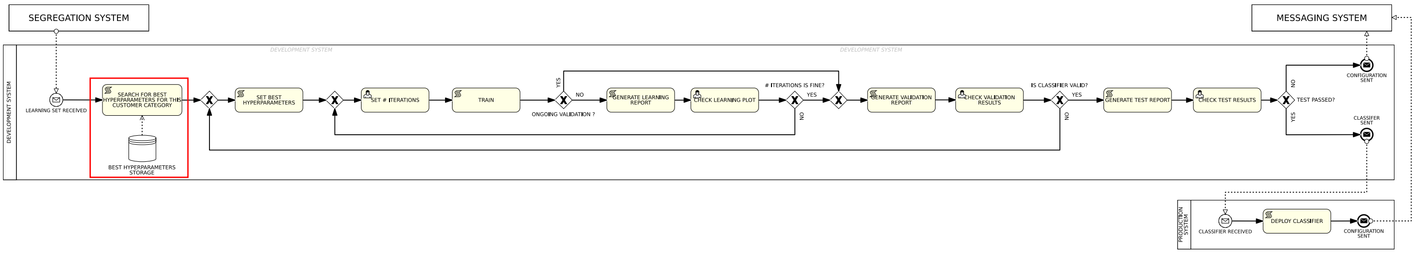
As a handoff improvement, we decided to leverage the clustering of similar patients in the following way: In case there is data imbalance and/or missing data coverage, we check in our storage if there is enough data to cover the imbalance/missing coverage, and if there is, we can simply move forward to the generation of the learning set. This will dramatically reduce the number of times that a message will need to be sent through the Messaging System to reconfigure the Segregation System, leading to a great reduction in human tasks performed.



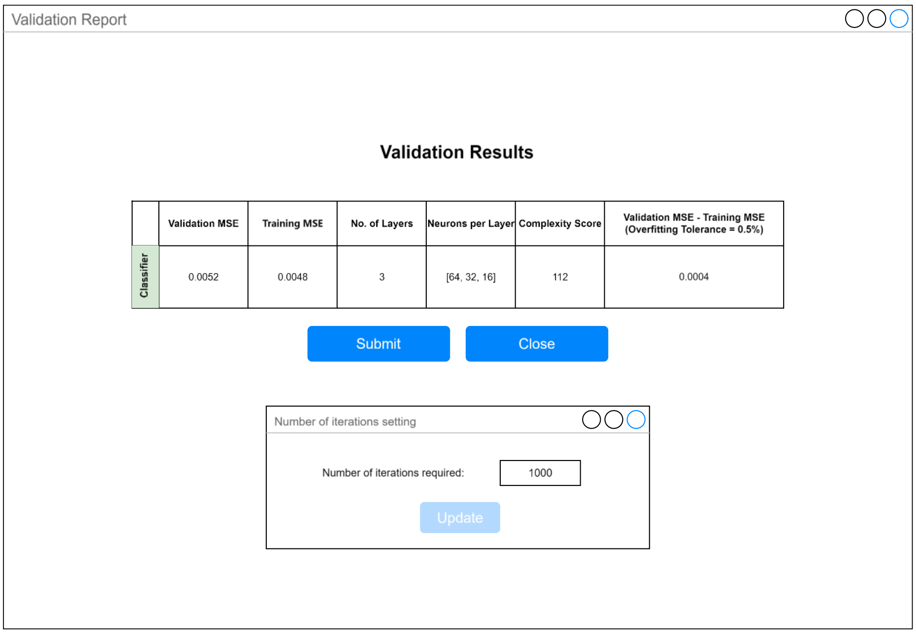
Service level improvement – Best hyperparameters' are known (Arduini)

At the service level, we streamlined the workflow within a single system, specifically the development system. This was achieved based on the assumption that 'best hyperparameters' are already available for developing new classifiers, making the grid search for finding optimal hyperparameters no longer necessary.

The **changes** introduced include a storage lookup to retrieve these **hyperparameters**. Additionally, the task <u>*'set hyperparameters'* was removed</u>, and the second <u>gateway *'ongoing validation?'* is no longer required</u>, as it is assumed that the hyperparameters will be correct from the start. Consequently, there is no need to retrain in search of better hyperparameters.
The **number of iterations** for an initial test is also retrieved from storage; however, unlike the hyperparameters, these values are not always optimal for a new classifier, though they do provide a good starting point. For this reason, the workflow <u>retains the *'check learning plot'*</u> step to verify whether the value is appropriate or needs adjustment. As a result, the success rates of the <u>gateway *'# iterations is fine?'* have now increased</u>.



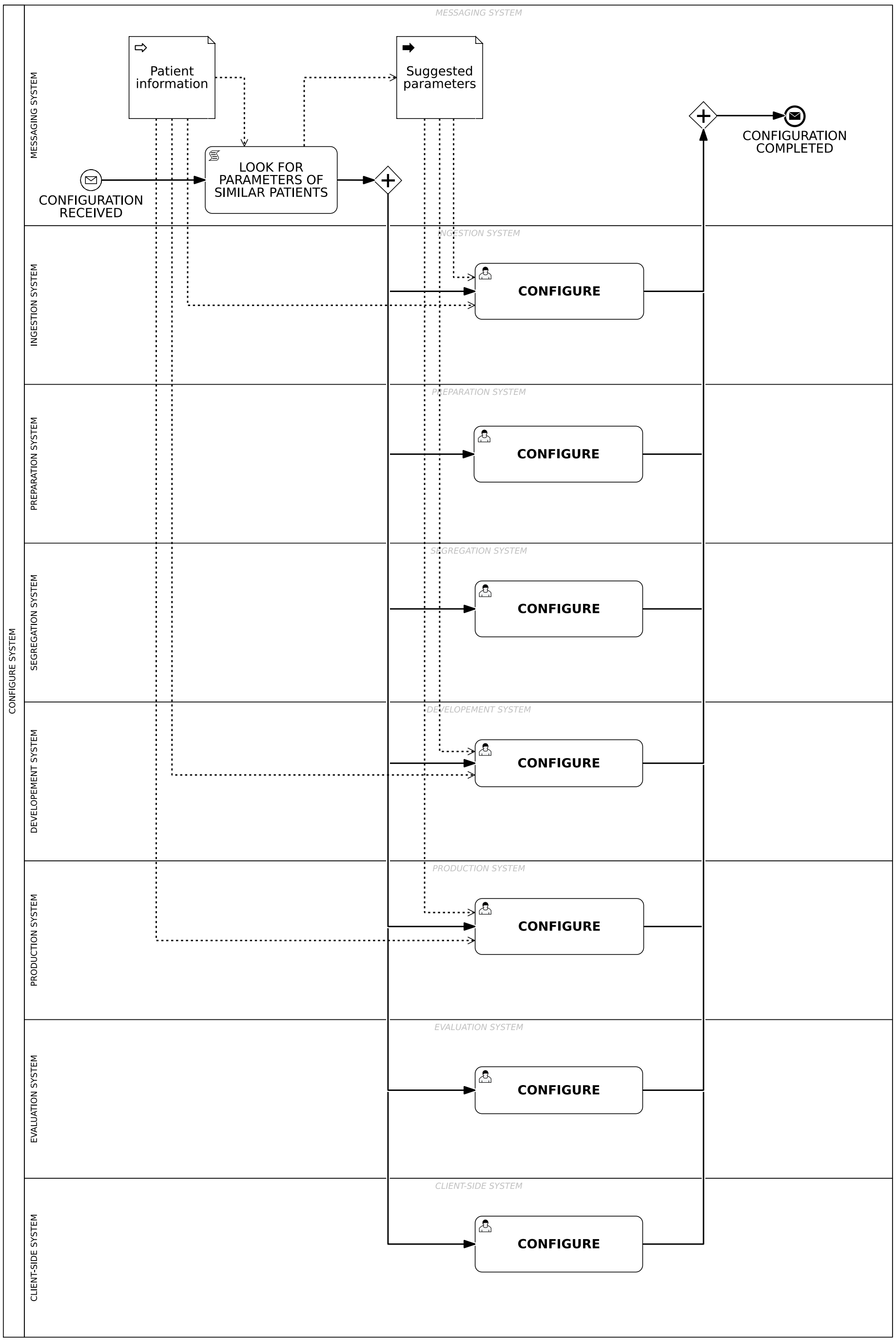
Since the hyperparameters will be correct from the start, the task of '*Check validation results*' becomes less costly, as it will now involve evaluating the performance of only one classifier instead of five, as was the case in the AS-IS version.



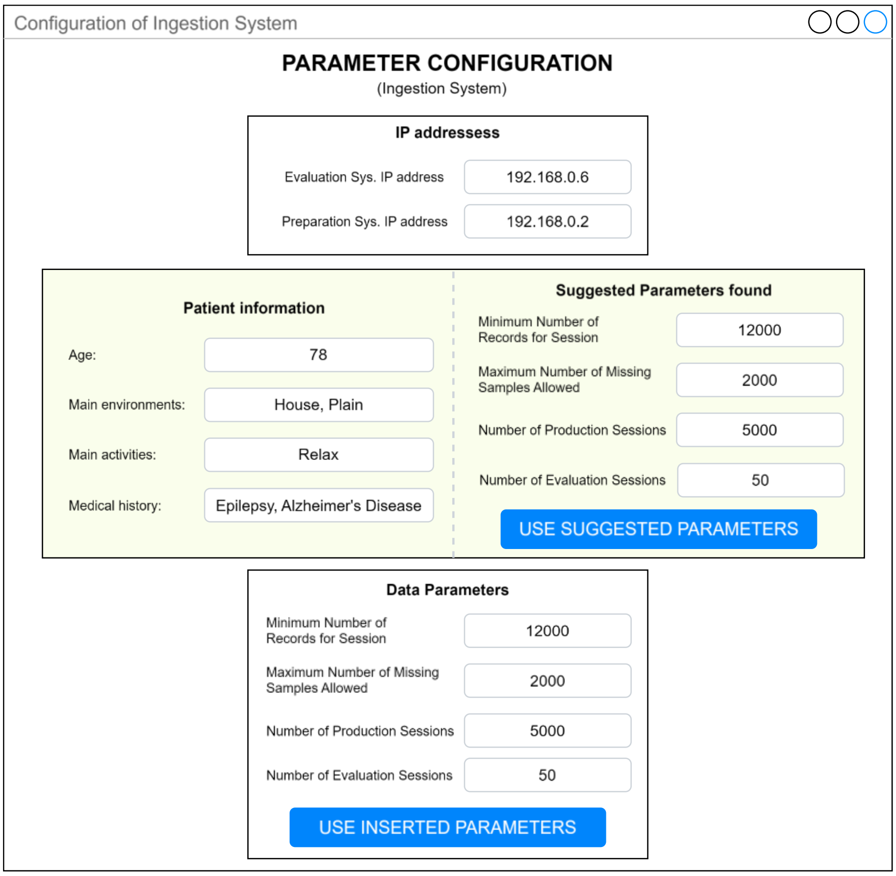
STEP	COST CALCULATION	SC
1) USER opens the validation result form	1x1x1.75	1,75
2) SYSTEM displays the report for the resulting classifier trained with the best hyperparameters, highlighting the classifier's ID in green if it meets the tolerance criteria or in red if it does not		
3) USER checks the color of the classifier	1x2x1.75	3,5
4) IF color is red	0.02	
4.1) SYSTEM shows a warning message: "No selectable classifier. Check parameters." The "Number of iterations setting" tab opens and the "Update" button is enabled once the new value is inserted.		
4.2) USER updates the Number of iterations required	0.02x(4x3)x1.75	0,42
4.3) USER clicks on the "Update" button	0.02x1x1.75	0,035
5) ELSE (color is green)	0.98	
5.1) USER clicks on "Submit" to validate the classifier	0.98x1x1.75	1,715
5.2) SYSTEM shows a confirmation message: "Classifier validated successfully"		
6) USER closes the validation results form	1x1x1.75	1,715
	HUMAN TASK COST	8,715

Task level improvement - Streamlining Tasks with Known Best Hyperparameters (Crispino)

Starting from the assumption of clustering patients along with their information, instead of defining the number of layers and neurons based on specific rules, we can adopt parameters already used in a similar network within the same category. This adjustment reduces the cognitive effort required to determine these parameters, thereby lowering the overall cost of configuring the development system and ingestion system. Our hypothesis is that in 90% of cases, the same parameters from a comparable network will be reused, while in the remaining 10%, the parameters will be manually configured as was done in the AS-IS model.



