*Security and Safety*, Vol. 3, 2024020 (2024)
https://doi.org/10.1051/sands/2024020
sands.edpsciences.org

Security and Safety in Artificial Intelligence

Research Article

OPEN ACCESS

Other Fields

# A requirements model for AI algorithms in functional safety-critical systems with an explainable self-enforcing network from a developer perspective

Christina Klüver[1,*], Anneliesa Greisbach[1], Michael Kindermann[2], and Bernd Püttmann[3]

[1] *CoBASC Research Group, Essen 45130, Germany*
[2] *Pepperl+Fuchs Group, Mannheim 68307, Germany*
[3] *TÜV Nord Group, Essen 45307, Germany*

Received: 30 August 2024 / Revised: 30 October 2024 / Accepted: 30 October 2024 / Published online: 31 October 2024

**Abstract** The requirements for ensuring functional safety have always been very high. Modern safety-related systems are becoming increasingly complex, making also the safety integrity assessment more complex and time-consuming. This trend is further intensified by the fact that AI-based algorithms are finding their way into safety-related systems or will do so in the future. However, existing and expected standards and regulations for the use of AI methods pose significant challenges for the development of embedded AI software in functional safety-related systems. The consideration of essential requirements from various perspectives necessitates an intensive examination of the subject matter, especially as different standards have to be taken into account depending on the final application. There are also different targets for the "safe behavior" of a system depending on the target application. While stopping all movements of a machine in industrial production plants is likely to be considered a "safe state", the same condition might not be considered as safe in flying aircraft, driving cars or medicine equipment like heart pacemaker. This overall complexity is operationalized in our approach in such a way that it is straightforward to monitor conformity with the requirements. To support safety integrity assessments and reduce the required effort, a Self-Enforcing Network (SEN) model is presented in which developers or safety experts can indicate the degree of fulfillment of certain requirements with possible impact on the safety integrity of a safety-related system. The result evaluated by the SEN model indicates the achievable safety integrity level of the assessed system, which is additionally provided by an explanatory component.

**Keywords** Functional safety, Safety-critical systems, Requirements for AI methods, Explainable self-enforcing networks (SEN)

**Citation** Klüver C, Greisbach A, Kindermann M and Püttmann B. A requirements model for AI algorithms in functional safety-critical systems with an explainable self-enforcing network from a developer perspective. Security and Safety 2024; **3**: 2024020. https://doi.org/10.1051/sands/2024020

## 1 Introduction

Artificial Intelligence (AI) technologies are becoming increasingly important in technical applications. It is worth noting that with the support of systems like ChatGPT, even non-programmers can implement and use complex AI methods without having to be proficient with the methods. At a time when AI

* Corresponding author (email: cobasc@rebask.de)

methods are no longer a niche in academia and industry, but have reached the general population, standards and regulations are being discussed worldwide. Examples of fundamental AI standards from the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) include ISO/IEC 22989:2022 (AI concepts and terminology), ISO/IEC 23053:2022 (Framework for Machine Learning (ML) based AI systems), ISO/IEC TR 24372:2021 (Overview of computational approaches to AI), and ISO/IEC 23894:2023 (Information technology – Artificial intelligence – Guidance on risk management).

By summarizing various methods under the umbrella of AI, it is necessary to specify exactly which method is used, given the fact that not every method meets the requirements in safety-critical systems (SCS). This applies in particular to the reliability, transparency, and accountability of methods based on Neural Networks (NN), such as Deep Learning (DL). In addition, as soon as safety is affected, specific regulations must be followed, especially in functional safety-related systems (ISO/IEC TR 5469:2024), or in SCS respectively, which are meanwhile enshrined in law, *e.g.*, through the European AI ACT [1–3].

This leads to increased overall complexity as diverse requirements need to be met for different applications. Even if the same AI method is used, it must be considered that the standards in the area of *e.g.*, aerospace are different from those for automated driving or mining. In addition, explaining the results or decisions made by an AI method must ensure that it meets the needs of all involved stakeholders.

These are just some of the aspects to consider, *e.g.*, the European "Machinery Regulation 2023/1230", which will come into force in January 2027, explicitly includes "software that performs a safety function" as a "safety component" [4, p. L 165/3 (19)]. This creates further challenges for any developer, which are not fully foreseeable at this stage [5, 6].

This dynamic has implications at different levels: Standards are being adopted in all areas and in some cases still need to be specified, auditors of safety-critical systems need clear guidance, and companies need to keep track in order to meet the (still unclear) requirements in time.

A model that contains the essential requirements and allows the degree of their fulfillment to be assessed can assist both software developers and, ideally, evaluators. We are confronted with these challenges from different perspectives as developers, assessors, and contributors to the development of regulations, and in order to address them, we have developed a basic model of the requirements for using AI methods in SCS, which can be easily adapted and extended to other areas.

Methodologically, this means taking a close look at the standards and requirements, filtering out the essential ones for a specific use case, and assessing their significance for the respective functional safety or security levels. Since the level of safety to be ensured varies, an "optimal" degree of fulfillment is defined for each level; these serve as "reference types" for all new inputs. As this is not a data-driven model, these assessments are based on intensive literature research, our own expertise, and discussions.

The technical method used is a self-organized learning neural network that learns to map a wide range of requirements to a corresponding safety level – the basic model. Such an approach only makes sense if the evaluation performed by experts is also reproducible in the technology used. Therefore, we use a Self-Enforcing Network (SEN) with this property.

The SEN model also incorporates explainability through Shapley values and provides transparency into how the system satisfies safety criteria. This approach provides a novel solution to the complexity of verifying AI-based safety-critical systems and ensures that safety requirements are met. By exploring recent developments in AI safety frameworks and the role of explainability in safety-critical applications, this paper contributes to the development of reliable, compliant AI systems in areas where safety is of high concern.

To concretize this approach, the use of AI methods in functionally safety-critical systems related to machinery is examined in more detail. The focus is on methods based on Neural Networks, which are considered particularly critical for use in SCS.

As mentioned before, the development of the model requires an intensive exploration of important aspects of fundamental standards in functional safety-critical systems, which are considered below. The resulting requirements for AI methods lead to recent developments of quality models and frameworks for risk mitigation and safety improvement, which are presented in Section 3, along with important concepts of explainability and the derived "knowledge model" for SEN. In Section 4 the Self-Enforcing Network is described in more detail. The concept of explainability based on Shapley Values (SV) is also introduced, which enriches SEN to an *intrinsically explainable* network. The presentation of the results of SEN using the implemented and learned "knowledge" follows in Section 5. In selected examples, the degree

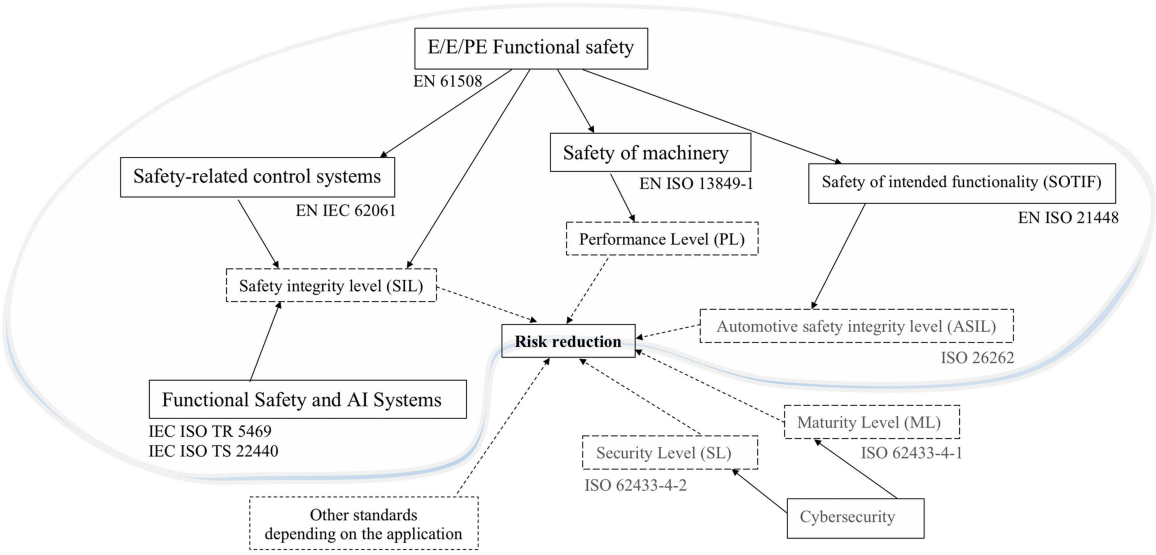


**Figure 1.** Overview of the standards. The circled standards are of particular importance for the model

of fulfillment of the requirements is entered from the developer's perspective. The SEN indicates which Safety Integrity Level (SIL) can be achieved depending on the used (AI) method and the architecture of the overall system. The explanation component visually displays selected requirements and their degree of fulfillment in order to achieve the highest safety integrity level.

## 2 Norms and requirements for safety functions and AI-methods

As a starting point for selecting a set of requirements for the use of AI methods in SCS, we focus on the standards EN 61508:2010, EN ISO 13849-1:2023 and EN IEC 62061:2021.

Requirements for the evaluation of functional safety can be derived from the basic safety publication EN 61508. This standard deals with the functional safety of electric, electronic, or programmable electronic systems for use in safety-related systems. The standard describes how to qualify safety functions that are used for risk reduction in applications displaying hazards that could result in harm to persons, property, or the environment. The term safety integrity is introduced to categorize the dangerous failure rate of a safety function. The safety integrity level (SIL) is divided into four levels based on the random dangerous residual failure rates for safety-related systems as one parameter. It specifies the corresponding numerical ranges of the target failure rate [7]. In addition, the achievable SIL level also relies on *e.g.*, the measures taken to exclude systematic failures.

Other important standards include EN ISO 13849-1, which provides a more simplified approach to qualify safety-related parts of machine control systems using any technology or any combination of different technologies (*e.g.*, electrical/ electronic, hydraulic, pneumatic, mechanical) and uses Performance Levels (PL) to classify the safety integrity level. EN IEC 62061 is another functional safety standard specifying requirements for safety-related control systems for machinery also using the safety integrity level (SIL) to classify the safety integrity.

Besides the "accidental faults" covered by requirements specified in functional safety standards there are also "intentional faults" that could have a negative influence on the achievable safety integrity. Protecting systems against "intentional faults" is addressed by requirements in cybersecurity standards like the IEC 62443 series.

An overview of the standards discussed here, and the method used to assess risk is shown in Figure 1. The circled elements are explained in more detail below and are also part of the SEN model. The area covered refers to all functional safety standards – cybersecurity and other risk mitigation standards are not covered in detail.

