

ICS 35.240.01

**Künstliche Intelligenz –
Quantifizierung von Unsicherheiten im Maschinellen Lernen;
Text Englisch**

Artificial Intelligence –
Uncertainty quantification in machine learning; Text in English

Intelligence Artificielle –
Quantification de l'incertitude dans l'apprentissage automatique; Texte en anglais

Gesamtumfang 46 Seiten

Dieses Dokument wurde durch die im Vorwort genannten Verfasser erarbeitet und verabschiedet.

Contents

	Page
Foreword	4
Introduction	6
1 Scope	7
2 Normative references	7
3 Terms and definitions	7
4 Uncertainties in machine learning	13
4.1 The use of “uncertainty” in existing AI standardization	13
4.2 Elements of an ML system	13
4.3 Uncertainty in machine learning ontology	15
4.4 Sources of uncertainty in ML models	17
4.5 Uncertainty calibration	18
5 Applications of uncertainty quantification in ML	19
5.1 Motivation	19
5.2 Runtime monitoring	19
5.3 Transfer learning	20
5.4 Out-of-distribution detection	20
5.5 Active learning	21
5.6 Reinforcement learning	21
5.7 Fusion	21
6 Uncertainty quantification approaches and properties	21
6.1 Introduction and motivation	21
6.2 Probabilistic models	22
6.3 Uncertainty quantification methods	22
6.3.1 Introduction	22
6.3.2 Model-agnostic methods	22
6.3.3 Model-specific methods	23
6.4 Properties of uncertainty quantification approaches	23
7 Requirements and recommendations	23
7.1 System dependencies	23
7.2 Situation awareness	24
7.3 Sources	24
7.4 Factors	24
7.5 Separation of concerns	25
7.6 Agnosticism	25
7.7 Statistical confidence	25
7.8 Calibration	25
7.9 Overconfidence	25
7.10 Interpretability, transparency and human-understandability of the uncertainty quantification results	26
7.11 Explainability, transparency and human-understandability of the uncertainty quantification approach	26
7.12 System integration	26
Annex A (informative) Types of uncertainty: aleatoric and epistemic	27
Annex B (informative) Exemplary uncertainty calibration metrics and methods	28
Annex C (informative) Use cases	34
C.1 Uncertainty quantification for sensor fusion during landing of an aerial vehicle	34
C.1.1 Use case definition	34
C.1.2 Solution via uncertainty quantification	34
C.1.3 Examples of relevant requirements and recommendations	37
C.2 Uncertainty quantification for traffic sign recognition	38
C.2.1 Use case definition	38

C.2.2 Solution via uncertainty quantification	38
Bibliography	41

Figures

Figure 1 — Overview of uncertainty quantification ontology	15
Figure 2 — Overview of sources and layers of uncertainty	18
Figure B.1 — Reliability diagram for examining uncertainty calibration for categorical variables	33
Figure B.2 — Reliability diagrams for examining uncertainty calibration for continuous random variables [70]	33
Figure C.1 — Drone landing use case	34
Figure C.2 — Drone landing system architecture	36
Figure C.3 — Exemplary drone camera image with probabilistic landing pad detection [74]	36
Figure C.4 — Exemplary drone state and uncertainty evolution without and with POD fusion over time	37
Figure C.5 — Schematic uncertainty wrapper — Additional external information sources for the uncertainty wrapper and its outcomes	39
Figure C.6 — Decision tree example considering semantic factors [43]	40

Tables

Table A.1 — Overview of aleatoric and epistemic uncertainty	27
Table B.1 — Common metrics for measuring uncertainty calibration of an ML model for categorical and continuous random variables	28
Table B.2 — Common (post-hoc) uncertainty calibration methods for the correction of possibly uncalibrated uncertainty estimates of an ML model for categorical and continuous random variables	30

Foreword

This DIN SPEC has been developed according to the PAS procedure. The development of a DIN SPEC according to the PAS procedure is carried out in DIN SPEC-consortiums and does not require the participation of all stakeholders.