Task: Ingestion System Configuration



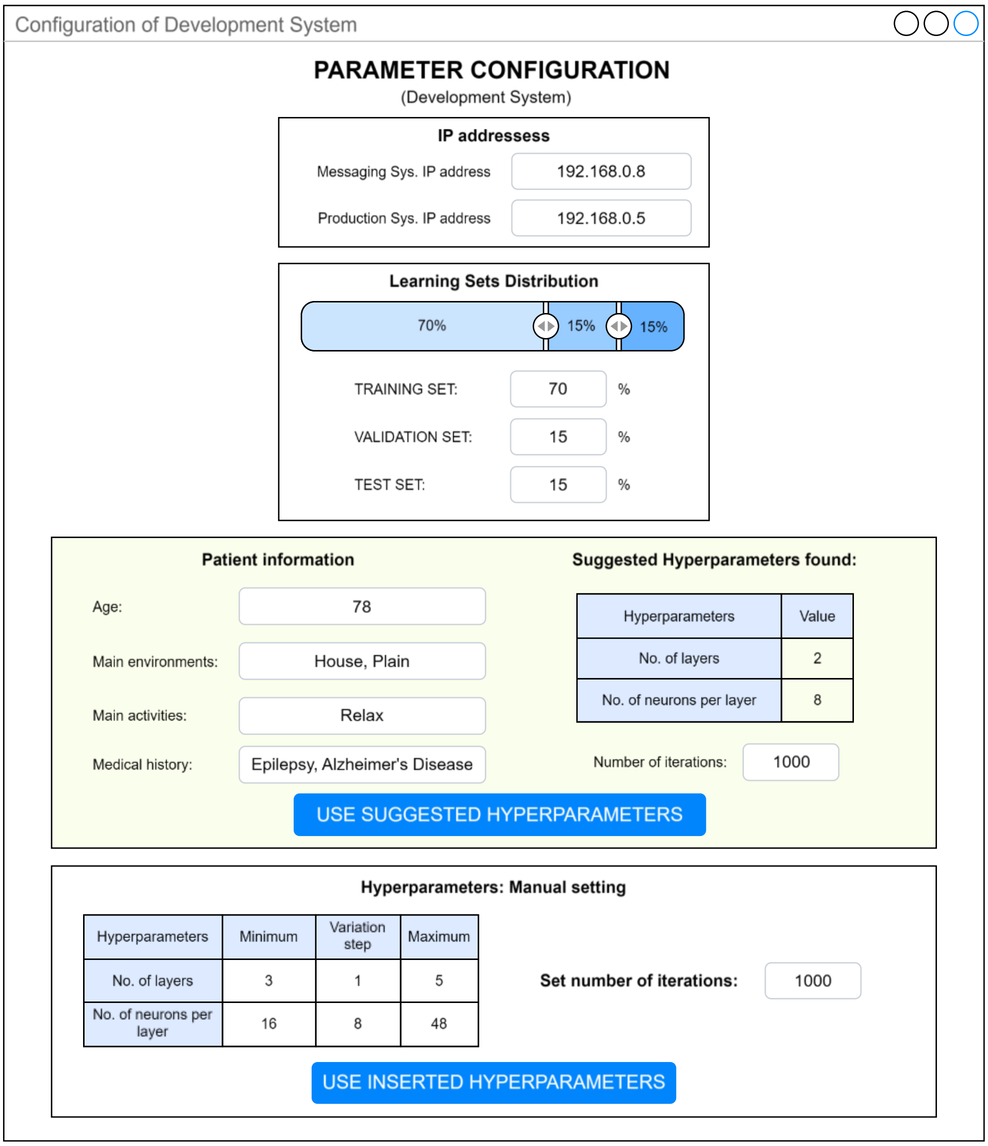
STEP	COST CALCULATION	SC
1) USER opens the configuration form for the Ingestion System	1x1x1	1
2) SYSTEM displays the configuration form		
3) FOR EACH IP address to set:	2	
3.1) USER looks at the IP address	2x(1+3)x1	8
3.2) USER enters the IP address	2x1x1	2
4) IF there are suggestions for the parameters	0.9	
4.1) USER selects USE SUGGESTED PARAMETERS	0.9x1x1x1	0,9
5) ELSE	0.1	
5.1) USER inserts a value for the minimum number of records to collect for a session	0.1x4 x3 x1	1,2
5.2) USER inserts a value for the maximum number of missing samples allowed	0.1x4 x3 x1	1,2
5.3) USER inserts a value for the number of production sessions	0.1x4 x3 x1	1,2
5.4) USER inserts a value for the number of evaluation sessions	0.1x4 x3 x1	1,2
5.5) USER selects USE INSERTED PARAMETERS	1x1x1	1
6) SYSTEM displays a confirmation		
7) USER closes the form	1x1x1	1
	HUMAN TASK COST	18,7

The cost shown in red (+3) indicates the IT technician's cost for assigning the IP address to the Evaluation System and the Preparation System.

The cost shown in green (x3) indicates that this is a multi-criteria choice:

- at element 5.2) the USER must consider: total number of samples, rate of lost samples in the client-side systems (helmet, environment, activity), rate of arrival of samples.
- at element 5.3) USER must consider: age of the patient, illnesses/medical history of patient, preferred activities of the patient

Task: Development System Configuration



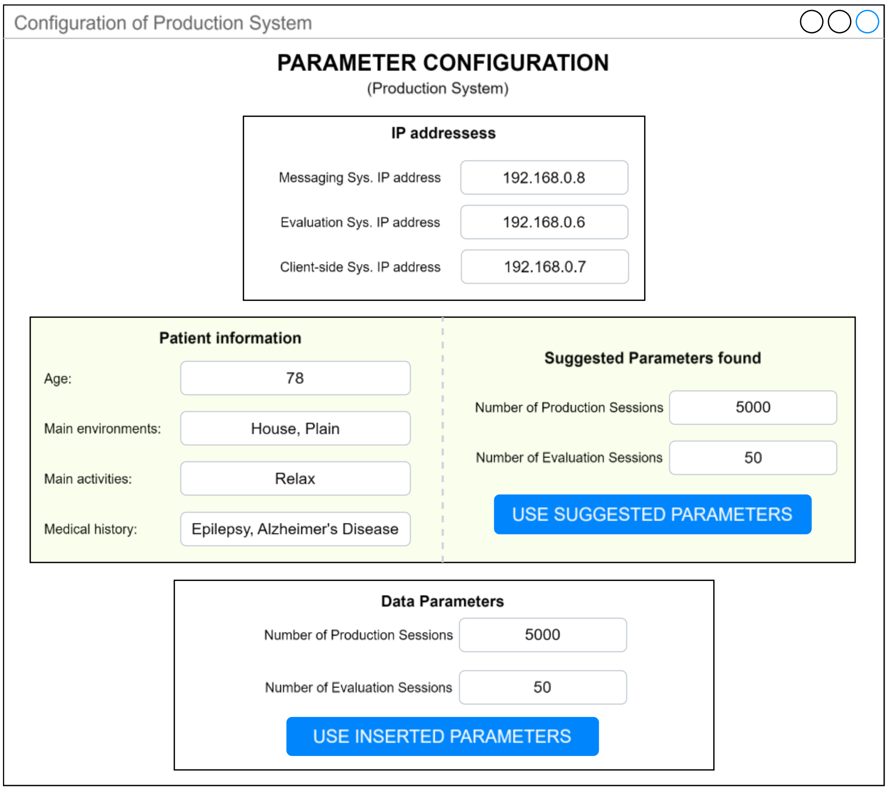
STEP	COST CALCULATION	SC
1) USER opens the configuration form for the Development System	1x1x1.75	1,75
2) SYSTEM shows the configuration form		
3) FOR EACH IP address to set:	2	
3.1) USER looks at the IP address	2x1x1.75	3,5
3.2) USER enters the IP address	2x1x1.75	3,5

4) USER enters the size of training set	1x4 x3 x1.75	21
5) USER enters the size of validation set	1x4 x3 x1.75	21
6) USER enters the size of test set	1x1x1.75	1,75
7) IF hyperparameters suggestions are available	0.9	
7.1) USER selects USE SUGGESTED HYPERPARAMETERS	0.9x1x1x1.75	1,575
8) ELSE	0.1	
8.1) FOR EACH value of the hyperparameters table	6	
8.1.2) USER sets the hyperparameters	0.1x6x4 x3 x1.75	12,6
8.2) USER sets the number of iterations	0.1x4 x3 x1.75	2,1
8.3) USER selects USE INSERTED HYPERPARAMETERS	0.1x1x1.75	0,175
9) SYSTEM shows a confirmation		
19) USER closes the configuration form	1x1x1.75	1,75
	HUMAN TASK COST	77,525

The cost shown in green (x3) indicates that this is a multi-criteria choice:

- at elements 4), 5) and 6) the USER must consider: general conventions, problem domain and number of sessions available. The choice of one of the three has cost 1 since it is automatically resized based on the previous two choices;
- at element 7.1.2), instead, the USER must consider: conventions based on domain knowledge, available computational power and previous experience with similar cases.
- at element 7.2), instead, the USER must consider: number of sessions in the training set, architecture of the prototype neural networks and previous experience with similar cases.

Task: Production System Configuration



STEP	COST CALCULATION	SC
1) USER opens the configuration form for the Production System	1x1x1	1
2) SYSTEM displays the configuration form		
3) FOR EACH IP address to set:	3	
3.1) IF setting the IP address of the Client-side system	0.33	
3.1.1) USER looks at the Client-side IP address	0.33x3x(1+3)x1	3,96
3.2) ELSE	0.67	
3.2.1) USER looks at the IP address	0.67x3x1x1	2,01
3.3) USER enters the IP address	3x1x1	3
4) IF suggested parameters are available	0.9	
4.1) USER selects USE SUGGESTED PARAMETERS	0.9x1x1x1	0,9
5) ELSE	0.1	
5.1) USER enters the number of production sessions	0.1x4 x3 x1	1,2
5.2) USER enters the number of evaluation sessions	0.1x4 x3 x1	1,2
5.3) USER selects USE INSERTED PARAMETERS	1x1x1	1
6) SYSTEM displays a confirmation		
7) USER closes the form	1x1x1	1
	HUMAN TASK COST	19,97

The cost shown in red (+3) indicates the IT technician's cost for assigning the IP address to the Client-side System.

The cost shown in green (x3) represents a multi-criteria choice. In this case, the USER must consider: the patient's age, medical history/illnesses, and the patient's habitual activities.

Collapsed Workflow

The biggest changes with respect to the AS-IS collapsed workflow are the following:

- There are now 2 new gateways: *'IMBALANCE CORRECTED?'* and *'COVERAGE CORRECTED?'*
  - These help us reduce the number of times that a token is sent back to configuration, since we assume the imbalance/coverage is corrected 95% of the time.
- The Development part of the workflow is now simplified:
  - there is no longer a gateway *'ONGOING VALIDATION (2)?'* that sends the token back through the workflow; this is because we now have immediate access to the best hyperparameters for a classifier, and thus we don't need to check multiple combinations of hyperparameters.

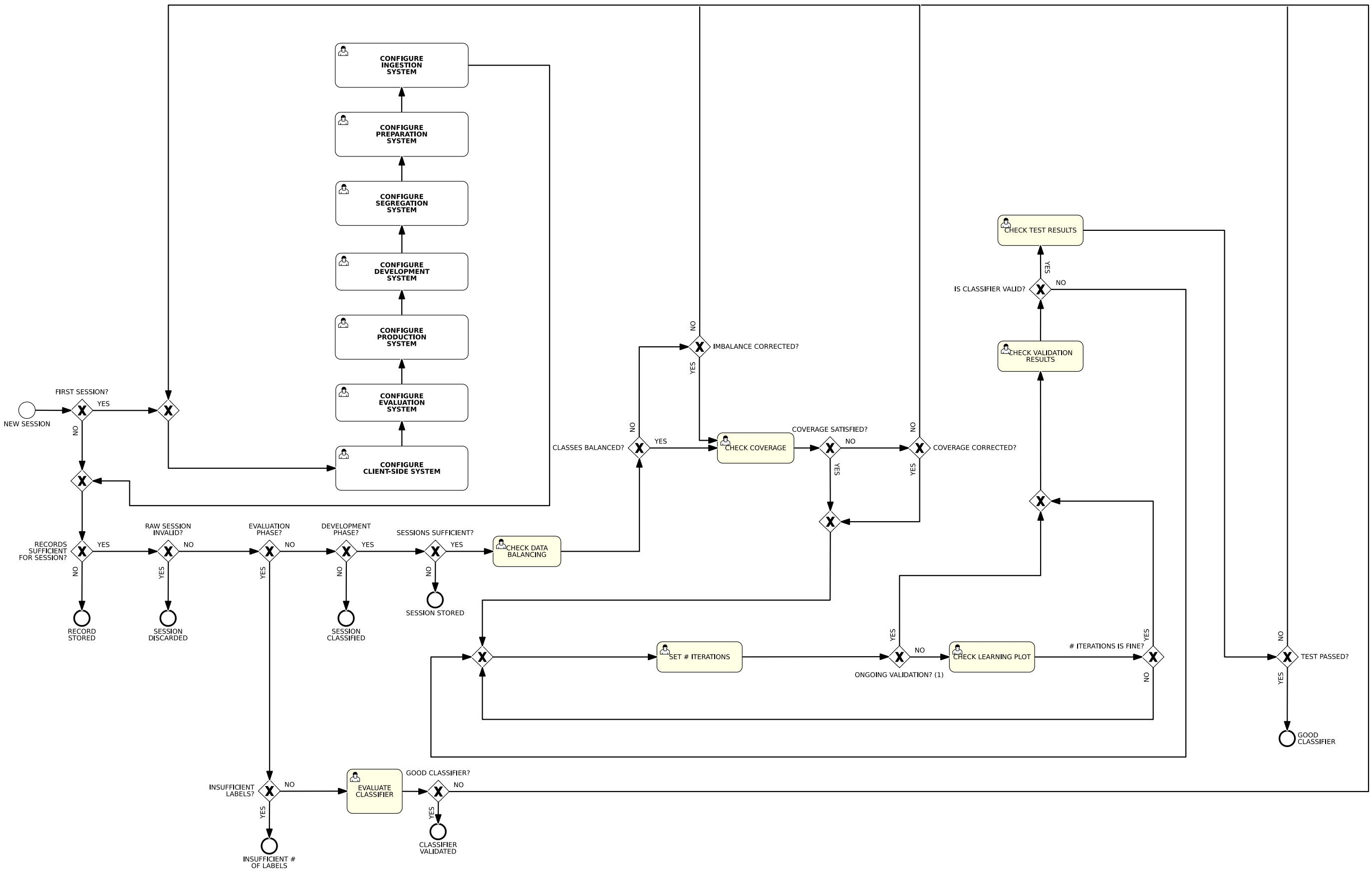


Table of updated gateway percentages

EXCLUSIVE GATEWAY	TRUE (%)	FALSE (%)	COMMENT
IMBALANCE CORRECTED?	95	5	We assume that the clustering of similar patients allows us to correct the class imbalance 95% of the time
COVERAGE CORRECTED?	95	5	As above, we assume that the clustering of similar patients allows us to correct the missing data coverage 95% of the time
ONGOING VALIDATION?	50	50	We do not need to check multiple possible hyperparameters for our classifiers, and we assume that 1 in 2 times we need to check the learning plot $$\frac{5 \times 1}{5 \times 2} = 50\%$$
# ITERATIONS IS FINE?	90	10	We assume that the # of iterations is now correct 90% of the time, thanks to information from similar patients