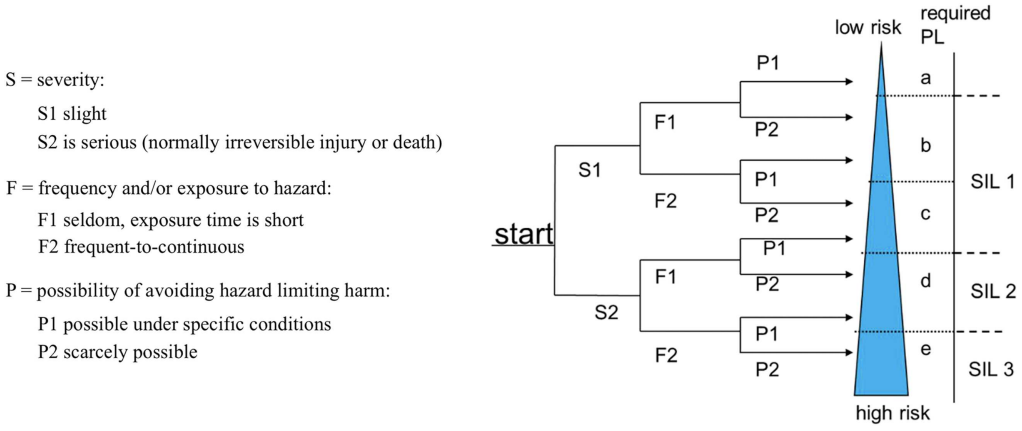


**Figure 2.** Tree for determining the risk reduction and relation between PL and SIL ([8], p. 14)

## 2.1 Determination of required safety level

The relationship between safety and risk of failure can be derived using a risk graph. Risk is the combination of (estimated) probability and severity of occurrence of a hazardous event. Safety is the absence of unacceptable risk, but there is no absolute safety, so safety has a probabilistic characteristic.

The standards EN 61508, EN ISO 13849 1, and EN IEC 62061 use different designations, but have similar parameters for determining the necessary contribution to risk reduction by the individual safety levels.

EN ISO 13849-1 defines three factors to be evaluated for the risk assessment: severity of injury (S), frequency and/or exposure times to hazard (F), and the possibility of avoiding or limiting harm (P). The resulting necessary Performance Level (PL) for the safety function is derived from the available performance levels PL a to PL e, with PL e being the most stringent. The final PL is determined following the tree (as shown in Figure 2).

EN IEC 62061 defines the following four factors to evaluate the risk value: The severity of injury (Se), the frequency and duration of exposure to the hazard (Fr), the likelihood of a hazardous event occurring (Pr), and the possibility to avoid or reduce damage (Av). EN 61508 specifies the factors consequence of the dangerous event (C), the frequency and exposure time (F), the possibility of avoiding the dangerous event (P), and the probability of the unwanted occurrence (W) to evaluate the risk value.

The risk assessment adds the factor severity to the likelihood of a hazardous event. The SIL requirements for the safety function of the control system are derived from the calculated result. While process industry applications only require safety functions when the usual process controls fail, machines need the safety function continuously as risk is imminent during the time of use. The standard EN IEC 62061 is limited to SIL 1 to SIL 3 as machine hazards never cause catastrophic events.

The necessary risk reduction for an application can be defined via PL from EN ISO 13849 or SIL from EN IEC 62061 or IEC 61508 standards [9]. Figure 2 shows both Performance Level (PL) and Safety Integrity Level (SIL). These measures can be related to each other.

The risk assessment procedure and the risk assessment requirements are also adapted depending on the application of the safety function used in. Examples of these adaptions are EN 50129 for safety-related systems in railways, IEC 61511 for the process industry, and DIN EN IEC 60601 for medical devices.

## 2.2 Safety integrity levels for Artificial Intelligence systems

Safety Integrity Levels for Artificial Intelligence (AI-SIL) are currently proposed by [10], using a Level of Rigour" (LoR). An overview of the proposed AI-SIL assessment model is shown in Figure 3.

The assessment method considers the complexity of the input (entropy) and the complexity of the output (non-determinism). To determine the AI SIL, these factors are combined with the assigned SIL. It is important to perform a "functional decomposition" to divide the overall function into conventional (non-AI-based) components and AI-based components.

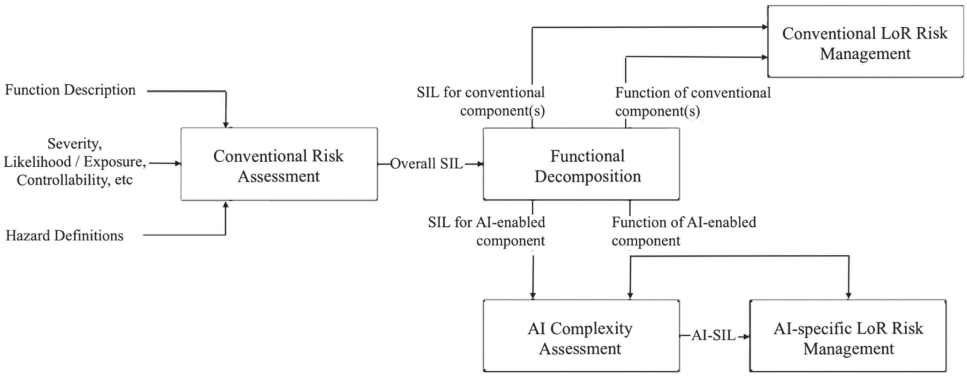


**Figure 3.** AI-SIL assessment method (reprinted with permission from [10], p. 400).

A more general framework for AI Safety Levels (ASL), as a type of Safety Integrity Level (SIL) [11], is proposed *e.g.*, by the Future of Life Institute [12] to classify the levels of capability of AI systems, which correspond to SIL levels.

The assessment method considers the complexity of the input (entropy) and the complexity of the output (non-determinism). To determine the AI SIL, these factors are combined with the assigned SIL. It is important to perform a "functional decomposition" to divide the overall function into conventional (non-AI-based) components and AI-based components.

A more general framework for AI Safety Levels (ASL), as a type of Safety Integrity Level (SIL) [11], is proposed *e.g.*, by the Future of Life Institute [12] to classify the levels of capability of AI systems, which correspond to SIL levels.

## 2.3 Functional safety architectures

To evaluate systems according to functional safety requirements, the standards EN 61508, EN ISO 13849 or EN IEC 62061 provide sets of rules that specify generally applicable requirements for the design of safety functions including the architecture.

A safety-related overall system usually consists of several subsystems. The following block diagram in Figure 4, taken from EN ISO 13849-1, shows a "typical structure" of safety systems or subsystems.

The system can be divided into input (I), logic (L), and output (O). To describe the safety function, the input and output interface must be specified and the behavior at the output must be described depending on the conditions at the input. Particular attention must be paid to the "safe state" at the input and output. The output of this safety system must, with a certain degree of reliability, bring a system to a state where a hazard is no longer given – the safe state.

The behavior of the safety-related system (SRS) in the event of malfunctions of the subsystems or subcomponents belonging to the overall system must also be considered. The possible faults must be divided into "dangerous faults" and "safe faults" and limiting the rate of "dangerous faults" to an "acceptable level" is the aim of functional safety requirements. To this end, hardware reliability requirements have been specified in functional safety standards. In the circuit structure shown in Figure 4, a dangerous fault in a subsystem might lead to dangerous behavior in the whole system.

Figure 5 shows extended circuit architectures. In addition to the "functional channel" a "second channel" (on the left) consists of a "Test Equipment" (TE) and an output to establish the "safe state" (Output Test Equipment (OTE)). If the "test equipment" detects something unexpected in the behavior of the "functional channel", it switches the system to a "safe state" via the OTE. According to the functional safety requirements, it is important that this second channel is as independent as possible from the functional channel to ensure that the "second channel" can bring the system to a "safe state" under all possible conditions in the functional channel.

This circuit concept increases the safety integrity of the system but still contains an area in the input section of the functional channel, where faults can lead to dangerous system behavior. A further increase in the safety integrity of a system can be achieved by using a system with a fully redundant safety path

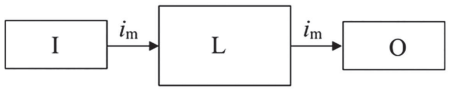


**Figure 4.** 1-channel safety system with $i_m$ = interconnecting means, and $m$ = monitoring/testing

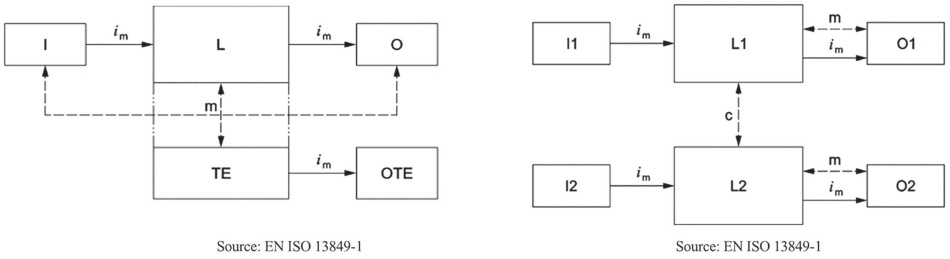


**Figure 5.** Left: 1-channel safety system with additional second shutdown path with $i_m$ = interconnecting means, and $m$ = monitoring/testing; on the right 2-channel safety system, with $c$ = cross monitoring

[13, 14], as shown in Figure 5 on the right. Here there are two sub-circuits, each of which can execute the safety function independently of the other sub-circuit. This means that random hardware faults in one signal path cannot cause the system to "fail" as the second path will still work.

The functional safety standards distinguish between random hardware faults and systematic faults. While the architecture improvements shown in Figure 5 reduce the dangerous failure rate caused by random faults, they do not reduce the probability of systematic system faults. Root causes for systematic faults are *e.g.*, "conceptual weaknesses" in the hardware or software design and can also be triggered by "environmental conditions", such as electromagnetic radiation in the system environment.

If both channels of a redundant safety system are designed identically, there is a greater probability that a systematic fault will cause the same undesired behavior in both channels under the same conditions. As a result, such systematic faults generally cannot be safely controlled by identical redundant circuits.

A method to combat systematic errors is designing redundant systems based on the diversity principle. Here, the individual channels or parts of the channels are set up "differently". Achieving this involves using various circuit technologies, concepts in hardware circuitry or software architecture, involving different developers for individual channels, and if necessary for system support in the final application after development completion. Even if AI-based algorithms are not permitted for functional safety circuits according to traditional functional safety requirements, they could reduce the probability of dangerous errors through diversity in circuit design.

### 2.4 Standardization in functional safety systems and AI-Methods

Standardization in the AI field of technology poses several challenges. This is certainly due to the fact that the term AI covers a large number of technologies and methods, each of which has fundamentally different properties and possible evaluation criteria.

There is currently also a gap between the current state of AI standardization at the international level and the desired state of EU-wide harmonized standards required for compliance with the AI Act. In [15], the AI standardization activities of ISO/IEC JTC 1/SC 42 for computational approaches are mapped to the high-risk AI requirements in the AI Act for an overview. The progress of this standard will also have an impact on the European Aviation Safety Agency (EASA), *e.g.*, which is considering introducing three ML application levels in aerospace [16].

On the German side, the standardization working group AK 914.0.11 of the German Commission for Electrical, Electronic & Information Technologies (DKE) of DIN and VDE [17] has developed a set of technical regulations that approach this topic from a different angle. In principle, three main issues are to be addressed: the selection of the AI method or technique to be used for the specific application; the collection and selection of data on which to train the system; the method of decision making based on the AI method and training data.

This distinction seems appropriate, as separate rule sets could be created for each of these topics to prove the suitability of the AI system. These are not specified here; rather, it is pointed out that the

establishment of certain properties is necessary, but the way to achieve this is still to be worked out. In particular, with respect to methods and technology, it is likely that a restriction to certain methods or technical implementations is advantageous, such as the development of a set of rules for special neural networks.