This document has been developed and adopted by the initiator(s) and authors named below:

- Industrieanlagen-Betriebsgesellschaft mbH
Dr. Lukas Höndorf, Rafal Kulaga, Bastian Bernhardt, Suman Subedi
- Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS
Dr. Maximilian Poretschkin, Elena Gina Haedecke
- ai.dopt
André Blum
- DEKRA Digital & Product Solutions
Dr. Jörg Firnkorn
- e:fs TechHub GmbH
Dr. Fabian Küppers
- Fraunhofer-Institut für Experimentelles Software Engineering IESE
Dr. Michael Kläs, Dr. Rasmus Adler
- Fraunhofer-Institut für Kognitive Systeme IKS
Karsten Roscher
- Giesecke+Devrient
Jonas Goltz
- Helsing GmbH
Jacobus Bachmüller, Moritz Krah
- Leibniz Fachhochschule
Matthias Neumann-Brosig
- Bundesnetzagentur für Elektr., Gas, Telekommunikation, Post und Eisenbahnen (BNetzA)
Taras Holoyad
- Institut für Arbeitsschutz der Deutschen Gesetzlichen Unfallversicherung (IFA)

Paul Martin Fechtner

At present, there are no standards covering this topic in the body of German Standards.

DIN SPECS are not part of the body of German Standards.

A draft of this DIN SPEC has not been published.

Despite great efforts to ensure the accuracy, reliability and precision of technical and non-technical information, the DIN SPEC-consortium cannot give any explicit or implicit assurance or warranty in respect of the accuracy of the document. Users of this document are hereby made aware that the consortium cannot be held liable for any damage or loss. The application of this DIN SPEC does not release users from the responsibility for their own actions and is applied at their own risk.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. DIN shall not be held responsible for identifying any or all such patent rights.

Provision of this document free of charge as a PDF via the Beuth WebShop has been financed in advance.

For current information on this document, please go to DIN's website (www.din.de) and search for the document number in question.

Introduction

Rapid advancements in machine learning (ML) have led to the adoption of ML algorithms in applications in almost every industrial sector and scientific fields. ML methods are also becoming increasingly important in the development of real-world and often safety-critical systems. Robust and safe operation of ML elements in such systems is essential and can be supported by uncertainty quantification.

Uncertainty quantification is a broad and extensively researched discipline, often linked with fields such as e.g., numerical modeling and simulation. Uncertainty quantification allows to estimate the uncertainty associated with models, algorithms, and predicted results [1]. The estimation of uncertainty is also of major interest and importance in machine learning.

Due to the complexity of many ML applications, training data (often) cannot represent the entire operational design domain, and scenarios will occur that were insufficiently covered during the development of the ML model. High quality uncertainty estimates can support better runtime monitoring. Uncertainty quantification is, therefore, one of the building blocks of general safety of artificial intelligence. Apart from that, uncertainty quantification is also highly relevant to other aspects of ML such as transfer learning, active learning, reinforcement learning, or data fusion [2].

Uncertainty quantification in ML (and especially deep learning) is already an established but still very fast-growing field of research. While scientific advancements have created a solid base for broader adoption of uncertainty quantification in ML, there is a lack of standardization work that reflects the state of the art and helps stakeholders to apply uncertainty quantification in practice. To address this issue, this document defines essential terminology (Clause 3), introduces important aspects of uncertainty quantification (Clause 4), describes selected applications of uncertainty quantification (Clause 5), provides an overview of uncertainty quantification approaches and general characteristics (Clause 6) as well as formulates requirements and guidelines for uncertainty quantification in ML (Clause 7). Lastly, the document is supplemented with informative annexes that include additional explanations (Annex A and Annex B) and presentation of use case examples (Annex C).

1 Scope

This document specifies general guidelines and requirements for the development and use of methods for the quantification of uncertainties in machine learning (ML).

This document defines fundamental terminology for uncertainty quantification in ML and specifies the purpose, use, and necessity of these quantifications.

This document provides an overview of available uncertainty quantification approaches in ML and their characteristics, and describes selected applications.