To-BE: stats

Scenario Statistics

	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	0 seconds	5.2 minutes	0.2 seconds
Process instance cycle times excluding off-timetable hours	0 seconds	5.2 minutes	0.2 seconds
Process instance costs	0 EUR	0 EUR	0 EUR

Activity Durations, Costs, Waiting times, Deviations from Thresholds

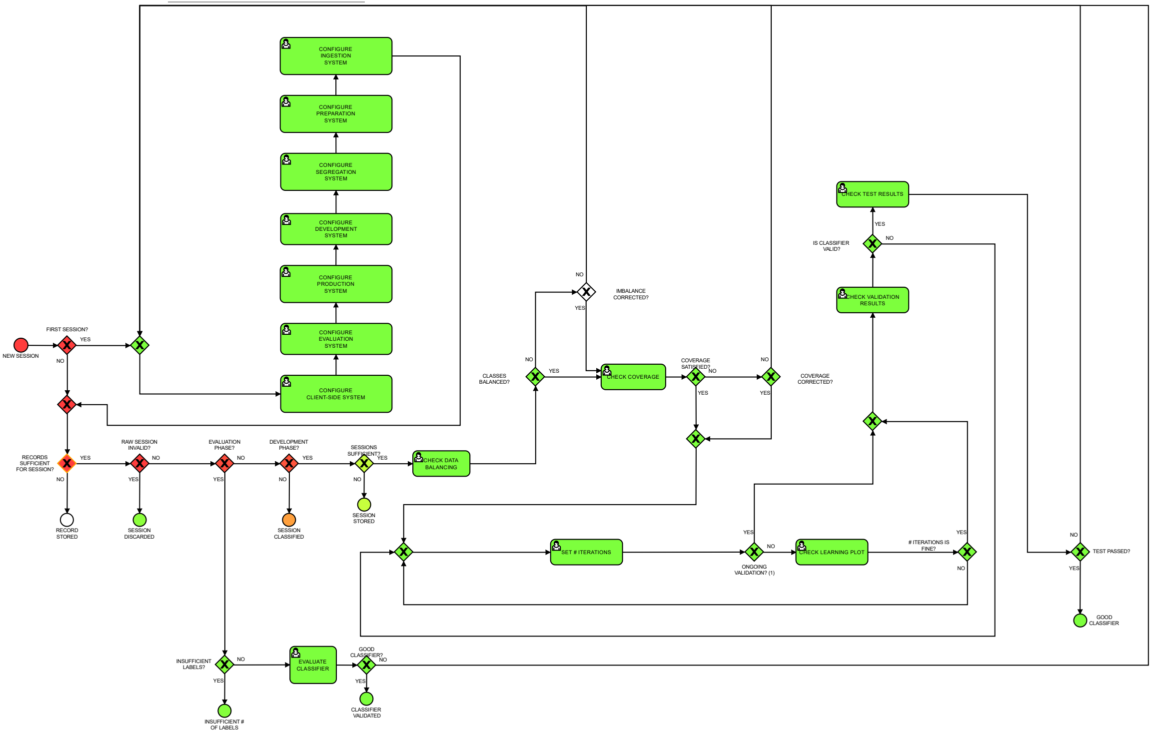
Name		**Waiting time**			**Duration**			**Duration over threshold**			**Cost**			**Cost over threshold**		
	Count	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
CHECK COVERAGE	7	0 s	0 s	0 s	1.4 m	1.5 m	1.5 m	0 s	0 s	0 s	0	0	0	0	0	0
CHECK DATA BALANCING	7	0 s	0 s	0 s	36.2 s	37.6 s	38.6 s	0 s	0 s	0 s	0	0	0	0	0	0
CHECK LEARNING PLOT	1	0 s	0 s	0 s	11.1 s	11.1 s	11.1 s	0 s	0 s	0 s	0	0	0	0	0	0
CHECK TEST RESULTS	7	0 s	0 s	0 s	10.5 s	10.7 s	11 s	0 s	0 s	0 s	0	0	0	0	0	0
CHECK VALIDATION RESULTS	8	0 s	0 s	0 s	4.8 s	5 s	5.2 s	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
CLIENT-SIDE SYSTEM	1	0 s	0 s	0 s	20.2 s	20.2 s	20.2 s	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
DEVELOPMENT
SYSTEM	1	0 s	0 s	0 s	1.3 m	1.3 m	1.3 m	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
EVALUATION
SYSTEM	1	0 s	0 s	0 s	40.7 s	40.7 s	40.7 s	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
INGESTION
SYSTEM	1	0 s	0 s	0 s	18.8 s	18.8 s	18.8 s	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
PREPARATION
SYSTEM	1	0 s	0 s	0 s	1.5 m	1.5 m	1.5 m	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
PRODUCTION
SYSTEM	1	0 s	0 s	0 s	20.1 s	20.1 s	20.1 s	0 s	0 s	0 s	0	0	0	0	0	0
CONFIGURE
SEGREGATION
SYSTEM	1	0 s	0 s	0 s	38.8 s	38.8 s	38.8 s	0 s	0 s	0 s	0	0	0	0	0	0
EVALUATE
CLASSIFIER	5	0 s	0 s	0 s	25.9 s	26.9 s	27.8 s	0 s	0 s	0 s	0	0	0	0	0	0
SET # ITERATIONS	9	0 s	0 s	0 s	20.3 s	21.2 s	22 s	0 s	0 s	0 s	0	0	0	0	0	0

Heatmap of counts

07/12/24, 11:15 bimp.cs.ut.ee/simulator/heatmapviewer/

Heatmap

Heatmap based on Counts



Legend

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

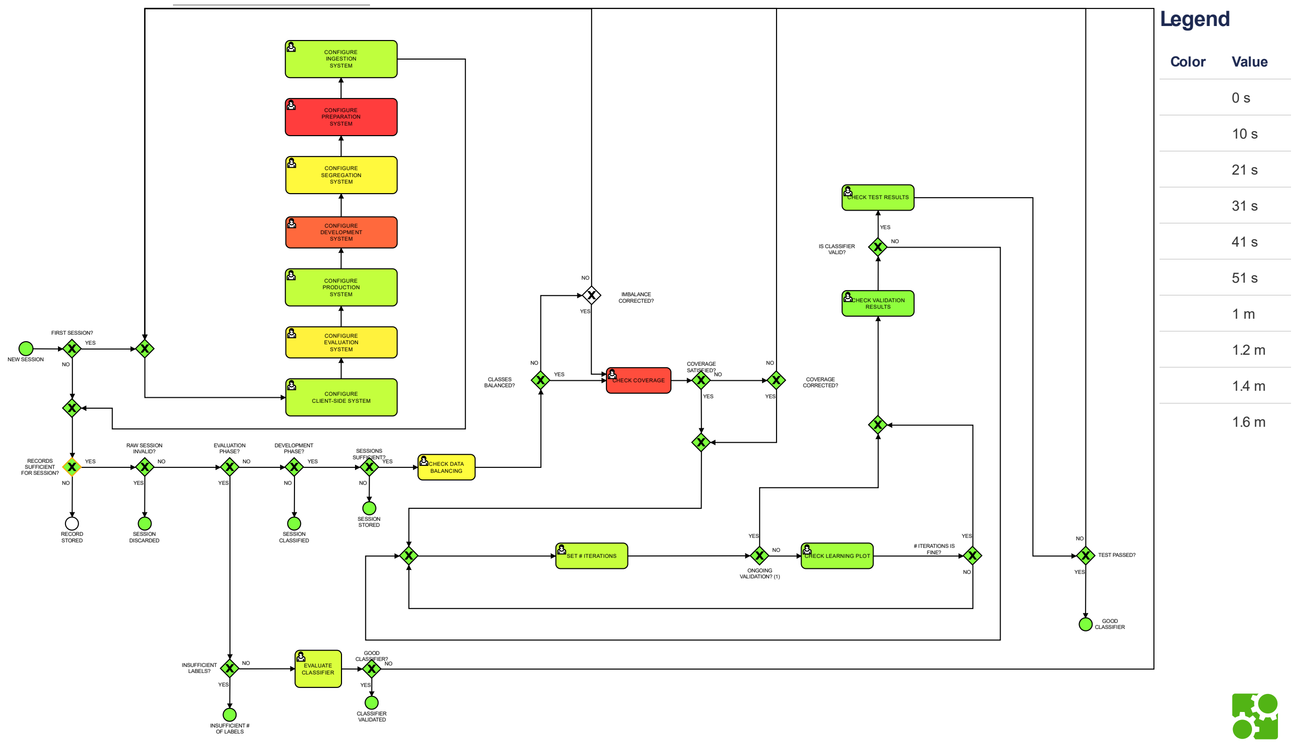
https://bimp.cs.ut.ee/simulator/heatmapViewer/ 1/1

Heatmap of durations

07/12/24, 11:15 bimp.cs.ut.ee/simulator/heatmapviewer/

Heatmap

Heatmap based on Durations



https://bimp.cs.ut.ee/simulator/heatmapViewer/ 1/1

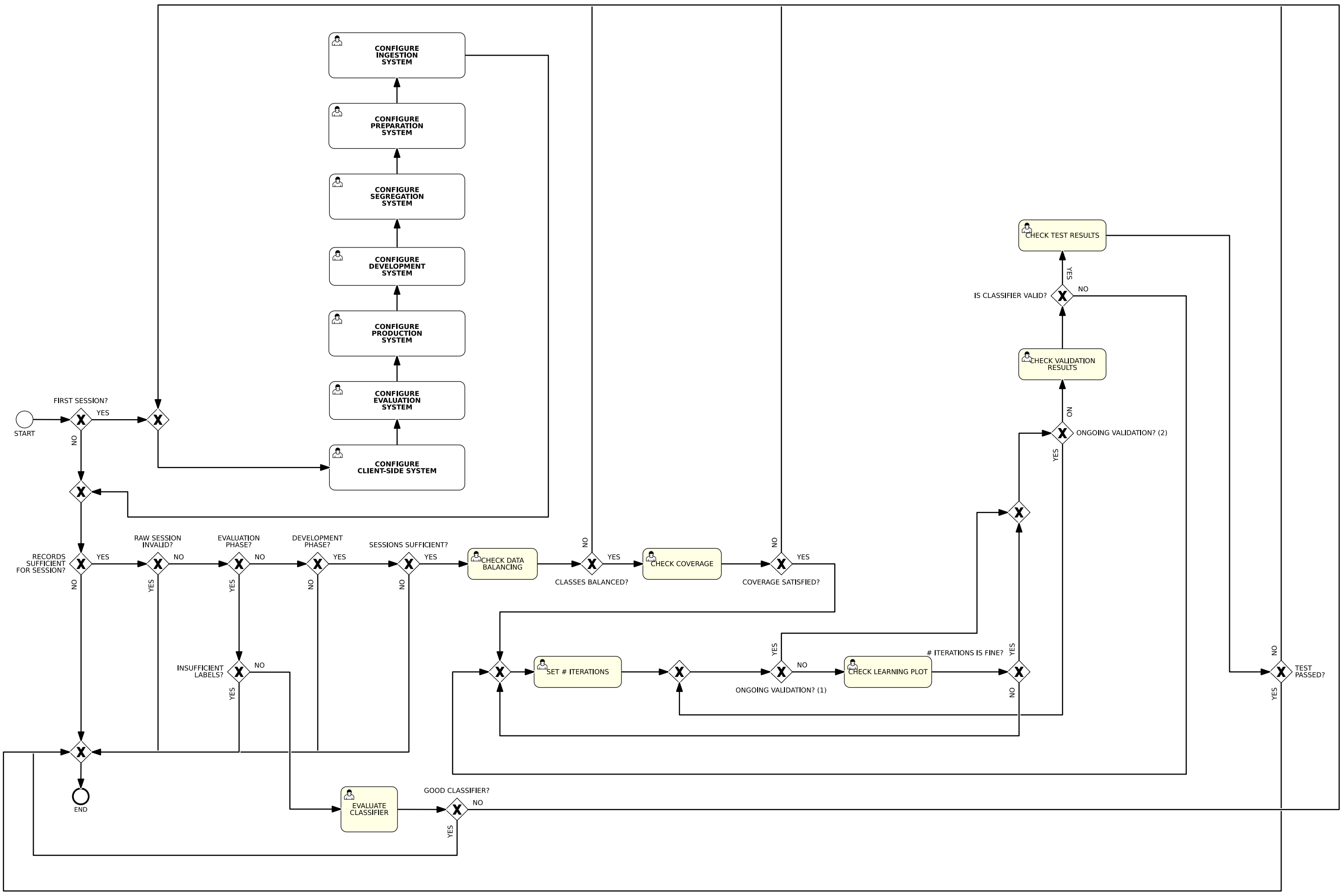
Comparative Discussion

As we can see, the maximum lifetime of a token in the workflow decreased significantly, as did the average lifetime, dropping from 10.4 minutes to 5.2 minutes and from 0.8 seconds to 0.2 seconds, respectively.

- Thanks to the Handoff level improvements, some tokens that would have been sent back to the configuration step are instead moved forward in the workflow.
- The Service level improvements allow us to skip the repetition of many human tasks involved in classifier development, saving a significant amount of time.
- The Task level improvement reduces the average duration of crucial configuration tasks, resulting in faster session processing.

Process Mining

Normative Process Model (Lucchesi, Speciale)



We initially simulated the Normative Model using BIMP with 100 input tokens. Unfortunately, this initial simulation did not provide sufficient data to continue our analysis of the model. Specifically, certain paths within the model and their associated tasks were executed too infrequently, preventing us from introducing violations into the resulting simulation log. For example, the task *'SET #ITERATION'* was executed in only two cases, whereas we intended for it to be executed more frequently.