Another important issue is the classification of usage into "usage levels" [18, 19]. AI is often not used directly in a safety function, but rather as a diagnostic function. Sometimes a conventional security function is implemented in parallel, sometimes AI is used only as a tool in the development of a safety system. In all these use cases, it may be easier to justify the suitability of an AI system for the intended use with evidence.

The extent to which traditional proof of suitability is possible, conditionally possible, or impossible also plays a role. Gaps in the chain of reasoning are often to be expected due to the special properties of an AI system, but there are also ways of providing evidence that does not come from the set of verification methods already described in standards, but whose suitability for proving certain necessary properties can be regarded as equivalent.

IEC ISO TR 5469:2024, "Functional Safety and AI Systems", published in January 2024, which can be seen as a bridge between artificial intelligence and the IEC 61508 safety standard, covers all the above-mentioned possibilities for the use of AI in SRS [IEC ISO TR 5469:2024]: "use of AI inside a safety related function to realize the functionality; use of non-AI safety-related functions to ensure safety for an AI controlled equipment; use of AI systems to design and develop safety-related functions".

However, IEC 61508:2010 is interpreted such that it does not recommend the use of AI methods, even for diagnostics, solutions as part of safety-related systems are based on performing system decompositions [10] so that *e.g.*, a non-AI-based deterministic monitor adopts its safety requirements together with those of the AI-based component it is monitoring [20]. This will change in the next edition as there is a high interest in using these methods in safety-relevant applications or at least giving hints on establishing suitable rulesets to qualify such systems for use in safety applications.

Interim conclusion: the functional safety standards mentioned are basic standards that generally describe how safety functions for risk reduction in SCS should be designed. The methods and techniques given in these standards and the required properties to achieve cannot directly be transferred to AI-based SCS, but it is still advantageous to assess these details to find equally appropriate methods and techniques to achieve appropriate properties where this is possible. The IEC technical report IEC ISO TR 5469:2024 highlights where it would be possible to use methods and techniques from conventional standards for evaluation of safety functions directly, while revealing other areas where new solutions must be found. The TR is based on creative dialogue between safety and AI experts to develop new and equally acceptable requirements and evaluation criteria.

The following is a brief description of challenges and some of the most promising concepts for evaluating AI systems for use in safety-critical systems.

## 3 Challenges of using and implementing AI in functional safety-critical systems

Compared to traditional software development approaches, the process of creating AI-based software is different and more demanding because existing techniques and tools, such as those used for testing according to the standards required for safety-critical systems (SCS), are not directly applicable [21]. In recent years, investments have been made in the development of general frameworks and tools, depending on the AI methods, in order to meet these requirements. Due to the large number of publications, we focus on recent developments that affect SCS in the selected examples.

### 3.1 Investigations in quality models and frameworks for AI-Systems

Quality models for AI products, such as ISO/IEC 25059:2023 (will be replaced by ISO/IEC AWI 25059 Software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Quality model for AI systems) need to be extended, *e.g.*, to meet the requirements of the AI Act. In [22], a corresponding extension is proposed (Figure 6).

As shown in Figure 6, the requirements for the quality of a product increase as soon as AI methods are used, with a fundamental focus on functional adaptability, robustness, control, and transparency. To

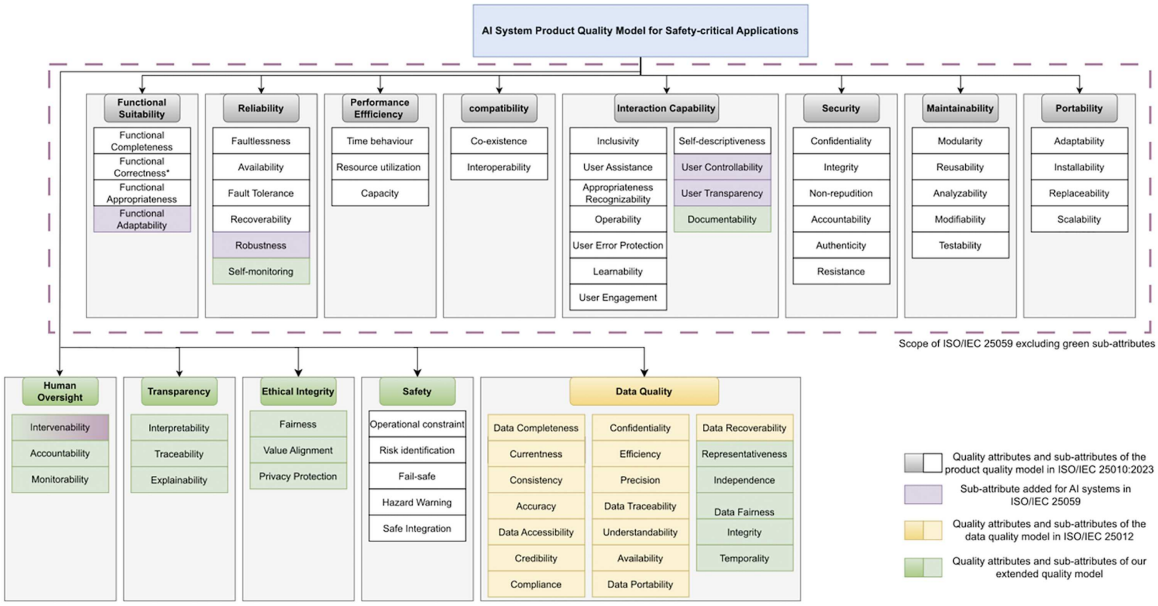


**Figure 6.** Extended Quality Model for AI products for safety-critical applications (reprinted with permission from [22], p. 982).

meet the requirements, for example, libraries for testing deep learning methods [23, 24] and trustworthy ML-based [25], or assurance case-centered methods [23] are proposed for systems engineering. In addition, data quality plays a particularly important role in *e.g.*, functional safety-critical machinery systems. When using AI methods, special criteria must be considered, such as noise or failures in sensor data, which are essential for the correct detection of problems in machines [26].

All of these efforts are aimed at minimizing risk and ensuring the safety of AI models. Various frameworks are intended to provide support:

The "AI TRiSM" tool assists organizations in mitigating AI risks, meeting privacy, security, and ethical requirements [27].

S.A.F.E. [28], an integrated AI risk management model, is proposed to meet the requirements of Sustainability, Accuracy, Fairness and Explainability, employs metrics and Key Intelligence Risk Indicators (KAIRI) to measure and monitor risk and trustworthiness over time.

HARA (hazard and risk analysis), a known standard framework, defines how to assess and evaluate the risk posed by electronics (in road vehicles). In particular, it addresses the risk of failure of electrical, electronic and software systems, an important step in meeting the requirements of ISO 26262 [29].

The "Box-Jenkins framework for safe AI" recognizes the Machine Learning (ML) model, estimates the specific parameters, and validates the model at different levels [30].

A "MLOps" process (mixed ML application development and operation) supports the continuous development and safety assurance of ML-based systems in the railway sector [31]. In this process, the system technology, the safety assurance, and the ML life cycle are defined in a workflow.

"SAFEXPLAIN" [32] provides traceability for DL methods, safety patterns to satisfy criticality and fault tolerance requirements, library implementations according to safety requirements, and computer platform configurations to handle non-determinism in Critical Autonomous AI-based Systems (CAIS). Such tools are particularly important for licensing or certifying AI methods [33, 34] for licensing high-risk artificial intelligence].

"Guaranteed Safe AI" is a proposal for a framework for the assurance of robust and reliable AI systems [12]. The framework consists of a "world model" that provides a mathematical description of how the AI system affects the outside world, a mathematical description of what effects are acceptable in a "safety specification", and a "verifier" that should provide a proof that the AI satisfies the safety specification relative to the world model.

CHESSIoT, a model-driven environment for engineering industrial IoT systems (based on Fault Tree Analysis (FTA), CHESS, and Failure Propagation Transformation Calculus (FPTC)), enables the estimation of the failure behavior of the entire system [35].

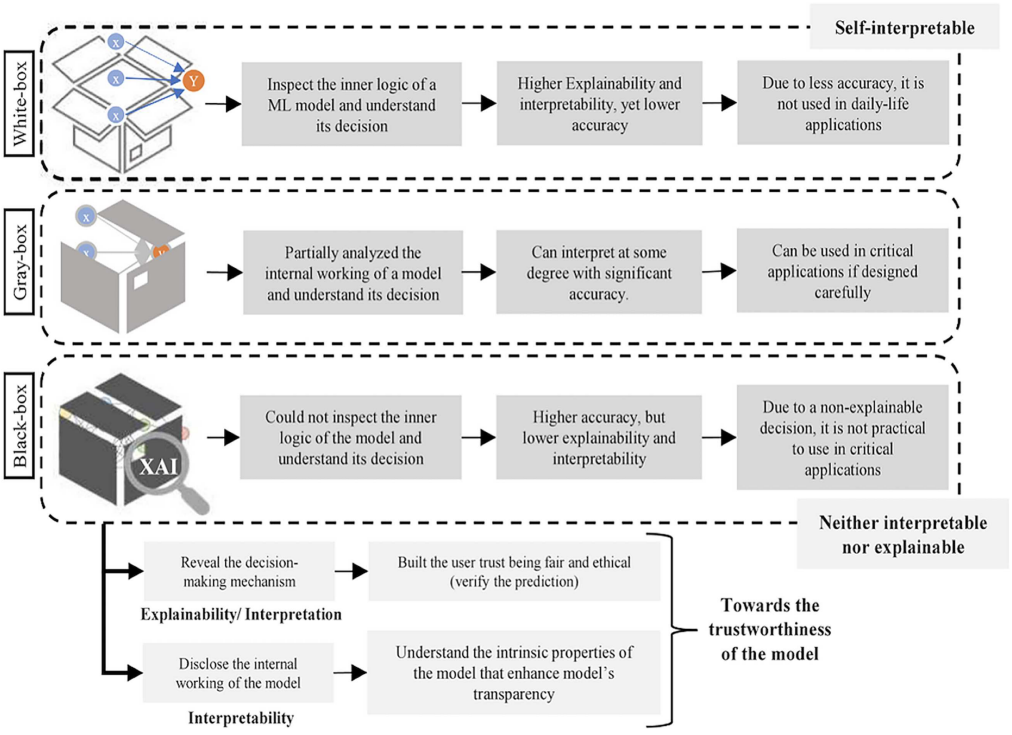


**Figure 7.** Comparison of different models in terms of their explanatory power ([45], p. 3)

The "attack intelligence framework" identifies cyber-physical attacks using various ML and DL algorithms. Explainable AI (xAI) is used to improve the traceability of the attack attribution module and to extract attack intelligence [36].

Further, actually developed frameworks should ensure trustworthiness for high-risk AI-based systems [37], consistency, reliability, explainability, and safety (CRISP) [38], safety of intended functionality (SOTIF) [19, 39], or improvement for the SCS in terms of uncertainty prediction and performance [40], to name only a few.

Explainability is crucial for the use of AI methods in the SCS, which is why the key aspects are presented in the following.

### 3.2 Explainability of AI-systems

Numerous approaches have emerged, with respect to explainability, in [41], referred to as the "Tower of Babel in Explainable Artificial Intelligence" [42–44].

As mentioned above, many methods are sublimated under AI, which can generally be divided into three classes (Figure 7) in terms of their ability to be explainable or interpretable [45].

As shown in Figure 7, black-box models, like most neural network-based methods, cannot be used in SCS unless they are interpretable by adding an explanatory component. Choosing an explanatory method is also a difficult task, as it is necessary to determine which type of explanation and result is appropriate for a given problem [43, 46]. In Figure 8, some of the key aspects of an xAI taxonomy are summarized.

The *scope* describes the area to which the explanation refers. It can refer to the functioning of the entire model (*global xAI*) or to the explanation of the background of a specific decision (*local xAI*) [47–49].

The *stage* is divided into *ante-hoc* and *post-hoc* explaining methods. [50]. An ante-hoc explainability method provides explanations during training and is given by the design of the algorithm. It is particularly applicable for classical ML methods, as long as the models are small and manageable. This method is also called *intrinsic xAI*, which is model-specific and explainable due to its internal structure. In this case, the learning method can either be transparent (algorithmic transparency), the technical functionality can be clear (simulability), or the algorithm can be decomposed into its individual parts (decomposability)

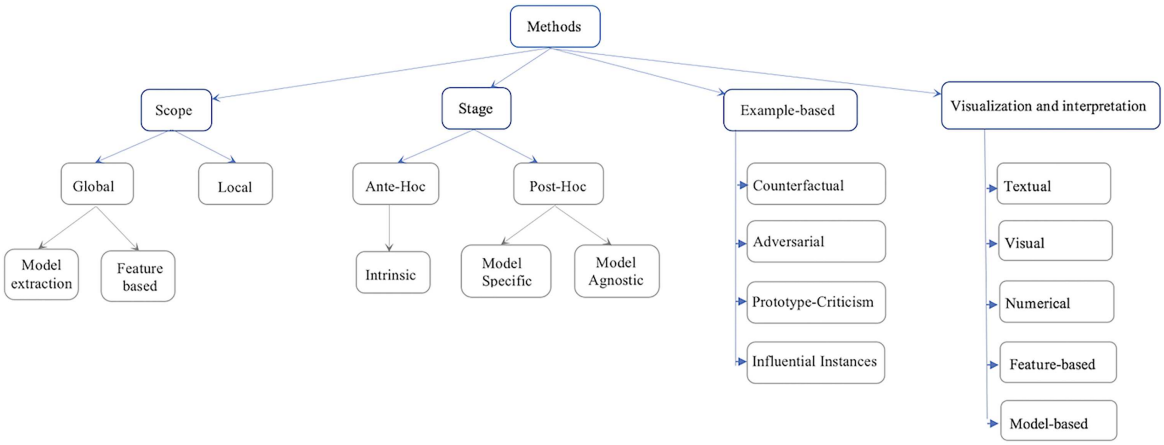


**Figure 8.** xAI taxonomy

[51, 52]. Post-hoc methods can be either *model-specific* or *model-agnostic*. In the first case, the methods are developed for a specific model. In model agnostic xAI, an explanation is given after training, which allows portability to other models [53].

The *example-based* methods can include *counterfactuals* as a possibility to explain predictions, or *adversarial* examples to provide insight into the internal structure of algorithms, find their weaknesses, and improve interpretability [43]. *Prototypes* are selected representative instances from the data, and *criticisms* are instances that are not well represented by these prototypes. Identifying and analyzing *influential instances* helps to find problems with the data, debug the model, and better understand the behavior of the model. Other aspects of the taxonomy of explanatory methods distinguish whether the basic *functions* are important to obtain information about a model, or whether the *result*, *e.g.*, the calculation of feature importance (see Section 4), is essential [49].

The explanations (*visualization and interpretation*) themselves are in turn dependent on the method used and the objective ([54, 55]; *e.g.*, [56] for autonomous driving). Several techniques and tools are meanwhile used to support explainability [19, 53, 57, 58], or to define different phases of XAI in a typical machine learning development process [59].

In [60], the Situation Awareness Framework for Explainable AI (SAFE-AI) provides metrics for evaluating the quality of explanations. Explainability in SCS is considered from different perspectives, such as regulations and standards [61]. For safety-critical systems, explainability is not enough for a system to be trustworthy. Various requirements such as robustness, reliability, traceability, data quality, and other factors must be demonstrated [25, 43, 62, 63].

This brief insight into current developments, the visible global efforts to regulate the use of AI methods and the numerous existing standards on safety-related systems make uncertainties regarding the requirements to be fulfilled obvious.

From the aspects described so far, which are not to be considered complete and which must be addressed when using AI in SCS, the requirements that also serve as "knowledge" for the Self-Enforcing Network (SEN, Section 4) are summarized below.

### 3.3 The "knowledge" of the model

Regardless of the possible combinations for controlling or monitoring machines with embedded AI methods, there are numerous software development requirements that need to be met. Some selected topics that must be regarded when using AI methods in SCS are briefly presented, most of which have been collected by [64].

*General software requirements*

- Verification: verification methods and formal verification (*e.g.*, rule extraction), dynamic and static verification (tests) (critical in CSC with embedded AI).
- Validation: Model, solution checking [65] (critical in CSC with embedded AI).

- Safety arguments/Reliability safety: qualitative requirements. Collection of various documentation for the development process and the final product.
- Requirements of software safety standard (DO-178C): Development Assurance Level (DAL): Five levels of software criticality: A (most critical) through E (least critical) [66].

*Requirements for AI methods (considering Articles 8 – 15 of the AI Act)*

- Adaptability: flexible structure adaptable to different accident scenarios.
- Situation awareness: a key factor in performance improvement and error reduction.
- Coherence: guidelines for model development to ensure consistency.
- Corner-case robustness: reaction to input in safety-critical autonomous systems.
- Robustness guarantee: System robustness through testing, formal verification, robustness guarantees, and other countermeasures (required in Article 10, AI Act).
- Systematical testing/quality of test data: data quality and testing are key to confidence building (ISO/IEC/IEEE 29119-4, which will be replaced by ISO/IEC AWI TS 29119-11; required in Article 17 of the AI Act).
- Handling or avoiding combinatorial explosions: critical on high-dimensional data.
- Method for dependability assessment: service that can be trusted [67]; critical for software-intensive SCS.
- Method for reliability analysis: the probability that the application will satisfactorily fulfill its expected function without failure.
- Availability: the specific functions must be executed within a certain period of time.
- Proper operation: testing the functional operation of safety-related systems.
- Method for fault diagnosis: detailed fault diagnosis to quickly identify process anomalies and component failures and determine the source of failures (*e.g.*, [68]).
- Process-based safety arguments: designed specifically for NN used in safety-related applications [66]; for reinforcement learning, *e.g.*, [69].
- Evidence-based safety arguments: framework for safety case development method based on goal structuring notation (GSN) [70–73].
- Hazard and risk analysis: *e.g.*, HARA; methods for DL are found *e.g.*, in [74].
- Coding standards: traceability and efficiency of the source code [66]; for AI methods, further specification is required (*e.g.*, [75]).
- Part of the system: when the control and monitoring software is integrated into the machine, the requirements are different from those of the diverse monitoring.
- No external access to the system: important SCS prerequisite.
- Not accessible for modification by user: important SCS prerequisite.
- System diagnosis: less requirements if the method is used only for diagnosis.
- Performance level (PL): PLa – PLe.
- System certainty: system trust building.
- White Box AI: transparency (required by Articles 13 and 52 of the AI Act).
- Interpretability: deployers must be able to interpret the system's results and use them appropriately (Article 13 AI Act). Methods of interpretability are needed for large datasets.
- Explanation: explanation of the results adapted to the stakeholder's expertise.

These requirements, which do not claim to be exhaustive, have been incorporated into the model, along with additional information and requirements that which are necessary to obtain the possible classification for safety-related applications:

*Possible methods:*

- Classical algorithms: rule-based, *e.g.*, Petri nets.
- Machine Learning (ML): *e.g.*, Bayesian nets.
- Deep Learning (DL).
- AI – Expert systems: rule-based, explanatory.
- AI – Neural Networks: flatten, deterministic.

*Characteristics:*
- Deterministic: makes it easier to verify the reliability.
- Statistics (probabilistic): complicates verification of reliability.
- High complexity: complicates reliability testing; reduces potential applications.
- Constant behavior: ensure that the software will always behave the same (as deterministic).

*Architecture:*
- Redundancy: simultaneous use of two functions allows comparison,
- Completeness: the method is part of the system and evaluated as well.
- Diversity: redundancy also allows for finding systematic failures.
- Detectability: method is used for diagnostics or monitoring and is not necessarily part of the machine.

*Machinery:*
- Safety circuit: conventional yes or no (1 or 0).
- Architecture (whole system): 1-channel, 2-channel, 3-channel.
- Functional safety (machinery): required yes or no (1 or 0).
- Safety rating: how reliable is the functionality regarding safety (SIL 1 to SIL 3 or PL a to e or other)?

For each of the requirements listed above, the required degree of fulfillment for the respective SIL was determined by experts and based on practical experience.

The developed model can be easily implemented in the Self-Enforcing Network (SEN), which is briefly described below, as well as the explanatory component.

## 4 The explainable Self-Enforcing Network (SEN) and the requirements model

SEN, developed by the research group CoBASC (Computer Based Analysis of Socio-technical Complexity) [76], is a self-organized learning network. The peculiarity of SEN is that the weight values are not randomly generated as in other types of neural networks. In this case, it is a two-layered feed-forward network in which the attributes $a$ (requirements) are mapped to the objects $o$ (SIL). The vector containing the attributes is the input vector of the NN and the objects are the output vector. The stated assessments of the required degree of fulfillment are recorded in the so-called "semantic matrix" (sm). Figure 9 shows the architecture of a SEN.

The learning rule called Self-Enforcing Rule (SER) transforms the values of the semantic matrix $v_{\mathrm{sm}}$ 1:1 into the weight matrix $w_{\mathrm{ao}}$ of the network, i.e., the values and the learning process can be reconstructed at any time (as shown in Eq. 1). Concretely, this means that the values in the semantic matrix are verified against the learning rule. If a value $v_{\mathrm{sm}}$ in the connection between an attribute (input neuron) and an object (output neuron) is 0, this value also remains 0 in the weight matrix. If a value is unequal to 0, it is multiplied by the learning rate $c$, a constant value, according to the learning rule. Since the weight values are not randomly generated but are adjusted by the learning rule, a learning rate $c$ of 0.1 and one learning step is in most cases sufficient.

A cue validity factor (cvf) is defined for each attribute $a$ (requirement) when the model is created. The cue validity factor influences the strength of an attribute's effect on the network's activation and is additionally calculated by the learning rule in the weight matrix (as shown in Eq. 1).

$$w_{ao}{=}c{*}v_{sm}{*}cvf_{a} \tag{1}$$

The SEN has a number of different activation functions that can be set according to the type of problem or the size of the data volumes. The final activation $a_{\mathrm{j}}$ (as shown in Eq. 2) of the neurons (objects) is determined for this task by what is called the Enforcing Activation Function (EAF).

$$a_j = \sum_{i=1}^{n} \frac{w_{ao} * a_i}{1 + |w_{ao} * a_i|} \tag{2}$$

The network learns the data in a self-organized manner, meaning that no target is given, and orders them according to their similarity. New input data is then classified with the learned data and visualized in two

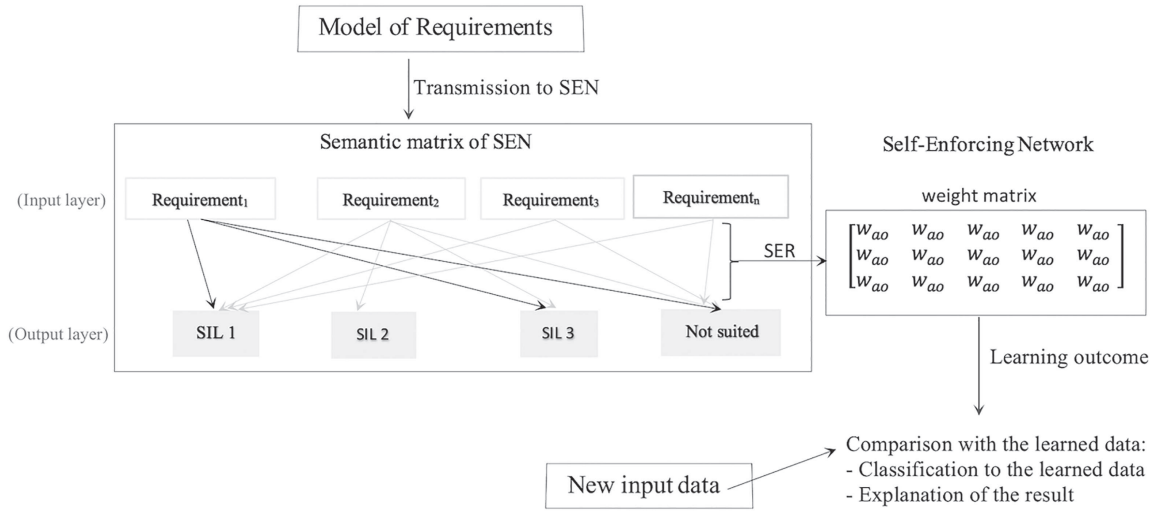


**Figure 9.** Architecture of SEN

different ways, which is another special feature of SEN. The similarity result is calculated by the highest activation value of a neuron (object), which is output in a "ranking" list, and the smallest Euclidean "distance", which is typical for self-organized learning networks.

These calculations are important to ensure that a model is developed in a reliable way. For example, the highest activated neuron by a new input vector (or the highest probability for a class, depending on the DL model) gives the result. This calculation is known to lead to incorrect classifications because higher values in the weight matrix influence the activation result. As a result, the activated neurons may differ from those determined by Euclidean distance, which also accounts for smaller differences.

In the validation phase, i.e., testing the same training data, it is necessary that both calculations give the same result in the first place. If this is not the case, the model needs to be rethought. Of course, new data may produce different classification results from the two calculations; in this case, it is indicated that there is no conclusive result.

### 4.1 Examples of practical applications of SEN in SCS

For these special features, the SEN is being tested, *e.g.*, in a project with the German Weather Service (DWD) to provide Air Traffic Management (ATM) at Frankfurt Airport with runway selection recommendations based on weather forecasts [77]. In another project, SEN is being used as a prototype for diverse monitoring of a safety-critical system in mines [78].

In the first case, the SEN learns the "ideal" wind conditions for the respective runways in the form of "reference types", which are determined based on data (weather forecasts and runway selections) and experts. Based on new weather forecasts, SEN makes a recommendation for a specific runway. As it is impossible to include all wind constellations in the learning process, the reference types have proved their worth, as SEN can check which runway is the most suitable. The evaluation can only be carried out by the expert decisions.

In the second case, fault-free processes in the conveyor are learned according to the same principle. The tower winding machine is monitored and controlled by a large number of sensors in accordance with the "Technical Requirements for Shaft and Inclined Conveyor Systems" (Technische Anforderungen a Schacht- und Schrägförderanlagen – TAS). Two control systems are installed in the shaft and the safety-critical signals are read in on two channels and monitored for parity. The conveyor machine is programmed using function blocks [78] and corresponds to a 2-channel safety system (Figure 5). As a redundant method, SEN has a monitoring function and is evaluated against the implemented methods.

In both cases, experts also determine the most important characteristics of the processes, which are reinforced in the SEN by the cvf. By creating reference types, the amount of data to be learned is significantly reduced without any loss of quality, which means that new input data can be processed very

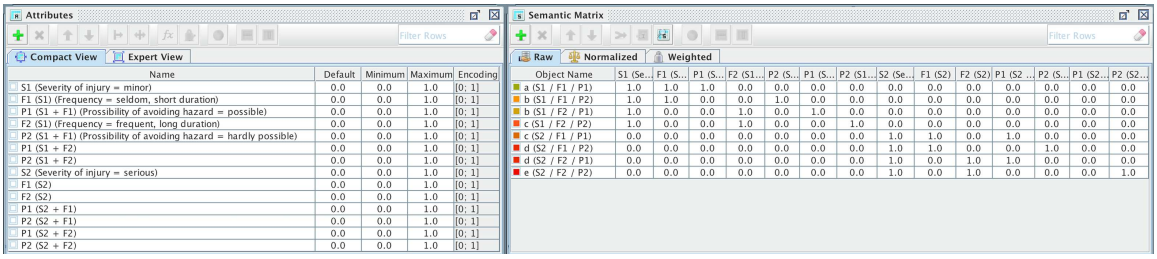


**Figure 10.** The definition of the attributes and objects in the SEN tool

fast. As the weight matrix is not generated at random, but according to the specific learning rule, experts can always reconstruct the results.

Because these areas are classified as safety-critical and the amount of data to be analyzed is very large, the SEN must prove that it meets the requirements for a support system and a redundancy monitoring system. In addition to the aforementioned standards and requirements that apply to AI methods in the SCS, specific requirements must also be met.

In aerospace, *e.g.*, DO-178B applies to software development, which is why this standard has been also integrated into the model. For the use of AI methods, the mentioned developments for the ML application levels [16] are being pursued. In the mining sector, there are demands for updates to ensure the safety of control systems (*e.g.*, [79]). For *e.g.*, mine hoist command systems are safety-related electrical control systems (SRECS) using multiple safety instrumented systems (SIS), the use of a SIL or PL rating is strongly recommended by [80]. There are currently no explicit regulations governing the use of AI methods in the mining sector. With the coming into force of the AI Act and the Machinery Regulation 2023/1230, it is expected that regulations for AI in functional safety-critical systems will be adapted across the board.

Consequently, developers of AI methods and companies designing or using safety-critical systems need to continuously monitor developments in order to meet all requirements in time. These examples illustrate the current complexity. On the one hand, the AI software itself must meet all safety requirements in order to reliably support existing systems or assist humans. On the other hand, developments in standards must be monitored, and the development status as well as the degree of compliance with the requirements must be documented.

The application examples involve data-driven models that are enhanced with human expertise to ensure reliability and traceability. To verify the developmental stage or the level of compliance with the requirements, the model is developed manually only by experts. The following demonstrates how the SEN operates in this context.

### 4.2 Simplified example for determining the PL

To illustrate how the SEN works, the risk graph (Section 2) is used to determine the Performance Level (PL). The attributes used are severity (S1 and S2), frequency (F1 and F2), and possibility (P1 and P2) of hazard avoidance, which are assigned to the corresponding objects (PL) for each path in the semantic matrix (Figure 10). For each PL, the value 1.0 is entered in the semantic matrix for the components if they correspond to a PL (as the connection between the attribute and object), otherwise the values are 0.0.

In addition to a simple input option for the attributes and for the semantic matrix, the SEN tool offers several visualizations of the results, of which the ranking and the distances are shown below.

If a linear activation function that adds the incoming signals, a learning rate of 1.0, a cue validity factor (cvf) of 1.0 for all attributes, and one learning iteration are selected, the learning rule results in a weight matrix that corresponds exactly to the semantic matrix. As mentioned before, this ensures the traceability of the learning process and the data, as shown in Figure 11.

Since the weights correspond to the values in the semantic matrix, one learning step is sufficient to learn the PLs and their paths in the risk matrix.

After learning, new inputs can be presented to the network. In this case, the example is from [81] who used the SISTEMA software to determine the PL in the risk matrix using the specified $S$, $F$, and $P$, for the implementation of an energy storage safety system (Figure 12).

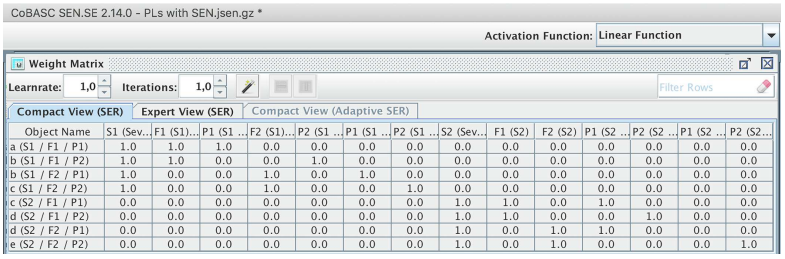


**Figure 11.** The weight matrix of SEN and selected parameters

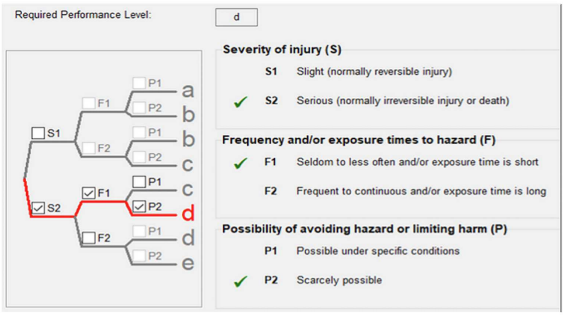


**Figure 12.** The result of SISTEMA after specifying S, F, and P, indicates that PL d is required for the problem ([81], p. 111). Developed by the DGUV (German Social Accident Insurance), the software assists in evaluating safety according to the ISO 13849-1 standard (for Windows OS only)

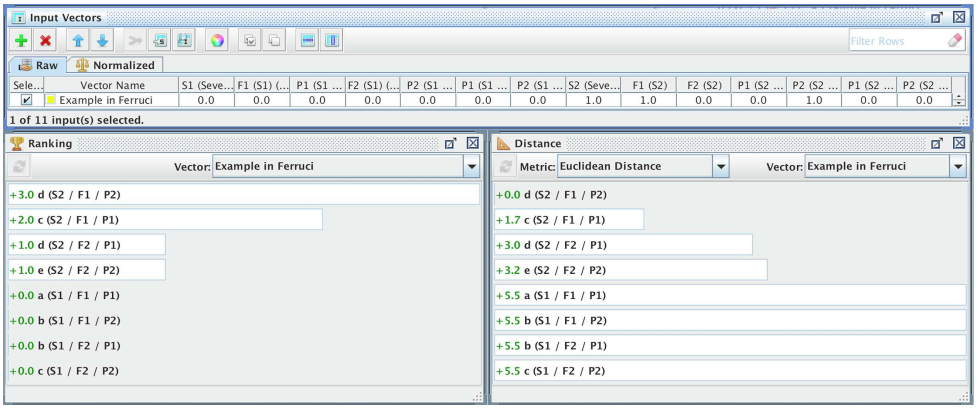


**Figure 13.** Input vector (on the top) and the results of SEN

This information is given as an input vector into the SEN tool. Figure 13 shows the computed result in the ranking and in the distances.

The input vector is shown in the top image, with the calculations and visualizations of the rankings and distances below. The result is clear: the highest activation with a value of 3.0 (by the linear function for three components with a value of 1.0) and the distance with a value of 0.0 shows the PL d in first place, with the brackets indicating that the entries are S2, F1, and P2.

The result shows the overall activation value of an object (PL) or the value of the distances, but it is not clear which attributes ($S$, $F$, $P$) are responsible for this result. Only the explanatory component reveals the importance of each attribute.

## 4.3 The explainability of SEN

The explanatory component [54, 77] uses the calculation of Feature Importance (FI) based on the concept of Shapley values (SV), which was adapted to "explainable AI" [28, 55]. The FI indicates the impact of the components (attributes) on the output.

SV, introduced to game theory by Shapley [82], calculates the effects of cooperation and individual performance on the outcome of a game.

Given a set of players participating in the game, all possible player sets $S$ are formed, containing a subset $n$ of the entire number of players $N$. The influence $\Phi$ of player $i$ on the outcome of the game $v$ is calculated for each player set. The total contribution of the player outcome $\Phi_{\mathrm{i}}(v)$ is the sum of all partial influences resulting from coalitions in the respective player sets (as shown in Eq. 3).

$$\Phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(n-1-|S|)!}{n!} \left(v(S \cup \{i\}) - v(S)\right) \tag{3}$$

In [77] is demonstrated, that the calculation at SEN can be simplified compared to the calculation at other models (*e.g.* [28]) because of the weight matrix of SEN. This has the effect that the weight values directly reflect the importance, of each attribute, and thus the FI $\Phi$ of attribute i has a corresponding effect on the output v (as shown in Eq. 4):

$$\Phi_i(\mathrm{v}) \;=\; \mathrm{v(i)} \tag{4}$$

According to the xAI taxonomy, SEN is considered an *intrinsically* explainable AI because of its internal operation, as well as its instant identification of FI. The determination and visualization of FI can be read directly by decomposing a vector into its individual components (attributes), the sum of which also results in the final activation value in the ranking of an object, which is also important for traceability.

### 4.4 The SEN-model and FI-Visualization

As mentioned earlier, the task of SEN is to learn the 44 selected requirements (Section 3) that are necessary to achieve a particular SIL. Conversely, it should also learn which software developments do not sufficiently satisfy the requirements and are therefore unsuitable for safety-related systems.

#### *4.4.1 Methodology notes*

Again "ideal" reference types are defined, one each for SIL 1–3 and one for an inappropriate method in functional SCS. "Ideal" means that the requirements can hardly be met in reality but should be aimed at.

For example, all requirements are assigned a value of 1.0 to achieve SIL 3. On the other hand, ML and DL methods were assigned values of 0.0 according to the strict regulations of IEC 61508, apart from AI methods such as expert systems with an explanatory component (with a value of 1.0) and explanatory flat and deterministic NN (with a value of 0.4). The PL has a value of 3.5 and the architecture was assigned a value of 3, which represents a 3-channel architecture.

The values for SIL 2 and SIL 1 have been reduced accordingly, as the requirements for these are not quite as high. Finally, for the hypothetical case, most of the requirements are not met, i.e., they have a value of 0.0.

#### *4.4.2 Parameters*

All attributes with general requirements have a cue validity factor (cvf) of 0.5; the "explainability" attribute and the attributes that represent the whole system and are decisive for the SIL have a value of 1.0. These requirements are therefore considered more important for the assignment to an SIL (Table 2 in Section 5).

The data in the semantic matrix (these are the assessments of the requirements that must be fulfilled in an SIL) are normalized in the value range between $-1$ and 1. Normalization in this range is particularly advantageous for data-driven models of real-world problems to achieve better differentiation in the outputs.

Further parameters are a learning rate of 0.1, a learning iteration of 1, and the enforcing activation function (EAF).

#### *4.4.3 FI-Visualization*

The Visualization also represents a special feature, as the FI results are displayed related to a reference type [77]. This means that the FIs for the reference types are first determined. The FIs for new input data are then mapped by comparison, so that the matches and deviations

**Table 1.** Validation results

| | Ranking | Distance |
|---|---|---|
| SIL 1 | 1.26 | 0.00 |
| SIL 2 | 0.82 | 0.00 |
| SIL 3 | 2.68 | 0.00 |
| Not suitable for SCS | 2.84 | 0.00 |

**Table 2.** Requirements and degree of fulfillment for 4 different developmental stages and methods

| Requirements | xAI, deterministic NN | DL, without xAI | DL, beginnings of xAI | Rule-based algorithm |
|---|---|---|---|---|
| Explanation | 1.0 | 0.0 | 0.4 | 1.0 |
| Deterministic | 1.0 | 0.0 | 0.0 | 1.0 |
| Statistics (probabilistic) | 0.0 | 0.8 | 0.4 | 0.0 |
| High complexity | 0.0 | 0.8 | 0.8 | 0.0 |
| Constant behavior | 1.0 | 0.8 | 0.4 | 1.0 |
| Hazard and risk analysis | 0.0 | 0.0 | 0.1 | 1.0 |
| Coding standards | 0.8 | 0.0 | 0.7 | 1.0 |
| No external access | 1.0 | 1.0 | 1.0 | 1.0 |
| Not accessible for modification by user | 1.0 | 0.0 | 1.0 | 1.0 |
| Diagnosis application | 1.0 | 0.0 | 1.0 | 0.0 |
| Diverse | 1.0 | 0.0 | 1.0 | 1.0 |
| Architecture (whole system) | 3.0 | 3.0 | 3.0 | 2.0 |

are visible for each component. Since a complete representation is no longer meaningful for humans with large amounts of data, it is determined which attributes are particularly important for a decision or recommendation. (See https://www.eurocontrol.int/sites/default/files/2024-04/20240429-flyai-forum-session-3-zinkhan-greisbach-zurmaar-kluver.pdf for an illustration of the procedure for the "Runway Recommendation Project".)

For the subsequent presentation of the results, a simplified model is used in which SIL 3, with the highest requirements, serves as a reference for the FI.

## 5 Results

First, the reference types are validated by presenting them as input data. In the case of NN, this always verifies that the model or data is unambiguously learned and classified. The activation values (ranking) respectively the distances from Table 1 result from the validation of the learned vectors. A positive activation value for "Ranking" and a value of 0.0 for "Distance" is required to declare the validation successful.

Due to the normalization between $-1$ and 1, the activation of SIL 2 is correspondingly lower. The two poles "SIL 3" and "Not suitable for SCS" have the highest final activation. As expected, the distances have for all reference types the value 0.0, as the validation data is identical to the learning data.

Next, a user can specify in the input vector for each requirement the degree of fulfillment, the development stage, or define if a property is true or not (*e.g.*, a value of 1.0 if a deterministic NN is used, the value 0.0 for complexity, the value 3.0 for a 3-channel architecture of the whole system, etc.).

It is very important to mention that for the SIL the user can of course not give an indication, since the SIL is supposed to be the result of the SEN, hence the value in the input vector for this attribute is always 0.0.

Table 2 shows, for illustrative purposes, only those values of the selected attributes that are most important to explain which SIL can be reached in a SCS with a specific method. The rating is done for a flat neural network with an explanatory component (xAI, deterministic NN, like the SEN presented here), two different developmental stages using a Deep Learning (DL) method, and, in comparison to NN-based methods, for a rule-based algorithm (as used *e.g.*, for control systems in mining).

The possible SILs determined by SEN and the corresponding FI visualizations are as follows:

For the developmental stage of a deterministic NN with an explanatory component, the result is shown in Figure 14.

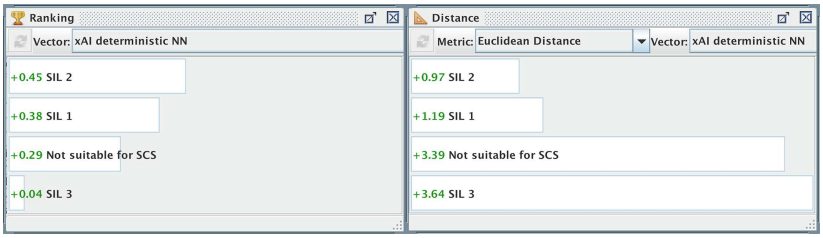


**Figure 14.** Result of SEN for the input vector "xAI deterministic NN"

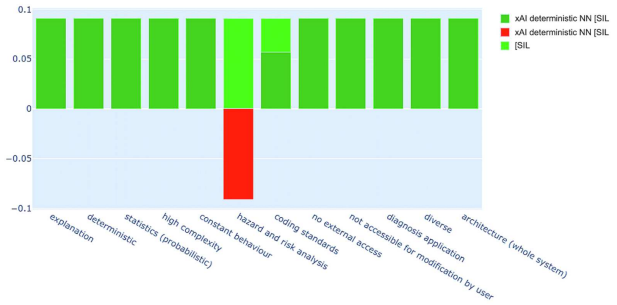


**Figure 15.** Shapley values for the input vector "xAI deterministic NN", equivalent to the described xAI SEN

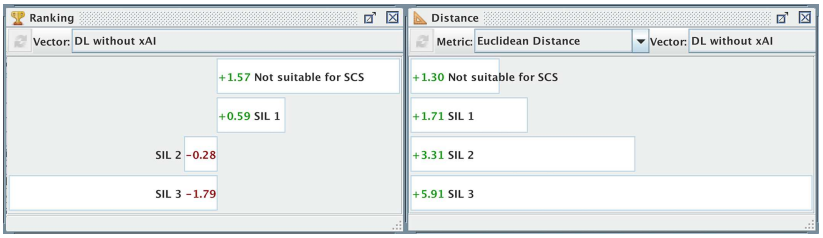


**Figure 16.** Result of SEN for the input vector "DL without xAI"

SEN indicates in the first place for both calculations (ranking and distance) that the classification occurs to SIL 2 but with a lower activation level than for the validated vector in Table 1 and at a greater distance. However, it also means that some requirements are not sufficiently met, but which ones are not specified.

The calculation of the FI provides a visual result of the explanatory component for the missing requirements to achieve SIL 3; for illustration purposes, the results are shown in Figure 15, again in excerpts for selected requirements and for the intended application of the method.

The specified requirements for SIL 3 are shown in light green. Since all the requirements must be met in their entirety for SIL 3, they all have the same value. A dark green color means that a new input vector is 100% compliant with the learned requirements. Based on the results, it is immediately apparent that the "hazard and risk analysis" requirement (shown in red) is not met at all, and the "coding standards" requirement is only partially satisfied. Note, that current code standards cannot be fully met because they cannot be automatically applied to AI methods.

When using a DL method without an explanatory component (DL without xAI), the model is not suitable for SCS, as clearly shown in Figure 16. This is due to the higher complexity of the method and the probabilistic elements. Accordingly, the results of the FI, as shown in Figure 17.

It is immediately clear from the FI that most of the requirements are not being met at this stage of development.

Of interest is the result for the developmental stage using a DL method with investigations in explainability, hazard, and risk assessment (DL, beginnings of xAI – Table 2). In this case, the results of the SEN are not equal in the first place (Figure 18).

Deviations of the inputs from the requirements are shown in the explanation component with respect to SIL 3 in Figure 19.

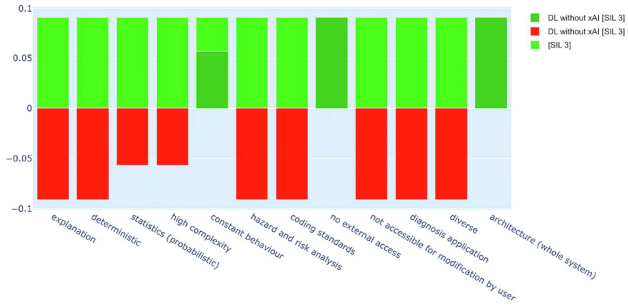


**Figure 17.** FI for the input vector "DL without xAI" with respect to SIL 3

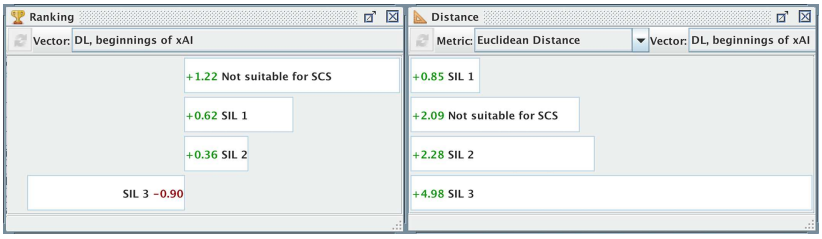


**Figure 18.** Result of SEN for the input vector "DL, beginnings of xAI"

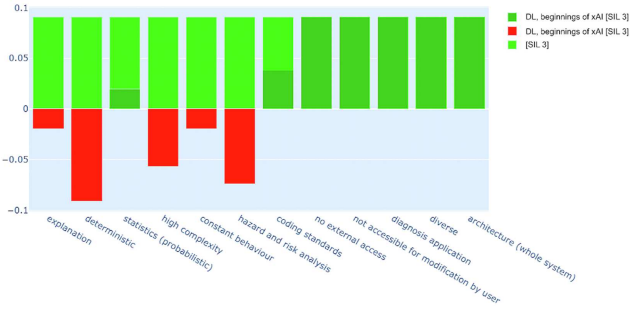


**Figure 19.** FI for the input vector "DL, beginnings of xAI"

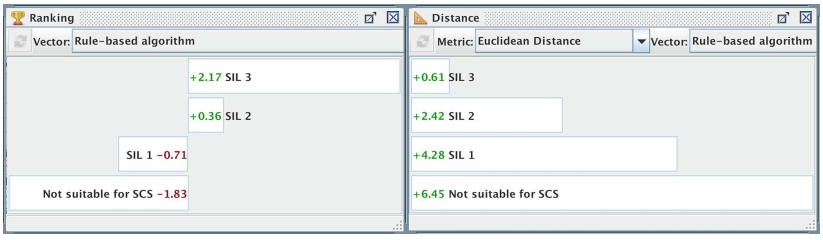


**Figure 20.** Result of SEN for the input vector "Rule-based algorithm"

There is no clear SIL assignment in Figure 19. However, the distance calculation in SEN with a value of 0.85, assigns the developmental stage of the method to SIL 1, according to the conditions met. On the other hand, the strongest activation (ranking) in the first argues that the method is still unsuitable for SCS, although the activation value of 1.22 is lower than that of 2.84 for the evaluation vector "Not suitable for SCS" and near to the final activation of SIL 1 in Table 1. From a ranking perspective, its use in safety-critical systems is still not recommended due to the high complexity and probabilistic features of the method.

The visualization of the FI in Figure 20 is also not entirely clear in this case and could be misinterpreted; the combination of several visualizations and the specification of numerical values is advantageous, as shown in this case.

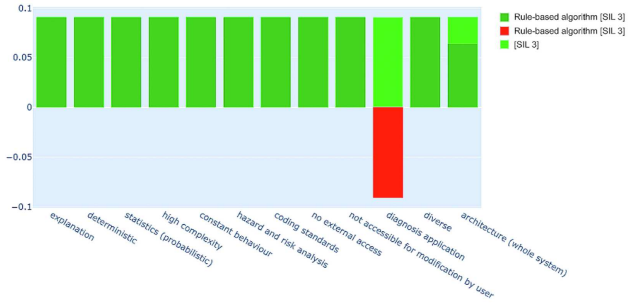


**Figure 21.** FI for the input vector "Rule-based algorithm"

The basic model developed to test the suitability of an AI method based on NN provides comprehensible results. In contrast, a rule-based method is specified as an input vector in which the algorithm has a control function in a 2-channel architecture, corresponding to an algorithm used in the monitoring of conveyor machines. In this case, the functional safety of machines is required; accordingly, a PLe must be achieved that has a value of 5.0 in the semantic matrix and is also expected in the input vector. In Figure 20, the result is again unambiguous.

SEN indicates with an activation value of 2.17 (ranking) and a distance of 0.61 that this algorithm conforms to requirements for SIL 3. The FI gives more insights (Figure 21).

As the base model was primarily designed for applications of AI methods that are currently not allowed to be used for control, but only redundantly for monitoring or diagnostics, the FI shows a deviation in this component. In addition, a 3-channel architecture is expected, but the algorithm is implemented in a 2-channel safety system. However, it is also immediately apparent in the FI that the requirements for SIL 3 are largely fulfilled.

This result shows the first limitation of the basic model. It has been developed primarily for NN-based methods, and the requirements or possible applications of these algorithms have been considered. In order to be able to verify other methods, the requirements for diverse applications need to be extended.

Another limitation is that the selection and evaluation of the requirements were not determined by a broad group of experts, which means that subjective influences cannot be ruled out. To generalize the model, more experts should be involved to obtain an objective evaluation. However, the model presented can serve as a basis for further adaptation.

In additional to this application within the functional safety discipline, it can be concluded that a SEN can be used to support evaluations in many other engineering disciplines where requirements are clearly specified with more or less clear acceptance criteria. In the area of cyber security, the well-known IEC 62443 series of standards, with Part 4–1 on requirements for the lifecycle of secure product development and Part 4–2 on technical security requirements, provides a clear framework for achieving characteristic security levels SLs of a component over its entire lifecycle. Substituting the ruleset used above with the requirements of this standard would be a promising candidate for further use of SEN networks in regard to fulfilling standard requirements. A further discipline is SOTIF as used in the automotive industry. Here, the framework of requirements as described in ISO 26262 with reachable ASILs also is based on clear requirements that could be used as a set of rules within a SEN network to evaluate whether a certain ASIL has been achieved for the intended function. The model is transferrable to a lot of other engineering disciplines by changing the set of attributes accordingly.

## 6 Conclusion and future work

There is currently a highly dynamic development both at the level of standards and laws and in the implementation of AI methods. In this article, we have selected the most important aspects and requirements for AI methods in SCS from both the literature and practical experience and evaluated them in relation to SILs.

The current model, which considers only general terms, provides a general overview of important aspects. It therefore offers a compact set of necessary requirements that can be easily adapted and expanded.

In the SEN, the information provided by users regarding the degree of fulfillment of requirements is evaluated, assigned to a SIL, and specified by the Shapley Values (SV). The examples show the potential of such a model, where all stakeholders can select and evaluate requirements and desired outcomes, considering regulatory requirements and standards. Progress or unfulfilled requirements can be easily and transparently tracked.

Only the relevant requirements can be selected, specified, and evaluated for the specific case. Behind each generic term there are numerous requirements that need to be identified for a concrete case; the approach presented here makes it possible to maintain an overview.

In the next step, the model is varied so that the fulfillment of the requirements can be checked based on a given SIL, as is done in the case of functional safety assessment. In this case, the "target SIL" is defined and then it is verified that this target is achieved.

### Conflict of Interest

The author declares no conflict of interest.

### Data Availability

No data are associated with this article.

### Authors' Contributions

Anneliesa Greisbach and Christina Klüver contributed to the explanatory Self-Enforcing Network, while Michael Kindermann and Bern Püttmann provided the standards and all aspects of functional safety-critical systems. The model was developed based on the expertise of all authors.

### Acknowledgements

We thank the reviewers for their constructive comments.

### Funding

This research received no external funding.

## References

[1] Bengio Y, Hinton G and Yao A et al. Managing extreme AI risks amid rapid progress. Science 2024; **384**: 842–845.

[2] Wörsdörfer M. Mitigating the adverse effects of AI with the European Union's artificial intelligence act: Hype or hope? Glob Bus Organ Excell 2024; **43**: 106–126.

[3] Fraunhofer IKS, Heidemann L and Herd B et al. The European Artificial Intelligence Act. Whitepaper-EU-AI-Act-Fraunhofer-IKS-4.pdf. 2024.

[4] European Union. Regulation (EU) 2023/1230 of the European Parliament and of the Council of 14 June 2023 on machinery and repealing Directive 2006/42/EC of the European Parliament and of the Council and Council Directive 73/361/EEC. 2023; **66**, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2023:165:FULL

[5] de Koning M, Machado T and Ahonen A et al. A comprehensive approach to safety for highly automated off-road machinery under Regulation 2023/1230. Safety Sci 2024; **175**: 106517

[6] Castellanos-Ardila JP, Punnekkat S, Hansson H and Backeman P. Safety argumentation for machinery assembly control software. In: International Conference on Computer Safety, Reliability, and Security, Springer, 2024, pp. 251–266.

[7] Cao Y, An Y, Su S and Sun Y. Is the safety index of modern safety integrity level(SIL) truly appropriate for the railway? Accid Anal Prev 2023; **192**: 107267.

[8] Malm T, Venho-Ahonen O and Hietikko M et al. From risks to requirements: Comparing the assignment of functional safety requirements, 2015.

[9] Okoh P and Myklebust T. Mapping to IEC 61508 the hardware safety integrity of elements developed to ISO 26262. Safety and Reliability (Taylor & Francis, 2024), pp. 1–17.

[10] Diemert S, Millet L, Groves J and Joyce J. Safety integrity levels for artificial intelligence. In: International Conference on Computer Safety, Reliability, and Security, Springer, 2023, pp. 397–409.

[11] Dalrymple D, Skalse J and Bengio Y et al. Towards guaranteed safe AI: A framework for ensuring robust and reliable AI systems, 2024 ArXiv preprint [arXiv: 2405.06624], 2024.

[12] Future of Life Institute. AI Governance Scorecard and Safety Standards Policy. Evaluating proposals for AI governance and providing a regulatory framework for robust safety standards, measures and oversight, 2023. https://futureoflife.org/wp-content/uploads/2023/11/FLI_Governance_Scorecard_and_Framework.pdf

[13] Abbasinejad R, Hourfar F, Kacprzak D, Almansoori A and Elkamel A. SIL calculation in gas processing plants based on systematic faults and level of maturity. Proc Safety Environ Protect 2023; **174**: 778–795.
[14] Shubinsky I, Rozenberg E and Baranov L. Safety-critical railway systems. Reliability Modeling in Industry 4, Elsevier, 2023, pp. 83–122.
[15] Golpayegani D, Pandit HJ and Lewis D. To be high-risk, or not to be–semantic specifications and implications of the AI act's high-risk ai applications and harmonised standards. In: Paper presented at: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, 2023: pp. 905–915.
[16] European Union Aviation Safety Agency (EASA). EASA Concept Paper: guidance for Level 1 AND 2 machine learning applications Issue 02. 2024, https://horizoneuropencpportal.eu/sites/default/files/2024-06/easa-concept-paper-guidance-for-level-1-and-2-machine-learning-applications-2024.pdf
[17] DIN, DKE: German Standardization Roadmap on Artificial Intelligence. 2022. www.din.de/go/roadmap-ai.
[18] Bacciu D, Carta A, Gallicchio C and Schmittner C. Safety and Robustness for Deep Neural Networks: An Automotive Use Case. In: International Conference on Computer Safety, Reliability, and Security, Springer, 2023, pp. 95–107.
[19] Perez-Cerrolaza J, Abella J and Borg M et al. Artificial intelligence for safety-critical systems in industrial and transportation domains: A survey. ACM Comput Surv 2024; **56**: 1–40.
[20] Brando A, Serra I and Mezzetti E et al. On neural networks redundancy and diversity for their use in safety-critical systems. Computer 2023; **56**: 41–50.
[21] Oveisi S, Gholamrezaie F and Qajari N et al. Review of artificial intelligence-based systems: evaluation, standards, and methods. Adv. Stand Appl Sci 2024; **2**: 4–29.
[22] Kelly J, Zafar SA and Heidemann L et al. Navigating the EU AI Act: A methodological approach to compliance for safety-critical products. In: IEEE Conference on Artificial Intelligence (CAI) 2024; 979–984, doi: 10.1109/CAI59869.2024.00179.
[23] Wei R, Foster S and Mei H et al. ACCESS: Assurance case centric engineering of safety–critical systems. J Syst Soft 2024; **213**: 112034
[24] Zhang X, Jiang W and Shen C et al. A Survey of deep learning library testing methods, ArXiv preprint [arXiv: 2404.17871], 2024.
[25] Mattioli J, Sohier H and Delaborde A et al. An overview of key trustworthiness attributes and KPIs for trusted ML-based systems engineering. AI and Ethics; **4**: 15–25.
[26] Iyenghar P. Exploring the impact of dataset accuracy on machinery functional safety: Insights from an AI-Based predictive maintenance system. ENASE 2024: 484–497, DOI: 10.5220/0012683600003687.
[27] Habbal A, Ali MK and Abuzaraida MA. Artificial Intelligence Trust, risk and security management (AI TRiSM): Frameworks, applications, challenges and future research directions. Exp Syst Appl 2024; **240**: 122442.
[28] Giudici P, Centurelli M and Turchetta S. Artificial Intelligence risk measurement. Exp Syst Appl 2024; **235**: 121220.
[29] Bjelica MZ. Systems, Functions and Safety: A Flipped Approach to Design for Safety, Springer Nature, 2023.
[30] Morales-Forero A, Bassetto S and Coatanea E. Toward safe AI. AI Soc 2023; **38**: 685–696.
[31] Zeller M, Waschulzik T, Schmid R and Bahlmann C. Toward a safe MLOps process for the continuous development and safety assurance of ML-based systems in the railway domain. AI and Ethics; **4**: 123–130.
[32] Abella J, Perez J and Englund C et al. SAFEXPLAIN: Safe and explainable critical embedded systems based on AI. In: 2023 Design, Automation and Test in Europe Conference & Exhibition (DATE), 2023, pp. 1–6.
[33] Tambon F, Laberge G and An L et al. How to certify machine learning based safety-critical systems? A systematic literature review. Automated Software Eng 2022; **29**: 38.
[34] Malgieri G and Pasquale F. Licensing high-risk artificial intelligence: Toward ex ante justification for a disruptive technology. Computer Law & Security Review 2024; **52**: 105899.
[35] Ihirwe F, Di Ruscio D, Di Blasio K, Gianfranceschi S and Pierantonio A. Supporting model-based safety analysis for safety-critical IoT systems. J Comput Languages 2024; **78**: 101243.
[36] Al-Hawawreh M, Moustafa N. Explainable deep learning for attack intelligence and combating cyber–physical attacks. Ad Hoc Net 2024; **153**: 103329.
[37] Stettinger G, Weissensteiner P and Khastgir S. Trustworthiness Assurance Assessment for High-Risk AI-Based Systems. IEEE Access 2024; **12**: 22718–22745.
[38] Gaur M and Sheth A. Building trustworthy NeuroSymbolic AI Systems: Consistency, reliability, explainability, and safety. AI Mag 2024; **45**: 139–155.
[39] Wang H, Shao W and Sun C et al. A Survey on an emerging safety challenge for autonomous vehicles: Safety of the intended functionality. Engineering 2024; **33**: 17–34.
[40] Ahamad S and Gupta R. Uncertainty modelling in performability prediction for safety-critical systems. Arab J Sci Eng 2024: 1–15, https://doi.org/10.1007/s13369-024-09019-0.

[41] Schneeberger D, Röttger R and Cabitza F et al. The tower of babel in explainable artificial intelligence (XAI). In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. Springer Nature, 2023, pp. 65–81.
[42] Seed W and Omlin C. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. Knowl Based Syst 2023; **263**: 110273.
[43] Hassija V, Chamola V and Mahapatra A et al. Interpreting black-box models: a review on explainable artificial intelligence. Cog Comput 2024; **16**: 45–74.
[44] Tursunalieva A, Alexander DL and Dunne R. Making sense of machine learning: A Review of interpretation techniques and their applications. Appl Sci 2024; **14**: 496.
[45] Ali S, Abuhmed T and El-Sappagh S et al. Explainable artificial intelligence (XAI): What we know and what is left to attain trustworthy artificial intelligence. Inf Fus 2023; **99**: 101805
[46] Saranya A and Subhashini R. A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends. Decis Anal J 2023; **7**: 100230.
[47] Das A and Rad P. Opportunities and challenges in explainable artificial intelligence (xai): A survey, ArXiv preprint [arXiv: 2006.11371], 2020.
[48] Schwalbe G and Finzel B. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. Data Mining Knowl Discov 2023; **38**: 3043–3101.
[49] Guidotti R, Monreale A and Ruggieri S et al. A survey of methods for explaining black box models. ACM Comput Surv (CSUR) 2018; **51**: 1–42.
[50] Islam MR, Ahmed MU, Barua S and Begum S. A systematic review of explainable artificial intelligence in terms of different application domains and tasks. Appl Sci 2022; **12**: 1353.
[51] Arrieta AB, Díaz-Rodríguez N and Del Ser J et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fus 2020; **58**: 82–115.
[52] Mittelstadt B, Russell C and Wachter S. Explaining explanations in AI. In: Proceedings of the conference on fairness, accountability, and transparency 2019, pp. 279–288.
[53] Cao S, Sun X and Widyasari R et al. A Systematic literature review on explainability for machine/deep learning-based software engineering research, ArXiv preprint [arXiv: 2401.14617], 2024
[54] Greisbach A and Klüver C. Determining feature importance in self-enforcing networks to achieve explainable AI (xAI). In: Proceedings 32 Workshop Computational Intelligence, Karlsruhe, KIT Scientific Publishing, 2022, pp. 237–256.
[55] Li M, Sun H, Huang Y and Chen H. Shapley value: from cooperative game to explainable artificial intelligence. Auton Intell Syst 2024; **4**: 1–12.
[56] Atakishiyev S, Salameh M, Yao H and Goebel R. Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions, IEEE Access, 2024.
[57] Minh D, Wang HX, Li YF and Nguyen TN. Explainable artificial intelligence: a comprehensive review. Artif Intell Rev 2022; **55**: 3503–3568
[58] Sharma NA, Chand RR and Buksh Z et al. Explainable AI frameworks: Navigating the present challenges and unveiling innovative applications. Algorithms 2024; **17**: 227.
[59] Dwivedi R, Dave D and Naik H et al. Explainable AI (XAI): Core ideas, techniques, and solutions. ACM Comput Surv 2023; **55**: 1–33
[60] Sanneman L and Shah JA. The situation awareness framework for explainable AI (SAFE-AI) and human factors considerations for XAI systems. Int J Human–Comput Inter 2022; **38**: 1772–1788.
[61] Nannini L, Balayn A and Smith AL. Explainability in ai policies: A critical review of communications, reports, regulations, and standards in the EU, US, and UK. In: Proceedings of the 2023 ACM Conference on fairness, accountability, and transparency 2023, 1198–1212.
[62] Rech P. Artificial neural networks for space and safety-critical applications: Reliability issues and potential solutions. IEEE Transactions on Nuclear Science, 2024.
[63] Petkovic D. It is Not "Accuracy vs. Explainability"–We need both for trustworthy AI systems. IEEE Trans Technol Soc 2023; **4**: 46–53.
[64] Wang Y and Chung SH. Artificial intelligence in safety-critical systems: a systematic review. Indus Manag Data Syst 2022; **122**: 442–470.
[65] Cabitza F, Campagner A and Malgieri G et al. Quod erat demonstrandum? – Towards a typology of the concept of explanation for the design of explainable AI. Exp Syst Appl 2023; **213**: 118888.
[66] Baron C and Louis V. Framework and tooling proposals for Agile certification of safety-critical embedded software in avionic systems. Comput Indus 2023; **148**: 103887.
[67] Guiochet J, Machin M and Waeselynck H. Safety-critical advanced robots: A survey. Robot Auton Syst 2017; **94**: 43–52.
[68] Gaurav K, Singh BK and Kumar V. Intelligent fault monitoring and reliability analysis in safety–critical systems of nuclear power plants using SIAO-CNN-ORNN. Multimedia Tools Appl 2024; **83**: 61287-61311.

[69] Rodvold DM. A software development process model for artificial neural networks in critical applications. In: IJCNN'99, International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339) 1999, Vol. 5, pp. 3317–3322.
[70] Eilers D, Burton S, Schmoeller da Roza F and Roscher K. Safety assurance with ensemble-based uncertainty estimation and overlapping alternative predictions in reinforcement learning, 2023.
[71] Weaver R, McDermid J and Kelly T. Software safety arguments: Towards a systematic categorisation of evidence. In: International System Safety Conference, Denver, CO 2002.
[72] Schwalbe G and Schels M. Concept enforcement and modularization as methods for the ISO 26262 safety argumentation of neural networks, 2020.
[73] Chelouati M, Boussif A, Beugin J and El Koursi E-M. Graphical safety assurance case using Goal Structuring Notation (GSN)–challenges, opportunities and a framework for autonomous trains. Reliabil Eng Syst Safety 2023; **230**: 108933.
[74] Fahmy H, Pastore F, Briand L and Stifter T. Simulator-based explanation and debugging of hazard-triggering events in DNN-based safety-critical systems. ACM Trans Softw Eng Methodol 2023; **32**: 1–47.
[75] Ahmad K, Abdelrazek M and Arora C et al. Requirements engineering for artificial intelligence systems: A systematic mapping study. Inf Softw Technol 2023; **158**: 107176.
[76] Klüver C and Klüver J. Self-organized learning by self-enforcing networks. In: Advances in Computational Intelligence: 12th International Work-Conference on Artificial Neural Networks, IWANN 2013, LNCS 7902, Springer, 2013, pp. 518–529.
[77] Zinkhan D, Greisbach A, Zurmaar B, Klüver C and Klüver J. Intrinsic explainable self-enforcing networks using the icon-d2-ensemble prediction system for runway configurations. Eng Proc 2023; **39**: 41.
[78] Klüver C, Werner C, Nowara P, Castel B and Israel R. Self-enforcing networks for monitoring safety-critical systems: A prototype development. In: Klüver C, & Klüver J (eds.) *New algorithms for practical problems*: Springer Vieweg, 2025 (in German).
[79] Figiel, A. and Klačková, I. Safety requirements for mining systems controlled in automatic mode. Acta Montan Slovaca 2020; **25**
[80] Galy, B. and Giraud, L. