INTERNATIONAL STANDARD

ISO/IEC 22989

First edition
2022-07

Information technology — Artificial intelligence — Artificial intelligence concepts and terminology

Technologies de l'information — Intelligence artificielle — Concepts et terminologie relatifs à l'intelligence artificielle

Reference number
ISO/IEC 22989:2022(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 42, *Artificial Intelligence*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

Advancements in computing capacity, reduction of costs of computation, availability of large amounts of data from many sources, inexpensive online learning curricula and algorithms capable of meeting or exceeding human level performance in particular tasks for speed and accuracy have enabled practical applications of AI, making it an increasingly important branch of information technology.

AI is a highly interdisciplinary field broadly based on computer science, data science, natural sciences, humanities, mathematics, social sciences and others. Terms such as “intelligent”, “intelligence”, “understanding”, “knowledge”, “learning”, “decisions”, “skills”, etc. are used throughout this document. However, it is not the intention to anthropomorphize AI systems, but to describe the fact that some AI systems can rudimentarily simulate such characteristics.

There are many areas of AI technology. These areas are intricately linked and developing rapidly so it is difficult to fit the relevance of all technical fields into a single map. Research of AI includes aspects such as aspects including “learning, recognition and prediction”, “inference, knowledge and language” and “discovery, search and creation”. Research also addresses interdependencies among these aspects[23].

The concept of AI as an input and output process flow is shared by many AI researchers, and research on each step of this process is ongoing. Standardized concepts and terminology are needed by stakeholders of the technology to be better understood and adopted by a broader audience. Furthermore, concepts and categories of AI allow for a comparison and classification of different solutions with respect to properties like trustworthiness, robustness, resilience, reliability, accuracy, safety, security and privacy. This enables stakeholders to select appropriate solutions for their applications and to compare the quality of available solutions on the market.

As this document does provide a definition for the term AI in the sense of a discipline only, the context for its usage can be described as follows: AI is a technical and scientific field devoted to the engineered system that generates outputs such as content, forecasts, recommendations or decisions for a given set of human-defined objectives.

This document provides standardized concepts and terminology to help AI technology to be better understood and used by a broader set of stakeholders. It is intended for a wide audience including experts and non-practitioners. The reading of some specific clauses can however be easier with a stronger background in computer science. These concerns are described primarily Clauses 5.10, 5.11 and 8, which are more technical than the rest of the document.

Information technology — Artificial intelligence — Artificial intelligence concepts and terminology

1 Scope

This document establishes terminology for AI and describes concepts in the field of AI.

This document can be used in the development of other standards and in support of communications among diverse, interested parties or stakeholders.

This document is applicable to all types of organizations (e.g. commercial enterprises, government agencies, not-for-profit organizations).

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1 Terms related to AI

3.1.1
AI agent
automated (3.1.7) entity that senses and responds to its environment and takes actions to achieve its goals

3.1.2
AI component
functional element that constructs an *AI system* (3.1.4)

3.1.3
artificial intelligence
AI
<discipline> research and development of mechanisms and applications of *AI systems* (3.1.4)

Note 1 to entry: Research and development can take place across any number of fields such as computer science, data science, humanities, mathematics and natural sciences.

3.1.4
artificial intelligence system
AI system
engineered system that generates outputs such as content, forecasts, recommendations or decisions for a given set of human-defined objectives

Note 1 to entry: The engineered system can use various techniques and approaches related to *artificial intelligence* (3.1.3) to develop a *model* (3.1.23) to represent data, *knowledge* (3.1.21), processes, etc. which can be used to conduct *tasks* (3.1.35).

Note 2 to entry: AI systems are designed to operate with varying levels of *automation* (3.1.7).

3.1.5
autonomy
autonomous
characteristic of a system that is capable of modifying its intended domain of use or goal without external intervention, control or oversight

3.1.6
application specific integrated circuit
ASIC
integrated circuit customized for a particular use

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.193, modified — Acronym has been moved to separate line.]

3.1.7
automatic
automation
automated
pertaining to a process or system that, under specified conditions, functions without human intervention

[SOURCE: ISO/IEC 2382:2015, 2121282, modified — In the definition, "a process or equipment" has been replaced by "a process or system" and preferred terms of "automated and automation" are added.]

3.1.8
cognitive computing
category of *AI systems* (3.1.4) that enables people and machines to interact more naturally

Note 1 to entry: Cognitive computing tasks are associated with *machine learning* (3.3.5), speech processing, *natural language processing* (3.6.9), *computer vision* (3.7.1) and human-machine interfaces.

3.1.9
continuous learning
continual learning
lifelong learning
incremental training of an *AI system* (3.1.4) that takes place on an ongoing basis during the operation phase of the AI system life cycle

3.1.10
connectionism
connectionist paradigm
connectionist model
connectionist approach
form of cognitive modelling that uses a network of interconnected units that generally are simple computational units

3.1.11
data mining
computational process that extracts patterns by analysing quantitative data from different perspectives and dimensions, categorizing them, and summarizing potential relationships and impacts

[SOURCE: ISO 16439:2014, 3.13, modified — replace "categorizing it" with "categorizing them" because data is plural.]

3.1.12
declarative knowledge
knowledge represented by facts, rules and theorems

Note 1 to entry: Usually, declarative knowledge cannot be processed without first being translated into *procedural knowledge* (3.1.28).

[SOURCE: ISO/IEC 2382-28:1995, 28.02.22, modified — Remove comma after “rules” in the definition.]

3.1.13
expert system
AI system (3.1.4) that accumulates, combines and encapsulates *knowledge* (3.1.21) provided by a human expert or experts in a specific domain to infer solutions to problems

3.1.14
general AI
AGI
artificial general intelligence
type of *AI system* (3.1.4) that addresses a broad range of *tasks* (3.1.35) with a satisfactory level of performance

Note 1 to entry: Compared to *narrow AI* (3.1.24).

Note 2 to entry: AGI is often used in a stronger sense, meaning systems that not only can perform a wide variety of tasks, but all tasks that a human can perform.

3.1.15
genetic algorithm
GA
algorithm which simulates natural selection by creating and evolving a population of individuals (solutions) for optimization problems

3.1.16
heteronomy
heteronomous
characteristic of a system operating under the constraint of external intervention, control or oversight

3.1.17
inference
reasoning by which conclusions are derived from known premises

Note 1 to entry: In AI, a premise is either a fact, a rule, a model, a feature or raw data.

Note 2 to entry: The term "inference" refers both to the process and its result.

[SOURCE: ISO/IEC 2382:2015, 2123830, modified – Model, feature and raw data have been added. Remove “Note 4 to entry: 28.03.01 (2382)”. Remove “Note 3 to entry: inference: term and definition standardized by ISO/IEC 2382-28:1995”.]

3.1.18
internet of things
IoT
infrastructure of interconnected entities, people, systems and information resources together with services that process and react to information from the physical world and virtual world

[SOURCE: ISO/IEC 20924:2021, 3.2.4, modified – “…services which processes and reacts to…” has been replaced with “…services that process and react to…” and acronym has been moved to separate line.]

3.1.19
IoT device
entity of an *IoT system* (3.1.20) that interacts and communicates with the physical world through sensing or actuating

Note 1 to entry: An IoT device can be a sensor or an actuator.

[SOURCE: ISO/IEC 20924:2021, 3.2.6]

3.1.20
IoT system
system providing functionalities of *IoT* (3.1.18)

Note 1 to entry: An IoT system can include, but not be limited to, IoT devices, IoT gateways, sensors and actuators.

[SOURCE: ISO/IEC 20924:2021, 3.2.9]

3.1.21
knowledge
<artificial intelligence> abstracted information about objects, events, concepts or rules, their relationships and properties, organized for goal-oriented systematic use

Note 1 to entry: Knowledge in the AI domain does not imply a cognitive capability, contrary to usage of the term in some other domains. In particular, knowledge does not imply the cognitive act of understanding.

Note 2 to entry: Information can exist in numeric or symbolic form.

Note 3 to entry: Information is data that has been contextualized, so that it is interpretable. Data is created through abstraction or measurement from the world.

3.1.22
life cycle
evolution of a system, product, service, project or other human-made entity, from conception through retirement

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.23]

3.1.23
model
physical, mathematical or otherwise logical representation of a system, entity, phenomenon, process or data

[SOURCE: ISO/IEC 18023-1:2006, 3.1.11, modified – Remove comma after “mathematical” in the definition. "or data" is added at the end.]

3.1.24
narrow AI
type of *AI system* (3.1.4) that is focused on defined *tasks* (3.1.35) to address a specific problem

Note 1 to entry: Compared to *general AI* (3.1.14).

3.1.25
performance
measurable result

Note 1 to entry: Performance can relate either to quantitative or qualitative findings.

Note 2 to entry: Performance can relate to managing activities, processes, products (including services), systems or organizations.

3.1.26
planning
<artificial intelligence> computational processes that compose a workflow out of a set of actions, aiming at reaching a specified goal

Note 1 to entry: The meaning of the “planning” used in AI life cycle or AI management standards can be also actions taken by human beings.

3.1.27
prediction
primary output of an *AI system* (3.1.4) when provided with *input data* (3.2.9) or information

Note 1 to entry: Predictions can be followed by additional outputs, such as recommendations, decisions and actions.

Note 2 to entry: Prediction does not necessarily refer to predicting something in the future.

Note 3 to entry: Predictions can refer to various kinds of data analysis or production applied to new data or historical data (including translating text, creating synthetic images or diagnosing a previous power failure).

3.1.28
procedural knowledge
knowledge which explicitly indicates the steps to be taken in order to solve a problem or to reach a goal

[SOURCE: ISO/IEC 2382-28:1995, 28.02.23]

3.1.29
robot
automation system with actuators that performs intended *tasks* (3.1.35) in the physical world, by means of sensing its environment and a software control system

Note 1 to entry: A robot includes the control system and interface of a control system.

Note 2 to entry: The classification of a robot as industrial robot or service robot is done according to its intended application.

Note 3 to entry: In order to properly perform its *tasks* (3.1.35), a robot makes use of different kinds of sensors to confirm its current state and perceive the elements composing the environment in which it operates.

3.1.30
robotics
science and practice of designing, manufacturing and applying robots

[SOURCE: ISO 8373:2012, 2.16]

3.1.31
semantic computing
field of computing that aims to identify the meanings of computational content and user intentions and to express them in a machine-processable form

3.1.32
soft computing
field of computing that is tolerant of and exploits imprecision, uncertainty and partial truth to make problem-solving more tractable and robust

Note 1 to entry: Soft computing encompasses various techniques such as fuzzy logic, machine learning and probabilistic reasoning.

3.1.33
symbolic AI
AI (3.1.3) based on techniques and *models* (3.1.23) that manipulate symbols and structures according to explicitly defined rules to obtain inferences

Note 1 to entry: Compared to *subsymbolic AI* (3.1.34), symbolic AI produces declarative outputs, whereas subsymbolic AI is based on statistical approaches and produces outputs with a given probability of error.

3.1.34
subsymbolic AI
AI (3.1.3) based on techniques and *models* (3.1.23) that use an implicit encoding of information, that can be derived from experience or raw data.

Note 1 to entry: Compared to *symbolic AI* (3.1.33). Whereas symbolic AI produces declarative outputs, subsymbolic AI is based on statistical approaches and produces outputs with a given probability of error.

3.1.35
task
<artificial intelligence>action required to achieve a specific goal

Note 1 to entry: Actions can be physical or cognitive. For instance, computing or creation of *predictions* (3.1.27), translations, synthetic data or artefacts or navigating through a physical space.

Note 2 to entry: Examples of tasks include classification, regression, ranking, clustering and dimensionality reduction.

3.2 Terms related to data

3.2.1
data annotation
process of attaching a set of descriptive information to data without any change to that data

Note 1 to entry: The descriptive information can take the form of metadata, labels and anchors.

3.2.2
data quality checking
process in which data is examined for completeness, bias and other factors which affect its usefulness for an *AI system* (3.1.4)

3.2.3
data augmentation
process of creating synthetic samples by modifying or utilizing the existing data

3.2.4
data sampling
process to select a subset of data samples intended to present patterns and trends similar to that of the larger *dataset* (3.2.5) being analysed

Note 1 to entry: Ideally, the subset of data samples will be representative of the larger *dataset* (3.2.5).

3.2.5
dataset
collection of data with a shared format

EXAMPLE 1 Micro-blogging posts from June 2020 associated with hashtags #rugby and #football.

EXAMPLE 2 Macro photographs of flowers in 256x256 pixels.

Note 1 to entry: Datasets can be used for validating or testing an AI *model* (3.1.23). In a *machine learning* (3.3.5) context, datasets can also be used to train a *machine learning algorithm* (3.3.6).

3.2.6
exploratory data analysis
EDA
initial examination of data to determine its salient characteristics and assess its quality

Note 1 to entry: EDA can include identification of missing values, outliers, representativeness for the task at hand – see *data quality checking* (3.2.2).

3.2.7
ground truth
value of the target variable for a particular item of labelled input data

Note 1 to entry: The term ground truth does not imply that the labelled input data consistently corresponds to the real-world value of the target variables.

3.2.8
imputation
procedure where missing data are replaced by estimated or modelled data

[SOURCE: ISO 20252:2019, 3.45]

3.2.9
input data
data for which an *AI system* (3.1.4) calculates a predicted output or inference

3.2.10
label
target variable assigned to a sample

3.2.11
personally identifiable information
PII
personal data
any information that (a) can be used to establish a link between the information and the natural person to whom such information relates, or (b) is or can be directly or indirectly linked to a natural person

Note 1 to entry: The "natural person" in the definition is the PII principal. To determine whether a PII principal is identifiable, account should be taken of all the means which can reasonably be used by the privacy stakeholder holding the data, or by any other party, to establish the link between the set of PII and the natural person.

Note 2 to entry: This definition is included to define the term PII as used in this document. A public cloud PII processor is typically not in a position to know explicitly whether information it processes falls into any specified category unless this is made transparent by the cloud service customer.

[SOURCE: ISO/IEC 29100:2011/Amd1:2018, 2.9]

3.2.12
production data
data acquired during the operation phase of an *AI system* (3.1.4), for which a deployed *AI system* (3.1.4) calculates a predicted output or *inference* (3.1.17)

3.2.13
sample
atomic data element processed in quantities by a *machine learning algorithm* (3.3.6) or an *AI system* (3.1.4)

3.2.14
test data
evaluation data
data used to assess the performance of a final *model* (3.1.23)

Note 1 to entry: Test data is disjoint from *training data* (3.3.16) and *validation data* (3.2.15).

3.2.15
validation data
development data
data used to compare the performance of different candidate *models* (3.1.23)

Note 1 to entry: Validation data is disjoint from *test data* (3.2.14) and generally also from *training data* (3.3.16). However, in cases where there is insufficient data for a three-way training, validation and test set split, the data is divided into only two sets – a test set and a training or validation set. Cross-validation or bootstrapping are common methods for then generating separate training and validation sets from the training or validation set.

Note 2 to entry: Validation data can be used to tune hyperparameters or to validate some algorithmic choices, up to the effect of including a given rule in an expert system.

3.3 Terms related to machine learning

3.3.1
Bayesian network
probabilistic *model* (3.1.23) that uses Bayesian *inference* (3.1.17) for probability computations using a directed acyclic graph

3.3.2
decision tree
model (3.1.23) for which *inference* (3.1.17) is encoded as paths from the root to a leaf node in a tree structure

3.3.3
human-machine teaming
integration of human interaction with machine intelligence capabilities

3.3.4
hyperparameter
characteristic of a *machine learning algorithm* (3.3.6) that affects its learning process

Note 1 to entry: Hyperparameters are selected prior to training and can be used in processes to help estimate model parameters.

Note 2 to entry: Examples of hyperparameters include the number of network layers, width of each layer, type of activation function, optimization method, learning rate for neural networks; the choice of kernel function in a support vector machine; number of leaves or depth of a tree; the K for K-means clustering; the maximum number of iterations of the expectation maximization algorithm; the number of Gaussians in a Gaussian mixture.

3.3.5
machine learning
ML
process of optimizing *model parameters* (3.3.8) through computational techniques, such that the *model's* (3.1.23) behaviour reflects the data or experience

3.3.6
machine learning algorithm
algorithm to determine *parameters* (3.3.8) of a *machine learning model* (3.3.7) from data according to given criteria

EXAMPLE Consider solving a univariate linear function $y = \theta_0 + \theta_1 x$ where y is an output or result, x is an input, θ_0 is an intercept (the value of y where x=0) and θ_1 is a weight. In *machine learning* (3.3.5), the process of determining the intercept and weights for a linear function is known as linear regression.

3.3.7
machine learning model
mathematical construct that generates an *inference* (3.1.17) or *prediction* (3.1.27) based on input data or information

EXAMPLE If a univariate linear function ($y = \theta_0 + \theta_1 x$) has been trained using linear regression, the resulting model can be $y = 3 + 7x$.

Note 1 to entry: A machine learning model results from training based on a *machine learning algorithm* (3.3.6).

3.3.8
parameter
model parameter
internal variable of a *model* (3.1.23) that affects how it computes its outputs

Note 1 to entry: Examples of parameters include the weights in a neural network and the transition probabilities in a Markov model.

3.3.9
reinforcement learning
RL
learning of an optimal sequence of actions to maximize a reward through interaction with an environment

3.3.10
retraining
updating a *trained model* (3.3.14) by *training* (3.3.15) with different *training data* (3.3.16)

3.3.11
semi-supervised machine learning
machine learning (3.3.5) that makes use of both labelled and unlabelled data during *training* (3.3.15)

3.3.12
supervised machine learning
machine learning (3.3.5) that makes only use of labelled data during *training* (3.3.15)

3.3.13
support vector machine
SVM
machine learning algorithm (3.3.6) that finds decision boundaries with maximal margins

Note 1 to entry: Support vectors are sets of data points that define the positioning of the decision boundaries (hyper-planes).

3.3.14
trained model
result of *model training* (3.3.15)

3.3.15
training
model training
process to determine or to improve the parameters of a *machine learning model* (3.3.7), based on a *machine learning algorithm* (3.2.10), by using *training data* (3.3.16)

3.3.16
training data
data used to train a *machine learning model* (3.3.7)

3.3.17
unsupervised machine learning
machine learning (3.3.5) that makes only use of unlabelled data during *training* (3.3.15)

3.4 Terms related to neural networks

3.4.1
activation function
function applied to the weighted combination of all inputs to a *neuron* (3.4.9)

Note 1 to entry: Activation functions allow neural networks to learn complicated features in the data. They are typically non-linear.

3.4.2
convolutional neural network
CNN
deep convolutional neural network
DCNN
feed forward neural network (3.4.6) using *convolution* (3.4.3) in at least one of its layers

3.4.3
convolution
mathematical operation involving a sliding dot product or cross-correlation of the input data

3.4.4
deep learning
deep neural network learning
<artificial intelligence> approach to creating rich hierarchical representations through the *training* (3.3.15) of *neural networks* (3.4.8) with many hidden layers

Note 1 to entry: Deep learning is a subset of *ML* (3.3.5).

3.4.5
exploding gradient
phenomenon of backpropagation *training* (3.3.15) in a neural network where large error gradients accumulate and result in very large updates to the weights, making the *model* (3.1.23) unstable

3.4.6
feed forward neural network
FFNN
neural network (3.4.8) where information is fed from the input layer to the output layer in one direction only

3.4.7
long short-term memory
LSTM
type of *recurrent neural network* (3.4.10) that processes sequential data with a satisfactory performance for both long and short span dependencies

3.4.8
neural network
NN
neural net
artificial neural network
<artificial intelligence> network of one or more layers of *neurons* (3.4.9) connected by weighted links with adjustable weights, which takes input data and produces an output

Note 1 to entry: Neural networks are a prominent example of the *connectionist approach* (3.1.10).

Note 2 to entry: Although the design of neural networks was initially inspired by the functioning of biological neurons, most works on neural networks do not follow that inspiration anymore.

3.4.9
neuron
<artificial intelligence> primitive processing element which takes one or more input values and produces an output value by combining the input values and applying an *activation function* (3.4.1) on the result

Note 1 to entry: Examples of nonlinear activation functions are a threshold function, a sigmoid function and a polynomial function.

3.4.10
recurrent neural network
RNN
neural network (3.4.8) in which outputs from both the previous layer and the previous processing step are fed into the current layer

3.5 Terms related to trustworthiness

3.5.1
accountable
answerable for actions, decisions and performance

[SOURCE: ISO/IEC 38500:2015, 2.2]

3.5.2
accountability
state of being *accountable* (3.5.1)

Note 1 to entry: Accountability relates to an allocated responsibility. The responsibility can be based on regulation or agreement or through assignment as part of delegation.

Note 2 to entry: Accountability involves a person or entity being accountable for something to another person or entity, through particular means and according to particular criteria.

[SOURCE: ISO/IEC 38500:2015, 2.3, modified — Note 2 to entry is added.]

3.5.3
availability
property of being accessible and usable on demand by an authorized entity

[SOURCE: ISO/IEC 27000:2018, 3.7]

3.5.4
bias
systematic difference in treatment of certain objects, people or groups in comparison to others

Note 1 to entry: Treatment is any kind of action, including perception, observation, representation, *prediction* (3.1.27) or decision.

[SOURCE: ISO/IEC TR 24027:2021, 3.3.2, modified – remove oxford comma in definition and note to entry]

3.5.5
control
purposeful action on or in a process to meet specified objectives

[SOURCE: IEC 61800-7-1:2015, 3.2.6]

3.5.6
controllability
controllable
property of an *AI system* (3.1.4) that allows a human or another external agent to intervene in the system's functioning

3.5.7
explainability
property of an *AI system* (3.1.4) to express important factors influencing the *AI system* (3.1.4) results in a way that humans can understand

Note 1 to entry: It is intended to answer the question "Why?" without actually attempting to argue that the course of action that was taken was necessarily optimal.

3.5.8
predictability
property of an *AI system* (3.1.4) that enables reliable assumptions by *stakeholders* (3.5.13) about the output

[SOURCE: ISO/IEC TR 27550:2019, 3.12, "by individuals, owners, and operators about the PII and its processing by a system" has been replaced with "by stakeholders about the outputs".]

3.5.9
reliability
property of consistent intended behaviour and results

[SOURCE: ISO/IEC 27000:2018, 2.55]

3.5.10
resilience
ability of a system to recover operational condition quickly following an incident

3.5.11
risk
effect of uncertainty on objectives

Note 1 to entry: An effect is a deviation from the expected. It can be positive, negative or both and can address, create or result in opportunities and threats.

Note 2 to entry: Objectives can have different aspects and categories and can be applied at different levels.

Note 3 to entry: Risk is usually expressed in terms of risk sources, potential events, their consequences and their likelihood.

[SOURCE: ISO 31000:2018, 3.1, modified — Remove comma after "both" in Note 1 to entry. Remove comma after "categories" in Note 2 to entry.]

3.5.12
robustness
ability of a system to maintain its level of performance under any circumstances

3.5.13
stakeholder
any individual, group, or organization that can affect, be affected by or perceive itself to be affected by a decision or activity

[SOURCE: ISO/IEC 38500:2015, 2.24, modified — Remove comma after "be affected by" in the definition.]

3.5.14
transparency
<organization> property of an organization that appropriate activities and decisions are communicated to relevant *stakeholders* (3.5.13) in a comprehensive, accessible and understandable manner

Note 1 to entry: Inappropriate communication of activities and decisions can violate security, privacy or confidentiality requirements.

3.5.15
transparency
<system> property of a system that appropriate information about the system is made available to relevant *stakeholders* (3.5.13)

Note 1 to entry: Appropriate information for system transparency can include aspects such as features, performance, limitations, components, procedures, measures, design goals, design choices and assumptions, data sources and labelling protocols.

Note 2 to entry: Inappropriate disclosure of some aspects of a system can violate security, privacy or confidentiality requirements.

3.5.16
trustworthiness
ability to meet *stakeholder* (3.5.13) expectations in a verifiable way

Note 1 to entry: Depending on the context or sector, and also on the specific product or service, data and technology used, different characteristics apply and need verification to ensure *stakeholders'* (3.5.13) expectations are met.

Note 2 to entry: Characteristics of trustworthiness include, for instance, reliability, availability, resilience, security, privacy, safety, accountability, transparency, integrity, authenticity, quality and usability.

Note 3 to entry: Trustworthiness is an attribute that can be applied to services, products, technology, data and information as well as, in the context of governance, to organizations.

[SOURCE: ISO/IEC TR 24028:2020, 3.42, modified — Stakeholders' expectations replaced by stakeholder expectations; comma between quality and usability replaced by "and".]

3.5.17
verification
confirmation, through the provision of objective evidence, that specified requirements have been fulfilled

Note 1 to entry: Verification only provides assurance that a product conforms to its specification.

[SOURCE: ISO/IEC 27042:2015, 3.21]

3.5.18
validation
confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled

[SOURCE: ISO/IEC 27043:2015, 3.16]

3.6 Terms related to natural language processing

3.6.1
automatic summarization
task (3.1.35) of shortening a portion of *natural language* (3.6.7) content or text while retaining important semantic information

3.6.2
dialogue management
task (3.1.35) of choosing the appropriate next move in a dialogue based on user input, the dialogue history and other contextual *knowledge* (3.1.21), to meet a desired goal

3.6.3
emotion recognition
task (3.1.35) of computationally identifying and categorizing emotions expressed in a piece of text, speech, video or image or combination thereof

Note 1 to entry: Examples of emotions include happiness, sadness, anger and delight.

3.6.4
information retrieval
IR
task (3.1.35) of retrieving relevant documents or parts of documents from a *dataset* (3.2.5), typically based on keyword or *natural language* (3.6.7) queries

3.6.5
machine translation
MT
task (3.1.35) of automated translation of text or speech from one *natural language* (3.6.7) to another using a computer system

[SOURCE: ISO 17100:2015, 2.2.2]

3.6.6
named entity recognition
NER
task (3.1.35) of recognizing and labelling the denotational names of entities and their categories for sequences of words in a stream of text or speech

Note 1 to entry: Entity refers to concrete or abstract thing of interest, including associations among things.

Note 2 to entry: "Named entity" refers to an entity with a denotational name where a specific or unique meaning exists.

Note 3 to entry: Denotational names include the specific names of persons, locations, organizations and other proper names based on the domain or application.

3.6.7
natural language
language that is or was in active use in a community of people and whose rules are deduced from usage

Note 1 to entry: Natural language is any human language, which can be expressed in text, speech, sign language, etc.

Note 2 to entry: Natural language is any human language, such as English, Spanish, Arabic, Chinese or Japanese, to be distinguished from programming and formal languages, such as Java, Fortran, C++ or First-Order Logic.

[SOURCE: ISO/IEC 15944-8:2012, 3.82, modified — "and the rules of which are mainly deduced from the usage" replaced by "and its rules are deduced from usage. Removed comma after "Chinese" in Note 2 to entry 3.6.8]

3.6.8
natural language generation
NLG
task (3.1.35) of converting data carrying semantics into *natural language* (3.6.7)

3.6.9
natural language processing
NLP
<system> information processing based upon *natural language understanding* (3.6.11) or *natural language generation* (3.6.8)

3.6.10
natural language processing
NLP
<discipline> discipline concerned with the way systems acquire, process and interpret *natural language* (3.6.7)

3.6.11
natural language understanding
NLU
natural language comprehension
extraction of information, by a functional unit, from text or speech communicated to it in a *natural language* (3.6.7), and the production of a description for both the given text or speech and what it represents

[SOURCE: ISO/IEC 2382:2015, 2123786, modified – Note to entry has been removed, hyphen in natural-language has been removed, NLU has been added.]

3.6.12
optical character recognition
OCR
conversion of images of typed, printed or handwritten text into machine-encoded text

3.6.13
part-of-speech tagging
task (3.1.35) of assigning a category (e.g. verb, noun, adjective) to a word based on its grammatical properties

3.6.14
question answering
task (3.1.35) of determining the most appropriate answer to a question provided in *natural language* (3.6.7)

Note 1 to entry: A question can be open-ended or be intended to have a specific answer.

3.6.15
relationship extraction
relation extraction
task (3.1.35) of identifying relationships among entities mentioned in a text

3.6.16
sentiment analysis
task (3.1.35) of computationally identifying and categorizing opinions expressed in a piece of text, speech or image, to determine a range of feeling such as from positive to negative

Note 1 to entry: Examples of sentiments include approval, disapproval, positive toward, negative toward, agreement and disagreement.

3.6.17
speech recognition
speech-to-text
STT
conversion, by a functional unit, of a speech signal to a representation of the content of the speech

[SOURCE: ISO/IEC 2382:2015, 2120735, modified — Note to entry has been removed.]

3.6.18
speech synthesis
text-to-speech
TTS
generation of artificial speech

[SOURCE: ISO/IEC 2382: 2015, 2120745]

3.7 Terms related to computer vision

3.7.1
computer vision
capability of a functional unit to acquire, process and interpret data representing images or video

Note 1 to entry: Computer vision involves the use of sensors to create a digital image of a visual scene. This can include images, such as images that capture wavelengths beyond those of visible light such as infrared imaging.

3.7.2
face recognition
automatic pattern recognition comparing stored images of human faces with the image of an actual face, indicating any matching, if it exists, and any data, if they exist, identifying the person to whom the face belongs

[SOURCE: ISO 5127:2017, 3.1.12.09]

3.7.3
image
<digital> graphical content intended to be presented visually

Note 1 to entry: This includes graphics that are encoded in any electronic format, including, but not limited to, formats that are comprised of individual pixels (e.g. those produced by paint programs or by photographic means) and formats that comprised of formulas (e.g. those produced as scalable vector drawings).

[SOURCE: ISO/IEC 20071-11:2019, 3.2.1]

3.7.4
image recognition
image classification process that classifies object(s), pattern(s) or concept(s) in an *image* (3.7.3)

4 Abbreviated terms

API	application programming interface
CPS	cyber-physical systems
CPU	central processing unit
CRISP-DM	cross-industry process model for data mining
DNN	deep neural network
DSP	digital signal processor
FPGA	field-programmable gate array
GPU	graphics processing unit
HMM	hidden Markov model
IT	information technology
KDD	knowledge discovery in data
NPU	neural network processing unit
OECD	organisation for economic co-operation and development
POS	part of speech

5 AI concepts

5.1 General

The interdisciplinary study and development of AI systems aim at building computer systems able to perform tasks that normally require intelligence. AI-enabled machines are intended to perceive certain environments and take actions that fulfil their demands.

AI uses techniques from many fields, such as computer science, mathematics, philosophy, linguistics, economics, psychology and cognitive science.

Compared to most conventional non-AI systems, there is a number of interesting features that are shared by some or all AI systems:

a) Interactive — inputs to AI systems are generated by sensors, or through interactions with humans, with outputs which can result in stimulating an actuator or providing responses to humans or machines. An example can be object recognition as a result of an AI system being presented with an image of an object.

b) Contextual — some AI systems can draw on multiple sources of information, including both structured and unstructured digital information, as well as sensory inputs.

c) Oversight — AI systems can operate with various degrees of human oversight and control, depending on the application. An example is a self-driving vehicle with varying levels of automation.

d) Adaptive — some AI systems are engineered to utilize dynamic data in real time and retrain to update their operation based on new data.

5.2 From strong and weak AI to general and narrow AI

From a philosophical point of view, the feasibility of machines possessing intelligence has been debated. This debate has led to introduce two different kinds of AI: the so-called weak AI and strong AI. In weak AI, the system can only process symbols (letters, numbers, etc.) without ever understanding what it does. In strong AI, the system also processes symbols, but truly understands what it does. The denominations "weak AI" and "strong AI" are mostly important to philosophers but irrelevant to AI researchers and practitioners.

Following this debate, the qualifications of "narrow AI" vs "general AI" appeared, which are more suitable to the AI field. A "narrow AI" system is able to solve defined tasks to address a specific problem (possibly much better than humans would do). A “general AI” system addresses a broad range of tasks with a satisfactory level of performance. Current AI systems are considered as “narrow”. It is not yet known whether “general” AI systems will be technically feasible in the future.

5.3 Agent

It is possible to look at AI systems from an agent paradigm point of view since some applications of AI aim at simulating human intelligence and human behaviour. Defined as an engineering discipline, AI can be seen as the domain that tries to build artificial agents exhibiting rational behaviour. The agent paradigm establishes a clear line separating the agent and the environment in which it evolves. The agent paradigm is illustrated in Figure 1.

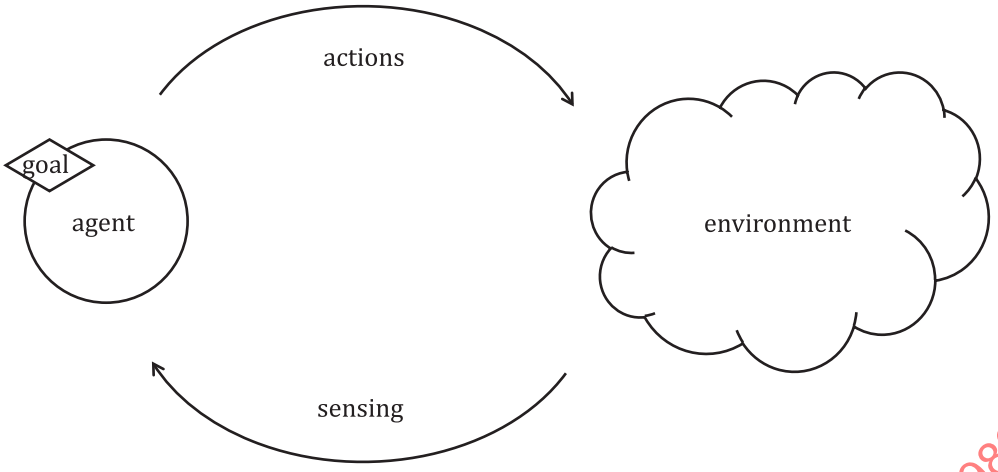

Figure 1 — The agent paradigm

An AI agent interacts with its environment through sensors and actuators, taking actions that maximize its chance of successfully achieving its goals.

Environments have different characteristics depending on the task being undertaken, and these characteristics impact the level of problem-solving difficulty.

In this paradigm, several types of AI agents can be defined, depending on their architecture[29]:

— reflex agents, which rely only on the current situation to choose an action;

— model-based agents, which rely on a model of their environment that allows them to consider the results of their available actions;

— goal or utility-based agents, which rely on an internal utility function that allows them to choose actions that achieve goals, and among goals, look for the most desirable ones;

— learning agents, which can gather information on their environment and learn to improve their performance.

Several sophisticated and high-level architectures based on different theories have been developed to implement agents.

5.4 Knowledge

The AI-specific meaning of "knowledge" warrants a more detailed discussion, due to the prevalence of that concept in the document and in the field.

While in other domains, the term can be associated to cognitive capabilities, in the context of AI it is a purely technical term that refers to contents, not capabilities. The knowledge concept is part of the data-information-knowledge hierarchy, according to which data can be used to produce information, and information can be used to produce knowledge. In the context of AI, these are purely technical, non-cognitive processes.

Knowledge differs from information in that information is observed by the system, while knowledge is what the system retains from such observations. Knowledge is structured and organized; it abstracts away from the specificities of individual observations. Depending on the goal, the same information can lead to different knowledge.

Knowledge differs from its representation in that the same knowledge can have different representations: it can appear under different concrete forms, each with their own pros and cons, but they all have the same meaning.

These distinctions have a technical impact, as some AI approaches, methods and other study topics rely entirely on the ability to produce different knowledge for the same information, or different representations for the same knowledge.

5.5 Cognition and cognitive computing

Cognition comprises the acquisition and processing of knowledge through reasoning, exclusive or shared experience, learning and perception. It encompasses concepts such as attention, the formation of knowledge, memory, judgment and evaluation, reasoning and computation, problem solving and decision making, comprehension and production of language.

Cognitive computing is among the subdisciplines of AI[27]. It aims to implement cognition using capabilities such as pattern identification from the processing of huge amounts of information. It enables people to interact more naturally with machines. Cognitive computing tasks are associated with machine learning, speech processing, natural language processing, computer vision and human-machine interfaces.

5.6 Semantic computing

Semantic computing addresses the matching of computational content semantics to human intentions. It provides representations for describing information and uses these representations for retrieving, managing, manipulating and creating content (such as text, video, audio, process, function, device and network). Semantic description of content enables uncertainty reduction in cognitive processes and logical reasoning on information. This in turn helps to achieve information enrichment, deconfliction, summarization and comparison. Therefore, semantic computing is an approach that combines prior information and learning.

5.7 Soft computing

Soft computing is a methodology that combines various techniques that can tolerate imprecision, uncertainty and partial truth to solve complex problems. Conventional computing methods are generally applied to find precise and rigorous solutions to problems. However, such solutions can be unsuitable or alternatively can result in extreme complexity. Soft computing is built on the understanding that the real world is often imprecise and uncertain. As a result, attempting to define precise solutions to real-world problems can often have associated costs and complexity. Thus, soft computing aims to leverage the tolerance for imprecision, uncertainty and partial truth to achieve tractable, robust and low-cost solutions[24]. Examples of soft computing techniques are fuzzy systems, evolutionary algorithms, swarm intelligence and neural network systems.

5.8 Genetic algorithms

Genetic algorithms simulate natural selection by creating and evolving a population of individuals (solutions) for optimization problems. The creation of new solutions based on an initial population is inspired from genetic mutations. The chromosome (set of "genes") is represented as a string of 0s and 1s. Once an initial population of chromosomes is generated, the first step is just to calculate the fitness of each chromosome. The fitness function value quantifies the optimality of a solution by ranking it against the other solutions. If the solution created is not optimal, then a pair of chromosomes is selected for exchanging parts (crossover) and creating two offspring chromosomes. In the next step, a mutation randomly changes at least one gene in the chromosomes. The initial population is replaced with the new population and a new iteration starts. GA iterations end when one of the termination criteria (usually a predefined number of iterations) is satisfied. In the end, the fittest chromosomes are retained[25].

5.9 Symbolic and subsymbolic approaches for AI

In the discipline of AI, many different points of view with different paradigms exist. There is no classification that will establish a clear-cut distinction between different types of AI. Nevertheless, it is possible to provide some dimensions along which AI systems can be positioned.

Since the foundation of AI as a discipline, two paradigms have developed in competition: symbolic AI and subsymbolic AI.

Symbolic AI involves encoding knowledge with symbols and structures and it mostly uses logics to model reasoning processes. In this paradigm, information is encoded explicitly using a formal representation, whose syntax is processable by a computer and whose semantics is meaningful to humans.

The other approach is subsymbolic AI, using the connectionist paradigm. This paradigm is not based on symbolic reasoning; rather, it relies on the implicit encoding of knowledge. This implicit knowledge representation is predominantly based on statistical approaches to the processing of experience or raw data. Examples of this type of AI system are various machine learning systems, including the different forms of deep neural networks.

Modern AI systems typically contain elements of both symbolic AI and subsymbolic AI. Such systems are called hybrid AI.

5.10 Data

Data are central to many AI systems. Many of them are designed to process data, and it is often necessary to use data for testing purposes. In the case of machine learning systems, their whole life cycle relies on the availability of data.

Data can come in structured form (e.g. relational databases) or unstructured form (e.g. emails, text documents, images, audio and files). Data are a key aspect of AI systems and they go through processes including:

— data acquisition, in which the data are obtained from one or more sources. Data can be sourced within the organization or brought in from outside. The suitability of the data needs to be assessed, for example, whether it is biased in some ways or whether it is broad enough to be representative of the expected operational data input;

— exploratory data analysis, where the data characteristics are examined for patterns, relationships, trends and outliers. Such analysis can guide later steps such as training and verification;

— data annotation, in which the significant elements of the data are added as metadata (e.g. information about data provenance or labels to aid with training a model);

— data preparation, in which the data are put into a form that can be used by the AI system;

— filtering, which is the removal of unwanted data. The effects of the filtering need to be carefully examined to avoid the introduction of unwanted bias and other issues;

— normalization, which is the adjustment of data values to a common scale so that they are mathematically comparable;

— de-identification or other processes, which can be required if the dataset includes personally identifiable information (PII) or is associated with individuals or organizations, before the data can be used by the AI system (e.g. see ISO/IEC 20889);

— data quality checking, in which the contents of the data are examined for completeness, for bias and other factors that affect its usefulness for the AI system. Checking for data poisoning is crucial to ensure that training data have not been contaminated with data that can cause harmful or undesirable outcomes;

— data sampling, in which a representative subset of the data is extracted;

— data augmentation, in which the data that are available in too small quantities undergo several kinds of transformations in order to expand the dataset;

— data labelling, in which datasets are labelled, which means that samples are associated with target variables. Labels are often needed for test data and validation data. Some ML approaches also rely on the availability of labels for training the model (see 5.11.1 and 5.11.3).

Depending on the use case and on the approach used, data in AI system can be involved in several ways:

— production data is the data processed by the AI system in the operation phase. not all ai systems involve production data, depending on the use case, but this is independent of the technical design and approach of the ai system.

— test data is the data used to evaluate the performance of the ai system, before its deployment. it is expected to be similar to production data, and proper evaluation needs test data to be disjoint from any data used during development. all ai approaches warrant evaluation but depending on the task it is not always adequate to use test data.

— validation data corresponds to data used by the developer to make or validate some algorithmic choices (hyperparameter search, rule design, etc.). it has various names depending on the field of ai, for instance in natural language processing it is typically referred to as development data. there are cases when no validation data are needed.

— training data is used specifically in the context of machine learning: it serves as the raw material from which the machine learning algorithm extracts its model to address the given task.

NOTE 1 In software assessment frameworks, validation is the process of checking whether certain requirements have been fulfilled. It is part of the evaluation process. In AI-specific context, the term "validation" is used to refer to the process of leveraging data to set certain values and properties relevant to the system design. It is not about assessing the system with respect to its requirements, and it occurs before the evaluation stage.

NOTE 2 In software assessment frameworks, "test" refers to various diverse processes such as searching for bugs, performing unit tests and measuring computation time. Its meaning in AI refers specifically to statistical evaluation of the system performance against a dedicated dataset.

5.11 Machine learning concepts

5.11.1 Supervised machine learning

Supervised machine learning is defined as “machine learning that makes use of labelled data during training” (3.3.12). In this case, ML models are trained with training data that include a known or determined output or target variable (the label). The value of the target variable for a given sample is also known as the ground truth. Labels can be of any type including categorical, binary or numeric values, or structured objects (e.g. sequences, images, trees or graphs) depending on the task. Labels can be part of the original dataset but in many cases, they are determined manually or through other processes.

Supervised learning can be used for classification and regression tasks, as well as for more complex tasks pertaining to structured prediction.

For information on supervised machine learning see ISO/IEC 23053.

5.11.2 Unsupervised machine learning

Unsupervised machine learning is defined as “machine learning that makes use of unlabelled data during training” (3.3.17).

Unsupervised machine learning can be useful in cases such as clustering where the objective of the task is to determine points of commonality among the samples in the input data. Reducing the dimensionality of a training dataset is another application of unsupervised machine learning where the most statistically relevant features are determined irrespective of any label.

For information on unsupervised machine learning see ISO/IEC 23053.

5.11.3 Semi-supervised machine learning

Semi-supervised machine learning is defined as “machine learning that makes use of both labelled and unlabelled data during training” (3.3.11). Semi-supervised machine learning is a hybrid of supervised and unsupervised machine learning.

Semi-supervised machine learning is useful in cases where labelling all the samples in a large training dataset would be prohibitive from a time or cost perspective. Refer to ISO/IEC 23053 for further details about semi-supervised machine learning.

5.11.4 Reinforcement learning

Reinforcement learning is the process of training an agent(s) interacting with its environment to achieve a predefined goal. In reinforcement learning, a machine learning agent(s) learns through an iterative process of trial and error. The goal of agent(s) is to find the strategy (i.e. build a model) for obtaining the best rewards from the environment. For each trial (successful or not), an indirect feedback is provided by the environment. The agent(s) then adjusts its behaviour (i.e. its model) based on this feedback. Refer to ISO/IEC 23053 for further information on reinforcement learning.

5.11.5 Transfer learning

Transfer learning refers to a series of methods where data intended for solving one problem is leveraged to apply the knowledge gained from it to a different problem. For example, information gained from recognizing house numbers in a street view can be used to recognize handwritten numbers. Refer to ISO/IEC 23053 for further details about transfer learning.

5.11.6 Training data

Training data consists of data samples used to train a machine learning algorithm. Typically, the data samples relate to some particular topic of concern and they can consist of structured or unstructured data. The data samples can be unlabelled or labelled.

In the latter case, the label is used to guide the process of training the machine learning model. For example, where the input data are images and the aim is to decide whether an image shows a cat, the label can be "true" for an image that is of a cat and "false" for an image that is not of a cat. This allows the trained model to represent a statistical relationship between attributes of a training data sample and the target variable.

The number of data samples in the training data and selection of appropriate features contribute to how well the resulting ML model fits the distribution of the data or target variable. However, there is a trade-off of the computational time and resources required for computing if the dataset is extremely large.

5.11.7 Trained model

This document defines a trained model as the result of model training which in turn is defined as the process to establish or to improve the parameters of a machine learning model, based on a machine learning algorithm, by using training data. A machine learning model is a mathematical construct that generates an inference, or prediction, based on input data or information. The trained model should be usable by an AI system to make predictions based on production data from the area of interest. Various standardized formats exist to store and transmit the trained model as a set of numbers.

5.11.8 Validation and test data

To assess the trained model, it is common to split the data acquired for developing a model into datasets for training, validation and test.

Validation data are used during and after training to tune hyperparameters. The test set is used to verify that the model has learned what it was supposed to. Both consist of data that are never shown

to the model during training. If one were to use training data for those purposes, the model is capable of "remembering" the correct prediction without actually processing the data sample. To avoid overestimating model performance, test data are not shown during tuning either.

When using cross-validation, data are split such that each data sample is used for both training and validation. This approach emulates the use of a larger dataset, which can improve model performance. Sometimes, insufficient data are available to allow for separate training, validation and test sets. In such cases, data are split only into two sets, namely 1. training or validation data, and 2. test data. Separate validation and training datasets are then generated from the training or validation data, for example via boot-strapping, or cross-validation.

5.11.9 Retraining

5.11.9.1 General

Retraining consists of updating a trained model by training with different training data. It can be necessary due to many factors, including the lack of large training datasets, data drift and concept drift.

In data drift, the accuracy of the model's predictions decays over time due to changes in the statistical characteristics of the production data (e.g. image resolution has changed, or one class has become more frequent in data than another). In this case, the model needs to be retrained with new training data that better represents the current production data.

In concept drift, the decision boundary moves (e.g. what is legal and what is not tends to change when new laws are published), which also degrades the accuracy of predictions, even though the data have not changed. In the case of concept drift, the target variables in the training data need to be relabelled and the model retrained.

When retraining an existing model, a specific consideration is to overcome or minimize the challenges of so-called catastrophic forgetting. Many machine learning algorithms are good at learning tasks only if the data is presented all at once. As a model is trained on a particular task, its parameters are adapted to solve the task. When new training data is introduced, adaptations based on those new observations overwrite knowledge that the model had previously acquired. For neural networks, this phenomenon is known as "catastrophic forgetting", and is considered one of their fundamental limitations.

5.11.9.2 Continuous learning

Continuous learning, also known as continual learning or lifelong learning, is incremental training of a model that takes place on an ongoing basis while the system is running in production. It is a special case of retraining, where model updates are repeated, occur with high frequency, and do not involve any interruption of operation.

In many AI systems, the system is trained during the development process before the system is put into production. This is similar in nature to standard software development, where the system is built and tested fully before it is put into production. The behaviour of such systems is assessed during the verification process and is expected to be unchanged during the operation phase.

AI systems that embody continuous learning involve the incremental update of the model in the system as it operates during production. The data input to the system during operation is not only analysed to produce an output from the system, but also simultaneously used to adjust the model in the system, with the aim of improving the model on the basis of the production data. Depending on the design of the continuous learning AI system, there can be human actions required in the process, for example data labelling, validating the application of a specific incremental update or monitoring the AI system performance.

Continuous learning can help dealing with limitations of the original training data and it can also help deal with data drift and concept drift. However, continuous learning brings significant challenges in ensuring that the AI system still operates correctly as it learns. Verification of the system in production

is necessary, as is the need to capture the production data to become part of the training dataset in case the AI system is updated at some future point.

Due to the risk of catastrophic forgetting, Continuous learning implies the ability to learn over time by accommodating new observations made on current data, while retaining previous knowledge.

The characteristics of continuous learning include:

— learning over time in dynamic environments (ideally in the open world);

— augmenting the previously learned knowledge by learning new knowledge to improve performance (either through new data or reasoning on existing knowledge);

— discovering new aspects of the task to be learned and learning them incrementally;

— learning on the job or learning while the system is running in production.

5.12 Examples of machine learning algorithms

5.12.1 Neural networks

5.12.1.1 General

Neural networks attempt to simulate intelligent capability in observing, learning, analysing and decision making for complex problems. Hence, the design of neural networks draws inspiration from the way neurons are connected in the brains of humans and animals. The structure of neural networks is composed of interconnected processing elements, called neurons. Each neuron receives several inputs and generates only one output. They are organized into layers, where the output of one layer becomes the input to the next layer. Each connection has an assigned weight related to the importance of the input. The neural network "learns" by training with known inputs, comparing actual output with the expected one and using the error to adjust weights. Thus, the links which produce correct answers are strengthened and those which generate incorrect answers are weakened.

This document defines deep learning as an approach to creating rich hierarchical representations through the training of neural networks with many hidden layers. This process allows the neural network to progressively refine the final output. Deep learning can reduce or eliminate the need for feature engineering as the most relevant features are identified automatically. Deep learning can require significant time and computing resources.

There are many neural network "architectures" (essentially, arrangements of neurons) and this is an active area of research with new neural network architectures continuing to be introduced. Examples of NN architectures include:

— feed forward neural network;

— recurrent neural network;

— convolutional neural network.

These NN architectures are described in 5.12.1.2 through 5.12.1.4

NOTE Refer to ISO/IEC 23053 for further information on NNs.

5.12.1.2 Feed forward neural network

FFNN is the most straightforward neural network architecture. It feeds information from the input to the output in one direction only. There are no connections between the neurons within the same layer. Two adjacent layers can be typically 'fully connected' in that each neuron in one layer has a connection to each neuron in the subsequent layer.

5.12.1.3 Recurrent neural network

5.12.1.3.1 General

RNNs[21] deal with inputs that appear in an ordered sequence, i.e. the ordering of the inputs in the sequence matters. Examples of such inputs include dynamic sequences like sound and video streams, but also static sequences like text or even single images. RNNs have nodes that both take input information from the previous layer and also factor in information from themselves from a previous pass. RNNs have a stateful property influenced by past learning. RNNs are widely used in speech recognition, machine translation, time series forecasting and image recognition. Refer to ISO/IEC 23053 for further information on RNN.

5.12.1.3.2 Long short-term memory network

An LSTM network is a form of RNN designed for problems that require remembering information with both longer and shorter time differences, making them suitable for learning long-term connections. They have been introduced to solve the vanishing gradient problem in RNNs associated with back-propagation[22].

LSTM networks can learn complex sequences, such as writing like Shakespeare or composing music. Refer to ISO/IEC 23053 for further information on LSTM.

5.12.1.4 Convolutional neural network

A CNN is a neural network that includes at least one layer of convolution to filter useful information from inputs. Common uses include image recognition, video labelling and natural language processing. Refer to ISO/IEC 23053 for further information on CNN.

5.12.2 Bayesian networks

Bayesian networks are graphical models used for generating predictions on the dependencies between variables. They can be used to derive probabilities for the causes or variables that can contribute to the outcome. This causality is very useful in applications such as medical diagnosis. Other applications where Bayesian networks are useful include data analysis, addressing incomplete data and mitigating overfitting of models to data. Bayesian networks rely on Bayesian probability: the probably of an event is dependent on the extent of belief in that event. Further information on Bayesian networks can be found in[20] and in ISO/IEC 23053.

5.12.3 Decision trees

Decision trees use a tree structure of decisions to encode possible outcomes. Decision tree algorithms are widely used for classification and regression. The tree is formed of decision nodes and leaf nodes. Each decision node has at least two branches, whereas leaf nodes represent the final decision or classification. Generally, the nodes are ordered in terms of the decision that gives the strongest predictor. Input data needs to be split into various factors in order to determine the best outcome. Decision trees are analogous to flow charts where at each decision node a question can be asked to determine which branch to proceed to.

5.12.4 Support vector machine

SVM is a machine learning method widely used for classification and regression. SVMs mark data samples into two different categories and then assigns new data examples to one category or the other. SVM are maximum-distance classification algorithms. They define a hyperplane to separate two classes above and below it, providing the maximal distance between the classifying plane and the closest data points. The points that are closest to the border are called support vectors. The orthogonal distance between support vectors and the hyperplane is half of the margin of SVM. The training of an SVM involves maximizing the margin subject to the data from the different categories that are on the

opposite side of the hyperplane. SVMs also use kernel functions to map data from the input space into a higher-dimensional (sometimes infinite) space, in which the classifying hyperplane will be chosen.

These hard-margin SVM are rarely used in practice. A hard-margin classifier only works if the data is linearly separable. With just one data sample on the wrong side of the hyperplane, the classifier cannot be solved.

In contrast, soft-margin classifiers allow data samples to violate the margin (i.e. to be situated on the wrong side of the hyperplane). Soft-margin classifiers attempt to achieve maximal margin while limiting margin violations.

Categorization of unlabelled data and use in prediction and pattern recognition are examples of the application of SVM. When using SVM for regression, the objective is the reverse of SVM classifier. In SVM regression, the objective is to fit as many data instances as possible inside the margin, while limiting margin violations (those samples outside the margin).

5.13 Autonomy, heteronomy and automation

AI systems can be compared based on the level of automation and whether they are subject to external control. Autonomy is at one end of a spectrum and a fully human-controlled system on the other, with degrees of heteronomy in between. Table 1 shows the relationship between autonomy, heteronomy and automation, including the null case of no automation.

Table 1 — Relationship between autonomy, heteronomy and automation

		Level of automation	Comments
Automated system	Autonomous	6 - Autonomy	The system is capable of modifying its intended domain of use or its goals without external intervention, control or oversight.
	Heteronomous	5 - Full automation	The system is capable of performing its entire mission without external intervention
		4 - High automation	The system performs parts of its mission without external intervention
		3 - Conditional automation	Sustained and specific performance by a system, with an external agent being ready to take over when necessary
		2 - Partial automation	Some sub-functions of the system are fully automated while the system remains under the control of an external agent
		1 - Assistance	The system assists an operator
		0 - No automation	The operator fully controls the system

NOTE In jurisprudence, autonomy refers to the capacity for self-governance. In this sense, also, "autonomous" is a misnomer as applied to automated AI systems, because even the most advanced AI systems are not self-governing. Rather, AI systems operate based on algorithms and otherwise obey the commands of operators. For these reasons, this document does not use the popular term autonomous to describe automation[30].

Relevant criteria for the classification of a system on this spectrum include the following:

— the presence or absence of external supervision, either by a human operator ("human-in-the-loop") or by another automated system;

— the system's degree of situated understanding, including the completeness and operationalizability of the system's model of the states of its environment, and the certainty with which the system can reason and act in its environment;

— the degree of reactivity or responsiveness, including whether the system can notice changes in its environment, whether it can react to changes, and whether it can stipulate future changes;

— whether its operation persists up until or beyond the completion of a particular task or the occurrence of a particular event in the environment (e.g. relevant to achieving a goal, timeouts or other mechanisms);

— the degree of adaptability to internal or external changes, necessities or drives;

— the ability to evaluate its own performance or fitness, including assessments against pre-set goals;

— the ability to decide and plan proactively in respect to system goals, motivations and drives.

Instead of substituting for human work, in some cases the machine will complement human work, which is called human-machine teaming. This can happen as a side-effect of AI development or a system can be developed specifically with the goal of creating a human-machine team. Systems that aim to complement human cognitive capabilities are sometimes referred to as intelligence augmentation.

Overall, the presence of accountable supervision during operation can assist in ensuring that the AI system works as intended and avoids unwanted impacts on stakeholders.

5.14 Internet of things and cyber-physical systems

5.14.1 General

AI is increasingly used as a component in embedded systems like internet of things and cyber-physical systems, either for analysing streams of information about the physical world arising from sensors, or for making predictions and decisions about physical processes that are used to send appropriate commands to actuators to control or influence those physical processes.

5.14.2 Internet of things

IoT is an infrastructure of interconnected entities, systems and information resources together with services which process and react to information from the physical world and the virtual world (3.1.8). Essentially, an IoT system is a network of nodes with both sensors, which measure properties of physical entities then transmit data relating to those measurements, and actuators, which change properties of physical entities in response to a digital input.

Medical monitoring and monitoring the state of the atmosphere are examples of IoT systems, where the output of the system is information that is intended to assist human beings in their activities (e.g. warnings to medical staff, weather forecasts for humans).

IoT systems involve networked IT systems interacting with physical entities. Foundational to IoT systems are digital IoT devices, in the form of sensors and actuators, that interact with physical entities. A sensor measures one or more properties of one or more physical entities and outputs data that can be transmitted over a network. An actuator changes one or more properties of a physical entity in response to a valid input, received over a network. Both sensors and actuators can be in many forms, such as thermometers, accelerometers, video cameras, microphones, relays, heaters, robots or industrial equipment for manufacturing or process control. See ISO/IEC 30141 for more information.

AI can play an important role in the context of IoT systems. This includes the analysis of incoming data and decision making which can assist in achieving the goals of the system, such as the control of physical entities and physical processes, by providing contextual, real-time and predictive information.

5.14.3 Cyber-physical systems

CPS are systems similar to IoT, but where control logic is applied to the input from sensors in order to direct the activities of actuators and thereby influence processes taking place in the physical world.

A robot is an example of a CPS system, where sensor input is directly used to control the activities of the robot and perform actions in the physical world.

Robotics encompasses all activities relating to the design, assembly, production, control and usage of robots for different kinds of applications. A robot is composed of electronic, mechanical, firmware and software components tightly interacting with each other to achieve the goals set for a specific application. Robots are usually comprised of sensors to assess their current situation, processors to provide control through analysis and the planning of actions and actuators to realize the actions. Industrial robots set in manufacturing cells are programmed to repeat in a precise way the same trajectories and actions over and over without deviation. Service robots or collaborative robots need to adapt to changing situations and dynamic environments. Programming this flexibility is intractably challenging because of all the variability involved.

AI system components can serve as part of the control software and the planning process through the "sense, plan, act" paradigm, thus enabling robots to adjust when obstacles appear or when target objects have moved. Coupling robotics and AI system components enables automated physical interaction with objects, environment and people.

5.15 Trustworthiness

5.15.1 General

Trustworthiness of AI systems refers to characteristics which help relevant stakeholders understand whether the AI system meets their expectations. These characteristics can help stakeholders verify that:

— AI systems have been properly designed and validated in conformance with state-of-the art rules and standards. This implies quality and robustness assurance;

— AI systems are built for the benefits of the relevant stakeholders who have aligned objectives. This implies awareness of the workings of AI algorithms and an understanding of the overall functioning by stakeholders. It also implies qualification or certification assurance of AI development and operation in conformance with legal requirements and sectorial standards when available;

— AI systems are provided with proper identification of responsible and accountable parties;

— AI systems are developed and operated with consideration for appropriate regional concerns.

For further information refer to ISO/IEC TR 24028.

5.15.2 AI robustness

For AI systems, robustness can describe their ability to maintain their level of performance, as intended by their developers, under any circumstances. An example of robustness is the ability of a system to perform within acceptable limits despite external or harsh environmental conditions. Robustness can encompass other attributes such as resilience and reliability. The proper operation of an AI system relates to, or can lead to, the safety of its stakeholders in a given environment or context (see ISO/IEC TR 24028).

For example, a robust ML-based AI system can have the ability to generalize to noisy inputs, such as an absence of overfitting. To achieve robustness, one option is to train the model or models using large training datasets including noisy training data (see ISO/IEC TR 24028).

Robustness properties demonstrate the ability (or inability) of the system to have comparable performance on atypical data as opposed to the data expected in typical operations, or on inputs dissimilar to those on which it has been trained (see ISO/IEC TR 24029-1).

When processing input data, an AI system is expected to generate predictions (its outputs) within some acceptable, consistent or effective range. Even if these outputs are not ideal, a system can still be considered robust. An AI system whose outputs do not fall within this acceptable, consistent or effective range when processing input data cannot be considered robust.

Robustness can be considered differently for different types of AI systems such as:

— Robustness of a regression model is the ability to have acceptable metrics of amplitude of response on any valid input.

— Robustness of a classification model is the ability to avoid inserting new classification errors when moving from typical inputs to inputs within a certain range of those.

5.15.3 AI reliability

Reliability is the ability of a system or an entity within that system to perform its required functions under stated conditions for a specific period of time (see ISO/IEC 27040).

Reliability of an AI system refers to the ability that enables it to provide required prediction (3.1.27), recommendation and decision consistently correctly during its operation stage (6.2.6).

Reliability can be affected by at least the robustness, generalizability, consistency and resilience of an AI system. All inputs and environment settings meeting stated criteria are supposed to be processed correctly during its functioning. Some of the inputs can be different with the ones used within its development stage, but can happen when the system is used. Backup of an AI system or a component also improve reliability, which would provide business logic implementations that behave the same with the original. It works when the AI system gets failed.

Reliability can support an AI system's functional safety, in the sense that automatic protections and operations are required (by stakeholders) for the system or part of it against defined failure.

5.15.4 AI resilience

Resilience is the ability of the system to recover operational condition quickly following an incident. Fault tolerance is the system's ability to continue to operate when disruption, faults and failures occur within the system, potentially with degraded capabilities.

With AI systems, as with other software system types, hardware faults can affect the correct execution of the algorithm.

Reliability relates to resilience, but the expected service levels and expectations are different, with resilience expectations possibly lower as defined by stakeholders. System with resilience can offer a degraded level of operation under some types of failure which can be acceptable to stakeholders. Resilient systems should also include approaches for recovery as needed.

5.15.5 AI controllability

Controllability is the property of an AI system that an external agent can intervene in its functioning. Controllability can be achieved by providing reliable mechanisms by which an agent can take over control of the AI system.

A key aspect of controllability is the determination of which agent(s) can control which components of the AI system (e.g. the service provider or product vendors, the provider of the constituent AI, the user or an entity with regulatory authority).

Further information on controllability can be found in ISO/IEC TR 24028:2020, 10.4.

5.15.6 AI explainability

Explainability is the property of an AI system that means that the important factors influencing a decision can be expressed in a way that humans can understand. Explainability can be particularly important when the decisions being made by an AI system affect one or multiple humans. Humans are liable to distrust a decision unless they can gain an understanding of how the decision was reached, especially where the decision is in some way adverse to them at a personal level (e.g. refusal of a loan application).

Explainability can also be a useful means of validating the AI system, even where the decisions do not directly affect humans. For example, if an AI system is analysing an image of a scene to identify entities in that scene, it can be useful to see an explanation of the reasons for a decision made about the content of the scene, as a way of checking that what is identified is indeed what is being claimed. There are counterexamples from the history of AI systems where no such explanations were available, and it was discovered that the AI system was identifying some entities in a scene based on spurious correlations that existed in the training data.

Explainability can be easier for some types of AI system than for others. Deep learning neural networks can be problematic since the complexity of the system can make it hard to provide a meaningful explanation of how the system arrives at a decision.

Rule-based algorithms, such as symbolic methods or decision trees, are often considered to be highly explainable, as those rules directly provide some explanation. However, the explanation can be less understandable when such models grow in size and complexity.

5.15.7 AI predictability

Predictability is the property of an AI system that enables reliable assumptions by stakeholders about the output. Predictability plays an important role in the acceptability of AI systems and is often mentioned in debates about ethics with regard to AI systems. Trust in the technology is often based on predictability: a system is trusted if users can infer how the system will behave in a particular situation, even if the users cannot explain the factors behind the system behaviour. On the contrary, users can stop trusting a system if the system starts to work surprisingly in situations where the correct answer seems obvious.

However, there are several issues with a naïve notion of predictability which is based on the idea that a human should be able to predict the behaviour of an AI system:

— A definition directly based on human understanding is inherently subjective. A definition of predictability should use objective, quantifiable criteria.

— It should be possible to establish trust in an AI system even if one (human) cannot predict its precise behaviour in all situations. A statistical guarantee of the appropriateness of its behaviour can be more useful. The rationale behind this statement is that many machine learning approaches produce necessarily unpredictable results.

Predictability is associated with accuracy. Methods to improve accuracy can reduce the likelihood that AI systems generate unpredictable outputs.

5.15.8 AI transparency

Transparency of AI systems supports human centred objectives for the system and is a topic of ongoing research and discussion. Providing transparency about an AI system can involve communicating appropriate information about the system to stakeholders (e.g. goals, known limitations, definitions, design choices, assumptions, features, models, algorithms, training methods and quality assurance processes). Additionally, transparency of an AI system can involve informing stakeholders about the details of data used (e.g. what, where, when, why data is collected and how it is used) to produce the system and the protection of personal data along with the purpose of the system and how it was built and deployed. Transparency can also include informing stakeholders about the processing and level of automation used to make related decisions.

NOTE That disclosure of some information in pursuit of transparency can run counter to security, privacy or confidentiality requirements.

5.15.9 AI bias and fairness

In a general sense, the meaning of the term bias depends on its context.

In AI, the term bias refers to the idea that different cases call for different treatment. In this sense, bias is that which allows machine learning systems to judge that one situation is different from another and to behave differently accordingly. As such, bias is fundamental to the machine learning process and to adapting behaviour to the particular situation at hand.

In social context, however, the term bias often refers to the notion that certain differences in treatment are unfair. To avoid confusion, in the context of AI, instead of bias, the term unfairness is used to refer to unjustified differential treatment that preferentially benefits certain groups more than others. Unfair AI system behaviour can lead to disrespect of established facts, beliefs and norms, leading to favouritism or discrimination.

While certain bias is essential for proper AI system operation, unwanted bias can be introduced into an AI system unintentionally and can lead to unfair system results. Sources of unwanted bias in AI systems are interrelated and include human cognitive bias, data bias and bias introduced by engineering decisions. Bias in training data is major source of bias in AI systems. Human cognitive biases can affect decisions about data collection and processing, system design, model training and other development decisions.

Minimizing unwanted bias in AI systems is a challenging goal, but its detection and treatment is possible. [ISO/IEC TR 24027:2021].

5.16 AI verification and validation

Verification is the confirmation that a system was built correctly and fulfils specified requirements. Validation is the confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled. Considerations in verification and validation include the following:

— Some systems are completely verifiable (e.g. all system components can individually be verified, as can the complete system).

— Some systems are partially verifiable and partially validatable (e.g. at least one system component can individually be verified, and the remaining system components can be validated, as can the complete system).

— Some systems are unverifiable but validatable (e.g. no system component can be verified, but all system components can be validated, as can the complete system).

— Some systems are unverifiable and partially validatable (e.g. no system component can be verified, but at least one system component can individually be validated).

— Some systems are unverifiable and unvalidatable (e.g. no system component can either be verified or validated).

5.17 Jurisdictional issues

AI systems can be deployed and operated in jurisdictions other than those in which the system was designed or manufactured. Developers and manufacturers of AI systems should be aware that applicable legal requirements can vary between jurisdictions.

For example, a car manufactured in one jurisdiction can be required to comply with different legal requirements to be authorized to enter a different jurisdiction.

Additionally, AI systems ordinarily require data to be gathered, processed and used during the development and operation stages of the AI system and disposed of during the retirement phase. Developers, manufacturers and users of AI systems should be aware that legal requirements for the collection, use and disposal of data can also vary between jurisdictions.

To mitigate the impact of varying legal requirements, developers and manufacturers of AI systems can make use of one or more of the following mitigations:

— Note the applicable legal requirements that can apply to the AI system during the preparation phase. This should also include legal requirements pertaining to the collection, use and disposal of data.

— Develop a plan for complying with the applicable legal requirements in the jurisdiction(s) in which the AI system is intended to be deployed and operated.

— Develop a plan to monitor compliance with legal requirements during the development, deployment, operation and retirement stages of the AI system.

— Develop a plan to monitor and respond to any changes in legal requirements.

— Adopt flexible design, deployment and operation approaches.

5.18 Societal impact

AI systems have a spectrum of risk, determined by the severity of the potential impact of a failure or unexpected behaviour. Relevant aspects for assessing the level of risk include the following:

— the type of action space the system is operating in (e.g. recommendations vs direct action in an environment);

— the presence or absence of external supervision;

— the type of external supervision (automated or manual);

— the ethical relevance of the task or domain;

— the level of transparency of decisions or processing steps;

— the degree of system automation.

For example, a system that only gives recommendations and cannot act on its own, in a domain that has no ethical relevance can be considered low risk. Conversely, an AI system can be considered high negative risk if its actions have direct impact on human lives, it operates without external supervision, and its decision-making is opaque.

NOTE For specific application domains, additional legal requirements, policies and standards can apply which can go beyond the impact analysis described in this subclause.

5.19 AI stakeholder roles

5.19.1 General

As shown in Figure 2, AI can involve several stakeholder roles and sub-roles. These roles and sub-roles are described in 5.19.2 through 5.19.7.

NOTE An organization or entity can take on more than one role or sub-role.

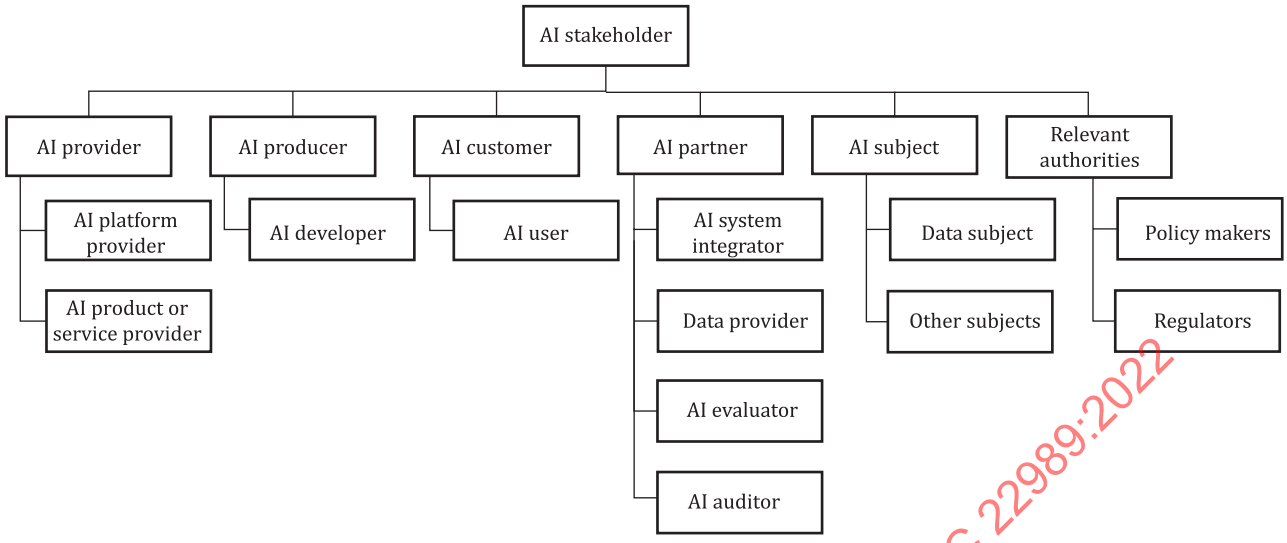

Figure 2 — AI stakeholder roles and their sub-roles

5.19.2 AI provider

5.19.2.1 General

An AI provider is an organization or entity that provides products or services that uses one or more AI systems. AI providers encompass AI platform providers and AI product or service providers.

5.19.2.2 AI platform provider

An AI platform provider is an organization or entity that provides services that enable other stakeholders to produce AI services or products.

5.19.2.3 AI service or product provider

An AI service or product provider is an organization or entity that provides AI services or products either directly usable by an AI customer or user, or to be integrated into a system using AI along with non-AI components.

5.19.3 AI producer

5.19.3.1 General

An AI producer is an organization or entity that designs, develops, tests and deploys products or services that use one or more AI system.

5.19.3.2 AI developer

An AI developer is an organization or entity that is concerned with the development of AI services and products. Examples of AI developers include, but are not limited to:

— Model designer: the entity that receives data and a problem specification and creates an AI model;

— Model Implementer: the entity that receives an AI model and specifies what computation to execute (the implementation to use and on what compute resources, for example CPU, GPU, ASIC, FPGA);

— Computation Verifier: the entity that verifies that a computation is being executed as designed;

— Model Verifier: the entity that verifies that the AI model is performing as designed.

5.19.4 AI customer

5.19.4.1 General

An AI customer is an organization or entity that uses an AI product or service either directly or by its provision to AI users.

5.19.4.2 AI users

An AI user is an organization or entity that uses AI products or services.

5.19.5 AI partner

5.19.5.1 General

An AI partner is an organization or entity that provides services in the context of AI. AI partners can perform technical development of AI products or services, conduct testing and validation of AI products and services, audit AI usage, evaluate AI products or services and perform other tasks. Examples of AI partner types are discussed in the following subclauses.

5.19.5.2 AI system integrator

An AI system integrator is an organization or entity that is concerned with the integration of AI components into larger systems, potentially also including non-AI components.

5.19.5.3 Data provider

A data provider is an organization or entity that is concerned providing data used by AI products or services.

5.19.5.4 AI auditor

An AI auditor is an organization or entity that is concerned with the audit of organizations producing, providing or using AI systems, to assess conformance to standards, policies or legal requirements.

5.19.5.5 AI evaluator

An AI evaluator is an organization or entity that evaluates the performance of one or more AI systems.

5.19.6 AI subject

5.19.6.1 General

An AI subject is an organization or entity that is impacted by an AI system, service or product.

5.19.6.2 Data subject

A data subject is an organization or entity that is affected by AI systems with following aspects:

— Subject of training data: where data pertaining to an organization or human is used in training an AI system, there can be implications for security and privacy, for the latter particularly where that subject is an individual human.

5.19.6.3 Other subjects

Other organizations or entities impacted by an AI system, service or product can be for example in the form of an individual or a community. For example, consumers who interacts with a social network that provides recommendations based on AI, drivers of vehicles with AI-based automation.

5.19.7 Relevant authorities

5.19.7.1 General

Relevant authorities are organizations or entities that can have an impact on an AI system, service or product.

5.19.7.2 Policy makers

These are organizations and entities that have the authority to set policies within an international, regional, national or industrial domain that can have an impact on an AI system, service or product.

5.19.7.3 Regulators

These are organizations and entities that have the authority to set, implement and enforce the legal requirements as intended in policies set forth by policy makers (5.17.9.2).

6 AI system life cycle

6.1 AI system life cycle model

The AI system life cycle model describes the evolution of an AI system from inception through retirement. This document does not prescribe a specific life cycle model but underlines some processes that are specific to AI systems that can occur during the system life cycle. Specific processes and timelines can occur during one or more of the life cycle stages and individual stages of the life cycle can be repeated during the system's existence. For example, it can be decided to repeat the "design and development" and "deployment" stages many times to develop and implement bug fixes and updates to the system.

A system life cycle model helps stakeholders build AI systems more effectively and efficiently. International Standards are useful in developing the life cycle model, including ISO/IEC 15288 for systems as a whole, ISO/IEC 12207 for software and ISO/IEC 15289 for system documentation. These International Standards describe life cycle processes for general systems, not specific to AI systems. Figure 3 provides an example of the stages and high-level processes that can be applied in the AI system life cycle. The stages and the processes can be iteratively carried out which is often required for AI system development and operation. There are various considerations that should be factored into the entire life cycle beginning to end. Examples of these considerations include:

— governance implications arising from either the development or use of AI systems;

— privacy and security implications due to the use of large amounts of data, some of which can be sensitive in nature;

— security threats that arise from data dependent system development;

— transparency and explainability aspects including data provenance and the ability to provide an explanation of how an AI system's output is determined.

Figure 3 shows an example of AI system life cycle model stages and high-level processes. Annex A shows how this AI system life cycle model maps to an AI system life cycle definition from the OECD.

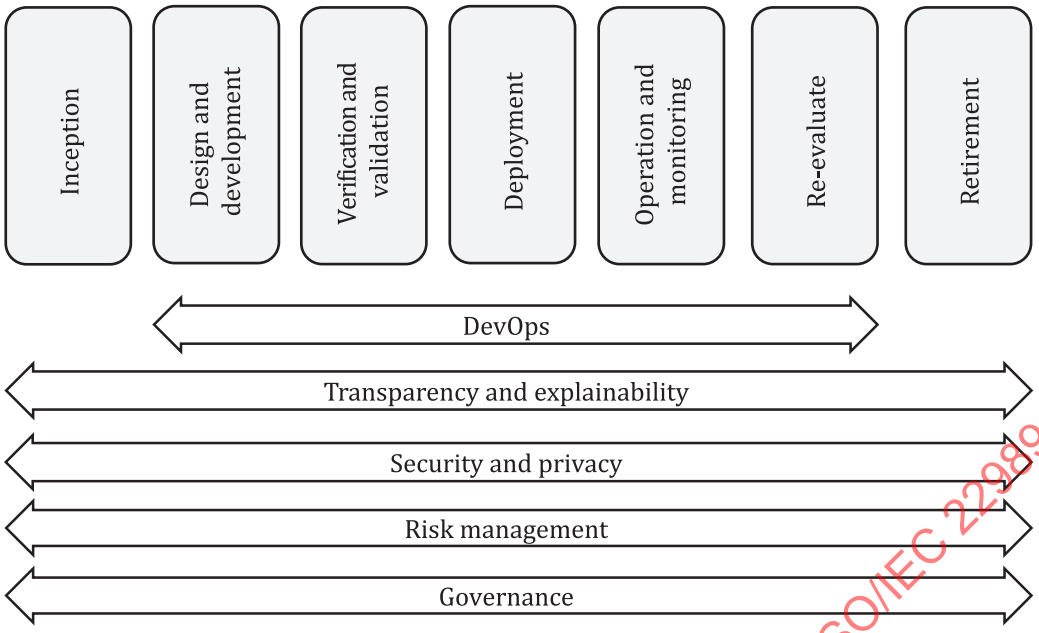

Figure 3 — Example of AI system life cycle model stages and high-level processes

AI systems differ from other types of systems, which can impact the life cycle model processes. For example:

— Most software systems are programmed to behave in precisely defined ways that are driven by their requirements and specifications. The AI systems based on machine learning use data-driven training and optimization methods to deal with widely varying inputs.

— Traditional software applications are usually predictable, which is less often the case for AI systems.

— Traditional software applications are also usually verifiable, while performance assessment of AI systems often requires statistical approaches and their verification can be challenging.

— AI systems typically need multiple iterations of improvement to achieve acceptable levels of performance.

Data management (encompassing processes and tools for data acquisition, data annotation, data preparation, data quality checking, data sampling and data augmentation) is a key aspect of AI systems.

The development and testing processes also differ for AI systems, since these processes are also data-based. This becomes more challenging for AI systems which use continuous learning (also known as continual learning or lifelong learning) where the system learns in the operation stage and where ongoing testing is required.

The release management process for AI systems is different from traditional software. While traditional software applications deal with code versioning and code diff functions, AI system releases include code and model diffs, as well as training data diffs if machine learning is used.

Some of the processes of the AI life cycle that differ from those in the traditional software life cycle are discussed in 6.2.

Figure 4 shows an example life cycle model for an AI system; different life cycle models are possible based on different development techniques. Figure 4 shows a series of stages of the life cycle and indicates some of the processes with each stage that are significant for AI systems and require special consideration, beyond the consideration required for the development of typical non-AI systems.

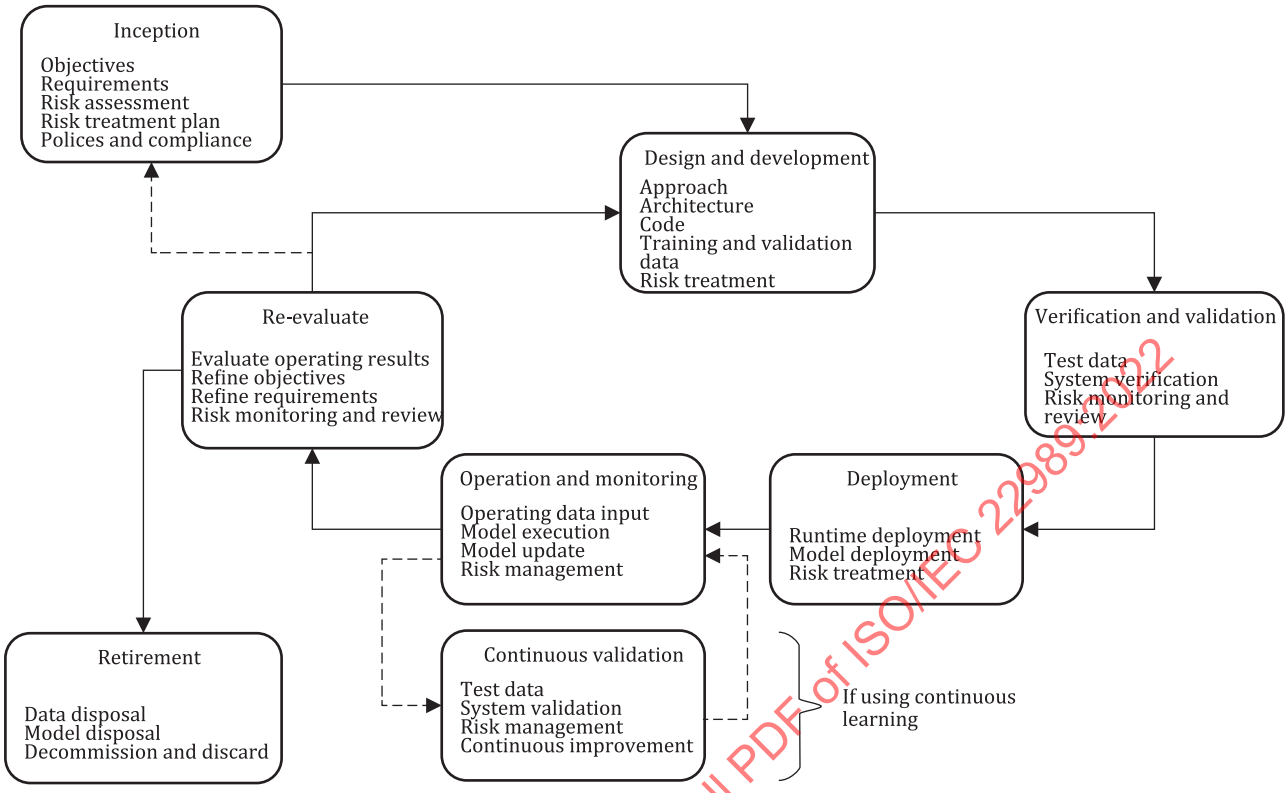

Figure 4 — Example AI system life cycle model with AI system-specific processes

As shown in Figure 4, the development and operation of AI systems tends to be more iterative in nature than for non-AI systems. AI systems tend to be less predictable and it typically takes some operational experience and adjustment of the AI system to meet its objectives.

6.2 AI system life cycle stages and processes

6.2.1 General

The processes described under each stage are representative examples as the specific processes will depend on the AI system. The processes can be performed in different orders and in some cases, in parallel.

These processes are not necessarily AI-specific themselves, but the stakes associated with AI make them of special importance in this context.

6.2.2 Inception

Inception occurs when one or more stakeholders decides to turn an idea into a tangible system. The inception stage can involve several processes and decisions that lead to a decision to proceed to the design and development stage. The inception stage can be revisited during the life cycle as new information is discovered in later stages. For example, it can be discovered that the system is not technically or financially feasible. Examples of processes that can occur during the inception stage include:

Objectives: Stakeholders should determine why an AI system needs to be developed. What problem does the system solve? What customer need or business opportunity does the system address? What are the metrics of success?

Requirements: Stakeholders should assemble a set of requirements for the AI system that spans the AI system life cycle. Failure to consider requirements for deployment, operation and retirement can result

in problems in the future. A multi-stakeholder approach including diverse subject-matter expertise can help to identify potential risks and unintended consequences of the system. Stakeholders should ensure the AI system requirements fulfil the AI system's objectives. Requirements should take into account that many AI systems are not predictable and the impacts this can have on achieving objectives. Stakeholders should consider factors of regulations and ensure the development and operation of AI system in compliance with relative compulsory policies.

Risk management: Organisations should assess risks in relation to AI during the whole life cycle of an AI system. The output of this activity should be a risk treatment plan. Risk management, including the identification, assessment and treatment of risk related to AI are described in ISO/IEC 23894.

Organizations should identify potential harms and benefits related to the AI system including conferring with representative users. This process can yield a set of values that can guide development of parts of the system including features, user interface, documentation and uses. Organizations should further study and refine the values to the extent they can become part of the system requirements. Legal, human rights, social responsibility and environmental frameworks can help in refining and describing the values.

In addition to the typical risks considered for a system, such as security and privacy, the plan should also address risks related to the values identified.

Transparency and accountability: Stakeholders should ensure that throughout the life cycle considerations such as data provenance, validity of data sources, risk mitigation efforts, processes and decisions implemented, are recorded to aid a comprehensive understanding of how the AI system outcomes are derived as well as for accountability purposes.

Cost and Funding: Stakeholders should forecast the costs of the AI system over the life cycle and ensure that funding is available.

Resources: Stakeholders should determine what resources are required to implement and complete each stage of the life cycle and they need to ensure the resources will be available when needed. Consideration should be given to the data that can be required for developing or evaluating an AI system. For ML based AI system, Particular consideration should be given to the training, validation and test data.

Feasibility: The inception stage leads to a decision of whether the AI system is feasible. A proof-of-concept can be conducted to determine whether the system meets requirements and objectives. Examples of requirements and objectives can include:

— addresses the defined problem,

— addresses a business opportunity or fulfilling a mission,

— meets specified capabilities and attributes.

If the AI system is deemed to be feasible, the stakeholders can decide to proceed to the development stage.

6.2.3 Design and development

The design and development stage creates the AI system and concludes with an AI system that is ready for verification and validation. During this stage, and particularly before the conclusion, stakeholders should ensure the AI system fulfils the original objectives, requirements and other targets identified during the inception stage. Examples of processes that can occur during the design and development stage include:

Approach: The stakeholders should determine an overall approach to designing the AI system, testing it and making it ready for acceptance and deployment. The approach stage can include consideration of whether both hardware and software will be needed, where to source components (e.g. develop from scratch, buy off-the-shelf hardware, use open source software).

Architecture: The stakeholders should determine and document the overall architecture of the AI system. The architecture and approach processes are related, and it can be necessary to iterate between the two.

Code: Software code for the AI system is developed or acquired.

Training data: AI systems embody acquired knowledge. Training data processing is a fundamental part of developing machine learning based AI systems (see 5.10).

Risk treatment: Organizations should implement processes and controls described in the risk treatment plan (see ISO/IEC 23894).

6.2.4 Verification and Validation

Verification and validation checks that the AI system from the design and development stage works according to requirements and meets objectives.

Examples of processes that can form part of verification and validation include:

Verification: The software is tested for functionality and bugs as is any hardware. Systems integration testing can also be done. A performance test can be conducted, checking whether the response time, delay or any other relevant performance characteristic of the AI system meets specific requirements.

An important aspect of AI systems is the need to verify that the AI capabilities work as designed. This requires the acquisition, preparation, and use of test data. Test data needs to be separate from any other data used during design and development and it also needs to be representative of input data that the AI system is expected to process.

Acceptance: Stakeholders deem the AI system to be functionally complete and at an acceptable level of quality and is ready to be deployed.

Risk monitoring and review: Organizations should review verification, test and validation results to be aware of events and conditions leading to risks according to the risk treatment plan (see ISO/IEC 23894).

6.2.5 Deployment

The AI system is installed, released or configured for operation in a target environment. Example processes of the deployment stage can include:

Target: AI systems can be developed in one environment and then deployed to another. For example, a self-driving system can be developed in a lab and then deployed in millions of automobiles. Other types of AI systems can be developed on client devices and then deployed to the cloud. For some AI systems, it is important to distinguish between the software components that are deployed and the model that can be deployed separately and which is used by the software at runtime. Software and model can be deployed independently.

Risk treatment: Organizations should review and improve processes and controls for risk management and potentially update the risk treatment plan (see ISO/IEC 23894).

6.2.6 Operation and monitoring

During the operation and monitoring stage the AI system is running and generally available for use.

Example processes that can occur during the operation and monitoring stage include:

Monitor: The AI system is monitored for both normal operation and also for incidents including unavailability, runtime failures or errors. These events are reported to relevant AI providers for action.

Repair: If the AI system fails or is experiencing errors, repairs to the system can be necessary.

Update: AI system software, models and hardware can be updated to meet new requirements and to improve performance and reliability.

Support: Users of the AI system are given any necessary support needed to successfully use the system.

Risk monitoring and review: Organizations should monitor AI systems during operation to assure and improve the quality and effectiveness of the risk management process (see ISO/IEC 23894).

6.2.7 Continuous validation

If the AI system uses continuous learning, the operation and monitoring stage is extended into an additional stage of continuous validation. In this stage, incremental training takes place on an ongoing basis while the system is running in production. The operation of the AI system is continually checked for correct operation using test data. It is also the case that the test data itself can require some updates to be more representative of current production data and therefore provide a more faithful evaluation of the AI system capabilities.

Risk management continuous improvement: Continuous validation should also be used to enable continuous improvement to risk management processes (see ISO/IEC 23894).

6.2.8 Re-evaluation

After the operation and monitoring stage, based on the results of the work of the AI system, the need for a reassessment can arise. Example processes that can occur during the re-evaluation stage include:

Evaluate operating results: The results of the system in operation should be evaluated and assessed against the objectives and the risks identified for the AI system.

Refine objectives: If the original objectives cannot be achieved by the AI system, or that the objectives need modification once experience of operating the system is available. This leads to refinement of the objectives.

Refine requirements: Operating experience can provide evidence that some of the original requirements are not valid in some ways and this can lead to the refinement of requirements, possibly with new requirements or the removal of some requirements.

Risk monitoring and review: Organizations should monitor events and conditions leading to risks as described in the risk treatment plan (see ISO/IEC 23894).

6.2.9 Retirement

At some point the AI system can become obsolete to the extent that repairs and updates are not good enough to meet new requirements. Example processes that can occur during the retirement stage include:

Decommission and discard: If the purpose of the AI system no longer exists, or a better approach has emerged, the AI system can be decommissioned and completely discarded. This can include the data associated with the system.

Replace: If the purpose of the AI system continues to be relevant, but a better approach has emerged, the AI system (or components of the AI system) can be replaced.

7 AI system functional overview

7.1 General

This document defines AI system as engineered system that generates outputs such as content, forecasts, recommendations or decisions for a given set of human-defined objectives. AI systems do not understand; they need human design choices, engineering and oversight. The degree of oversight

depends on the use case. At a minimum, oversight is typically present during training and validation. Such oversight is useful to ensure that the AI system is developed and used as intended, and that impacts on stakeholders are appropriately considered throughout the system life cycle.

Figure 5 depicts a functional view of an AI system, where inputs are processed using a model to produce outputs, and that model can be either built directly or from learning on training data. The parts drawn with dashed lines are for ML based AI Systems.

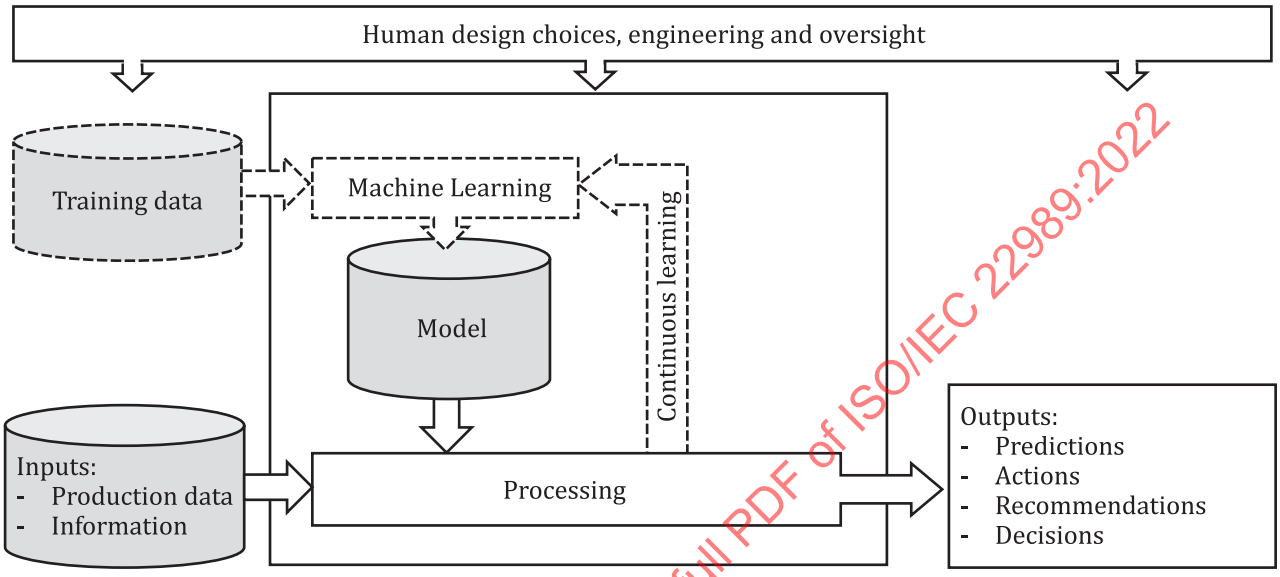

Figure 5 — AI system functional view

The purpose of this view is to provide a non-technical description of what AI systems do to achieve a result. To summarize, AI systems contain a model which they use to produce predictions and these predictions are in turn used to successively make recommendations, decisions and actions, in whole or in part by the system itself or by human beings.

7.2 Data and information

Data can be input to the AI system in production, in which case it is called production data. Input data can require preparation before it is presented to the AI system, such as the extraction of relevant features.

Input to an AI system can also be information instead of data, typically for optimization tasks where the only input needed is the information on what is to be optimized. Some AI systems do not require any input at all, but rather perform a given task on request (e.g. generating some synthetic image).

For ML, training data is used, to acquire some information about the domain of interest and the task to be addressed.

Data has other uses for the development and evaluation of AI systems (see 5.10).

7.3 Knowledge and learning

The model used by the AI system for its processing and for problem-solving is a machine-readable representation of knowledge.

There are two main types of such knowledge: declarative and procedural.

— Declarative knowledge is about what is. It is easy to verbalize and will translate into statements. For example, “the death cap mushroom is poisonous” is declarative knowledge.

— Procedural knowledge is about how to do something. It is often hard to verbalize. It will translate into procedures. For example, to know if a mushroom is poisonous, you can apply procedural knowledge: "If you have a mushrooms book, check into the book to see if you can identify your mushroom. If yes, the book will tell you the answer. If you can't, then go and see the pharmacist."

Knowledge has various possible representations from implicit to explicit ones.

Knowledge can also come from various sources, depending on the algorithms used: it can pre-exist, it can be acquired through sensing and learning processes, or it can be a combination of both.

Heuristic systems: AI systems that do not involve learning are called heuristic. Classical expert systems or reasoning systems equipped with a fixed knowledge base are good examples. In these cases, the system developers take advantage of human knowledge to provide reasonable rules for the AI system's behaviour.

ML based AI systems: AI systems that involve learning are said to be machine learning based. Learning entails computational analyses of a training dataset to detect patterns, build a model, and compare the output of the resulting model to expected behaviours. It is also known as training. The resulting knowledge base is a trained model based on a mathematical function and training set that represents a best approximation of the behaviour based on a given environment.

Continuous learning: AI systems also vary in terms of when and how data is acquired. In some cases, the knowledge base is static and provided at the outset, together with the system's pre-programmed components. In other cases, the knowledge base changes or adapts over time, with the information being updated over the course of its operation. Machine learning systems can be characterized by when, during their lifetime, learning occurs. In many cases, an initial training phase yields some approximation of the actual target function, and the system continues as-is without updating that internal representation based on new examples. An alternative approach, called lifelong or continuous learning, spreads the learning over time; the model is updated iteratively as new data is made available. In practice, models using lifelong learning usually implement a combination of both approaches; following an initial training phase in which the bulk of learning occurs, the model is refined with further data over time.

7.4 From predictions to actions

7.4.1 General

The result of input processing by the AI system can be of various natures, depending on the level of automation of the system. Depending on the use case, the AI system can produce only a raw, technical output (predictions), or it can take more effective steps in proposing or applying itself actions on the environment (recommendations, decisions, and finally actions).

In the case of classification, erroneous results are usually categorized as false positive or false negative errors. A false positive is described as a positive prediction when the real result is negative. A false negative is a result of the model incorrectly predicting a negative outcome. Users of AI systems need to understand the effects of an errant outcome including the possibility of a biased prediction. Such issues can directly reflect characteristics of the tools, processes or data used to develop the system.

A key point is that AI outputs are error prone. The output has a probability of being a correct, rather than being absolutely true. Both the system designers and the users of AI systems need to understand that such systems can produce incorrect outputs and the accountability implications of using such incorrect outputs.

7.4.2 Prediction

The term "prediction" refers to the very first output of an AI system.

AI systems make predictions by applying model to new data or situations. In the credit scenario in 7.4.3, an AI system was developed using previous loan records. To continue the example, when a new person

applies for a loan, their information is fed to the model which then produces an estimate of how likely that person is to repay a loan.

NOTE In artificial intelligence usage, prediction does not necessarily imply a statement about the future—it only refers to the output of an AI system, which can be a type of flower in an image, or a translation into another language.

7.4.3 Decision

Decisions correspond to choosing a specific course of action, with intention of applying it.

Decisions can be made either by the system itself or by human beings, based on the system outcomes. They can be made based on recommendations, or directly based on predictions.

For example, is a person predicted to be a good credit risk, a human loan officer can analyse that outcome together with this person's other information and the lender's situation, and then decide to approve this person's loan application. Alternatively, the system can make itself a recommendation to approve the loan and estimate the probability of that being the best course of action with respect to the lender's expectations, so that a loan officer who deems that probability acceptable decides to approve the loan. Or the loan application can be approved automatically, by applying system decision thresholds on such recommendations.

Human judgment and oversight are involved in various ways in that decision process. Human-defined thresholds are typically set by considering the risks associated with automating decisions. Even when decisions are fully automated, humans can use predictions to monitor the resulting decisions.

7.4.4 Action

Actions follow decisions, this is when the outcomes of the AI system start to affect the real word (whether physical or virtual).

Taking an action is the final step of applying information in an AI system. For example, in the credit application example in 7.4.2, once the person's loan is approved, the actions can include preparation of loan documents, getting signatures and issuing payments. Consider a robot. An action can be instructions to the robot's actuators to position its arms and hands. Depending on the AI system, the action can take place within the AI system boundary or outside of the AI system boundary.

8 AI ecosystem

8.1 General

Figure 6 represents an AI ecosystem in terms of functional layers. Large AI systems do not rely on a single technology, but rather on a mix of technologies developed over time. Such systems can use different technologies simultaneously, e.g. neural networks, symbolic models and probabilistic reasoning.

Each layer of Figure 6 uses the lower layers' resources for the implementation of its functions. Lighter shaded boxes are sub-components of a layer or function. The sizes of layers and sub-components are not indicative of importance.

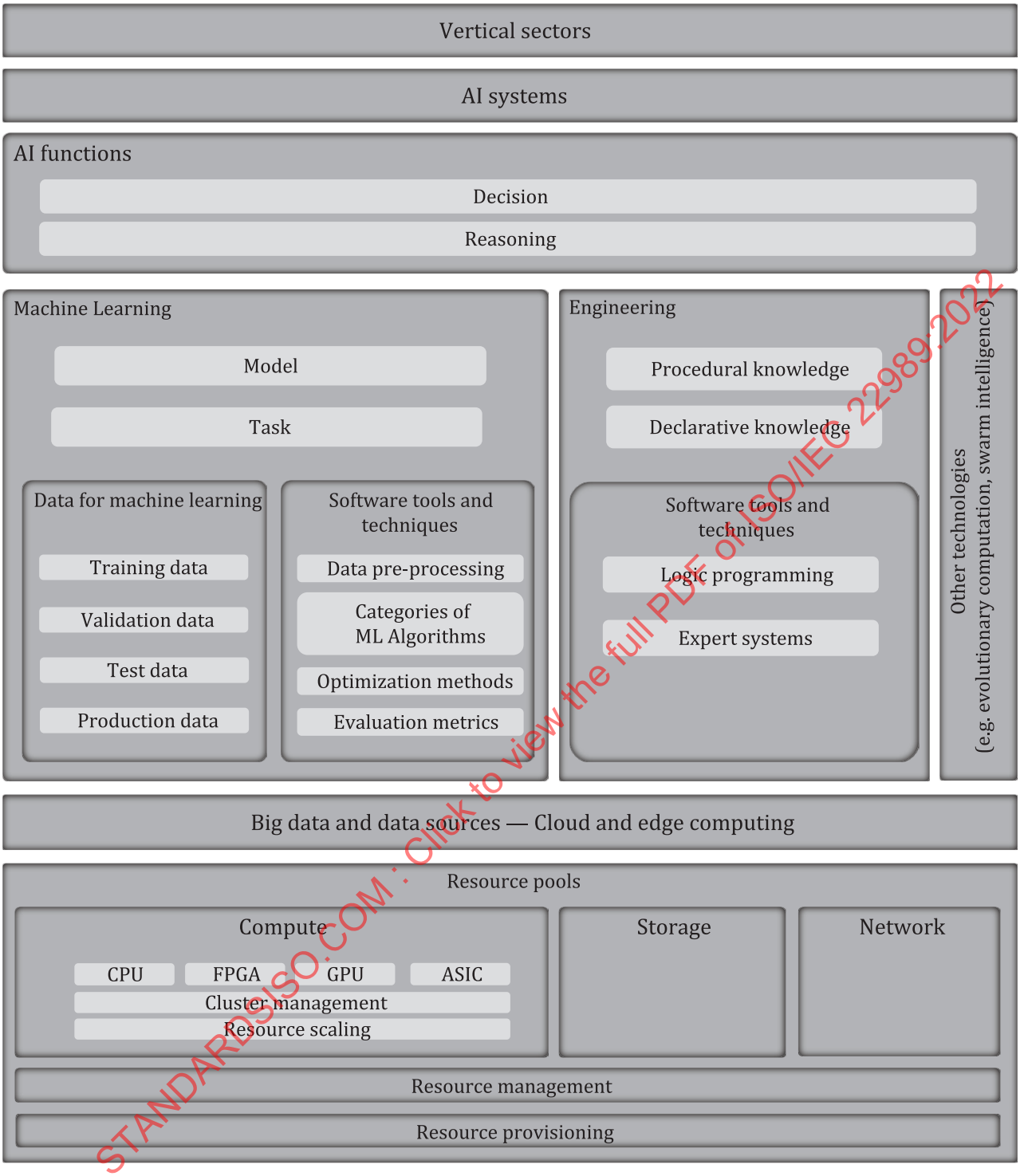

Figure 6 — AI ecosystem

Creating AI systems remains a topic of ongoing research. Meanwhile, the use of AI technology is becoming an inherent part of many industries, each with its own needs, values and legal constraints.

Specialized AI applications, such as those used for computer vision or for natural language processing, are themselves becoming the building blocks for implementation of different products and services. These applications are driving specialized AI system designs and, consequently, are setting the priorities for research and development.

AI technology often requires the use of significant compute, storage and network resources, for example during the training phase of a machine learning system. Such resources, as shown in Figure 6, can be provided efficiently using cloud computing.

The following subclauses describe the major components of the AI ecosystem as shown in Figure 6.

8.2 AI systems

AI systems can be used in numerous applications and to solve a multitude of tasks. Clause 9 describes examples of applications using AI such as image recognition, natural language processing and predictive maintenance. Clause 5 lists numerous task categories that AI systems can address.

AI systems follow a global functional path where information is acquired, either by hard coding (through engineering) or by machine learning to build a model of the domain. Then information, encoded in the form of a model, is applied at a reasoning level, where potential solutions are computed, then at a decision level, where a choice is made among the potential actions that can achieve the goal. The reasoning level includes spatial reasoning, temporal reasoning, common sense reasoning, computed policy application or any form of reasoning that can be coded. The decision level includes choice based on preferences or utilities among the actions.

8.3 AI function

Once the model is built, an AI function has the role to compute a prediction, a recommendation or more generally a decision that would help to reach the current goal of the AI system.

Reasoning is solely about applying the available data in the current situation to the model and asking the model what the possible options are.

Some examples of technologies which implement forms of reasoning include planning, Bayesian reasoning, automated theorem provers, temporal and spatial reasoning and ontology reasoners.

Among these possible options that would probably achieve the goal, the system still needs to decide which is the best.

Preferences and utilities come in play: an automated taxi will maximize the well-being of the client, a poker playing program will maximize its profit.

8.4 Machine learning

8.4.1 General

Machine learning is a process using computational techniques to enable systems to learn from data or experience. It employs a set of statistical methods to find patterns in existing data and to then use patterns to make predictions on production data.

In traditional computer programming, a programmer specifies the logic to solve a given problem by specifying exact computational steps using a programming language. In contrast, the logic of a machine learning model is in part dependent on the data used to train the model. Thus, the computations, or steps, needed to solve the problem are not determined a priori.

Also, in contrast to traditional computer programming, machine learning models can improve over time without being re-written by being re-trained on new, additional data and by using techniques to optimize model parameters and data features.

8.5 Engineering

8.5.1 General

In Engineering approaches by human experts, the processing relies solely on the expertise of the developer and their understanding of the task. Knowledge is not learned from data but through hardcoding by the developer based on their experience in a specific domain.

There are two main types of knowledge: declarative and procedural. See 7.3 for more details of both types of knowledge.

8.5.2 Expert systems

As the name implies, an expert system is an AI system that encapsulates knowledge provided by a human expert in a specific domain to infer solutions to problems.

An expert system consists of a knowledge base, an inference engine and a user interface. The knowledge base stores declarative knowledge of a specific domain, which encompasses both factual and heuristic information. The inference engine holds procedural knowledge: the set of rules and the methodology for reasoning. It combines facts provided by the user with information from the knowledge base.

Inference is done using predefined rules according to the expert and with logical statement evaluations. Classes of problems that can be solved using expert systems include classification, diagnosis, monitoring and prediction.

8.5.3 Logic programming

Logic programming is a form of programming based on programming languages that express formal logic. Prolog is an example of a logic programming language.

Formal logics for AI have been a significant research focus. Many kinds of formal logics target the modelling of human reasoning in various situations. Logic programming provides a framework to implement these models of human reasoning. AI agents need to be capable of reproducing different kinds of reasoning in a clearly specified, transparent and explainable fashion.

Logic programming with declarative statements coupled to strong natural language processing can lead an agent to reason by analogy, draw conclusions and generalize about objects and the environment.

EXAMPLE Apache Jena[19] is a semantic web framework that provides an inference engine.

8.6 Big data and data sources — cloud and edge computing

8.6.1 Big data and data sources

All ML systems use data. That data can take various forms. In some cases, the data used by ML systems is big data. This block in Figure 6 represents the sources, formats and the typical processing of big data regardless of its uses. This subclause further describes the major components shown in Figure 7.

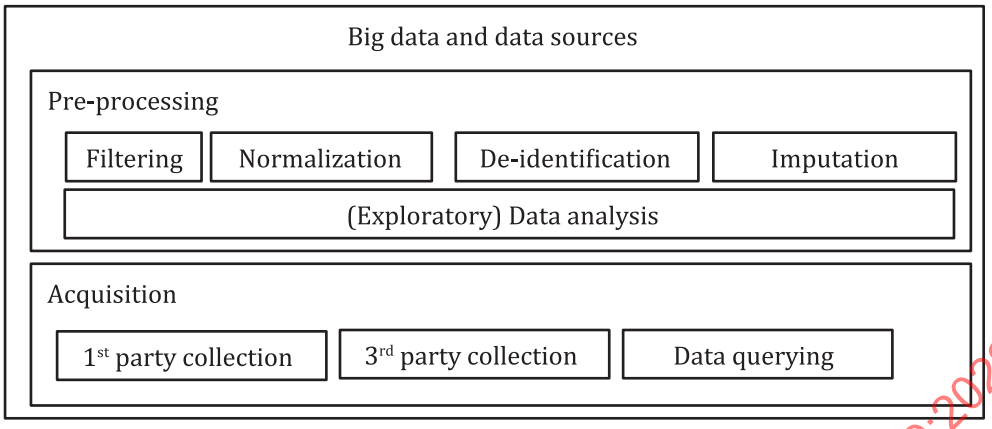

Figure 7 — Big data and data sources

Big data is extensive datasets whose characteristics in terms of volume, variety, velocity and variability require specialized technologies and techniques to process and realize value. For example, technologies have been developed specifically to enable distributed processing of large datasets using clusters of computers while using simple programming models. Additionally, storage and database technologies have been developed specifically to manage large data volumes that can be composed of other large volumes.

Big data has become important as organizations have increased the breadth and depth of data collection and therefore require specialized technologies and techniques to gain insights.

For more information about big data, see ISO/IEC 20546 and ISO/IEC 20547-3.

Big data has many uses for AI systems, and it is an enabler of many such systems. The availability of large collections of unstructured data in different application fields provide new insights as a result of using AI techniques such as knowledge discovery and pattern recognition. The availability of vast amounts of data for training results in improved machine learning models capable of being used a broad range of applications.

Data can be acquired by the same organization that uses it (1st party collection). For example, retailers use the transaction data they acquire from point-of-sale systems that they own. Data can also be acquired by 3rd parties such as research organizations and other data providers who collect data and then sell or share the data with other organizations that put it to use. Additionally, data can be acquired by querying and joining data from different datasets, both 1st and 3rd party.

Data can come from many sources such as:

— point of sale and other transactions;

— polls or surveys;

— statistical research;

— recorded observations;

— sensors;

— images;

— audio recordings;

— documents;

— interactions with systems.

8.6.2 Cloud and edge computing

Cloud computing is a paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on demand, see ISO/IEC 17788 and ISO/IEC 17789.

Cloud computing is commonly associated with large, centralized data centres, which have the capability of providing very large capacities of processing and data storage. Such large capacities can be essential to some parts of the AI life cycle, particularly when processing large datasets to train the AI systems and build the models used within them.

Edge computing is distributed computing in which processing and data storage takes place at or near the edge of the network where the nearness is defined by the system's requirements. The edge is the boundary between pertinent digital and physical entities, delineated by networked sensors and actuators (see ISO/IEC 23188).

Edge computing is largely about the placement and operation of software components and data storage. Where software components, such as those associated with AI systems, are dealing with IoT devices (sensors and actuators) there is often a need to minimize latencies and to produce results with significant time constraints (often termed real time), or a need for resilience so that a system can still function if communications are interrupted, or a need to protect the privacy of individual's data captured from edge devices. To achieve this can require that processing and data storage is done at or near the edge. For details, see ISO/IEC 23188.

However, it is important to understand that cloud computing can be deployed in many places within a distributed computing environment, including in places that are not centralized and that are near the edge. In this form, cloud computing can offer flexible and dynamic deployment of both software and data, using virtualized processing and virtualized data storage combined with resource pooling and rapid elasticity and scalability to enable appropriate placement and operation of components of AI systems.

It is commonly the case that edge computing systems are combined with centralized systems to create complete solutions, taking advantage of the capabilities of each type of system.

Three principal ML system designs combine cloud and edge: model training in the cloud, model training at the edge, and model training in cloud and at the edge.

a) Cloud services can be utilized as a centralized platform for the training of ML models (Figure 8). Due to resource restrictions of edge devices, computing and storage intensive tasks related to training, validation and maintenance of models are performed using a cloud infrastructure. Once trained, the model is deployed, applied, and, if necessary, updated on edge devices. Data from edge devices can be further used to perform training activities or, as in the case of reinforcement learning, provide feedback on the model quality.

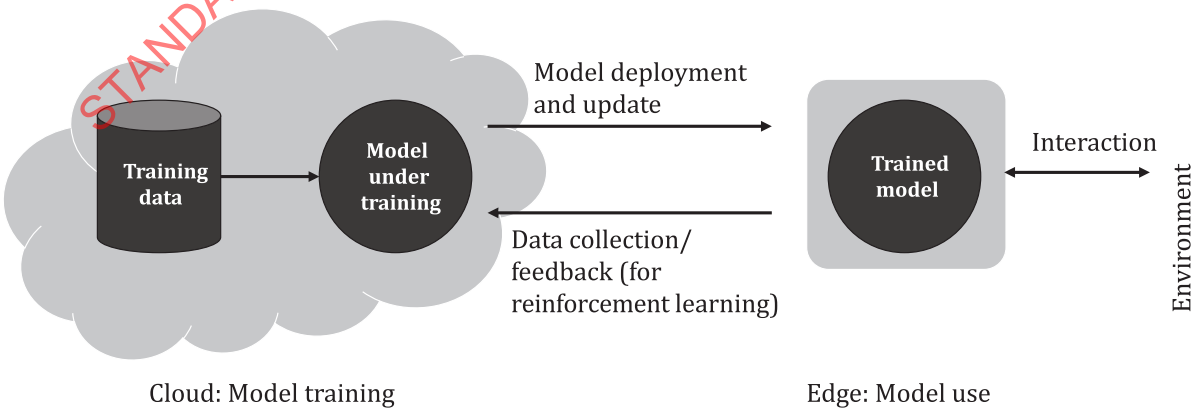

Figure 8 — Example of model training in the cloud