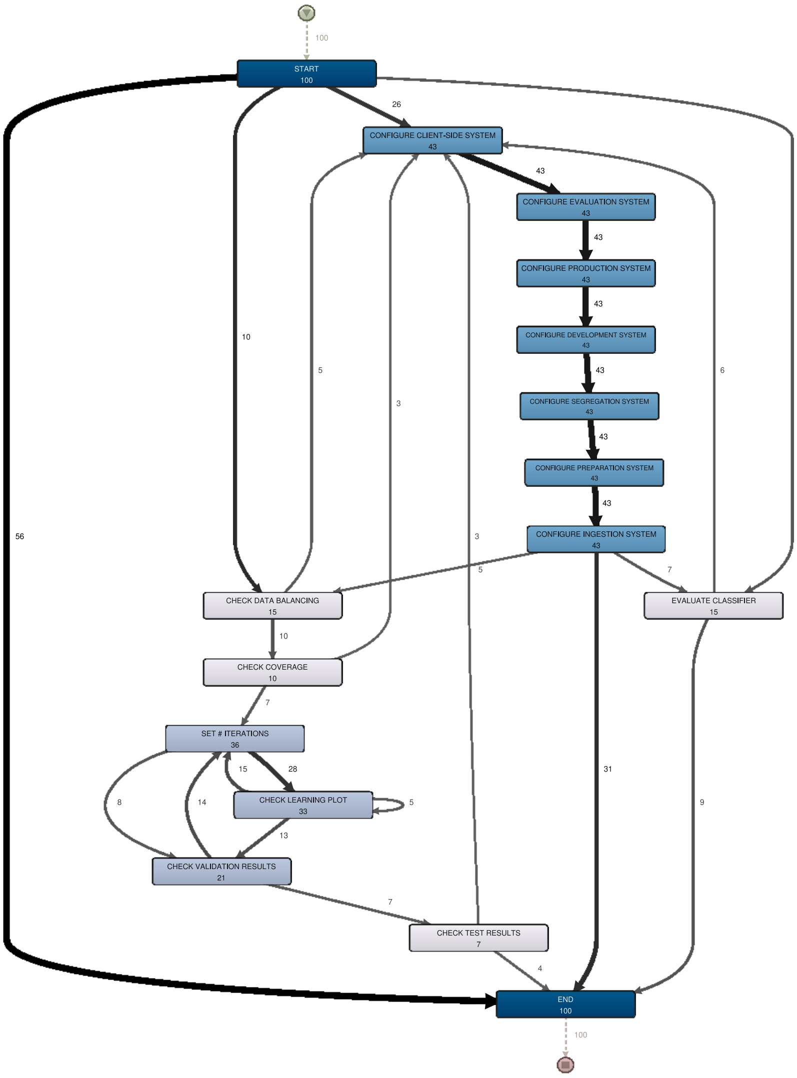
We were therefore compelled to adjust the settings of this initial simulation of the Normative Model. While the number of tokens remained unchanged at 100, we modified the percentages of certain gateways:

GATEWAY	YES	NO
FIRST SESSION?	30%	70%
RECORD SUFFICIENT FOR SESSION?	70%	30%
RAW SESSION INVALID?	30%	70%
EVALUATION PHASE?	40%	60%
DEVELOPMENT PHASE?	60%	40%
SESSIONS SUFFICIENT?	60%	40%
CLASS BALANCED?	60%	40%
COVERAGE SATISFIED?	60%	40%

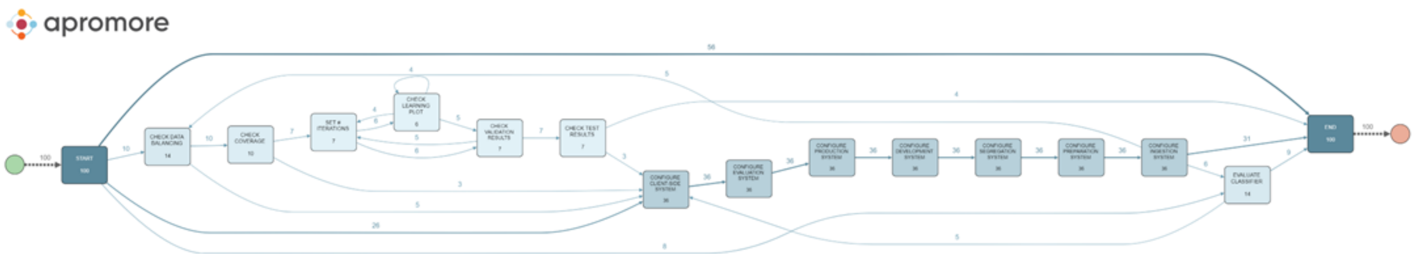
Transition Map Mining

Starting from the log generated using BIMP simulating the Normative Process Model, we used two tools: Apromore and ProM to mine transition maps, which are graphical representations of the business process, showing how activities are connected and the order in which they are executed.

Transition Map generated by DISCO



Transition Map generated by Apromore



Differences between the two Transition Maps

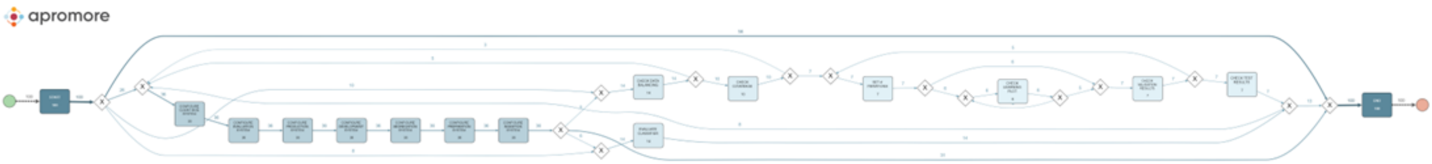
We can observe that tokens flow differently in the two Transition Maps, which is due to the distinct approaches used by the two tools to account for frequencies:

- Disco utilizes ***absolute frequency***, which <u>accounts for every token flow through a task</u>, including returns to the same task. For instance, if a token passes through a task three times, Disco will count all three occurrences. This approach is useful for highlighting loops or tasks executed multiple times within the same case. Consequently, the transition frequencies reported may be higher, particularly in processes with loops or recursive paths.
- Apromore, on the other hand, uses ***case frequency***, focusing on counting how many process instances (cases) visit a task, <u>disregarding returns to the same task</u> within the same case. If a token passes through a task three times, Apromore will count it only once for that case. This approach is beneficial for representing the overall behavior of processes without overemphasizing loops. As a result, the transition frequencies reported will generally be lower than those in Disco for recursive or repetitive processes.

BPMN Model Mining

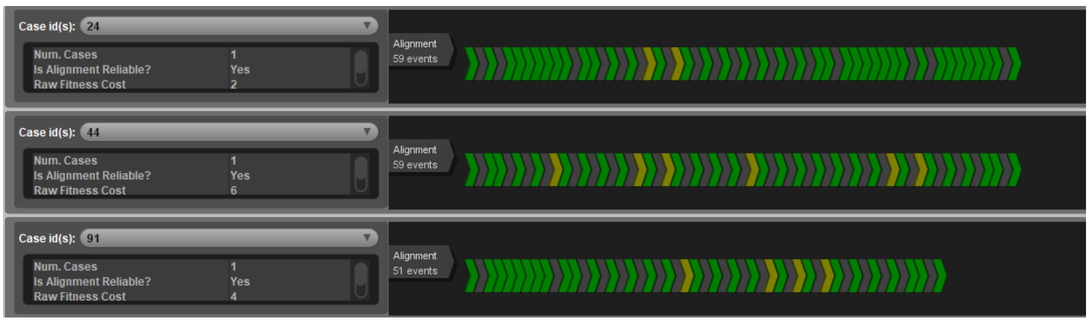
Starting from the log generated using BIMP to simulate the Normative Process Model, we used two tools: Apromore and ProM to mine BPMN models.

BPMN Model mined using Apromore

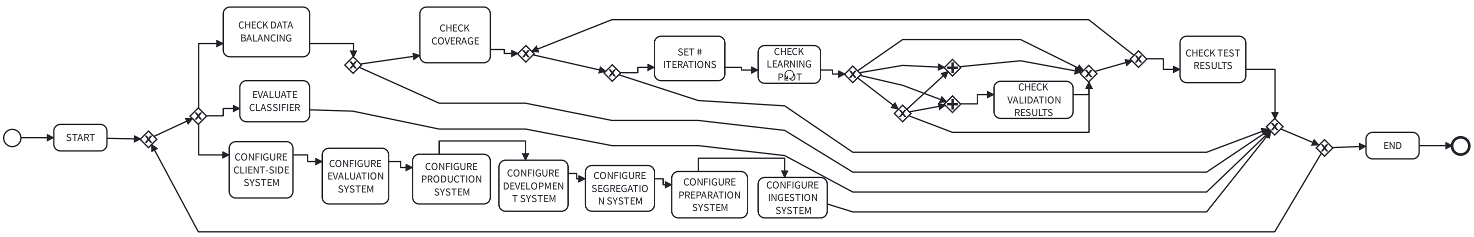


Differences with log

There is only one difference: in the BPMN model, it is not possible to transition directly from *'CHECK LEARNING PLOT'* to *'SET # OF ITERATIONS'*. This impacts four cases, three of which are depicted.



BPMN Model mined using ProM – Noise threshold 0

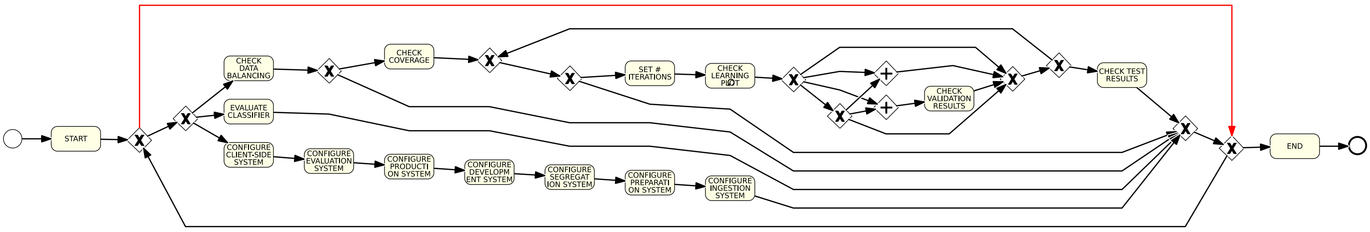


Differences with log

- The biggest difference is that there is no direct path from start to end, 56 tokens are affected by this.

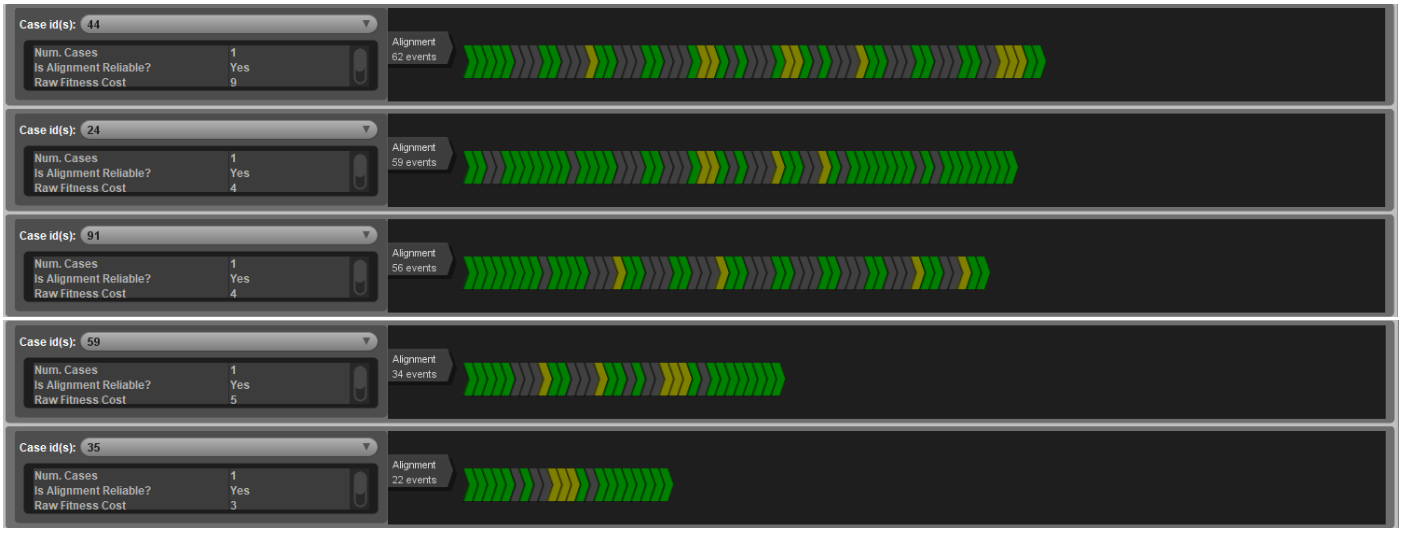
The fact that this path is ignored by the Inductive Miner algorithm used by ProM is quite intriguing. Probably, by changing the configuration of the algorithm (which we left as default), we could have ensured that it considered this path as relevant. It is indeed easy to notice that if there is no path from *'START'* to *'END'*, the fitness drops significantly, as this path is followed by 56 cases, resulting in a fitness value of 0.877.

Out of curiosity, we manually modified the BPMN model returned by the Inductive Miner by adding this missing path (highlighted in red).

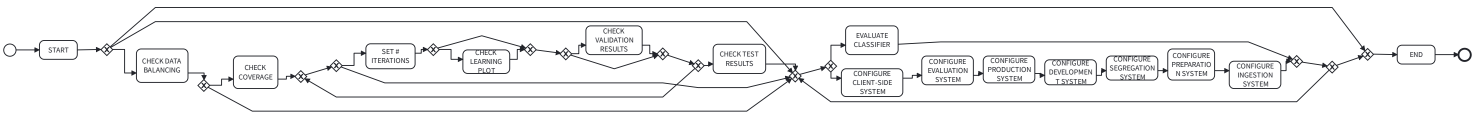


As expected, the fitness increased to 0.989, confirming that the presence of that path indeed improves the quality of the model.

- There are a small number of cases in the log that are not represented in the BPMN during the development phase. For instance, the BPMN lacks a path directly connecting *'SET # OF ITERATIONS'* to *'CHECK VALIDATION RESULT'*.

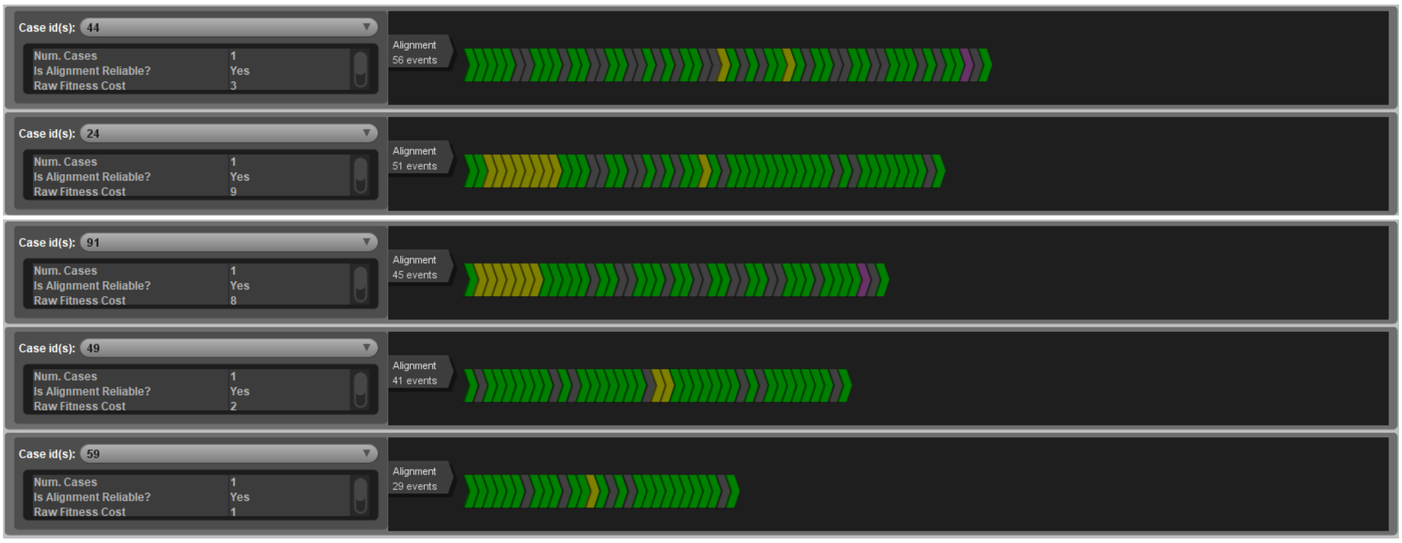


BPMN Model mined using ProM – Noise threshold 0.2



Differences with log

- There are a few connections in the BPMN that are missing. For example, it is not possible to return to *'CHECK DATA BALANCING'* after doing the configuration; this is what causes the long sequence of yellow events. Another yellow event is caused by the fact that in the mined BPMN it is not possible to loop on *'CHECK LEARNING PLOT'*. Also, there is no straight path from *'CHECK TEST RESULTS'* to *'END'*; this is what causes the purple event.



Four quality dimensions

Model	Fitness	Simplicity #sequenceFlow(edges) + #activities + #gateways	Precision	Generalization
Mined in ProM with 0 Noise threshold	0,877	42+16+13 = 71	0,60199	0,98991
Mined in ProM with 0.2 Noise threshold	0,988	40+16+14 = 70	0,62071	0,988
Mined in Apromore	0,994	43+16+15 = 74	0,76959	0,9895

- **Fitness:** the highest fitness was achieved by the Apromore-mined model. Fitness is the measure of how much of the behavior described in the log is captured by the model. This is also easily discernible by observing the three models: the one with the highest fitness is the one that resembles the most the original model;
- **Simplicity:** it is a measure that puts in perspective the three models from the point of view of the number of elements that compose the respective BPMNs. In other words, the less elements a model consists of, the better it is. As we can see from the table, the ProM-0.2-Noise model is the simplest one, closely followed by the ProM-0-Noise one. It is worth adding that there are no significative differences in the simplicity of the three models;
- **Generalization:** it is the measure of the capability of a model to abstract, avoiding overfitting. It is opposed to precision, since it considers as better a model that does not follow too much the event logs, starting from the assumption that they are not always complete, thus models should not focus too much on them, but rather use the logs to fit and learn a behavior general enough to have a margin of abstraction. There is no significative difference between the three mined models;
- **Precision:** the model mined in Apromore presents the highest precision. This is due to the fact that the conformance check does not detect extra behavior in that model (there are no events appearing in the mined BPMN that are not present in the log, although the precision cannot be 1 since the model does not capture some events appearing in the log). As we can see from the precision formula below, the number of events present in the log but not detected by the mined model contribute to lowering the numerator value, thus the precision result, while events detected by the mined model but not present in the log.

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

  - ε Is the number of unique events in a context of the log
  - en_L is the number of enabled activities in the model M
  - en_MIs the number of observed activities executed in a similar context in L

Introducing Violations into the Log (Arduini)

In this phase, we introduced three realistic and meaningful violations into the log file to observe how the conformance checking results change compared to the baseline scenario. In doing so, we referred to the same initial assumption made when improving the AS-IS model and transforming it into the TO-BE model, namely that our factory has available customer clusters with similar data. As we will see, this allows us to reuse hyperparameters (e.g., the number of layers and neurons) previously used to train classifiers for other clients in the same cluster, or even to reuse entire pre-trained neural networks that are 'ready for use'.

1) TRAINING PHASE SKIP

Assuming we have access to a pre-trained network from another customer with a similar profile, we reuse it, allowing us to skip the training phase required for the new classifier.

Tasks that can be excluded:

- *'SET # ITERATIONS'*
- *'CHECK LEARNING PLOT'*
- *'CHECK VALIDATION RESULTS'* (this task will only be removed when the log indicates a negative outcome that triggered a retraining of the classifier. However, one instance of this task will be retained to ensure a final validation check for quality assurance)

These tasks, which constitute the training phase, are skipped, proceeding directly to the *'CHECK VALIDATION RESULTS'* task.

Log modifications

Case ID: 84

84 START	
84 CHECK DATA BALANCING	Default Resource-000006
84 CHECK COVERAGE	Default Resource-000007
84 SET # ITERATIONS	Default Resource-000008
84 CHECK LEARNING PLOT	Default Resource-000009
84 SET # ITERATIONS	Default Resource-000010
84 CHECK LEARNING PLOT	Default Resource-000001
84 SET # ITERATIONS	Default Resource-000002
84 CHECK LEARNING PLOT	Default Resource-000003
84 SET # ITERATIONS	Default Resource-000004
84 CHECK VALIDATION RESULTS	Default Resource-000005
84 CHECK TEST RESULTS	Default Resource-000007
84 END	

→

84 START	
84 CHECK DATA BALANCING	Default Resource-000006
84 CHECK COVERAGE	Default Resource-000007
84 CHECK VALIDATION RESULTS	Default Resource-000005
84 CHECK TEST RESULTS	Default Resource-000007
84 END	

Case ID: 35

35 START	
35 CHECK DATA BALANCING	Default Resource-000007
35 CHECK COVERAGE	Default Resource-000008
35 SET # ITERATIONS	Default Resource-000009
35 CHECK LEARNING PLOT	Default Resource-000010
35 CHECK LEARNING PLOT	Default Resource-000001
35 CHECK VALIDATION RESULTS	Default Resource-000002
35 SET # ITERATIONS	Default Resource-000003
35 CHECK VALIDATION RESULTS	Default Resource-000004
35 CHECK TEST RESULTS	Default Resource-000005
35 CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000006
35 CONFIGURE EVALUATION SYSTEM	Default Resource-000008
35 CONFIGURE PRODUCTION SYSTEM	Default Resource-000010
35 CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000002
35 CONFIGURE SEGREGATION SYSTEM	Default Resource-000004
35 CONFIGURE PREPARATION SYSTEM	Default Resource-000006
35 CONFIGURE INGESTION SYSTEM	Default Resource-000008
35 END	

→

35 START	
35 CHECK DATA BALANCING	Default Resource-000007
35 CHECK COVERAGE	Default Resource-000008
35 CHECK VALIDATION RESULTS	Default Resource-000004
35 CHECK TEST RESULTS	Default Resource-000005
35 CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000006
35 CONFIGURE EVALUATION SYSTEM	Default Resource-000008
35 CONFIGURE PRODUCTION SYSTEM	Default Resource-000010
35 CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000002
35 CONFIGURE SEGREGATION SYSTEM	Default Resource-000004
35 CONFIGURE PREPARATION SYSTEM	Default Resource-000006
35 CONFIGURE INGESTION SYSTEM	Default Resource-000008
35 END	

Case ID: 91

91	START	
91	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000002
91	...	
91	CONFIGURE INGESTION SYSTEM	Default Resource-000008
91	CHECK DATA BALANCING	Default Resource-000009
91	CHECK COVERAGE	Default Resource-000010
91	SET # ITERATIONS	Default Resource-000001
91	CHECK LEARNING PLOT	Default Resource-000002
91	CHECK VALIDATION RESULTS	Default Resource-000003
91	SET # ITERATIONS	Default Resource-000004
91	CHECK LEARNING PLOT	Default Resource-000005
91	SET # ITERATIONS	Default Resource-000006
91	CHECK LEARNING PLOT	Default Resource-000007
91	CHECK VALIDATION RESULTS	Default Resource-000008
91	SET # ITERATIONS	Default Resource-000009
91	CHECK LEARNING PLOT	Default Resource-000010
91	SET # ITERATIONS	Default Resource-000001
91	CHECK LEARNING PLOT	Default Resource-000002
91	SET # ITERATIONS	Default Resource-000003
91	CHECK LEARNING PLOT	Default Resource-000004
91	SET # ITERATIONS	Default Resource-000005
91	CHECK LEARNING PLOT	Default Resource-000006
91	CHECK VALIDATION RESULTS	Default Resource-000007
91	SET # ITERATIONS	Default Resource-000008
91	CHECK LEARNING PLOT	Default Resource-000009
91	CHECK VALIDATION RESULTS	Default Resource-000010
91	CHECK TEST RESULTS	Default Resource-000001
91	END	

→

91	START	
91	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000002
91	...	
91	CONFIGURE INGESTION SYSTEM	Default Resource-000008
91	CHECK DATA BALANCING	Default Resource-000009
91	CHECK COVERAGE	Default Resource-000010
91	CHECK VALIDATION RESULTS	Default Resource-000010
91	CHECK TEST RESULTS	Default Resource-000001
91	END	

2) HYPERPARAMETERIZATION PHASE SKIP

Assuming we have access to the "best" hyperparameters used to develop classifiers for customers belonging to the same cluster, this allows us to skip the validation phase entirely.

Tasks that can be excluded:

- *'CHECK VALIDATION RESULT'*
- *'SET # ITERATIONS' & 'CHECK LEARNING PLOT'* (these two tasks will only be removed when their execution is triggered by the validation phase)

Skipping these tasks enables a direct transition to the testing phase, specifically to the *'CHECK TEST RESULTS'* task.

Log modifications

Case ID: 59

59	START	
59	CHECK DATA BALANCING	Default Resource-000009
59	CHECK COVERAGE	Default Resource-000010
59	SET # ITERATIONS	Default Resource-000001
59	CHECK LEARNING PLOT	Default Resource-000002
59	CHECK VALIDATION RESULTS	Default Resource-000003
59	SET # ITERATIONS	Default Resource-000004
59	CHECK LEARNING PLOT	Default Resource-000005
59	CHECK VALIDATION RESULTS	Default Resource-000006
59	SET # ITERATIONS	Default Resource-000007
59	CHECK LEARNING PLOT	Default Resource-000008
59	CHECK LEARNING PLOT	Default Resource-000009
59	CHECK VALIDATION RESULTS	Default Resource-000010
59	SET # ITERATIONS	Default Resource-000001
59	CHECK VALIDATION RESULTS	Default Resource-000002
59	CHECK TEST RESULTS	Default Resource-000003
59	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000004
59	CONFIGURE EVALUATION SYSTEM	Default Resource-000005
59	CONFIGURE PRODUCTION SYSTEM	Default Resource-000006
59	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000007
59	CONFIGURE SEGREGATION SYSTEM	Default Resource-000008
59	CONFIGURE PREPARATION SYSTEM	Default Resource-000009
59	CONFIGURE INGESTION SYSTEM	Default Resource-000010
59	END	

→

59	START	
59	CHECK DATA BALANCING	Default Resource-000009
59	CHECK COVERAGE	Default Resource-000010
59	SET # ITERATIONS	Default Resource-000001
59	CHECK LEARNING PLOT	Default Resource-000002
59	SET # ITERATIONS	Default Resource-000004
59	CHECK LEARNING PLOT	Default Resource-000005
59	SET # ITERATIONS	Default Resource-000007
59	CHECK LEARNING PLOT	Default Resource-000008
59	CHECK TEST RESULTS	Default Resource-000003
59	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000004
59	CONFIGURE EVALUATION SYSTEM	Default Resource-000005
59	CONFIGURE PRODUCTION SYSTEM	Default Resource-000006
59	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000007
59	CONFIGURE SEGREGATION SYSTEM	Default Resource-000008
59	CONFIGURE PREPARATION SYSTEM	Default Resource-000009
59	CONFIGURE INGESTION SYSTEM	Default Resource-000010
59	END	

