PhLMT0002 Model Election Log

Version 1.0

Documentation Information

Contract Number	101069595
Project Website	www.safexplain.eu
Contratual Deadline	31.03.2024
Dissemination Level	SEN
Nature	R
Author	Jon Imaz
Modified by	Javier Fernández
Reviewed by	Lorea Belategi
Approved by	Irune Agirre
Keywords	Functional Safety, AI, DL, Learning Management, DL Models

This project has received funding from the European Union's Horizon Europe programme under grant agreement number 101069595.

Table of Contents

1 Review / Modification History

Version	Date	Description Change
V1.0	04/12/2023	First version after complete internal review
V0.1	18/09/2023	First draft

*Note: The paragraphs/name of the project/Rev./Ref./history table in **blue** must be replaced with the information for the specific project. The paragraphs written in **red** are instructions that can be used as a guide, so they must be deleted.*

2 Objective

The objective of this document is to collect the specifications of the DL candidate models to be generated.

3 Scope

This template applies to the collection of models designed through the Artificial Intelligence - Functional Safety Management (AI-FSM).

4 Model Design

The deliverable generated from this template must include all the information related to the Model Design step. This template provides the minimum information that should be collected in this step.

The following table should be replicated for each of the candidate DL models:

Model design	<Model_ID>_<version>
Date	Date of design: Format YYYY/MM/DD (Year/month/day)
Responsible	The person who designs the model
Phase of the lifecycle	Learning Management
Framework used	Specify the framework used to train the model: tensorflow, pytorch, keras, etc.
Model Format	Training model depends on the DL training framework employed: PyTorch (.pth), Keras (.h5), ONNX (.onnx)
Model Functionality	Specify the functionality of the model: detection, classification, etc.
Model Architecture	Specify the architecture of the model considered, including information such as the typology of layers (LSTM, CNN, RNN, Dropout, etc.)
Hyperparameters	Specify the hyperparameters used to train the model, including information such as: • Number of hidden layers, number of nodes per layer, etc. • Type of activation function of each layer: linear, tanh, relu, sigmoid, etc. • Learning rate: determines the step size at which the optimization algorithm updates the model´s parameters during training. • Type of loss function: Mean Squared Error (MSE), Mean Absolute Error (MAE), Huber Loss, Binary Cross-entropy, Multi-class Cross-entropy/categorical Cross-entropy… • Batch size: It refers to the number of training instances in the batch or the number of instances used per gradient update (each update equivalent to an iteration). • Epochs: number of times the model evaluates the entire training dataset • Optimizer: SGD, ADAM, RMSProp, etc.

Techniques used	If necessary, specify information about techniques that have been used to avoid overtraining or improve the generalizability of the model, such as:
	• Early Stopping: it stops training when no improvement in the validation metric is observed for a predefined number of epochs. In this case, specify the parameters used (patience, tolerance, etc.)
	• Regularization techniques:
	  ○ L1 and L2 Regularization: These techniques add penalty terms to the loss function based on the magnitudes of model weights. They encourage smaller weights, reducing the risk of overfitting.
	  ○ Dropout: During training, randomly set a fraction of the input units to zero at each update. This prevents the model from relying too heavily on any specific feature, promoting more robust representations.
	• Learning Rate Scheduling:
	  ○ Learning Rate Annealing: Gradually reduce the learning rate during training. This can help the model converge more effectively and avoid overshooting minima.
	  ○ Cyclical Learning Rates: Periodically increase and decrease the learning rate within certain bounds. This can help the model escape local minima and find better solutions.

Pretrained models	Specify if the model is trained from scratch or the source of the initial parameters. In the case of using pre-trained models, specify the path to the folder where they are stored.

5 Acronyms and Abbreviations

Below is a list of acronyms and abbreviations employed in this document:

- AI - FSM – Artificial Intelligence - Functional Safety Management
- ODD – Operational Design Domain
- DL – Deep Learning
- MAE – Mean Absolute Error
- MSE – Mean Squared Error (MSE)

6 Bibliography

Add here the reference to used bibliography / references (if any).