BUREAU VERITAS MARINE & OFFSHORE

GUIDANCE NOTE

GUIDELINES FOR **MACHINE LEARNING SYSTEMS**

NI692 R00

EDITION OCTOBER 2025

BUREAU VERITAS

BUREAU VERITAS
MARINE & OFFSHORE
GUIDANCE NOTE

NI692 ROO OCTOBER 2025

NI692 *Guidelines for Machine Learning Systems*, edition October 2025, is a new document.

The PDF electronic version of this document available on the Bureau Veritas Marine & Offshore website **https://marine-offshore.bureauveritas.com/** is the official version and shall prevail if there are any inconsistencies between the PDF version and any other available version.

These rules are provided within the scope of the Bureau Veritas Marine & Offshore General Conditions, enclosed at the end of Part A of NR467, Rules for the Classification of Steel Ships. The latest version of these General Conditions is available on the Bureau Veritas Marine & Offshore website.

BUREAU VERITAS MARINE & OFFSHORE

Tour Alto
4 place des Saisons
92400 Courbevoie - France
+33 (0)1 55 24 70 00

marine-offshore.bureauveritas.com/rules-guidelines

Guidance Note
NI692

GUIDELINES FOR MACHINE LEARNING SYSTEMS

October 2025

Table of Content

Table of Content

Section 1 Overview of Machine Learning Systems

1 Application

1.1 Scope

1.1.1 The aim of this Guidance Note is to provide:

- an overview of Machine Learning (ML) systems life cycle and definitions (see Articles [1] to [6])
- recommendations for the assessment of ML systems (see Sec 2)
- an overview of current international or national regulations and standards frameworks applicable to ML systems (see App 1)
- examples of risk assessment of ML systems (see App 2).

1.2 Exclusions and limitations

1.2.1 This Guidance Note focuses exclusively on ML systems and does not cover symbolic AI (see [2.1.2] and [2.1.3]).

Integration of ML systems as components within larger systems is out of the scope of this Guidance Note.

This Guidance Note does not cover language models (see Tab 1).

ML systems with risk profile classified as unacceptable risk according to the AI Act of the European Union (see App 1, [2.1.2]) are out of the scope of this Guidance Note.

This Guidance Note does not cover cybersecurity for which reference is made to NR659 "Rules on Cybersecurity for the Classification of Marine Units".

2 Machine Learning system - Life cycle - Terminology and definitions

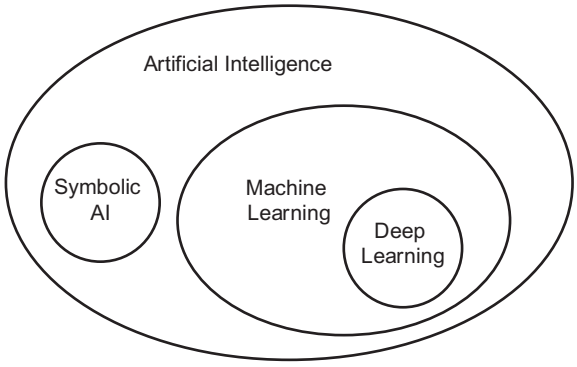
2.1 Taxonomy of Artificial Intelligence

2.1.1 Artificial Intelligence

AI encompasses various subsets, including Symbolic AI, Machine Learning (ML), and Deep Learning (DL) (see Fig 1).

Artificial Intelligence (AI) systems may be defined as structured sets of computational methods and models designed to process data, identify patterns, and generate outputs in alignment with predefined objectives.

Figure 1 : Taxonomy of Artificial Intelligence



2.1.2 Symbolic AI

Symbolic AI, also known as classical or logic-based AI, is a subset of AI that uses symbols and fixed logic to represent knowledge and perform reasoning.

Symbolic AI can be applied through rule-based systems, using if-then rules to process data and solve problems.

Due to their clear cause-and-effect relationships, they are predictable and straightforward, but they struggle with complex and non-linear data patterns.

For example, a rule-based system could be used for vessel compliance checks by applying a set of predefined rules to assess whether the design of a vessel and its equipments meet the corresponding requirements.

2.1.3 Machine Learning

Machine Learning is a subset of AI enabling systems to learn from data by automatically discovering patterns and handling complex and non-linear relationships without being explicitly programmed with rules.

Machine Learning (ML) systems use computational model that can adapt to unseen data and address evolving scenarios.

Continuous monitoring and dynamic adjustments are needed to handle unpredictable behaviours, and maintain effectiveness, reliability and accuracy of the ML system.

For example, a ML system could be employed for predictive maintenance in marine engines by analysing sensor data to detect patterns and forecast potential equipment failures.

2.1.4 Deep Learning

Deep Learning is a subset of ML that uses neural networks (structure of model inspired by the human brain) with multiple layers to learn and model complex patterns.

Deep Learning (DL) systems are particularly suited for tasks involving large datasets and real-time data with low inference.

They require constant monitoring and robust adjustment protocols to ensure performance stays within acceptable parameters and to manage any unexpected behaviours promptly.

For example, DL could be applied to vessel traffic monitoring, using deep neural networks to analyse radar or satellite data to detect and predict vessel movements, optimize routes, and prevent collisions.

2.2 Machine Learning system life cycle

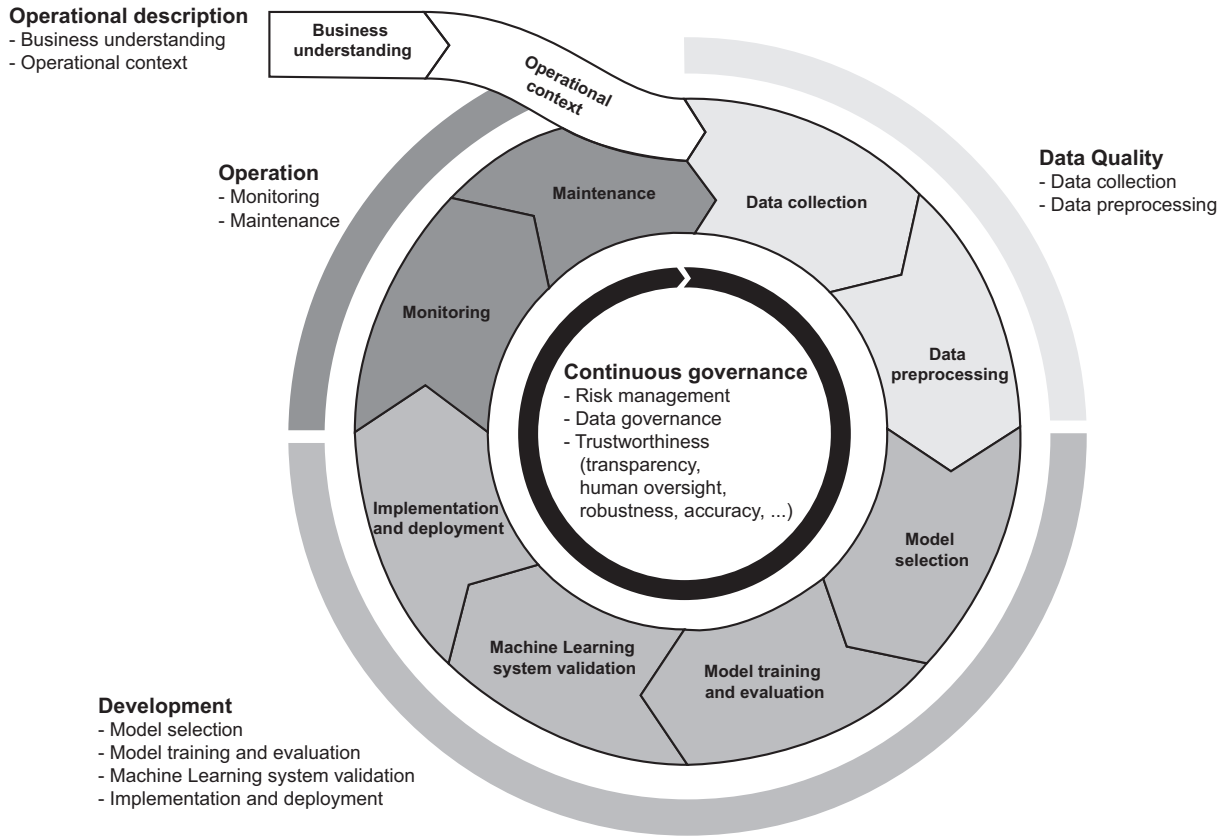
2.2.1 The life cycle of a ML system involves all stages from development and deployment to operation and maintenance.

It includes data collection and preprocessing (see Article [4]), ML system development and operation (see Article [5]), and ML system governance (see [2.2.2].

2.2.2 ML governance refers to the oversight of the ML system's behaviour and performance throughout its life cycle. It includes risk assessment and mitigation, data quality and governance, monitoring, maintenance, and alignment with trustworthiness principles (see [6]), in accordance with the ML system's operational context (see [3.2]).

2.2.3 An example of a supervised ML system life cycle is provided in Fig 2.

Figure 2 : Example of supervised Machine Learning system life cycle



3 Operational description

3.1 Roles and responsibilities: provider and deployer

3.1.1 An entity may act as both provider and deployer.

- Provider: entity responsible for developing a ML system and making it available for use.
- Deployer: entity responsible for adapting, operating and supervising a ML system.

3.2 Operational context - Terminology and definitions

3.2.1 Operational context

The operational context is a document constituted of the Concept of Operations, the Operational Envelope, and the Operational Design Domain. The Operational context defines the boundaries and domain within which the ML system is designed to operate.

3.2.2 Concept of Operations (ConOps)

The Concept of Operations (ConOps) provides high-level description of the ML system's objectives, intended domain of use, key functions, and operations. It sets out the boundaries within which the ML system is defined, developed, and tested.

3.2.3 Operational Envelope (OE)

The Operational Envelope (OE) describes the normal and degraded states in the ML system, the environmental and operational constraints, and anticipated failures:

- Normal state: state of the ML system when it is operating within its Operational Design Domain (ODD) and performing as intended without failures, anomalies, or environmental interference.
- Degraded state: state of the ML system when a failure has occurred or when environmental factors partially impair its function, but operation can continue without immediate risk or compromise. The ML system remains in the Operational Envelope.
- Fall-back state: state of the ML system when it is not in the Operational Envelope due to a failure or severe environmental conditions, and mitigation layers have been applied to transition it to a minimum risk condition.
  - Fall-back plans: manual procedures initiated after a ML system failure to restore operation.
  - Fail-safes: automatic actions triggered during a ML system failure to prevent further damage.
  - Redundancies: backup of ML system, input data and model to manage failures or inaccuracies.
- Contingency state: state of the ML system when it is not in the Operational Envelope and when the mitigation layers have failed to ensure a minimum risk condition.
  - Contingency plan: provides emergency procedures activated in contingency state to mitigate risks.

3.2.4 Operational Design Domain (ODD)

The Operational Design Domain (ODD) defines the operational and environmental conditions to be met for safe operations, and instructions under which the ML system should be operated to remain within the OE, and operated in emergency operations.

4 Data collection and preprocessing

4.1 Data quality

4.1.1 Data quality refers to the degree to which data is accurate, valid, unique, consistent, complete, up-to-date, available, and relevant for its intended purpose.

The reliability and outcomes of ML systems are directly influenced by the quality and relevance of the data they process.

Raw data may contain errors, inconsistencies, missing values, duplicates, or irrelevant information that introduce bias or noise. A larger volume of training data does not necessarily improve the model performance, especially for specific tasks. Excessive or irrelevant training data can reduce accuracy, and make the model miss nuanced distinction.