Case ID: 24

24	START	
24	CHECK DATA BALANCING	Default Resource-000001
24	...	
24	CONFIGURE PREPARATION SYSTEM	Default Resource-000007
24	CONFIGURE INGESTION SYSTEM	Default Resource-000008
24	CHECK DATA BALANCING	Default Resource-000009
24	CHECK COVERAGE	Default Resource-000010
24	SET # ITERATIONS	Default Resource-000002
24	CHECK LEARNING PLOT	Default Resource-000004
24	SET # ITERATIONS	Default Resource-000006
24	CHECK LEARNING PLOT	Default Resource-000008
24	SET # ITERATIONS	Default Resource-000010
24	CHECK VALIDATION RESULTS	Default Resource-000002
24	SET # ITERATIONS	Default Resource-000004
24	CHECK LEARNING PLOT	Default Resource-000005
24	CHECK LEARNING PLOT	Default Resource-000006
24	CHECK VALIDATION RESULTS	Default Resource-000007
24	SET # ITERATIONS	Default Resource-000008
24	CHECK LEARNING PLOT	Default Resource-000009
24	CHECK VALIDATION RESULTS	Default Resource-000010
24	CHECK TEST RESULTS	Default Resource-000001
24	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000002
24	...	
24	CONFIGURE INGESTION SYSTEM	Default Resource-000006
24	END	

→

24	START	
24	CHECK DATA BALANCING	Default Resource-000001
24	...	
24	CONFIGURE PREPARATION SYSTEM	Default Resource-000007
24	CONFIGURE INGESTION SYSTEM	Default Resource-000008
24	CHECK DATA BALANCING	Default Resource-000009
24	CHECK COVERAGE	Default Resource-000010
24	SET # ITERATIONS	Default Resource-000002
24	CHECK LEARNING PLOT	Default Resource-000004
24	SET # ITERATIONS	Default Resource-000006
24	CHECK LEARNING PLOT	Default Resource-000008
24	SET # ITERATIONS	Default Resource-000004
24	CHECK LEARNING PLOT	Default Resource-000005
24	SET # ITERATIONS	Default Resource-000008
24	CHECK LEARNING PLOT	Default Resource-000009
24	CHECK TEST RESULTS	Default Resource-000001
24	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000002
24	...	
24	CONFIGURE INGESTION SYSTEM	Default Resource-000006
24	END	

Case ID: 44

44	START	
44	CHECK DATA BALANCING	Default Resource-000001
44	CHECK COVERAGE	Default Resource-000002
44	SET # ITERATIONS	Default Resource-000003
44	CHECK LEARNING PLOT	Default Resource-000004
44	SET # ITERATIONS	Default Resource-000005
44	CHECK LEARNING PLOT	Default Resource-000006
44	CHECK VALIDATION RESULTS	Default Resource-000007
44	SET # ITERATIONS	Default Resource-000008
44	CHECK LEARNING PLOT	Default Resource-000009
44	SET # ITERATIONS	Default Resource-000010
44	CHECK LEARNING PLOT	Default Resource-000001
44	SET # ITERATIONS	Default Resource-000002
44	CHECK VALIDATION RESULTS	Default Resource-000003
44	SET # ITERATIONS	Default Resource-000004
44	CHECK LEARNING PLOT	Default Resource-000005
44	CHECK LEARNING PLOT	Default Resource-000006
44	SET # ITERATIONS	Default Resource-000007
44	CHECK VALIDATION RESULTS	Default Resource-000008
44	SET # ITERATIONS	Default Resource-000009
44	CHECK LEARNING PLOT	Default Resource-000010
44	CHECK LEARNING PLOT	Default Resource-000001
44	CHECK VALIDATION RESULTS	Default Resource-000002
44	SET # ITERATIONS	Default Resource-000003
44	CHECK LEARNING PLOT	Default Resource-000004
44	SET # ITERATIONS	Default Resource-000005
44	CHECK LEARNING PLOT	Default Resource-000006
44	SET # ITERATIONS	Default Resource-000007
44	CHECK LEARNING PLOT	Default Resource-000008
44	CHECK VALIDATION RESULTS	Default Resource-000009
44	SET # ITERATIONS	Default Resource-000010
44	CHECK VALIDATION RESULTS	Default Resource-000002
44	CHECK TEST RESULTS	Default Resource-000004
44	END	

→

44	START	
44	CHECK DATA BALANCING	Default Resource-000001
44	CHECK COVERAGE	Default Resource-000002
44	SET # ITERATIONS	Default Resource-000003
44	CHECK LEARNING PLOT	Default Resource-000004
44	SET # ITERATIONS	Default Resource-000005
44	CHECK LEARNING PLOT	Default Resource-000006
44	SET # ITERATIONS	Default Resource-000008
44	CHECK LEARNING PLOT	Default Resource-000009
44	SET # ITERATIONS	Default Resource-000010
44	CHECK LEARNING PLOT	Default Resource-000001
44	SET # ITERATIONS	Default Resource-000004
44	CHECK LEARNING PLOT	Default Resource-000005
44	SET # ITERATIONS	Default Resource-000009
44	CHECK LEARNING PLOT	Default Resource-000010
44	SET # ITERATIONS	Default Resource-000003
44	CHECK LEARNING PLOT	Default Resource-000004
44	SET # ITERATIONS	Default Resource-000005
44	CHECK LEARNING PLOT	Default Resource-000006
44	SET # ITERATIONS	Default Resource-000007
44	CHECK LEARNING PLOT	Default Resource-000008
44	CHECK TEST RESULTS	Default Resource-000004
44	END	

3) STREAMLINED RECONFIGURATION FOR PHASE CHANGE

Once the *'EVALUATE CLASSIFIER'* task is executed and yields a negative result, indicating that the model is no longer sufficiently accurate, reconfiguration becomes necessary. The purpose of this violation is to allow reconfiguration of only the Preparation System, changing the phase of the classifier to 'Development' as the current phase. This optimization helps avoid unnecessary reconfiguration of other systems, as they would only involve confirming previous settings.

Log modifications

Case ID: 49

49	START	
49	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000010
49	CONFIGURE EVALUATION SYSTEM	Default Resource-000002
49	CONFIGURE PRODUCTION SYSTEM	Default Resource-000004
49	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000006
49	CONFIGURE SEGREGATION SYSTEM	Default Resource-000008
49	CONFIGURE PREPARATION SYSTEM	Default Resource-000010
49	CONFIGURE INGESTION SYSTEM	Default Resource-000001
49	EVALUATE CLASSIFIER	Default Resource-000002
49	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000003
49	CONFIGURE EVALUATION SYSTEM	Default Resource-000004
49	CONFIGURE PRODUCTION SYSTEM	Default Resource-000006
49	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000007
49	CONFIGURE SEGREGATION SYSTEM	Default Resource-000008
49	CONFIGURE PREPARATION SYSTEM	Default Resource-000009
49	CONFIGURE INGESTION SYSTEM	Default Resource-000010
49	CHECK DATA BALANCING	Default Resource-000001
49	CHECK COVERAGE	Default Resource-000002
49	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000003
49	CONFIGURE EVALUATION SYSTEM	Default Resource-000004
49	CONFIGURE PRODUCTION SYSTEM	Default Resource-000005
49	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000007
49	CONFIGURE SEGREGATION SYSTEM	Default Resource-000009
49	CONFIGURE PREPARATION SYSTEM	Default Resource-000001
49	CONFIGURE INGESTION SYSTEM	Default Resource-000003
49	EVALUATE CLASSIFIER	Default Resource-000005
49	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000007
49	CONFIGURE EVALUATION SYSTEM	Default Resource-000009
49	CONFIGURE PRODUCTION SYSTEM	Default Resource-000010
49	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000001
49	CONFIGURE SEGREGATION SYSTEM	Default Resource-000002
49	CONFIGURE PREPARATION SYSTEM	Default Resource-000003
49	CONFIGURE INGESTION SYSTEM	Default Resource-000004
49	END	

→

49	START	
49	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000010
49	CONFIGURE EVALUATION SYSTEM	Default Resource-000002
49	CONFIGURE PRODUCTION SYSTEM	Default Resource-000004
49	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000006
49	CONFIGURE SEGREGATION SYSTEM	Default Resource-000008
49	CONFIGURE PREPARATION SYSTEM	Default Resource-000010
49	CONFIGURE INGESTION SYSTEM	Default Resource-000001
49	EVALUATE CLASSIFIER	Default Resource-000002
49	CONFIGURE PREPARATION SYSTEM	Default Resource-000009
49	CHECK DATA BALANCING	Default Resource-000001
49	CHECK COVERAGE	Default Resource-000002
49	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000003
49	CONFIGURE EVALUATION SYSTEM	Default Resource-000004
49	CONFIGURE PRODUCTION SYSTEM	Default Resource-000005
49	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000007
49	CONFIGURE SEGREGATION SYSTEM	Default Resource-000009
49	CONFIGURE PREPARATION SYSTEM	Default Resource-000001
49	CONFIGURE INGESTION SYSTEM	Default Resource-000003
49	EVALUATE CLASSIFIER	Default Resource-000005
49	CONFIGURE PREPARATION SYSTEM	Default Resource-000003
49	END	

Case ID: 5

5	START	
5	CHECK DATA BALANCING	Default Resource-000003
5	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000004
5	CONFIGURE EVALUATION SYSTEM	Default Resource-000005
5	CONFIGURE PRODUCTION SYSTEM	Default Resource-000006
5	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000007
5	CONFIGURE SEGREGATION SYSTEM	Default Resource-000008
5	CONFIGURE PREPARATION SYSTEM	Default Resource-000009
5	CONFIGURE INGESTION SYSTEM	Default Resource-000010
5	EVALUATE CLASSIFIER	Default Resource-000001
5	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000002
5	CONFIGURE EVALUATION SYSTEM	Default Resource-000004
5	CONFIGURE PRODUCTION SYSTEM	Default Resource-000005
5	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000006
5	CONFIGURE SEGREGATION SYSTEM	Default Resource-000007
5	CONFIGURE PREPARATION SYSTEM	Default Resource-000008
5	CONFIGURE INGESTION SYSTEM	Default Resource-000009
5	END	

→

5	START	
5	CHECK DATA BALANCING	Default Resource-000003
5	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000004
5	CONFIGURE EVALUATION SYSTEM	Default Resource-000005
5	CONFIGURE PRODUCTION SYSTEM	Default Resource-000006
5	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000007
5	CONFIGURE SEGREGATION SYSTEM	Default Resource-000008
5	CONFIGURE PREPARATION SYSTEM	Default Resource-000009
5	CONFIGURE INGESTION SYSTEM	Default Resource-000010
5	EVALUATE CLASSIFIER	Default Resource-000001
5	CONFIGURE PREPARATION SYSTEM	Default Resource-000008
5	END	

Case ID: 85

85	EVALUATE CLASSIFIER	Default Resource-000006
85	CONFIGURE CLIENT-SIDE SYSTEM	Default Resource-000008
85	CONFIGURE EVALUATION SYSTEM	Default Resource-000009
85	CONFIGURE PRODUCTION SYSTEM	Default Resource-000010
85	CONFIGURE DEVELOPMENT SYSTEM	Default Resource-000001
85	CONFIGURE SEGREGATION SYSTEM	Default Resource-000002
85	CONFIGURE PREPARATION SYSTEM	Default Resource-000003
85	CONFIGURE INGESTION SYSTEM	Default Resource-000004
85	END	

→

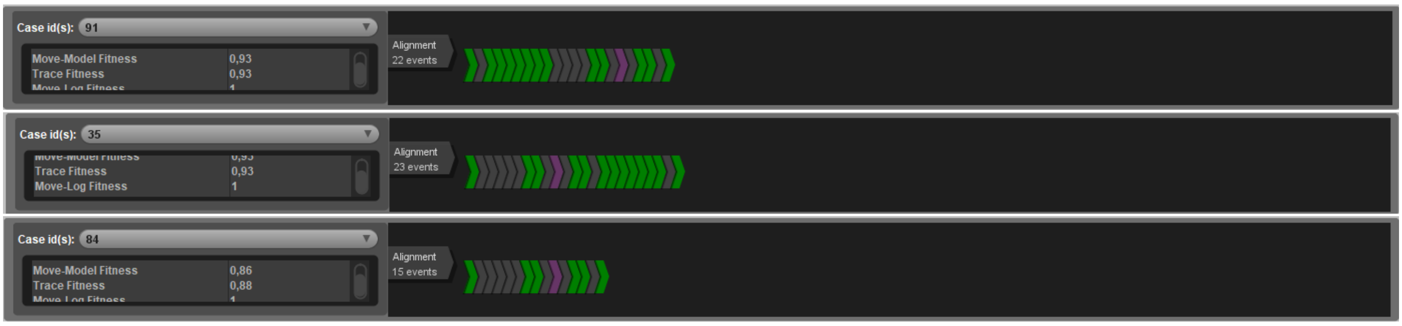
85	START	
85	EVALUATE CLASSIFIER	Default Resource-000006
85	CONFIGURE PREPARATION SYSTEM	Default Resource-000003
85	END	

Conformance Checking on ProM with Normative Model

Using the violated log, namely the one containing violations, we performed a conformance check with the original normative model to examine how the violated cases behave.

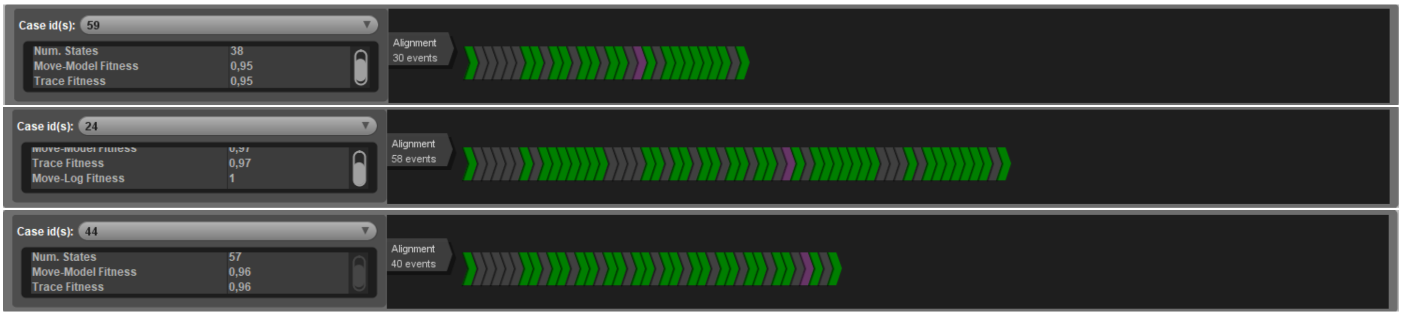
- **Case 84, 35, 91**

In these cases, we skip the entire training phase, meaning that we jump from *'CHECK COVERAGE'* directly to *'CHECK VALIDATION RESULTS'*. ProM interprets this as taking the shortest path from these two events, which is the one that only passes through "SET # ITERATIONS". Indeed, the purple event represents *'SET # ITERATIONS'*.



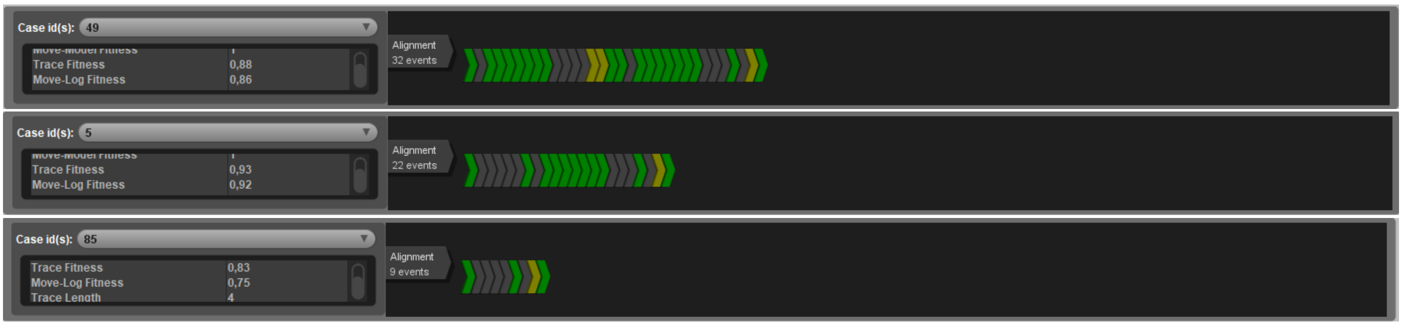
- **Case 59, 24, 44**

  In these cases, we skip the hyperparameter tuning phase, which eliminates the validation phase and the *'CHECK VALIDATION RESULTS'* event. Notably, the purple event corresponds to this skipped event.



- **Case 49, 5, 85**

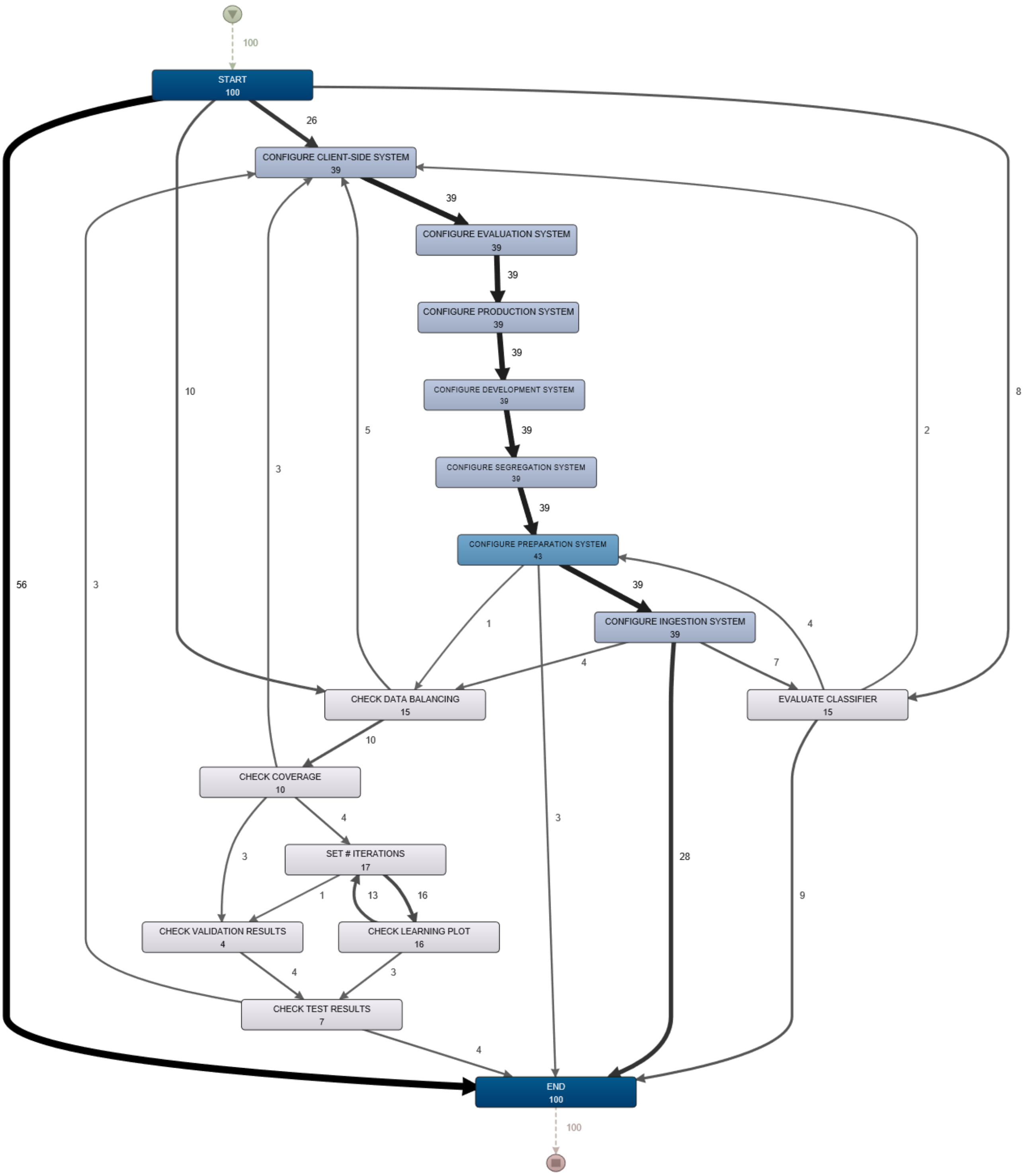
In these cases, we are only configuring the Preparation System following a classifier evaluation, and we skip every other configuration. This change is identified by the yellow events, which represent the configuration of the Preparation System. On case 49, the other yellow event is *'EVALUATE CLASSIFIER'.*



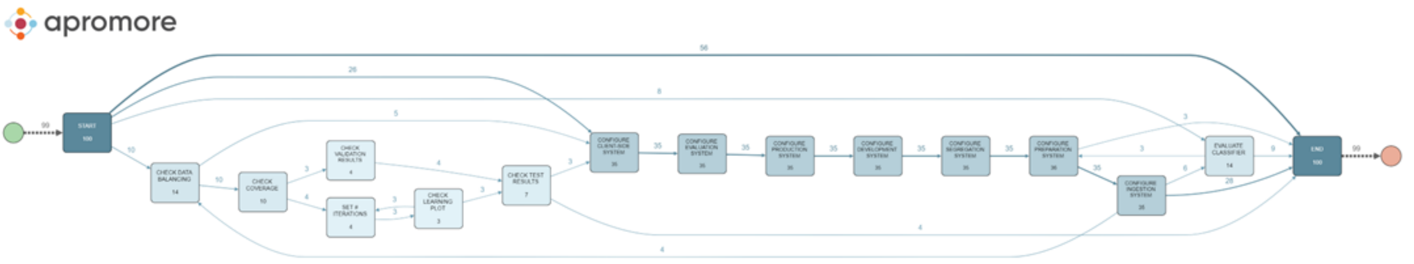
Transition map of violated log

We used two tools, Apromore and ProM, to mine transition maps from the violated log.

Transition map of violated log generated by Disco

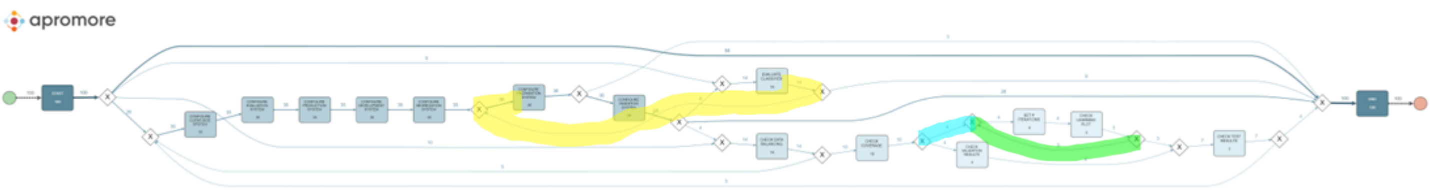


Transition map of violated log generated by Apromore

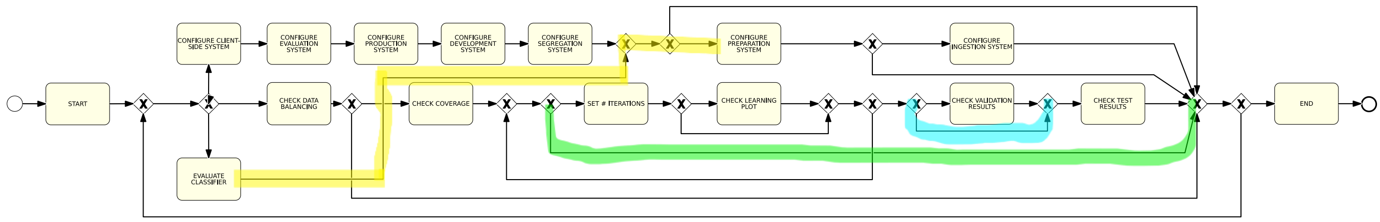


BPMN Model Mining from the violated log

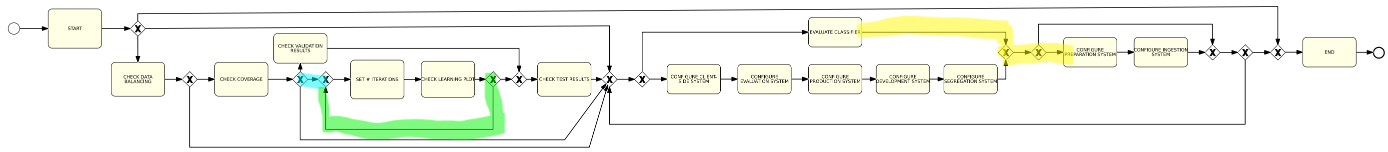
BPMN Model mined using Apromore on violated log



BPMN Model mined using ProM on violated log – Noise threshold 0



BPMN Model mined using ProM on violated log - Noise threshold 0.2



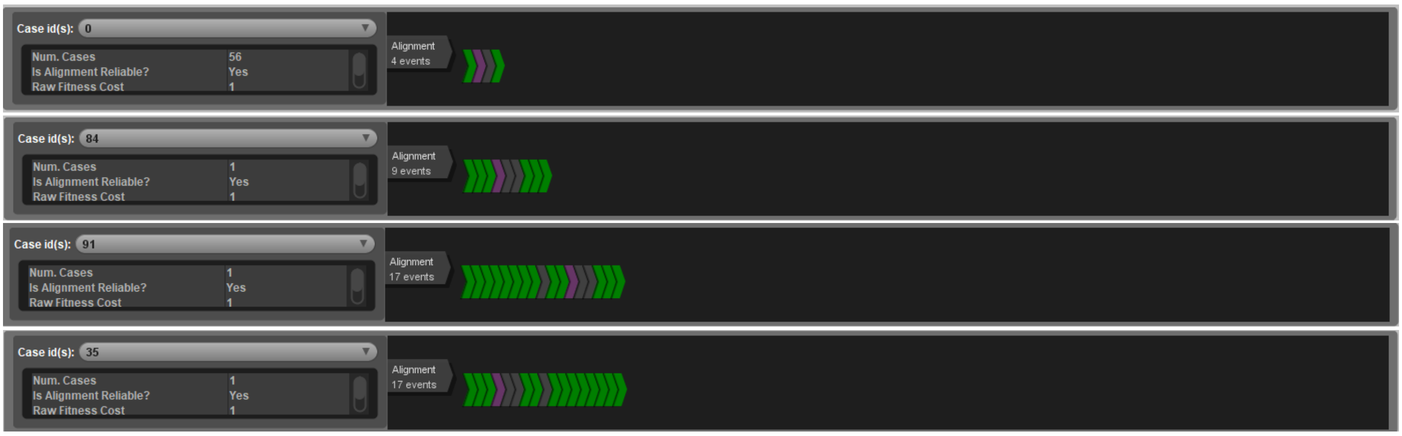
As we can see, both Apromore and ProM were able to capture the possibility of going from *'EVALUATE CLASSIFIER'* directly to *'CONFIGURE PREPARATION SYSTEM'* (highlighted in yellow). They were also able to capture the possibility of skipping *'CHECK VALIDATION RESULTS'* (highlighted in blue). Finally, all models also show the possibility of skipping the training phase (*'SET # ITERATIONS'* and *'CHECK LEARNING PLOT'*), highlighted in green.

Four quality dimensions

Model	Fitness	Simplicity #sequenceFlow(edges) + #activities + #gateways	Precision	Generalization
Mined in ProM with 0 Noise threshold	0,885	42+16+15 = 73	0,4943	0,9764
Mined in ProM with 0.2 Noise threshold	0,985	38+16+13 = 67	0,7331	0,9752
Mined in Apromore	0,995	43+16+15 = 74	0,7498	0,9803

As we saw for the initial mined models, the BPMN mined with ProM with 0 Noise threshold is the worst at reproducing the log behavior, while the 0.2 Noise Threshold and Apromore models have similar fitness.

Since precision is concerned with quantifying how much behavior a process model allows for that was never observed in the event log, we suspect that having many cases with purple events (meaning events that are only in the model) is what causes such a bad precision for the ProM model with 0 noise threshold. (see below)



Regarding simplicity, we can observe an interesting difference for the ProM-0.2-Noise model: it is significantly lower than the other two values, and this can be explained considering the non-zero noise threshold that makes the model ignore sufficiently infrequent paths.

Overall, the Generalization capability of the three models is very similar.

EXTRA - Alternative mining algorithms

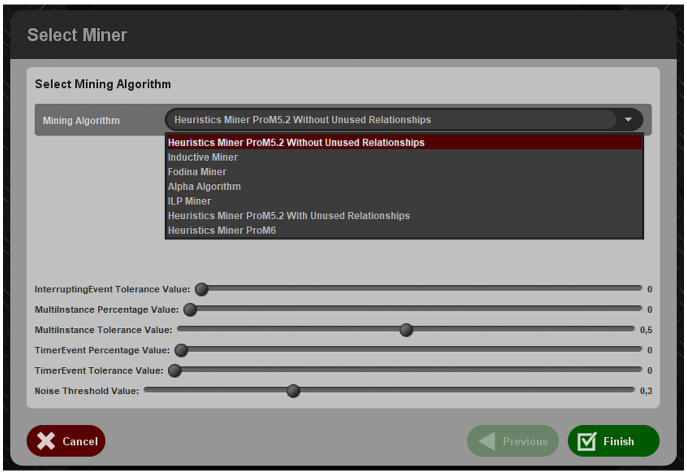
Since we finished the project earlier than expected and therefore had additional time to delve deeper into some of its aspects, we decided to further investigate the mining of BPMN models from the event log. In fact, using multiple mining algorithms to perform process discovery and compare the results is a very useful practice, as each process mining algorithm has specific characteristics, strengths, and limitations that can lead to different outcomes depending on the nature of the data and the complexity of the process.

This practice is <u>useful for addressing the trade-off between precision, generalization, and fitness</u>. A model that is too rigid might perfectly adhere to the data but fail to represent new future behaviors (overfitting), whereas a model that is too generic might miss important details of the process (underfitting). By comparing results from multiple algorithms, it is possible to identify a balance that provides an accurate yet sufficiently flexible and interpretable model.

Lastly, another advantage is the ability to identify potential inconsistencies in the data or unclear areas of the process. If two algorithms generate different models, this discrepancy may indicate the presence of noise in the log, infrequent activities, or complex behaviors that require further analysis.

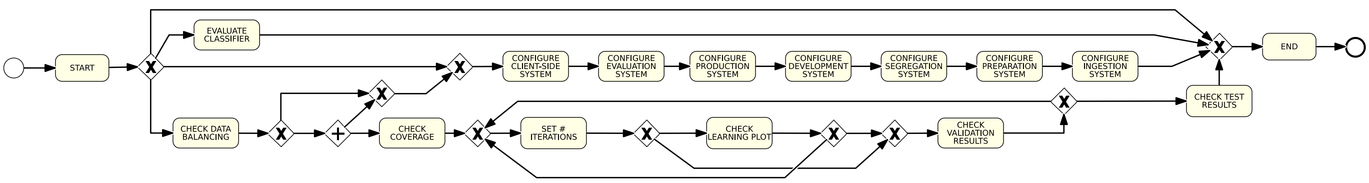
The mining algorithms analysed were found in the list provided by the "BPMN Miner" plugin on ProM, as reported in screenshot below. The algorthms chosen are:

- Fodina miner
- ILP miner
- Heuristics miner
- Heuristics miner with unused relationships
- Alpha miner



Fodina miner

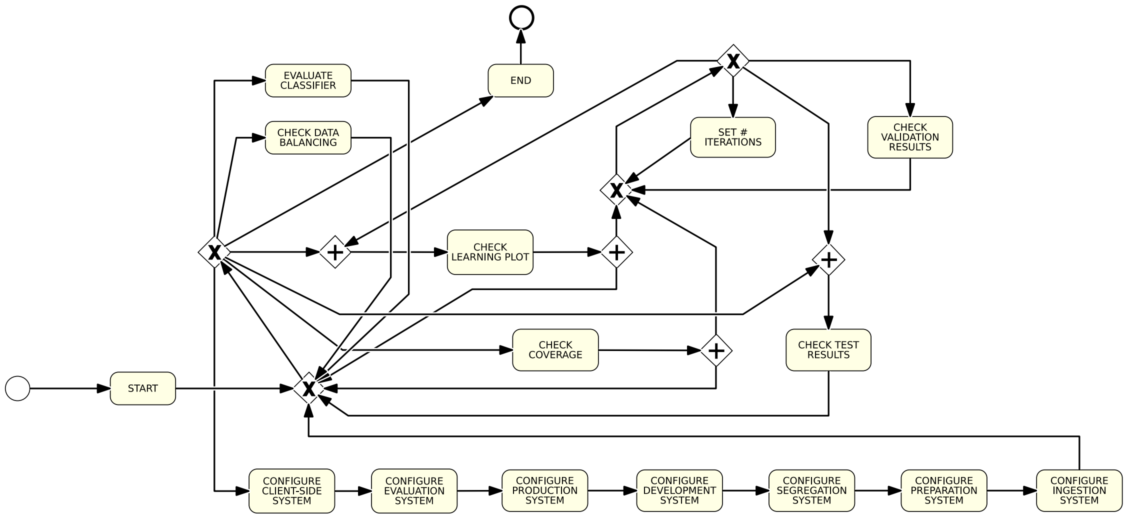
The Fodina Miner is a process mining algorithm designed to detect **parallelisms** and handle **noisy** or complex event logs. It stands out for its ability to identify activities executed in parallel, adapting better to the variability of real-world processes compared to more rigid algorithms like the Alpha Miner. However, one of its limitations is that it can generate detailed and **complex** models that are difficult to interpret, especially when logs contain many exceptions or rare paths.



At a first glance, the BPMN looks similar to the one mined on Apromore and ProM with the inductive miner. However, the Fodina miner has captured more loops between tasks, which are not numerous in the ProM mined BPMN. In general, it seems to have captured effectively the behavior and paths described in the log. The more significant difference with respect to the other mined BPMNs is the presence of a parallel gateway, which was not observed in the BPMNs produced by the inductive miner and Apromore.

ILP miner

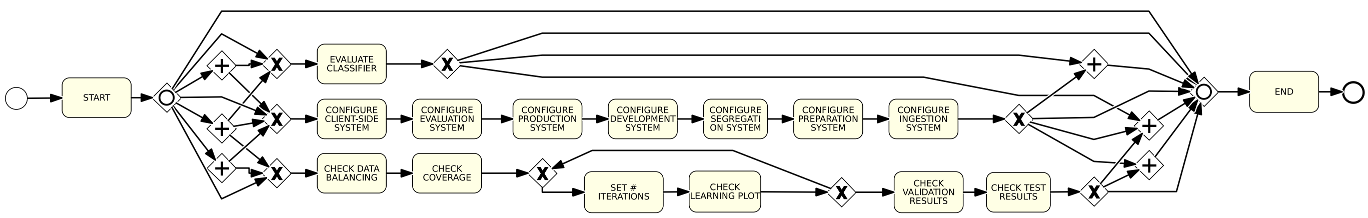
The ILP Miner is a process mining algorithm based on **integer linear programming** (ILP) to discover precise process models. Its main strength lies in its ability to generate models **free from invalid behaviors** (no overfitting), ensuring consistency with the event log. However, the ILP Miner suffers from **scalability** issues: it becomes inefficient with large or complex logs due to high computational demands. Additionally, it can produce **overly rigid** models, which struggle to handle noise or variability in real-world data.



As can be seen above, the BPMN mined by the ILP algorithm looks more complex than the others, in terms of edges crossing and loops represented. Looking more carefully, we can observe that the number of loops is very high with respect to the other BPMNs seen until now. In addition, we can observe a stronger presence of parallel gateways with respect to the BPMN obtained by the Fodina miner.

Heuristics miner

The Heuristics Miner is a process mining algorithm designed to handle **noise** and **uncertainty** in event logs. Its key strength lies in its ability to create **simplified and readable models** by focusing on the most frequent behavior while filtering out less significant paths. This makes it particularly useful for real-world, noisy data. However, its main limitation is that it may **ignore rare behaviors** and overemphasize frequent ones, potentially leading to an incomplete or **generalized** model that does not capture all process variations.



The basic heuristic algorithm considers all relationships between activities, including noisy or infrequent ones. It constructs the process model by:

- Analyzing the direct succession of activities based on event logs.

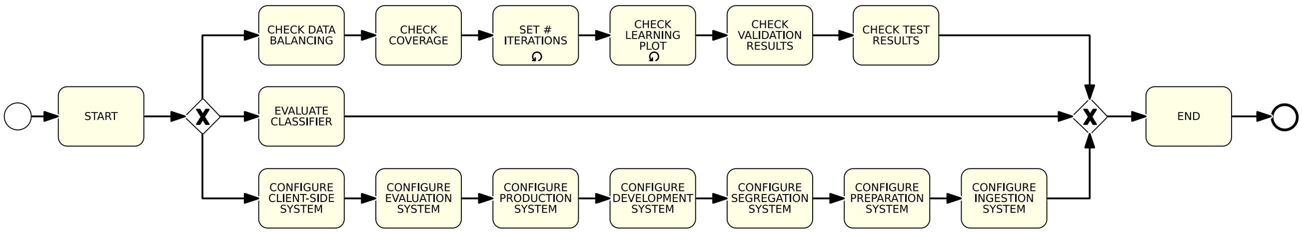
- Quantifying the strength of relationships between activities using measures such as dependency, frequency, or significance.
- Including all relationships that meet a predefined threshold, even if they are only marginally significant or rarely used.

This approach might include some noisy or unused relationships in the resulting model, leading to a potentially more complex or less precise model.
We can observe how it produces the most complex model with high fitness but low precision and generalization. It is suited for exploring complex and detailed processes but not ideal for simplification.

Heuristics miner with unused relationships

The Heuristics Miner with Unused Relationships is an extension of the standard Heuristics Miner, designed to provide additional insights into process models by including **unused relationships**, connections between activities that are identified in the event log but not utilized in the discovered model. This extension highlights **potential alternative paths** or relationships that could exist but are not part of the dominant behavior.

The key difference from the standard Heuristics Miner is that while the latter focuses only on the **most frequent paths** and filters out less significant connections, the extended version retains visibility of these unused relationships. This can help analysts identify **hidden behaviors** or evaluate whether certain relationships should be incorporated into the process model. However, including unused relationships may make the model **more complex** and harder to interpret, especially for noisy logs.

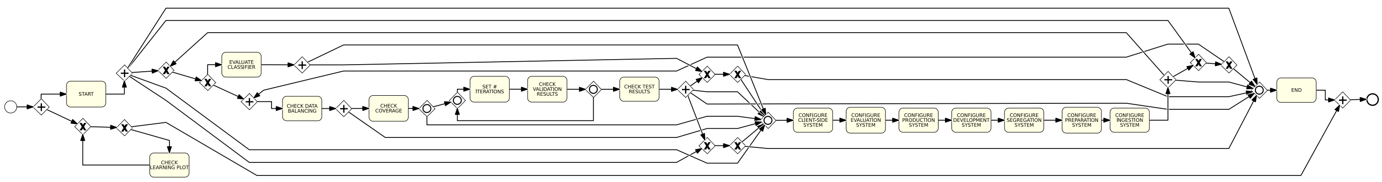


This version of the algorithm refines the process model by considering also unused relationships:

- It identifies and includes only those relationships that are part of the main behavioral patterns observed in the event logs.
- Unused relationships, those that are infrequent or irrelevant for the main process flow, are included in the model.

Alpha miner

The Alpha Miner is one of the earliest process mining algorithms, designed to discover **workflow models** from event logs. Its main strength is its **simplicity** and ability to detect basic patterns, such as **sequences**, **parallelism**, and **non-succession** in a structured manner.
However, the Alpha Miner has significant limitations: it cannot handle **noise** or **incomplete logs** and struggles with **short loops** or **complex behaviors**. It assumes perfectly structured event data, which makes it less suitable for real-world, unstructured processes. As a result, the discovered models may not always reflect the true behavior of the process.



For what regards this model, ProM could not compute precision and generalization score, thus they are not present in the following KPI table.

Four quality dimensions

Mining algorithm	Fitness	Simplicity #sequenceFlow(edges) + #activities + #gateways	Precision	Generalization
Inductive Miner with 0 Noise threshold	0,877	42+16+13 = 71	0,60199	0,98991
Inductive Miner with 0.2 Noise threshold	0,988	40+16+14 = 70	0,62071	0,988
Apromore	0,994	43+16+15 = 74	0,76959	0,9895
Fodina Miner	0,937	36+16+11 = 63	0,86642	0,98070
ILP Miner	0,998	33+16+8 = 57	0,60104	0,99334
Heuristics Miner	0.966	51+16+16 = 83	0,55415	0,64943
Heuristics Miner without Unused Relationships	0.84	21+16+2 = 39	0,87297	0,97552

- The **Fodina** model presents a high simplicity value but achieves excellent results in the other metrics. It presents the second highest precision and generalization, allowing us to say that it is one of the best models mined considering all measures together, although it is the third more complex among the four;
- The **ILP Miner** achieves the best fitness and generalization score, but its precision is rather low when compared to the ones achieved by the other models. Its complexity is in the average, making it difficult to choose the best model between this and the Fodina one;
- The **Heuristics** model is clearly the worst model regardless of the metric considered: it presents the highest simplicity and the lowest precision and generalization. Its variant, "**without unused relationships**", achieves the lowest simplicity recorded among all models, and very good results of precision and generalization, while the fitness is below the average but still acceptable.