This document specifies general technical requirements and recommendations for the uncertainty quantification in ML.

This document is applicable to research, development, and the usage of ML techniques including but not limited to medicine, earth observation, natural language processing (NLP), finance, computer vision, autonomous mobility and machinery.

This document does not specify requirements for uncertainty criteria necessary for safety functions and health protection, such as for the detection of persons in a hazardous area. This also applies to so-called assistance systems.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 22989:2022¹, *Information technology — Artificial intelligence — Artificial intelligence concepts and terminology*

DIN ISO 3534-1:2009-10, *Statistics — Vocabulary and symbols — Part 1: General statistical terms and terms used in probability (ISO 3534-1:2006); Text in German and English*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in DIN ISO 3534-1:2009-10, ISO/IEC 22989:2022, and the following apply.

ISO and IEC maintain terminological 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/>

DIN and DKE maintain terminological databases for use in standardization at the following addresses:

- DIN-TERMinologieportal: available at <https://www.din.de/go/din-term>
- DKE-IEV: available at <https://www.dke.de/DKE-IEV>

1 German version is under preparation. Stage at the time of publication: E DIN EN ISO/IEC 22989:2023-04.

3.1

AI system

artificial intelligence 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 to develop a model to represent data, knowledge, processes, etc. which can be used to conduct tasks.

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

[SOURCE: ISO/IEC 22989:2022, 3.1.4]

3.2

algorithmic transparency

<model> property of the model which enables the human to fully explore the model process by means of mathematical analysis and methods

3.3

aleatoric uncertainty

uncertainty (3.27) caused by stochasticity in observation stemming from the data generation process or data acquisition process or due to the lack of *measures* (3.14)

Note 1 to entry: There exist a grey area where the distinction between aleatoric uncertainty and epistemic uncertainty is ambiguous (see Annex A).

3.4

certainty

<ML-model-based information> belief that a statement based on an ML model output is correct

Note 1 to entry: The statement refers to the model output itself or statement created using the output.

3.5

confidence

statistical confidence

level of trust in the statistics or certainty

3.6

confidence level

value $(1 - \alpha)$ of the probability associated with a confidence interval or a statistical coverage interval

[SOURCE: ISO/IEC Guide 98-3:2008, C.2.29, modified — Note to entry was deleted]

3.7

decomposability

<model> property of the model which enables the human to comprehend each of the model parts without the need of additional tools

3.8

dependability

<of an item> ability to perform as and when required

Note 1 to entry: Dependability includes availability, reliability, recoverability, maintainability, and maintenance support performance, and, in some cases, other characteristics such as durability, safety and security.

Note 2 to entry: Dependability is used as a collective term for the time-related quality characteristics of an item.

[SOURCE: IEC, IEV ref 192-01-22]

3.9

epistemic uncertainty

uncertainty (3.27) caused by the lack of knowledge about the best model parameters due to insufficient amount of observations or improper training that can be decreased by enhancing the training process and/or collecting additional observations for the given *measures* (3.14)

Note 1 to entry: There exist a grey area where the distinction between aleatoric uncertainty and epistemic uncertainty is ambiguous (see Annex A).

3.10

fusion

deriving information by processing data from various sources

Note 1 to entry: Such data can be sensed data, aggregation of sensor data, or other processed data.

[SOURCE: ISO/IEC 29182-2:2013, 2.4.4, modified — “may” in Note 1 to entry was replaced by “can”]

3.11

input-quality uncertainty

uncertainty (3.27) caused by input-quality limitations during application

3.12

interpretability

<AI> level of understanding how the underlying (AI) technology works

[SOURCE: ISO/IEC TR 29119-11:2020, 3.1.42]

3.13

machine learning

ML

process of optimizing model parameters through computational techniques, such that the model's behaviour reflects the data or experience

[SOURCE: ISO/IEC 22989:2022, 3.3.5]

3.14

measure, noun

variable to which a value is assigned as the result of measurement

Note 1 to entry: The plural form “measures” is used to refer collectively to base measures, derived measures and indicators.

[SOURCE: ISO/IEC/IEEE 15939:2017, 3.15]

3.15

model-fit uncertainty

uncertainty (3.27) caused by (inherent) limitations of the trained ML model

Note 1 to entry: Limitations can be related to adaptability of feature characteristics or the variability of a model's feature characteristics.

3.16

operational design domain

ODD

target application scope

TAS

intended operational conditions

intended operational domain

operating conditions under which a given system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions

Note 1 to entry: The term is strongly related to and can sometimes be interchangeable to terms that are used outside the automotive context for similar purposes (e.g., intended use/purpose [ISO 14971:2019, 3.6] or target application scope defined as “common conditions under which the [ML model] is intended to be applied” [4])

[SOURCE: SAE J 3016:2021, modified — automotive-related terms “driving automation” and “, and/or the requisite or absence of certain traffic or roadway characteristics” removed]

3.17

out-of-distribution data

data from a distribution different from the distribution of data used during model development (including training, validation, calibration or testing data set)

Note 1 to entry: Often, the out-of-distribution data is encountered outside of the *operational design domain* (3.16).

3.18

probability distribution

distribution

probability measure (3.19) induced by a *random variable* (3.20)

Note 1 to entry: There are numerous, equivalent mathematical representations of a distribution including distribution function, probability density function, if it exists, and characteristic function. With varying levels of difficulty, these representations allow for determining the probability with which a random variable takes values in a given region.

Note 2 to entry: Since a random variable is a function on subsets of the sample space to the real line, it is the case, for example, that the probability that a random variable takes on any real value is 1. In many situations, it is much easier to deal directly with the random variable and one of its representations than to be concerned with the underlying probability measure. However, in converting from one representation to another, the probability measure ensures the consistency.

Note 3 to entry: A random variable with a single component is called a one-dimensional or univariate probability distribution. If a random variable has two components, one speaks about a two-dimensional or bivariate probability distribution, and with more than two components, the random variable has a multidimensional or multivariate probability distribution.

[SOURCE: DIN ISO 3534-1:2009-10, 2.11, modified — EXAMPLE was removed and “For the battery example, $P[X \geq 0] = 1$.” was deleted from Note 2 to entry.]

3.19

probability measure

$\wp$

non-negative function defined on the sigma algebra of events such that

a) $\wp(\Omega) = 1$, where Ω denotes the sample space

b) $\wp(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \wp(A_i)$ where $\{A_i\}$ is a sequence of pair-wise disjoint events

Note 1 to entry: A probability measure assigns a value from $[0, 1]$ for each event in the sigma algebra of events, with the value 1 representing certainty of occurrence. In particular, the probability measure associated with the null set is zero and the probability measure assigned to the sample space is 1.

Note 2 to entry: Property b) indicates that if a sequence of events has no elements in common when considered in pairs, then the probability measure of the union is the sum of the individual probability measures. As further indicated in property b), this holds if the number of events is countably infinite.

Note 3 to entry: The three components of the probability are effectively linked via random variables. The probabilities of the events in the image set of the *random variable* (3.20) derive from the probabilities of events in the sample space. An event in the image set of the random variable is assigned the probability of the event in the sample space that is mapped onto it by the random variable.

Note 4 to entry: The image set of the random variable is the set of real numbers or the set of ordered k -tuples of real numbers. (Note that the image set is the set onto which the random variable maps.)

[SOURCE: DIN ISO 3534-1:2009-10, 2.70, modified — EXAMPLE and the sentence “The value 0 corresponds to an event being impossible, while” from Note 1 to entry were deleted and replaced by “, with”. “ n -tuplets” was changed to “ k -tuples” in Note 4 to entry.]

3.20

random variable

function defined on a sample space where the values of the function are ordered k -tuples of real numbers

Note 1 to entry: An example of an ordered k -tuple is $(x_1, x_2, \dots, x_k)$. An ordered k -tuple is, in other words, a vector in k dimensions (either a row or column vector).

Note 2 to entry: Typically, the random variable has dimension denoted by k . If $k = 1$, the random variable is said to be one-dimensional or univariate. For $k > 1$, the random variable is said to be multi-dimensional. In practice, when the dimension is a given number, k , the random variable is said to be k -dimensional.

Note 3 to entry: A one-dimensional random variable is a real-valued function defined on the sample space that is part of a probability space.

Note 4 to entry: A random variable with real values given as ordered pairs is said to be two-dimensional. The definition extends the ordered pair concept to ordered k -tuples.

Note 5 to entry: The j^{th} component of a k -dimensional random variable is the random variable corresponding to only the j^{th} component of the k -tuple. The j^{th} component of a k -dimensional random variable corresponds to a probability space where events are determined only in terms of values of the component considered.

[SOURCE: DIN ISO 3534-1:2009-10, 2.10, modified — EXAMPLE was deleted and “ k -tuple” was replaced by “ k -tuple” in the definition.]

3.21

reliability

degree to which a system, product or component performs specified functions under specified conditions for a specified period of time

[SOURCE: ISO/IEC 25010:2011, 4.2.5, modified — Notes to entry were deleted]

3.22

scope-compliance uncertainty

uncertainty (3.27) caused by the possibility that the ML model provides an output although it is applied outside its *operational design domain* (3.16)

3.23

simulatability

<model> property of the model which enables the human to fully simulate the model and recreate the model's outputs without the need of additional tools

3.24

system

set of interrelated or interacting elements

[SOURCE: DIN EN ISO 9000:2015-11, 3.5.1]

3.25

transparency

<model> degree to which interpretability is supported through *algorithmic transparency* (3.2), *decomposability* (3.7), and *simulatability* (3.23) of the model

Note 1 to entry: This definition is based on the domains of interpretability introduced by [3].

Note 2 to entry: The opposite of model transparency is opacity. A black box model is opaque since its internals are unknown, yet a white box model is not necessarily transparent.

EXAMPLE 1 A white box decision tree model based on semantic (i.e., human-interpretable) features and a limited depth of e.g. four decision levels can be considered as a highly transparent model since it is

- a) algorithmic transparent, since the underlying concepts as binary decision making using information gain as a criteria is mathematically well understood,
- b) decomposable, since the inputs and variables and thresholds used in each node of the tree are human comprehensible,
- c) simulatable, since humans can follow up the complete decision making process and calculate the model results for a given input by themselves by applying at most four comparisons.

EXAMPLE 2 A white-box deep neural network with 10 million weights using the machine-encoded internal log messages with unknown meaning as features can be considered as a largely intransparent model since it

- a) has limited algorithmic transparency using approaches as stochastic gradient descent to determine the weights for its calculations,
- b) limited decomposability, since the meaning of the model's features is not semantically clear,
- c) and is not simulatable due to the millions of individual calculations and weights that need to be considered.

3.26

uncertainty

<information> state, even partial, of deficiency of information related to, understanding or knowledge of, an event, its consequence, or likelihood

Note 1 to entry: This uncertainty definition is derived from ISO Guide 73:2009, definition 1.1 on risk, Note 5.

Note 2 to entry: Uncertainty in the context of information is too broad to adequately address uncertainty in the scope of ML models and needs further refinement in the context of ML model-based information (see 3.27).

3.27

uncertainty

<ML model-based information> lack of *certainty* (3.4)

Note 1 to entry: Uncertainty in a statement can be caused by both

- i) information that indicates that the statement can be wrong, and
- ii) insufficient information that can justify its correctness.

Note 2 to entry: The statement based on the output of the ML model commonly refers to a failure mode of interest, e.g., “the relative deviation between the actual and model-predicted temperature is less than 1 %” or “an airplane that is visible on the input image is detected by the ML model”.

Note 3 to entry: In this document, unless stated otherwise, uncertainty refers to the total uncertainty viewed as combination of

- i) *aleatoric uncertainty* (3.3) and *epistemic uncertainty* (3.9), or
- ii) combination of uncertainty stemming from *model-fit uncertainty* (3.15), *input-quality uncertainty* (3.11) and *scope-compliance uncertainty* (3.22).

3.28

uncertainty quantification

act or result of measuring or judging degree of *uncertainty* (3.27)

4 Uncertainties in machine learning

4.1 The use of “uncertainty” in existing AI standardization

The foundational standard ISO/IEC 22989 refers to uncertainty in the definition of risk. ISO/IEC TR 24028 points out that uncertainty has to be dealt with when performing functional tests on AI systems. ISO/IEC TR 24029-1 discusses the use of uncertainty analysis for neural network models. ISO/IEC 23053 covers evaluation metrics for testing ML models. These metrics can be used to assess the performance and conformity of ML models. In more detail, metrics and methods for the performance assessment of ML models are described in ISO/IEC TS 4213.

To a certain extent, performance metrics for AI systems can express model-fit uncertainties (see 4.4) of an AI system, though they are incapable of quantifying *uncertainty* (3.27) entirely [5]. Usually, error rates or accuracy scores are specified for statistically expressing the deviation of the actual output from the correct output or the ideally expected output. However, error rates or accuracy scores do not quantify the uncertainty associated with a single prediction during the ML model operation.

Performance metrics can be used to quantify an expected level of performance. To what extent an AI system can maintain its level of performance in operation remains *uncertain* (3.26). The ability of an AI system to maintain its level of performance under any circumstances is the definition of “robustness” according to ISO/IEC 22989:2022. ISO/IEC TR 24029-1 provides background information about existing methods for robustness assessment of neural networks and introduces the term “uncertainty” in the context of interpolation stability in testing using formal methods. In ISO/IEC 24029-2, methodology for the use of formal methods to assess robustness of neural networks is introduced. ISO/IEC TR 24028 addresses methods and good practices related to development of neural network models which relates to the mitigation of epistemic uncertainties.

4.2 Elements of an ML system

ISO/IEC 22989 defines ML as the process of optimising model parameters through computational techniques, such that the model’s behaviour reflects the data or experience. Development of computer programs that can learn from data, as a field of ML, has been expanding with wide variety of emerging applications (ISO/IEC 23053). Availability of large amount of data and computation resources have enabled the progression of ML in various industry sectors. ISO/IEC 23053:2022 presents an introduction to the elements of an ML system breaking it down into ML tasks, ML models, ML data, and software tools and techniques, see ISO/IEC 23053:2022, Figure 1.

An ML task is defined as actions required to achieve a specific goal of the ML system to solve a problem. The problem is solved by utilizing a trained ML model on production data. Some examples of ML tasks are regression, classification, anomaly detection, dimensionality reduction, etc.

An ML model is comprised of a data structure and software to process the data structure. An ML model after being trained implements the target function and produces outputs that approximately represent the relevant statistical properties of the training data. ML algorithms utilise statistical optimisation methods (techniques) to perform this representation. The patterns learnt from the training data are reflected by the resultant mapping from the model inputs to the model outputs.

ML data comprises four mutually exclusive categories of datasets. The training dataset is used to train and estimate the ML model parameters. The validation dataset is used to select the best model according to a performance criterion. The test dataset is used to check the generalization capability of the model and evaluates the performance of the ML model on new data, and the production data is used during operation of the trained ML model to produce model prediction via inference.

ISO/IEC 23053:2022 summarizes ML algorithms, tools for data preparation, optimization methods, and ML evaluation metrics as software tools and techniques. ML algorithms define the computational structure of the ML model and the approach to be taken for training. They can be classified into neural networks, decision trees, support vector machines, etc. Tools for data preparation include, but are not limited to automated tools for preprocessing, data wrangling, filtering, cleaning, normalization of ML data. Optimization methods include methods used to fit an ML model to training data; examples are gradient descent methods and least square curve fitting. ML evaluation metrics are used to adequately express and examine the model performance for the model's specific use case. Some examples include precision, recall, f1 score, confusion matrix.