Data quality management needs to be maintained throughout the ML system life cycle, as evolving data sources, shifting distributions, and operational changes can affect the model behaviour over time.

4.2 Data collection - Terminology and definitions

4.2.1 Dataset

Structured collection of data that is measured, or generated, and is used as input.

4.2.2 Metadata

Data providing information on other data (e.g. source, type, description).

4.2.3 Development data

Data used to train, evaluate, and validate a model.

4.2.4 Training data
Largest portion of the dataset, which is used for training a model.

4.2.5 Test data
Portion of the dataset, which is used for evaluating the performance of a model.

4.2.6 Validation data
Portion of the dataset, which is used for validation of a model.

4.2.7 Production data
Real-world data used in a deployed model.

4.2.8 Label
Output that a model aims to predict, paired with an input feature (e.g. the label “tanker” or “bulk carrier” associated to an image in an image classification task).

4.2.9 Labelled data
Data that includes input features and corresponding output labels, used to train supervised learning models.

4.2.10 Unlabelled data
Data that includes only input features, used in unsupervised learning.

4.3 Data preprocessing - Terminology and definitions

4.3.1 Data preprocessing
Preparation and transformation of raw data before it is fed into a model (e.g. data integration, data cleaning, feature engineering, normalization, standardization, data partitioning).

4.3.2 Data integration
Process of combining data from multiple sources.

4.3.3 Data cleaning
Process of handling outliers, inconsistencies (e.g. errors, missing values), irrelevant attributes (e.g. errors, punctuation), and noise:

- Outlier: value that significantly differs from other observations (e.g. a value of 500 in a typical range of 50-100).
- Noise: data containing random or undesirable variations, errors, or inaccuracies that can negatively impact the performance of a model.
- White noise: random, uncorrelated data with a zero mean and constant variance. In some cases, some amounts of input noise can reinforce a model’s robustness and reduce overfitting, or extend data sample (e.g. data augmentation).
- Imputation: process of handling missing or incomplete data by substituting it with estimated values.

4.3.4 Data transformation
Process of applying operations on data to convert it into a suitable format for analysis and modeling (e.g. scaling, normalization, encoding categorical variables, feature extraction):

- Data scaling: process of adjusting numerical data to a comparable range so features contribute equally to the model's learning process:
  - Normalization: process of scaling features to a common scale, usually between 0 and 1.
  - Standardization: process of scaling features to have a mean of 0 and a standard deviation of 1.
- Data encoding: process of converting categorical data into numerical format that a model can train on (e.g. one-hot encoding, binary encoding).
- Data reduction: process of reducing the number of features without losing significant information (e.g. Principal Component Analysis).
- Feature engineering: process of selecting, extracting, transforming, and creating the most relevant features from raw data to align with intended ML system goal (e.g. calculating ship fuel efficiency by combining speed and engine power data).
- Embedding: process of transforming data into lower-dimensional space while preserving its meaning and relationships.
- Embedding analysis: process of analysing and visualizing embeddings to better understand relationships, detect anomalies, and improve feature selection (e.g. Principal Component Analysis applied to image embeddings).
- Data augmentation: process of artificially increasing the size and diversity of a dataset by applying transformations such as rotation, scaling, flipping, cropping, or noise injection.
- Class balancing: process of adjusting the number of instances in a class within an imbalanced dataset to ensure equal representation:
  - Oversampling: increases the number of instances in the minority class.
  - Undersampling: reduces the number of instances in the majority class.

4.3.5 Data partitioning

Process of dividing a dataset into distinct subsets to train, evaluate, and validate a model.

5 Machine Learning system development and operation

5.1 Models - Terminology and definitions

5.1.1 Model

Mathematical representation of a real-world process that makes analyses, predictions or decisions based on input data.

5.1.2 Task

Specific action or objective that a model is designed to accomplish.

5.1.3 Pattern

Recurring structure or sequence in data.

5.1.4 Parameters

Values that are learned from training data and determine how a model processes input to produce output (e.g. weights, bias, coefficients, support vectors, kernel matrices).

5.1.5 Hyperparameters

Top-level parameters influencing a model's learning process (e.g. learning rate, batch size, number of epochs, kernel type, number of trees).

5.1.6 Machine Learning algorithm

Set of mathematical procedures that a model follows to learn from data and generate outputs (e.g. linear regression, decision trees, neural networks).

5.1.7 Neural networks

Structure of model composed of interconnected nodes (neurons) arranged in layers, including an input layer, one or more hidden layers, and an output layer. These nodes are linked by adjustable weights, where each neuron processes input data through an activation function and passes the result to neurons in subsequent layers, enabling the network to learn patterns.

5.1.8 Architecture

Specific design of a neural network, defining how layers are structured and connected, how data is processed, and how features are extracted for a specific task (e.g. Convolutional Neural Networks, Region-based Convolutional Neural Network).

5.1.9 Hybrid model

Model combining multiple machine learning algorithms or learning approaches.

5.1.10 Learning approach

Strategy used to train a model:

- Supervised machine learning: the model learns patterns exclusively from labelled data. Each input is paired with a known correct output, helping the model to make accurate predictions.
- Unsupervised machine learning: the model identifies patterns independently, from unlabelled data.
- Semi-supervised machine learning: method using both labelled data to guide the learning process, and unlabelled data to improve the model's accuracy and generalization.
- Self-supervised machine learning: method using supervised machine learning algorithm on unlabelled data, where the model extracts the inherent structure of the data to generate implicit labels.
- Transfer learning: model trained for a specific task that is further trained on new data to adjust to a different task, preserving its acquired knowledge.
- Continuous learning: model continuously updated on new data.
- Reinforcement Learning (RL): method where an agent learns to make a sequence of decisions by interacting with an environment, receiving rewards or penalties (feedback) based on its actions.
- Agent: component of a RL system that can perform actions autonomously and interact with its environment.

Note 1: The prior list is not exhaustive as the field of ML is evolving rapidly. Other types of learning exist such as federated, multi-task, active, few-shot, meta-learning, zero-shot, curriculum, representation...

5.2 Models development and operation - Terminology and definitions

5.2.1 Training

Process of using data into a model, during which parameters are automatically adjusted to help the model learn underlying patterns and relationships in the data, while minimizing the error between the model's predictions and the ground truth (i.e. original values that a model aims to predict).

5.2.2 Retraining

Process of updating a model by training it on new data.

5.2.3 Fine-tuning

Process of adjusting parts of a pre-trained model for a new task or dataset.

5.2.4 Testing

Process of evaluating the model's success in performing the defined goal, and determining how well it generalizes to new, unseen data.

5.2.5 Generalization

Capacity of the model to perform well on unseen data.

5.2.6 Performance

Measure of the model's ability to accomplish its intended task.

5.2.7 Metrics

Quantifiable measures used to evaluate a model's performance across various aspects such as accuracy, robustness, efficiency and interpretability.

5.2.8 Validation

Process of evaluating a model using different methods to compare configurations and select the most optimal model (e.g. hyperparameter tuning, cross-validation).

5.2.9 Deployment

Process of implementing a trained model into a production environment.

5.2.10 Monitoring

Systematic and continuous tracking of a model's inputs, outputs, and internal processes to detect potential issues related to performance degradation, quality, risks, or unintended behaviour, to ensure alignment with expected goals over time.

5.2.11 Maintenance

Regular activities needed to maintain and improve a deployed model, including data updates, hyperparameter adjustments, technical issue resolution, infrastructure changes, as well as necessary updates to algorithms and datasets.

5.3 Classification of models

5.3.1 In addition to the definitions given in [5.1] and [5.2]:

- The Fig 3 provides an overview of learning approaches, tasks, and model algorithms.
- The Tab 1 provides common classification of models by task, along with typical ML algorithms and evaluation metrics.

Figure 3 : Learning approaches, tasks, and model algorithms

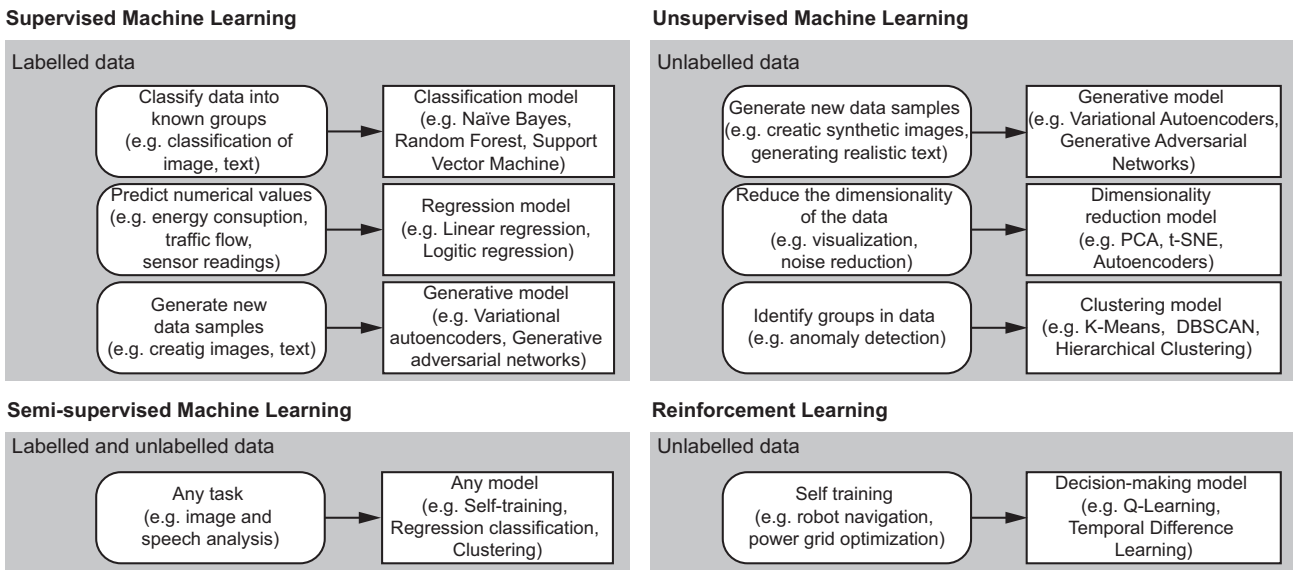


Table 1 : Examples of models

Task example	Definition	Machine Learning algorithm examples	Performance evaluation metrics examples
Anomaly detection	Identifies rare or unusual patterns in data that deviate from expected behaviour	• Autoencoders • Isolation forest	• F1-score • Precision-Recall Curve
Classification	Assigns input data to predefined categories or classes based on patterns learned from labelled training data	• Naïve Bayes • Random Forest • Support-Vector Machine (SVM) • k-Nearest Neighbours (k-NN)	• Accuracy • Precision • F1-score • Receiver Operating Characteristic curve (ROC)
Clustering	Groups similar data points together into clusters based on their inherent characteristics, without using predefined labels	• K-Means • Density-Based Spatial Clustering of Applications with Noise (DBSCAN) • Gaussian Mixture Models (GMM)	• Silhouette score • Calinski-Harabasz index
Computer vision	Analyses and interprets visual data, such as image classification or object detection	• Faster R-CNN • ResNet • U-Net	• Intersection over Union (IoU) • Mean Average Precision (mAP)
Decision-making	Independently analyses data, identifies patterns, and makes decisions or predictions without explicit human programming. It is designed to continuously update internal parameters based on new data and feedback, improving the performance over time.	• Q-Learning • Temporal Difference learning • State-Action-Reward-State-Action (SARSA)	• Cumulative reward • Average reward • Episode length • Success rate • Value function error
Dimensionality reduction	Transforms high-dimensional data (i.e. large number of features) into lower-dimensional representation (i.e. simplified form that retains essential patterns and relationships) to facilitate analysis and reduce overfitting	• Principal Component Analysis (PCA) • t-Distributed Stochastic Neighbour Embedding (t-SNE) • UMAP	• Explained variance ratio • Reconstruction error
Generative	Identifies the underlying patterns and distributions in the training data, and uses this knowledge to create new, synthetic examples that resemble the original data	• Variational Autoencoders, Generative Adversarial Networks (GANs) • Diffusion models	• Perplexity • Fréchet Inception Distance (FID) • Inception Score (IS)
Language understanding and generation	Processes and generates human language to perform tasks such as translation, question answering, and dialogue (e.g. language models, Large Language Models, foundation models)	• Generative Pre-training Transformer (GPT) • BERT	• Perplexity • BLEU • ROUGE
Regression	Predicts numerical output based on one or more input variables, using patterns learned from labelled data	• Linear regression • Polynomial regression	• Mean Squared Error (MSE) • R-squared • Mean Absolute Error (MAE)
Time Series Analysis	Identifies or learns patterns in temporal data to analyse and forecast data points ordered by time	• Auto-Regressive Integrated Moving Average (ARIMA) • Long Short-Term Memory (LSTM)	• Mean Absolute Error (MAE) • Mean Squared Error (MSE) • Mean Absolute Percentage Error (MAPE)

6 Trustworthiness and risks

6.1 Trustworthiness

6.1.1 Trustworthiness refers to the system's ability to be relied upon throughout its life cycle. It is defined by the interplay of multiple properties such as security, robustness, transparency, and explainability.

Trustworthiness characteristics may conflict, for example, a model may be accurate but not robust, or secure but biased.

The appropriate balance of these trade-off is highly contextual and should be considered in relation to the risks associated with the design or operation of the ML system.

6.2 Trustworthiness principles - Terminology and definitions

6.2.1 Accountability

Clear assignment of responsibilities to interested parties across all stages of the system life cycle.

6.2.2 Accessibility

Inclusive design providing equitable access to the system.

6.2.3 Explainability

Degree to which the model's internal logic and decision-making processes can be understood by humans.

6.2.4 Fairness

Impartial and equitable treatment or behaviour of the system, exempt of discrimination.

6.2.5 Human agency

Capacity of humans to make decisions and take actions autonomously in tasks affected by or involving AI.

6.2.6 Human oversight

Degree of human intervention in the system to ensure alignment with intended objectives:

- Controllability: extent to which the operator can intervene in the system's functioning.
- Human-In-The-Loop: human input in every decision cycle of the system.
- Human-On-The-Loop: supervision and capability for human intervention during the system's operation.
- Human-Out-Of-The-Loop: human intervention over the system when alerted.
- Human-Behind-The-Loop: human intervention after the system has completed its operations.

6.2.7 Privacy

Protection of personal information from unauthorized access or use.

6.2.8 Reproducibility

Consistent results under the same conditions.

6.2.9 Resilience

Ability to recover from unexpected events.

6.2.10 Robustness

Stable and reliable performance despite variation, noise, or perturbations in the data, or operating environment.

6.2.11 Safety

Ability of the system to safeguard information, data, functions, and its integrity when altered by unexpected failures or misuse.

6.2.12 Security

Ability of a system to identify, assess and mitigate risks to maintain its functions and confidentiality in the face of external changes and attacks.

6.2.13 Scalability

Ability to handle increasing amounts of data efficiently.

6.2.14 Traceability

Ability to track and document all processings of a system along its life cycle.

6.2.15 Transparency

Ability of the system to make its development, operations and decision-making processes accessible and understandable to everyone involved.

6.3 Risks - Terminology and definitions

6.3.1 Adversarial attack

Intentional modification of input data to cause the model to produce incorrect or unintended outputs.

6.3.2 Automation bias

Overreliance on model outputs by human operators.

6.3.3 Bias (in data and model behaviour)

Systematic difference in treatment introduced by the model or the dataset that may lead to unfair or unbalanced outcomes.

Note 1: Biases can have negative, neutral, or positive effects. Some biases are necessary for distinguishing between different input patterns (e.g. clustering or classification methods rely on certain biases to group inputs effectively), however, harmful biases should be identified and mitigated to ensure fairness.

6.3.4 Black-box model

Model whose internal workings or decision-making process is not fully transparent or explainable.

6.3.5 Confidentiality attack

Unauthorized access to protected or sensitive data through model outputs or interactions.

6.3.6 Data drift

Change in the distribution or characteristics of input data over time.

6.3.7 Data leakage

Unintended inclusion of target data in training data, leading to misleading generalization scores.

6.3.8 Data poisoning

Insertion of malicious or misleading samples into training data to manipulate or degrade the model's behaviour or outputs.

6.3.9 Hallucination

Generation of false or misleading content by the model, often presented with high confidence.

6.3.10 Inference attack

Extraction of sensitive or private information from the model's outputs.

6.3.11 Latency

Delay between receiving an input and producing an output.

6.3.12 Model decay

Gradual decline in model performance over time.

6.3.13 Model drift

Shift in model performance or behaviour due to evolving data patterns or changes in the environment.

6.3.14 Model poisoning

Embedding of hidden behaviours or triggers into a model during training (e.g. by altering weights or gradients), causing it to behave differently under specific conditions.

6.3.15 Out-of-distribution input

Input data that differs significantly from the training data, potentially leading to unreliable outputs.

6.3.16 Overfit

Excessive adaptation of the model to training data, resulting in poor generalization to new or unseen inputs.

6.3.17 Reward hacking

Exploitation of a reward function by the model or the agent to achieve high scores in unintended or undesirable ways.

6.3.18 Silent failure

Model failure without errors, alerts, or detectable anomalies.

6.3.19 Trade-off

Constraint in which improving one characteristic (e.g. accuracy, latency, interpretability) involves reducing another.

6.3.20 Underfit

Situation where the model is not complex enough or trained enough to capture patterns in the data.

Section 2 Machine Learning Systems Assessment

1 Documentation to be submitted for the assessment of a Machine Learning system

1.1 General

1.1.1 For the assessment of a ML system according to this guidance note, the documentation to be submitted is listed in Tab 1

This assessment should include the review of the documentation related to the ML system description, risk management, data quality and governance, model development, ML system development, and ML system operation.

1.1.2 If the ML system includes a hybrid model, all recommendations presented in this Guidance Note should be applied to each of the models.

Table 1 : Documentation to be submitted for the assessment of a Machine Learning system

	No.	Documentation	References
OPERATIONAL DESCRIPTION	1	Operational context	[2.1]
	2	Functional analysis	[2.2]
	3	Human oversight and automation	[2.3]
	4	Roles and responsibilities	[2.4]
RISK MANAGEMENT	5	Risk assessment	[3.1]
	6	Mitigation layers	[3.2]
	7	Bias assessment	[3.3]
	8	Impact assessment	[3.4]
	9	Risk, bias, and impact reassessment	[3.5]
DATA QUALITY and GOVERNANCE	10	Data collection	[4.1]
	11	Data preprocessing	[4.2]
MODEL DEVELOPMENT	12	Model design	[5.1]
	13	Model evaluation	[5.2]
MACHINE LEARNING SYSTEM DEVELOPMENT	14	Machine Learning system validation	[6.1]
	15	Technical environment	[6.2]
	16	Machine Learning system implementation	[6.3]
MACHINE LEARNING SYSTEM OPERATION	17	Monitoring	[7.1]
	18	Maintenance	[7.2]

2 Operational description

2.1 Operational context

2.1.1 The operational context (see Sec 1, [3.2.1]) is constituted of the Concept of Operations (see [2.1.2]), the Operational Envelope (see [2.1.3]), and the Operational Design Domain (see [2.1.4]).

2.1.2 The ConOps should document:

- tasks and functions of the ML system
- intended domain of use and foreseeable misuses
- limits of the ML system
- mode(s) of operation
- risk classification according to the AI Act (see App 1, [2.1.2]).

See example given in Tab 2.

Table 2 : Examples of ConOps

ConOps item	Description
Tasks and functions	• autonomous navigation and manoeuvring (with human supervision) • detect and avoid collisions with vessels, obstacles, and hazards in real-time (managed by the on-board sensors and processing units) • provide optimal route and speed based on weather, sea conditions, and traffic density (managed by the central server) • optimize fuel efficiency
Intended domain of use	• navigation in international waters under human supervision
Foreseeable misuses (i.e. maritime environment, constraints)	• inland navigation • ice navigation • operation in extreme weather (heavy fog, heavy snow, storm) • operation as fully autonomous without human supervision
Limits	• limited ability to interpret complex port traffic patterns • cannot operate effectively in severe weather conditions
Mode(s) of operation	• autonomous • remote from the Remote Operations Centre (ROC) • on-board
Risk classification	• high risk

2.1.3 The OE should document:

- the ML system's normal state
- the ML system's degraded state
- the ML system's fall-back state
- the ML system's fall-back plans
- the ML system's contingency state
- the ML system's contingency plans
- the type of interactions the ML system can handle with infrastructures and dynamic objects
- the environmental constraints
- the geographical constraints
- the operational constraints.

See example given in Tab 3.

Table 3 : Examples of Operational Envelope

OE item	Description
Normal state (i.e. functions during normal operations)	• detects and avoids collisions accurately • optimizes route navigation • operates within defined conditions • inputs (radar, LiDAR, GPS, cameras) are fully functional without anomalies • outputs (recommendations, actions) are fully functional without anomalies • stable weather conditions
Degraded state (i.e. triggers, capabilities during degraded states)	• minor failures (e.g. one sensor offline, outdated weather data) • moderate weather conditions affecting some sensors (e.g. heavy rain reducing camera effectiveness) • operates with redundancy or simplified model • recommendations and predictions are potentially less reliable
Fall-back state (i.e. triggers, consequences)	• critical failures (e.g. several sensors, processing units) • severe weather conditions impacting multiple sensors (e.g. storm affecting radar and LiDAR) • recommendations, predictions, and decisions are unreliable

OE item	Description
Fall-back plans (i.e. automatic and manual procedures)	• transition to manual operation mode • human expertise for unreliable predictions • automatic switch to backup models upon main model failure
Contingency state (i.e. triggers, capabilities during contingency state)	• severe system failure (e.g. hardware breakdown) • recommendations, predictions, and decisions are unavailable • unable to ensure minimum risk condition
Contingency plans	• emergency shut down of ML system • alert to ROC
Types of interactions (i.e. pedestrians, other vehicles)	Infrastructures: • port docking system • ship traffic management system Dynamic objects: • vessels • obstacles
Environmental constraints (i.e. temporal conditions, acceptable external factors)	• not operable in zero visibility (e.g. heavy fog) • operable in day and night conditions • operable in low-light conditions • operable in temperatures between -25°C and 45°C • operable in rain, fog, and moderate storms
Geographical constraints (i.e. acceptable maritime environments)	• not operable in polar regions • operable in high-traffic maritime routes • operable in international waters • operable in port area
Operational constraints (i.e. physical constraints, resource limitations, safety requirements)	• continuous power supply • continuous connectivity • operator availability

2.1.4 The ODD should document:

- the operational conditions to be met for operating the ML system
- the environmental conditions to be met for operating the ML system
- the instructions for normal operations
- the instructions for emergency operations (i.e. degraded, fall-back and contingency states)
- the foreseeable ML system malfunctions.

See example given in Tab 4.

Table 4 : Examples of Operational Design Domain

ODD item	Description
Operational conditions to be met for operating the ML system	• minimum of 90% data coverage from all sensors (radar, LiDAR, GPS, cameras) • ML model performance metrics within acceptable thresholds • real-time connection to shore-based support established
Environmental conditions to be met for operating the ML system	• visibility greater than 2 nautical miles • no extreme weather phenomena within 200 nautical miles • sea current speed and direction within predictable ranges
Instructions for normal operations	a) power on sensors, radar, and processing units b) verify availability and accuracy of data feeds (GPS, radar, camera, weather) c) monitor functions and actions (route changes, manoeuvring, collision avoidance) d) approve or override route suggestions e) monitor performance logs

ODD item	Description
Instructions for emergency operations (i.e. actions to be taken in the event of system failures, safety incidents, or other unexpected disruptions to ensure the ML system can be safely shut down, recovered, or transitioned to a safe state)	Degraded state: a) identify failures (e.g. camera offline) b) notify operators of failures c) switch to simplified model d) reduce vessel speed
	Fall-back state: a) transition to manual mode b) disable autonomous actions and route recommendations
	Contingency state: a) trigger alarms b) initiate emergency shut down c) alert Remote Operations Centre (ROC)
Foreseeable malfunctions	• misclassification of objects • delayed object detection • loss of input data • delayed input data

2.2 Functional analysis

2.2.1 A functional analysis should be conducted to identify and describe specific functions and sub-functions that rely on the ML system (see Tab 5).

Table 5 : Examples of functional analysis

Function	Description	Rely on ML system	Hardware item
F1	Provide user warning on possible collision of the ship with an external source of hazard	Yes	
F1.1	Detect possible target	Yes	
F1.1.1	Record and provide the computer with an image of the environment around the ship	No	Camera
F1.1.2	Preprocess the image	Yes	Processing unit - main
F1.1.3	Detect potential target on the image	Yes	Processing hardware - ML
F1.2	...		
F2	Monitor the system	Yes	Processing unit - main
F2.1	...		

2.3 Human oversight and automation

2.3.1 The degree of automation and human oversight of the system should be defined (see Tab 6 and Tab 7).

The degree of human oversight should be defined in accordance with the selected degree of system automation.

The compatibility between automation and human oversight may be mapped using the matrix representation (see Fig 1).

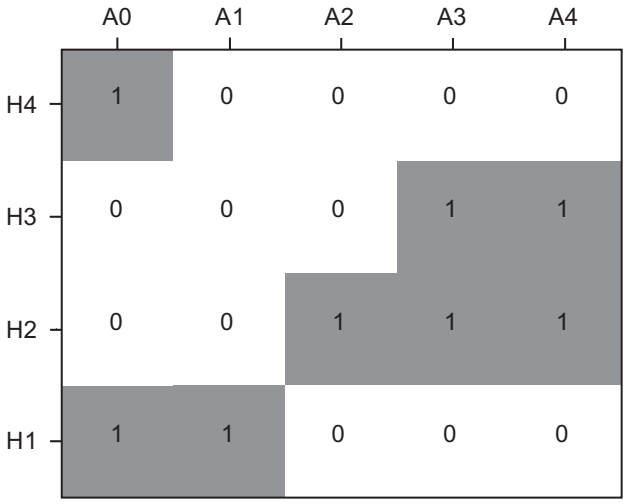
Table 6 : Automation degree

Automation degree		Definition
A0	Human operated	• Automated or manual operations are under human control • The ML system does not perform any actions
A1	Human directed	• The ML system provides analyses, forecasts, or recommendations to the operator • The ML system does not perform any actions
A2	Human delegated	• The ML system performs tasks and executes decisions automatically • The ML system decisions require explicit operator approval
A3	Human supervised	• The ML system operates autonomously • The ML system decisions do not require operator approval • The operator is always informed of the decisions and actions
A4	Full automation	• The ML system operates autonomously • The ML system decisions do not require operator approval • The operator is informed only in case of emergency

Table 7 : Human oversight degree

Human oversight degree		Action initiated by	Human controllability	Human monitoring	Human intervention
H1	Human-in-the-loop	Human	Yes	Yes	Real-time
H2	Human-on-the-loop	System	Yes	Yes	Real-time
H3	Human-out-of-the-loop	System	Yes	No	Real-time
H4	Human-behind-the-loop	System	No	No	Post-incident

Figure 1 : Matrix representation of human oversight and automation degree



2.3.2 Description of processes involving human-machine interactions should be provided.

The list of tasks (e.g. decision, action) should be provided, with indication of the autonomy and human oversight degrees (see Tab 8).

Table 8 : Examples of interactions between operator and ML system

Task	ML system output	Human task	Human oversight and automation
1	Identification of corrosion in bounding boxes	The operator is informed by the ML system that corrosion has been detected	A2 - H2
2	Corrosion classification	The operator confirms classification of corrosion proposed by the ML system	A1 - H1
3	Maintenance probability predictions and recommendations	The operator analyses probabilities, verifies recommendations and adjusts maintenance schedules	A0 - H4
4	...	...	...

2.4 Roles and responsibilities

2.4.1 Roles of providers and deployers may be:

- data provider: supplies raw data.
- data annotator: labels data for training.
- data engineer: prepares and processes data.
- bias analyst: examines datasets and outputs for potential biases.
- risk manager: assesses and mitigates risks in ML system.
- model provider (e.g. third-party suppliers): supplies pre-trained models.
- ML system provider: supplies functioning ML systems.
- model developer: develops, trains, evaluates and validates models.
- ML system developer: develops and deploys ML systems.
- operator / user (e.g. ship management, crew): interacts with ML system in operation and reports feedback.
- supervisor: oversees ML system operation, monitors performance, identifies and mitigates risks, has authority to intervene, and ensures alignment with operational context.
- ML system maintainer: updates and optimizes ML system based on incidents reported, user feedback, and technological advancements.

Note 1: Individuals may assume multiple roles within this framework.

2.4.2 Roles and responsibilities of all providers and deployers should be documented with sufficient level of details of responsibilities and tasks associated.

The documentation should include:

- role (see [2.4.1])
- responsibility
- accountability
- process stage where expertise is expected
- qualification
- any potential conflict of interest, and corresponding mitigation measures.

See example given in Tab 9.

Table 9 : Example of role and responsibilities

Role item	Description
Role	Data provider
Responsibility (e.g. function, tasks, expected expertise)	Provide historical data for development
Accountability	Accountable for the accuracy, completeness, and timeliness of provided data
Process stage of expertise	Data collection and data preprocessing
Qualification	Familiarity with maritime sensor systems, operational domain, and data risks
Conflict of interest and mitigation measure	Risk of biased data selection if data provider is involved in model evaluation
	Review of data quality by independent validation team

3 Risk management

3.1 Risk assessment

3.1.1 The risk assessment evaluates potential risks and hazards associated with the ML system, including mitigation layers and consequences on subsystems and components. It also identifies potential biases in the ML system design, data, or algorithms that could lead to unsafe or biased outcomes across the system's life cycle.

The risk assessment determines if adequate safety controls and mitigation layers are in place to ensure safe operation within acceptable risk levels.

3.1.2 Risks can emerge at any stage of the ML system life cycle from various sources such as third-party data, software, hardware, misuse or lack of transparency (e.g. previous preprocessing steps or identified risks may not be communicated at each stage of the ML life cycle). Therefore, the risk assessment should be applied to all stages of the ML system life cycle (see an example of risk assessment in App 2).

3.1.3 Each risk should be managed appropriately, with the corresponding metrics.

3.1.4 The risk assessment should include for each risk:

- hazards (e.g. erroneous ML system output, insufficient data analysis and preprocessing)
- causes (e.g. interpretation error, bias, poor quality of input data, data breaches)
- consequences (e.g. affected subsystems or performances)
- appropriate mitigation layers (see [3.2.1])
- if possible, severity of the impact (e.g. low, medium, high).

3.2 Mitigation layers

3.2.1 For each mitigation layer, the following elements should be documented:

- objectives of the mitigation layer
- justification for implementation
- implementation steps (i.e. tasks, methodologies)
- responsible roles (i.e. developer, supervisor, risk manager)
- resources required.

The rationale for each mitigation layer should be provided, and reasons for not implementing one should also be documented.

3.2.2 Mitigation measures should prioritize safety when the overall system is not behaving correctly.

3.2.3 The following mitigation layers should be implemented and documented:

- alerts for critical events requiring immediate attention
- fall-back plans and fail-safes
- redundancies (e.g. backup of ML system, input data, models)
- controllability measures (see [3.2.4])
- access restrictions (see [3.2.5])
- operator and supervisor training (see [3.2.6] and [3.2.7])
- human-machine interface (see [3.2.8])
- feedback processes (see [3.2.9])
- human agency measures (see [3.2.10])
- appropriate safety protocols related to domain and application
- for RL agents, circuit breakers
- for RL agents, reward function safeguards to prevent reward hacking.

3.2.4 Operators should be able to halt, shut down, or override the ML system at any time.

3.2.5 Mechanisms should be in place to ensure that access is restricted to authorized personnel only.

3.2.6 Operators should be trained to understand:

- intended uses and foreseeable misuses of the ML system
- functions and limitations of the ML system
- procedures for controlling the ML system (see [3.2.4])
- interpretation of outputs
- detection of anomalies, unexpected behaviour, or performance degradation
- feedback and reporting procedures for detected issues
- appropriate and inappropriate interactions with the ML system.

3.2.7 Supervisors should be trained to demonstrate clear understanding of the following:

- all items listed in [3.2.6]
- generic and context-specific ML system risks and vulnerabilities
- techniques for monitoring the ML system operation (e.g. quality of outputs, effectiveness)
- identification and mitigation of risks and automated bias.

3.2.8 A dedicated human-machine interface should be provided to operators and supervisors, displaying:

- real-time visualization of runtime evaluations (see [6.3.7])
- manual override, pause, and emergency shut-down functionalities
- OE state
- alert notifications for unexpected behaviours.

3.2.9 Feedback processes should be established to allow operators to:

- report anomalies or unexpected behaviours to supervisors and maintainers
- log significant events (including timeliness, incident details, notified operators, corrective measures applied).

3.2.10 Measures should be taken to prevent operator overconfidence and preserve human agency, such as:

- regular training sessions on ML system limitations and potential biases
- automated warning alerts for high-risk decisions
- human cross-verification for critical outputs
- monitoring of operator reliance on the ML system.

3.3 Bias assessment

3.3.1 The bias assessment evaluates the presence of systematic deviations in the ML system such as biases introduced by unbalanced datasets, underrepresentation of operational scenarios, or limited data diversity that may affect model robustness, reliability, or generalizability.

3.3.2 The bias assessment should include, for each identified bias:

- description of the metrics and methods employed for bias detection (e.g. statistical measures, human expertise)
- justification for the chosen metrics and methods
- bias
- causes
- consequences
- appropriate mitigation layers (see [3.2.1])
- notified interested parties (e.g. deployer, operator, data annotator).

3.4 Impact assessment

3.4.1 The impact assessment evaluates intended uses and foreseeable misuses of ML system on society and individual's lives, including impacts on the environment, human rights, and psychological well-being.

3.4.2 The impact assessment should include for each intended use and foreseeable misuse:

- impacts (e.g. effects on human rights, privacy violations)
- causes (e.g. lack of transparency, bias)
- consequences (e.g affected individuals or societies, loss of trust)
- mitigation layers (e.g. plans for managing failures, update process, transparency efforts)
- if possible, severity of the impact (e.g. low, medium, high).

3.5 Risk, bias, and impact reassessment

3.5.1 Risk, bias, and impact reassessment may be required to the satisfaction of the Society in the following cases:

- update in sensitivity of data
- update in complexity of the ML system
- new regulatory requirements
- new operational environment
- significant performance degradation.

3.5.2 For risk, bias, and impact, the deployer should specify:

- the frequency of reassessment (e.g. annually, for each maintenance update)
- the triggers for the need of a new or updated assessment (e.g. new regulations, integration of new data sources).

3.5.3 The risk, bias, and impact reassessment may be required at all stages of the ML system life cycle (i.e. data processing, development, and operation).

4 Data quality and governance

4.1 Data collection

4.1.1 All dataset collection (i.e. training, testing, validation, and production data) should be documented, justifiable, maintained and traceable (e.g. metadata, methodology for selection, known biases).

4.1.2 When collecting data, the provider and deployer should remain objective and avoid altering the original data values in ways that introduce bias or errors.

Bias can arise from actions such as:

- modifying values to make results appear more favourable
- including only certain data points that support a specific outcome while disregarding others
- rounding or truncating data in ways that misrepresents the true distribution.

4.1.3 The ML system may integrate data from various sources such as data from similar domain applications, and data from equivalent sources.

Noisy data not implemented during training phase, may be integrated into another set (i.e. to enhance robustness).

4.1.4 As models are typically trained on small portions of the actual data distribution, as far as possible, data selected for training models should represent the real-world operational domain and conditions.

4.1.5 Procedures should be specified for data storage, including the period for which it will be kept, and the secure methods for its disposal afterward.

4.1.6 The provider and deployer should document the metadata, including:

- data provenance
- description
- reasoning for dataset selection
- data rights
- data modalities (inputs and outputs)
- data type
- dataset size
- timeliness (collection and last update)
- prior preprocessing
- potential biases
- storage location of data
- retention period.

See example given in Tab 10.

Table 10 : Examples of metadata

Metadata item	Development data		Production data	
Dataset reference	Dataset 1A	Dataset 2A	Dataset 1B	Dataset 2B
Data provenance	• dataset: name AAA • author: company WWW	• dataset: name BBB • author: company XXX	• real-time sensor data (radar, LiDAR, GPS, cameras) • author: company YYY	• dataset: name CCC • author: company ZZZ
Description	Annotated dataset of 50000 images for vessel and obstacle detection	Automatic Identification System (AIS) vessel tracking dataset	Continuous sensor logs for navigational adjustments and fuel optimization	Real-time weather condition reports, including temperature, wind speed and other environmental factors
Reasoning for dataset selection	Fine-grained dataset for ship detection in high-resolution optical remote sensing images	Low error rate dataset, collected by industry-standard marine sensors	Sourced from actual operational environment	Real-time weather information with high accuracy
Data rights	• company WWW terms of use	• open licence for non-commercial use	• owned by operator • use for operational purposes only • subject to operator's terms	• API subscription with commercial usage rights • subject to third-party licensing
Data modalities inputs	• labelled images	• historical sensor data (radar, LiDAR, GPS) • annotated historical logs (weather, sea state, fuel metrics)	• real-time sensor data (radar, LiDAR, GPS, cameras, fuel metrics, speed, sea conditions) • real-time images and videos from cameras	• real-time external weather updates, wind speed, sea state, storm alerts
Data modalities outputs	• routes suggestions (GPS coordinates, navigational paths in map format) • collision bounding boxes and alerts • fuel efficiency metrics			
Data type	• categorical: object labels • numerical: pixel values for images • binary: video footage • string: images annotations	• float: distance, wave height, speed, fuel metrics, coordinates • string: weather conditions, sea conditions	• numerical: raw pixel values for images • binary: video footage • float: fuel metrics, speed, distance, coordinates • string: sea conditions	• float: wind speed • string: storm alerts, weather, sea conditions
Data size	4 TB	6 TB	1 GB/hour	500 MB/hour
Timeliness records for collection	Jan 2023 - Dec 2024	Jan 2022 - Dec 2024	Streaming initiated in Jan 2025	Jan 2025 - ongoing

Metadata item	Development data		Production data	
Timeliness records for last update	Dec 2024	Jan 2025	Real-time stream, ongoing	Real-time stream, ongoing
Prior preprocessing	No prior preprocessing known	Outliers and erroneous sensor readings have been removed from the dataset before collection	No prior preprocessing known	Raw data received directly from external APIs
Potential biases	Most data was gathered during daylight, which might affect model accuracy in night time conditions	Limited representation of smaller vessels	None identified	None identified
Storage location of data	• encrypted cloud infrastructure • local backups at ROC		• encrypted cloud infrastructure with real-time replication	
Retention period	Retained for 5 years		Retained for 12 months	

4.2 Data preprocessing

4.2.1 For each dataset (i.e. training, test, validation, and production), the provider and deployer should document all preprocessing and processing of data, including:

- data preprocessing (see [4.2.2] and [4.2.3]):
  - data quality (see [4.2.4] and Tab 11)
  - data integration
  - data cleaning
  - data transformation (e.g. feature engineering, data scaling, encoding, data reduction)
  - data partitioning (see [4.2.5])
- data processes:
  - bias assessment (see [3.3])
  - legitimate purpose for use of data (see [4.2.6])
  - labelling process (see [4.2.7])
  - access restriction measures (see [4.2.8])
  - confidentiality measures (see [4.2.9])
  - personal data handling process (see [4.2.10]).

4.2.2 For each preprocessing steps (see [4.2.1]), the following should be described:

- the purpose of each step (e.g. improve consistency, reduce noise, align formats)
- the method used (e.g. statistical validation, data profiling)
- the acceptance criteria or thresholds
- the justification for the selected criteria and method.

4.2.3 As far as practicable, identical preprocessing procedures as training data should be applied to production data to ensure consistent data are fed into the model.

4.2.4 The provider and deployer should define context-specific conditions for data quality, considering at minimum the dimensions defined in Tab 11.

4.2.5 The training set should not share instances with other datasets to prevent data leakage.

4.2.6 Preprocessing techniques should be limited to specified, explicit, and legitimate purposes.

4.2.7 When applicable, data labelling should be performed by personnel with the appropriate level of expertise in the relevant field covered by the ML system, and mechanisms such as review and cross-checking should be implemented, as far as practicable, to ensure accurate labelling.

4.2.8 Appropriate methods should be implemented to secure data against unauthorized access, unlawful processing, accidental loss, destruction, or damage.

4.2.9 Confidentiality measures should be implemented, such as:

- Data Protection Impact Assessments (DPIAs)
- data anonymization
- data encryption
- data pseudonymisation.

4.2.10 When applicable, individuals should be allowed to access to their personal data and the ability to rectify inaccuracies. Requests for the deletion of personal data should be facilitated through appropriate protocols.

Table 11 : Data quality dimensions

Data quality dimension	Condition examples
Completeness	Impute missing values
Accuracy and validity	Cross-check labels against verified sources
Uniqueness	Detect and remove duplicate entries
Consistency	Standardize formats (e.g. timestamps, units of measure)
Timeliness	Set latency thresholds for real-time input data
Availability	Implement fall-back mechanisms for missing or delayed real-time inputs
Relevance	Filter data based on operational domain
Level of detail and resolution	Filter resolution for low-granularity data
Volume	Apply data augmentation for underrepresented classes
Real-world representation	Integrate data from diverse operational and environmental scenarios

5 Model development

5.1 Model design

5.1.1 The choice of model should correspond to tasks objectives (see [2.1.2]), data characteristics (see [4.1.6]), and learning approach.

5.1.2 The provider should transparently, safely, and responsibly address the model's trade-offs, using a justifiable approach to weigh risks, impacts, costs, and benefits.

5.1.3 The provider should document the model characteristics, including:

- task, ML algorithm, architecture, and learning approach
- reasoning for model selection
- when applicable, details on model origin (e.g. pre-trained, transfer learning)
- description and justification of the development process (e.g. training loss curve, hyperparameter configuration)
- description of the model's capabilities and limitations
- model versioning
- risk and bias assessments (see [3.1] and [3.3]).

See example given in Tab 12.

5.1.4 For RL agents, the provider should document the model characteristics, including:

- all items listed in [5.1.3]
- agents and intended goals
- justification for reward functions and structures
- reward process based on the agent's performance
- environments
- actions and interactions
- states and transitions based on actions
- justification for balance between exploration and exploitation
- criteria and procedures for gradually expanding the exploration zone.

Table 12 : Examples of model characteristics documentation

Model item	Description
Task, ML algorithm, architecture and learning approach	CNN ResNet50 for image-based object detection and classification, supervised learning
Reasoning for model selection	Convolutional Neural Networks (CNN) selected for object detection due to high accuracy in detecting vessels and obstacles in maritime environments under varying weather and lighting conditions
Model origin	Developed model using pre-trained ResNet50 for feature extraction
Development process	• Model trained using cross-entropy loss and Adam optimizer (learning rate = 0,001 ; batch size = 32) over 50 epochs with early stopping • Evaluation during training used IoU and precision metrics • Based on validation results, the learning rate was reduced and class weighting was introduced to address imbalance • Final model selected via 5-fold cross-validation
Model capabilities and limitations	• Performs well in daylight and moderate weather • Reduced accuracy in low-light or obstructed scenes
Model versioning	• Version 2.1 • Retrained in 2025 with expanded dataset coverage
Risk and bias assessments	See [3.1], [3.3], and example given in App 2

5.2 Model evaluation

5.2.1 The model characteristics documentation (see [5.1.3] and [5.1.4]) and the evaluation report (see [5.2.3]) should be communicated to the deployer.

5.2.2 The provider should evaluate the model across the following dimensions:

- performance (e.g. accuracy, loss, regression error, error rate)
- robustness (e.g. adversarial testing, perturbation testing)
- reproducibility
- confidence (e.g. prediction confidence, uncertainty estimation)
- security (e.g. red teaming)
- model complexity (e.g. inference time, memory usage)
- explainability (e.g. SHAP, visualization tools)
- interpretability (e.g. human understanding of model outputs).

5.2.3 For each evaluation dimension (see [5.2.2]), the provider should conduct an evaluation report documenting:

- one or more appropriate metrics
- metric name and description
- justification for the relevance of the metric to the intended task
- possible range of values (e.g. between 0 and 1) and interpretation (e.g. higher values indicate better performance)
- criteria for acceptable scores (i.e. range, minimal value, maximal value) and justification
- scores.

See example given in Tab 13.

Table 13 : Example of evaluation report

Evaluation item	Metrics and description	Justification	Score range	Acceptable criteria	Scores
Performance	Accuracy: proportion of correct predictions over total predictions	Measures generalization across training, testing, and validation datasets	0 – 1 Higher indicates better generalization	≥ 0,90	• Training: 0,92 • Testing: 0,905 • Validation: 0,918 Values > 0,90 suggest reliable classification across conditions
	Intersection over Union (IoU): overlap between predicted and ground truth bounding boxes	Relevant for object detection precision	0 – 1 Higher means tighter bounding box alignment	≥ 0,75 (mean IoU)	• Mean: 0,77 • Min: 0,69 • Max: 0,84 Values > 0,75 indicate good spatial accuracy

Evaluation item	Metrics and description	Justification	Score range	Acceptable criteria	Scores
Robustness	Adversarial perturbation test: accuracy under controlled input perturbations with Fast Gradient Sign Method	Evaluates model resilience to noise or manipulation	0 – 1 Higher means model remains stable under adversarial noise	≥ 0,85	• Mean: 0,88 • Min: 0,83 • Max: 0,89 Values > 0,85 indicate strong robustness
	Out-of-distribution (OOD) accuracy: accuracy on inputs outside training distribution	Evaluates model behaviour under unexpected operational conditions	0 – 1 Higher means better generalization to unseen environments	≥ 0,80	• Mean: 0,82 • Min: 0,76 • Max: 0,84 Values > 0,80 suggest acceptable adaptability
...	...	...	...	...	...

6 Machine Learning system development

6.1 Machine Learning system validation

6.1.1 The deployer should define and conduct validation procedures to evaluate the ML system using production data. These procedures should demonstrate that the ML system performs as expected under real-world conditions and meets defined trustworthiness criteria (e.g. robustness, reliability, controllability, interpretability).

6.1.2 The following validation procedures should be defined:

- reliability evaluation (i.e. all evaluation dimensions listed in [5.2.2])
- controllability testing (i.e. operator ability to halt, override, and stop the ML system)
- real-world testing (i.e. trials under expected operational conditions)
- OE state testing (i.e. ML system behaviour under normal, degraded, fall-back, and contingency states)
- ODD condition testing (i.e. ML system behaviour under expected and unexpected environmental, geographical, and operational constraints)
- interoperability testing (i.e. compatibility with the broader system)
- operator testing (i.e. completeness and availability of technical documentation, training of operators).

6.1.3 Validation procedures may rely on one or more methods, including:

- statistical (i.e. quantitative analysis of system performance metrics under varying conditions)
- formal (i.e. mathematical models and logical reasoning to prove the system's properties and behaviours)
- empirical (i.e. extensive testing to observe the system's behaviour)
- simulation (i.e. controlled virtual environments to mimic real-world conditions and observe the system’s responses)
- evaluation (i.e. assessing the system based on defined quality attributes such as usability or reliability).

6.1.4 The deployer should provide a validation report. The report should include, for each validation procedure:

- validation procedure name (see [6.1.2])
- methods used (see [6.1.3])
- description of the methods (i.e. definition, methodology, elements tested)
- acceptable result criteria
- results and guidance on interpretation.

See example given in Tab 14.

6.2 Technical environment

6.2.1 The deployer should document the technical prerequisites, including:

- the device location of computing resources
- list of software and required version
- list of libraries and required version
- list of interactions, dependencies, and data exchange between different software components
- list of tools and equipment needed to troubleshoot, and support the ongoing operation of the ML system.

See example given in Tab 15.

6.2.2 Users should be clearly informed when interacting with a ML system.

6.2.3 Technical documentation on the ML system should include:

- ML system's intended purpose
- ML system use instructions (see [2.1.4] and Sec 1, [3.2.4])
- metadata (see [4.1.6])
- model characteristics (see [5.1.3] and [5.1.4])
- evaluation report (see [5.2.3])
- validation report (see [6.1.4])
- technical prerequisites (see Tab 15)
- risk, bias, and impact assessments (see [3.1], [3.3], and [3.4]).

6.2.4 The deployer should demonstrate that technical documentation (see [6.2.3]) is accessible to all users (e.g. "General information is available in the instruction manual, referenced as X1, accessible to all operators on main interface").

Table 14 : Example of validation report

Validation procedure	Methods and description	Acceptable result criteria	Results
Controllability Testing	Operator override simulation (empirical): test the ability of operators to interrupt, pause, or override the ML system in real-time scenarios. Methodology includes scripted interventions during simulated tasks.	• Control functions responsiveness ≤ 1 s • Override successfully interrupts ML system decisions • No missed or failed interventions	• All control functions triggered successfully • Average response time: 0,78 s • Max response time: 0,95 s • Intervention success rate: 100% • The ML system is fully controllable by operators • All override commands were executed within the required time frame, with no failures or delays • No critical failures observed
	Control latency benchmarking (statistical): measure the time delay between operator input and ML system response across multiple trials, and analyse the consistency and failure rate	• Mean latency ≤ 1 s • Standard deviation ≤ 0,2 s • 95th percentile ≤ 1,2 s	• Mean latency: 0,74 s • Standard deviation: 0,12 s • 95th percentile: 0,96 s • The ML system consistently responds to operator commands well within the required thresholds • Low standard deviation indicates stable responsiveness, and the 95th percentile confirms that even in rare cases, latency remains acceptable • No failures or missed responses were recorded
OE state testing	State transition simulation (empirical, simulation): evaluation of ML system stability, error handling, and recovery under normal, degraded, fall-back, and contingency states	• No critical failures in any state • Recovery time ≤ 2 s	• All states handled successfully • Average recovery time: 1,4 s • Max recovery time: 1,9 s • The ML system transitions smoothly between operational states, maintains safety, and recovers within acceptable time limits • No critical failures observed
...	...	...	...

Table 15 : Examples of technical prerequisites

Technical item	Description
Device location of computing resources	On-premise edge computing servers installed on vessel with cloud synchronization
Software	• Operating System name - v20.04 • software name - v3.9
Libraries	• library name - v2.17.0
Interactions, dependencies, data exchange	Sensors to ML system: • radar, LiDAR, GPS, and camera feeds streamed via Ethernet ML system to navigation system: • route recommendations transmitted via API ML system to cloud: • model updates and performance logs synchronized via satellite communication
Support tools	• diagnostic tool name • health monitoring tool name • technical manual

6.3 Machine Learning system implementation

6.3.1 The deployer should implement automated safeguards to be triggered in the following cases:

- runtime evaluation scores fall outside the acceptable criteria (see [6.3.8])
- input data fails to meet quality thresholds (see [4.2.4])
- input data is unavailable (e.g. sensor failure, communication loss)
- the ML system is operated outside its intended domain of use
- the ML system is halted by an operator
- the ML system is shut down
- anomalies or unexpected behaviours are detected.

6.3.2 For each trigger, the deployer should define:

- the detection method (e.g. threshold check, data profiling, anomaly detection)
- the automated response (e.g. operator notification, degraded state, suspension of autonomous actions)
- operators to be notified.

6.3.3 The deployer should implement automatic logging mechanisms throughout the ML system life cycle, including:

- runtime evaluations (see [6.3.7])
- operation logs (see [7.1.5])
- significant event logs (see [7.1.7])
- maintenance reports (see [7.2.8]).

6.3.4 The logging frequency (e.g. per decision cycle, per minute) should be defined by the deployer in accordance with the ML system's operational context, risk classification, and degree of automation.

6.3.5 The deployer should define retention periods for logs, which should be sufficient to support auditability and post-incident analysis.

6.3.6 Mechanisms should be in place to ensure that logs are securely stored and accessible to authorized personnel only.

6.3.7 Runtime evaluations should log scores for each metric (see [6.3.8]), output confidence scores, and input data quality checks.

Additionally, for RL agents, runtime evaluations should log agents, environments, states, actions, and rewards.

6.3.8 The deployer should define the acceptable criteria for runtime evaluation metrics, including:

- evaluation metrics
- acceptable criteria for each metric
- acceptable output confidence score
- for RL agents, boundaries for safe exploration zone
- for RL agents, limits for novel actions per episode.

6.3.9 The acceptable criteria establish the range, minimum, or maximum acceptable score for specified metrics, beyond which the ML system would be considered unreliable for its intended use.

6.3.10 The acceptable criteria should be justifiable and as stringent as possible.

6.3.11 The acceptable criteria are subject to the agreement of the Society, who may question or reject them to ensure the ML system meets appropriate security and safety measures and prevents any potential misuse (e.g. a maximal error rate score of 0.3 may be considered insufficient).

6.3.12 Any factors that might justify exceeding the acceptable criteria in specific situations should be specified (e.g. heavily blurred or partially cropped images in object detection system).

7 Machine Learning system operation

7.1 Monitoring

7.1.1 The deployer should operate the ML system in accordance with the operational context defined in [2.1].

7.1.2 The deployer should monitor the ML system, including:

- data quality of inputs (e.g. data quality checks, data quality for reinforcement learning activities, data drift)
- accuracy of outputs (e.g. performance degradation, model drift)
- domain of use (see [7.1.1])
- operators behaviour and interactions with the ML system (e.g. misuse, inappropriate interaction)
- for RL agents, agents, environments, states, actions, and rewards.

7.1.3 Processes to monitor, analyse, and evaluate the ML system should be documented and maintained.

7.1.4 Continuous learning activities and RL agents should be continuously supervised, interpreted and documented by a qualified human expert.

7.1.5 Logs of the ML system's operations should include:

- time range
- ML system actions
- Operational Envelope state
- environmental, geographical, and operational conditions
- significant event log, if any
- runtime evaluation (see [6.3.7]).

See example given in Tab 16.

7.1.6 All significant events (e.g. safeguard process activation, fails, data drift, ML system misuse, incorrect behaviour of the model) and suspected threats (e.g. unauthorized access attempts, adversarial attacks) should be documented.

Table 16 : Examples of operation log

Operation log item	Description
Time range	10:00:00 (GMT+1) 01/01/2025 - 10:59:59 (GMT+1) 01/01/2025
ML system actions	• route optimization • autonomous navigation • object detection
Operational Envelope state	• degraded state
Environmental, geographical, and operational conditions	• heavy rain • daytime • high-traffic maritime route
Significant event (e.g. exception, fault, triggered safeguard)	Significant event report 1 (see example given in Tab 17)

Runtime evaluation	Evaluation metrics and acceptable criteria:
	• latency <= 1.5 s
	• error rate <= 0.08
	• confidence >= 80%
	Scores:
	• latency = 1.2 s
	• error rate:
	  - mean = 0.064
	  - min = 0.02
	  - max = 0.07
	• confidence (per output):
	  - mean = 82%
	  - min = 81%
	  - max = 91%
	Input data quality checks:
	• data completeness = 92%
	• data availability = all sensors active
	• timeliness = within 1 s threshold

7.1.7 The operator should document the significant event report, including:

- time range of the event
- description of the significant event
- reason of incident
- affected subsystems
- potential impacts on each affected subsystems
- mitigation measures and action undertaken
- corrective actions for next maintenance
- notified operators.

See the example given in Tab 17.

7.1.8 The deployer should report relevant incidents necessitating corrective updates to the provider.

Table 17 : Examples of significant event report

Incident item	Incident 1
Time range	10:02:10 (GMT+1) 01/01/2025 - 10:02:40 (GMT+1) 01/01/2025
Description of the significant event	False positive collision warning
Reason of incident	Radar noise due to intense weather conditions
Affected subsystems	• collision avoidance system • autonomous navigation system • radar data
Potential impacts on affected subsystems	• increased frequency of false collision alerts making the collision avoidance system unreliable • autonomous navigation system temporarily disabled to prevent unpredictable behaviour • radar data is unreliable due to excessive noise
Mitigation measures and actions undertaken	• switch to manual control
Corrective actions for next maintenance	• update data preprocessing for radar filtering • retrain model with noisy data for robustness
Notified operators	• on board crew notified on January 1st 2025 • ML monitoring team notified on January 1st 2025

7.2 Maintenance

7.2.1 ML systems should be maintained regularly to ensure performance, consistency, and validity over time (e.g. implementation of new mitigations layers, retraining with new data, software update).

7.2.2 Schedule and methods for regular maintenance of ML systems should be defined and implemented.

7.2.3 Procedures should be put in place to improve the ML system based on operators' feedback.

7.2.4 As far as practicable, identical preprocessing procedures as training and production data should be applied to new data to ensure consistent data are fed into the model.

7.2.5 Maintenance updates should be in accordance with the operational context defined in [2.1].

When changes impact the operational context, or part of it, the ML system should be subject to a new approval.

7.2.6 A qualified human operator should monitor live updates of the model, retraining, new data injections, and outputs in accordance with the operational context defined in [2.1].

7.2.7 All updates should be justifiable, documented, and should go through each stage of the ML system life cycle (e.g. data quality checks in data preprocessing, model evaluation in development).

7.2.8 The deployer should document the maintenance report, including:

- time range
- update description
- reasoning for update
- notified operators
- evaluation report
- validation report
- risk, bias, and impact reassessment.

See example given in Tab 18.

Table 18 : Examples of maintenance report

Maintenance item	Description
Time range	00:00:00 (GMT+1) 15/01/2025 - 00:12:30 (GMT+1) 15/01/2025
Update description	Updated object detection model trained on extended dataset
Reasoning for update	Improve detection accuracy of small vessels at night and in moderate fog, based on operator feedback
Notified operators	• operators notified on January 7, 2025, via briefing and manual update • ML monitoring team updated with full maintenance report on Jan. 8, 2025
Evaluation report (i.e. updated results of model)	See [5.2.3] and example given in Tab 13
Validation report (i.e. updated results of ML system)	See [6.1.4] and example given in Tab 14
Risk, bias, and impact reassessments	See [3.5.1], [3.5.3], and example given in App 2

Appendix 1 Overview of Regulations, Standards and Recommendations

1 International regulations, standards and recommendations

1.1 Organisation for Economic Co-operation and Development (OECD)

1.1.1 The Organisation for Economic Co-operation and Development (OECD) adopted the "OECD, Recommendation of the Council on Artificial Intelligence, OECD/LEGAL/0449" in May 2019 and amended it in 2024.

The Recommendation sets out principles for policymakers such as investing in AI research, supporting an inclusive ecosystem, ensuring interoperable governance, enhancing human skills, and encouraging international cooperation for trustworthy AI.

It defines AI principles such as inclusivity, sustainability, promoting human rights, transparency, explainability, robustness, security, safety, and accountability.

1.2 Intergovernmental Forum for International Economic Cooperation (G20)

1.2.1 The Intergovernmental Forum for International Economic Cooperation (G20) AI Principles, adopted in June 2019 and updated in May 2024, are based on the OECD, Recommendation of the Council on Artificial Intelligence, OECD/LEGAL/0449.

The document defines principles such as transparency, explainability, robustness, security, safety, and accountability for responsible supervision of trustworthy AI systems.

The G20 AI Principles recommend to support inclusive growth, sustainable development, and well-being while prioritizing human-centered values and fairness.

1.3 United Nations Educational, Scientific and Cultural Organization (UNESCO)

1.3.1 The United Nations Educational, Scientific and Cultural Organization (UNESCO) adopted the Recommendation on the Ethics of Artificial Intelligence in November 2021.

The Recommendation focuses on promoting human rights, dignity, and environmental sustainability. An AI is considered ethical if it ensures gender equality, freedom of expression, healthcare practices, safeguards cultural heritage, addresses the implications of AI on jobs, and provides education in AI ethics.

1.4 International Maritime Organization (IMO)

1.4.1 As of January 2025, there are no specific international maritime regulations dedicated to AI systems.

1.5 International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC)

1.5.1 The following standards are listed for information and may be used for the development and assessment of ML systems:

- ISO/IEC 5259 Series Artificial intelligence - Data quality for analytics and machine learning (ML)
- ISO/IEC 8000 Series Data quality
- ISO/IEC 22989:2022 Information technology - Artificial intelligence - Artificial intelligence concepts and terminology
- ISO/IEC 23053:2022 Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)
- ISO/IEC 23894:2023 Information technology - Artificial intelligence - Guidance on risk management
- ISO/IEC 24027:2021 Information technology - Artificial intelligence (AI) - Bias in AI systems and AI aided decision making
- ISO/IEC 24029-1:2021 Artificial Intelligence (AI) - Assessment of the robustness of neural networks - Part 1: Overview
- ISO/IEC 24029-2:2023 Artificial Intelligence (AI) - Assessment of the robustness of neural networks - Part 2: Methodology for the use of formal methods
- ISO/IEC AWI 24029-3 Artificial Intelligence (AI) - Assessment of the robustness of neural networks - Part 3: Methodology for the use of statistical methods
- ISO/IEC 25058:2023 (E) Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Guidance for quality evaluation of artificial intelligence (AI) systems
- ISO/IEC 31000:2018 Risk management - Guidelines
- ISO/IEC 38505-1:2017 Information technology - Governance of IT - Governance of data - Part 1: Application of ISO/IEC 38500 to the governance of data
- ISO/IEC TR 38505-2:2018 Information technology - Governance of IT - Governance of data - Part 2: Implications of ISO/IEC 38505-1 for data management

- ISO/IEC TS 38505-3:2021 Information technology - Governance of data - Part 3: Guidelines for data classification
- ISO/IEC 42001:2023 Information technology - Artificial intelligence - AI Management System
- ISO/IEC DIS 42005:2024(E) Information technology - Artificial intelligence - AI system impact assessment

2 European regulations, proposals and recommendations

2.1 Artificial Intelligence

2.1.1 Assessment List for Trustworthy Artificial Intelligence (ALTAI)

The Assessment List for Trustworthy Artificial Intelligence (ALTAI) is a self-assessment tool published in July 2020 by the European Commission.

The ALTAI defines the most important principles for AI, such as human agency and oversight, robustness and safety, privacy and data governance, transparency, diversity and fairness, sustainability and societal well-being, and accountability.

The High-Level Expert Group on AI (HLEG), responsible for the ALTAI guideline, highlights the necessity to adopt both a vertical approach (domain-specific requirements) and a horizontal approach (universal requirements regardless of the sector domain or ML application) when assessing a ML system.

2.1.2 Artificial Intelligence Act

The regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) is a legislative act which entered into force in August 2024.

The AI Act introduces a risk-based approach to regulate AI systems, classifying them into different categories, as described in Tab 1.

The risk classification of AI systems depends on their functions, the intended purpose, and the context in which they are deployed.

The AI Act requires AI systems to be explainable (i.e. decision-making processes are understandable), transparent, under appropriate human oversight and traceable (i.e. users know what data is used to train the system). It requires data involved to be of high quality, unbiased, secured and compliant with current regulations.

Table 1 : AI Act - Risk classification of AI systems

General-Purpose AI (GPAI)	Versatile AI systems which can be adapted for various purposes, including high-risk and limited-risk applications (e.g. GPT models). Depending on their application, GPAI may pose systemic or non-systemic risks, influenced by factors such as the computing power required for training. They must adhere to transparency requirements and meet the obligations associated with the specific risk category of their use case. **Note 1:** As per Art. 56 of the AI Act, a Code of Practice detailing at least the aspects presented in Art. 53 and 55 is to be published in mid 2025. The AI Office is facilitating the development of this Code of Practice, which will provide a detailed framework for providers of general-purpose AI models to demonstrate compliance with the AI Act. **Note 2:** As mandated by Article 53(1)d) of the AI Act, the AI Office is also developing a template that general-purpose AI model providers can use to ensure compliance with the requirements for sufficiently detailed summary of training data.
Minimal risk	AI systems with negligible or no risk (e.g. spam filtering).
Limited risk	AI systems with minimal but notable risks (e.g. chatbots, recommendation systems). These systems must ensure transparency.
High risk	AI systems that significantly impact safety, fundamental rights, or critical operations (e.g. medical devices, autonomous vehicles). They require strict risk management, transparency, documentation, data governance, and human oversight.
Unacceptable risk	AI systems that are prohibited due to significant harm to individuals or society (e.g. real-time remote biometric identification in public spaces). These AI systems are prohibited under European law as contravening Union values and fundamental rights according to the AI Act.

2.2 Data

2.2.1 General Data Protection Regulation (GDPR)

The regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) is a legislative act which entered into force in May 2016.

The General Data Protection Regulation (GDPR) requires systems to ensure lawful basis and transparency regarding data processing, specifying what information is handled, how, by whom, for whom, and for what purpose.

2.2.2 Data Protection Impact Assessment (DPIA)

The Data Protection Impact Assessment (DPIA) is a process designed to assess and mitigate risks related to personal data processing at the earliest possible stage.

It is specifically required under Art. 35 of the GDPR when implementing new technologies, tracking locations, conducting large-scale surveillance, processing sensitive personal data, or using data for automated decision-making with significant impact.

The DPIA should provide a clear and systematic description of processing activities, detailing their purposes and legitimate interests. It should assess risks and potential impacts regarding the rights and freedoms of individuals' data.

2.2.3 Data Governance Act (DGA)

The regulation (EU) 2022/868 of the European Parliament and of the Council of 30 May 2022 on European data governance and amending Regulation (EU) 2018/1724 (Data Governance Act) is a legislative act which entered into force in June 2022.

The Data Governance Act (DGA) defines regulations to safely enable the sharing of sensitive data held by public bodies, and to regulate data sharing by private actors.

2.2.4 Data Act

The regulation (EU) 2023/2854 of the European Parliament and of the Council of 13 December 2023 on harmonised rules on fair access to and use of data and amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828 (Data Act) is a legislative act which entered into force in January 2024.

The Data Act, building on the GDPR and DGA, introduces the notion of "data spaces" and sets rules on how data generated by devices, services, and products should be accessed, used, and shared.

2.3 Other documents

2.3.1 European Committee for Standardization (CEN) and the European Committee for Electrotechnical Standardization (CENELEC)

The evaluation approach for AI systems, as designed by the European Committee for Standardization (CEN) and the European Committee for Electrotechnical Standardization (CENELEC), combines both vertical and horizontal methodologies.

The horizontal approach sets out universal requirements that apply across all sectors, it focuses on:

- risk management
- trustworthiness requirements (as defined by the CEN-CENELEC):
  - cybersecurity
  - transparency
  - robustness
  - accuracy
  - data quality and governance
  - human oversight
  - record-keeping.
- Machine Learning quality management
- conformity assessment.

The vertical approach is sector-specific, and encompasses tailored standards and guidelines to address the domain specific risks, operational domain and regulatory requirements.

2.3.2 Standardization Request

The European Commission has issued a standardization request to develop harmonized standards to support the implementation of the AI Act. As of July 2025, the following standards are in progress:

- AI Trustworthiness Framework
- AI Risk Management
- AI Quality Management System for regulatory purposes
- Concepts, measures and requirements for managing bias in AI systems
- Quality and governance of datasets in AI
- Cybersecurity specifications for AI systems
- AI Conformity Assessment.

2.3.3 Artificial Intelligence Roadmap 2.0

The Artificial Intelligence Roadmap 2.0, published by the European Union Aviation Safety Agency (EASA) in May 2023, is a Human-centric approach to AI in aviation.

The roadmap provides strategies for ensuring a safe, transparent, and ethical AI integration in aviation systems, prioritizing the well-being of passengers and operational personnel.

3 Overview of national regulations and guidelines

3.1 Common AI principles across nations

3.1.1 Numerous countries have developed AI regulations and guidelines focusing on risk management, transparency, human oversight, accountability, and ethical standards. Most frameworks encourage the responsible use of AI while safeguarding privacy, fairness, and security, especially in high-risk applications.

3.1.2 The Tab 2 provides a non-exhaustive overview of national AI regulations and guidelines:

Table 2 : Overview of national AI regulations and guidelines

Australia	Voluntary AI Safety Standard, Australian Government Department of Industry, Science and Resources, August 2024
Brazil	Bill No. 2338, of 2023, on how to use Artificial Intelligence, Federal Senate, 2023
Canada	Bill C-27, Digital Charter Implementation Act, 2022, House of Commons of Canada, Minister of Innovation, Science and Industry
China	New Generation Artificial Intelligence Development Plan (AIDP), China's State Council, July 2017
India	National Strategy for Artificial Intelligence, NITI Aayog (Government of the Republic of India), June 2018
Japan	Social Principles of Human-Centric AI, 2019 AI Guidelines for Business Ver1.0, Ministry of Internal Affairs and Communications and Ministry of Economy, Trade and Industry, April 2024
Korea	GPRN11-1721000-000393-01, National Strategy for Artificial Intelligence, The Government of the Republic of Korea, December 2019
Russia	National Strategy for the development of artificial intelligence for the period up to 2030, Decree of the President of the Russian Federation dated 10.10.2019 No. 490 on the development of artificial intelligence in the Russian Federation, as amended by the Decree of the President of the Russian Federation of 15.02.2024 No. 124
Singapore	Model Artificial Intelligence Governance Framework Second Edition, Info-communications Media Development Authority and the Personal Data Protection Commission, January 2020 Model AI Governance Framework for Generative AI, Info-communications Media Development Authority and AI Verify Foundation, May 2024
Switzerland	Guidelines on Artificial Intelligence for the Confederation, Federal Council, November 2020
Turkey	National Artificial Intelligence Strategy 2021-2025, Digital Transformation Office of the Presidency of the Republic of Turkey and the Ministry of Industry and Technology, August 2021 Recommendations on the Protection of Personal Data in the field of Artificial Intelligence, Turkish Data Protection Authority, September 2021 Ethics Guide of Generative Artificial Intelligence Use in the Scientific Research and Publication Process of Higher Education Institutions, Council of Higher Education, 2024
United Arab Emirates	National Strategy for Artificial Intelligence 2031, National Program for Artificial Intelligence, Minister of State for Artificial Intelligence, 2018
United Kingdom	A pro-innovation approach to AI regulation, Secretary of State for Science, Innovation and Technology by Command of His Majesty, March 2023
United States of America	The Artificial Intelligence Risk Management Framework (AI RMF 1.0), National Institute of Standards and Technology, U.S. Department of Commerce, January 2023 Blueprint for an AI Bill of Rights Making Automated Systems Work for the American People, the White House Office of Science and Technology Policy, October 2022

Appendix 2 Examples of Risk Assessment

1 General

1.1 Data risk assessment

1.1.1 With reference to Sec 2, [3.1], examples of data risk assessments to be performed for ML systems are listed in Tab 1.

1.2 Machine Learning development and operation risk assessment

1.2.1 With reference to Sec 2, [3.1], examples of ML development and operation risk assessments to be performed for ML systems are listed in Tab 2.

1.3 Human factors risk assessment

1.3.1 With reference to Sec 2, [3.1], examples of human factors risk assessments to be performed for ML systems are listed in Tab 3.

Table 1 : Examples of data risk assessment

Hazard	Causes	Consequences / Potential impacts on subsystems or components	Mitigation layers	References
Unrepresentative dataset	- Model trained on small portions of the actual data distribution	- Significant performance disparity between developed and deployed model - Data drift	- Training data is similar to real-world data - Oversampling or augmentation of representative data has been performed	Sec 2, [4.1.4] Sec 2, [4.2.1] Sec 2, [4.2.4]
Overfit	- Insufficient volume of data with regards of the model's complexity	- Model does not generalize well to unseen data	- Data augmentation methods have been implemented	Sec 2, [4.2.1] Sec 2, [4.2.4]
Data leakage	- Overlapping training, validation and test data	- Unrealistically high model performance leading to deployment failures	- Proper partitioning has been performed	Sec 2, [4.2.1] Sec 2, [4.2.5]
Insufficient data analysis and preprocessing	- Excessive volume of data	- Model does not generalize well to specific task	- Only data relevant for the intended task has been considered	Sec 2, [4.2.4]
Insufficient data analysis and preprocessing	- Insufficient volume of data	- Model does not generalize well to task	- Tests prove that data volume is sufficient to enable the model to learn the operational domain effectively	Sec 2, [4.2.4]
Insufficient data analysis and preprocessing	- Erroneous, incomplete, biased, noisy data - Poor quality of input data - Imbalanced dataset	- Poor quality of output data - Wrongful, biased output - Poor reliance on model's performance - Model does not generalize well	- Data quality checks prove that sufficient data preprocessing has been performed, and that data possesses an appropriate level of detail and resolution	Sec 2, [4.2.4]
Wrongful data labelling	- Unqualified annotator - Annotation errors	- Errors in dataset - Poor reliance on model's performance	- Annotation performed by qualified personnel - Cross-checking mechanisms have been implemented	Sec 2, [4.2.7]
Improper data processing	- Unauthorized or unlawful processing of data	- Accidental loss, destruction, or damage of data	- Data is accessible to authorized operators only	Sec 2, [4.2.8]
Data breach	- Unauthorized access to sensitive data	- Data theft	- Confidentiality and security measures have been implemented (encryption, strict access controls)	Sec 2, [4.2.8] Sec 2, [4.2.9]

Hazard	Causes	Consequences / Potential impacts on subsystems or components	Mitigation layers	References
Unlawful processing of data	- Non-compliance with data rights	- Legal repercussions	- Protocols for the retention, retention period, and secure disposal of data have been implemented - DPIAs have been performed - Data is anonymized, encrypted and pseudonymized - Individuals are able to access, modify and delete their data	Sec 2, [4.2.9] Sec 2, [4.2.8] Sec 2, [4.2.10]
Versioning issues	- Lack of dataset version control	- Lack of dataset version control	- Dataset versioning tools have been employed (e.g. Data Version Control)	Sec 2, [5.1.3]
Underfit	- Insufficient model complexity	- Model does not capture patterns in the data	- Tests prove that model is accurate and generalizes well	Sec 2, [5.2.2]
Misuse	- Wrong interpretation of data	- Poor reliance on model's performance - Poor human machine interaction	- Tests prove that data is interpretable and understandable	Sec 2, [5.2.2]
Inconsistent preprocessing	- Different preprocessing methods between datasets (e.g. training and production)	- Poor quality of output data - Wrongful, biased output - Poor reliance on model's performance - Model does not generalize well	- Similar data preprocessing techniques across training, test, validation, and production datasets have been performed - Similar data preprocessing techniques across training, test, validation, and production datasets have been performed	Sec 2, [4.2.4] Sec 2, [7.2.4]

Table 2 : Examples of ML development and operation risk assessment

Hazard	Causes	Consequences / Potential impacts on subsystems or components	Mitigation layers	Reference
Unexpected behaviour from the system	- Silent failure	- Undetected performance degradation - Delayed response to issues - Compromised performance	- Overall human supervision during operation - Alert mechanisms have been implemented	Sec 2, [2.3.1] Sec 2, [3.2.3]
Model failure	- Incorrect data processing - Lack of update - Deprecated, unsupported libraries, dataset or models	- Wrongful outputs - Dangerous actions - No reliance on model's performance - Data drift - ML system lifetime shortened	- Emergency shut-down, fall-back plan, fail-safes, and redundancies (Data Version Control) processes are implemented - Controllability measures are implemented - Continuous human supervision (e.g. ML system monitor, Human-In-Control, Human-In-The-Loop)	Sec 2, [3.2.3] Sec 2, [3.2.4] Sec 2, Tab 7
Unexpected behaviour from the system	- Significant drop in model performance - Wrongful outputs	- Poor reliance on model's performance	- Mitigation layers have been implemented - Methods for operators to provide feedback have been implemented	Sec 2, [3.2.3] Sec 2, [3.2.9]
Lack of interpretability in metrics' results	- Complex model outputs with unclear relevance to operators - Black-box model	- Lack of interpretability, explainability, understandability and maintenance	- Operators have been trained to interpret outputs - Methods have been implemented to ensure model is fully understandable by all operators	Sec 2, [3.2.6] Sec 2, [5.2.2]
Trade-off	- Improving one characteristic reduces another	- Reduced overall model performance - Biased outcomes	- Bias has been identified and mitigated - Model has been evaluated across multiple dimensions (accuracy, fairness, interpretability)	Sec 2, [3.3.2] Sec 2, [5.2.2]

Hazard	Causes	Consequences / Potential impacts on subsystems or components	Mitigation layers	Reference
Reward hacking	- Agent exploits reward function in unintended ways	- Agent achieves high rewards through unintended behaviours - Unsafe or inefficient navigation - Loss of trust in system reliability	- Reward functions have been reviewed and tested in varied scenarios - Continuous human supervision during operation	Sec 2, [5.1.4] Sec 2, [5.2.2]
Inference attack	- Unauthorized access to data	- Breach of confidentiality - Legal repercussions	- Procedures for securing access from unauthorized stakeholders have been implemented - Red teaming and adversarial tests have been performed to demonstrate security of the system	Sec 2, [5.2.2]
Misleading outputs	- Hallucination	- Misleading or false outputs - Biased outcomes - Loss of trust on model's performance	- Confidence thresholds have been implemented - Alert mechanisms have been implemented for critical outputs	Sec 2, [6.3.1] Sec 2, [6.3.7]
Wrongful outputs	- Model has not been properly tested in real-world conditions	- Significant drop in model performance - No reliance on model's performance	- Continuous real-world testing has been performed - Automatic monitoring processes have been implemented - Regular updates and retraining based on new data are performed	Sec 2, [6.1.2] Sec 2, [6.3.1] Sec 2, [7.1.3] Sec 2, [7.2.1]
Unexpected behaviour from the system Performance degradation ML system failure	- Lack of update	- Deprecated, unsupported libraries, dataset or models - Data drift - ML system lifetime shortened	- Automatic monitoring processes have been implemented - Frequent updates with new technologies are performed - Automatic maintenance processes have been implemented	Sec 2, [7.1.3] Sec 2, [7.2.1] Sec 2, [7.2.2]
Model drift	- Inconsistent update	- Wrongful outputs - No reliance on model's performance - Deprecated, unsupported libraries, dataset or models - Data drift - ML system lifetime shortened	- Changes and updates are in accordance with the operational context	Sec 2, [7.2.5]

Table 3 : Examples of human factors risk assessment

Hazard	Causes	Consequences / Potential impacts on subsystems or components	Mitigation layers	Reference
Conflict of interest (e.g. the ML system provider is also the training data provider)	- Biased objectives from the providers	- Unethical decisions - Unbalanced model outcomes	- Data have been selected objectively and impartially - Transparent communication and documentation	Sec 2, [2.4.2] Sec 2, [4.1.2] Sec 2, [4.1.6] Sec 2, [4.2.1]
ML system mishandling	- Overconfidence in ML system - Difficulties to interact with ML system	- Wrongful errors or actions	- Training and methods to avoid users put overconfidence in ML systems have been implemented - Users are aware they are interacting with a ML system - Effective human-machine interaction	Sec 2, [3.2.6] Sec 2, [3.2.10] Sec 2, [6.2.1] Sec 2, [6.2.2]

BUREAU VERITAS MARINE & OFFSHORE

Tour Alto
4 place des Saisons
92400 Courbevoie - France
+33 (0)1 55 24 70 00

marine-offshore.bureauveritas.com/rules-guidelines

BUREAU VERITAS
1828
BUREAU
VERITAS

Shaping a World of Trust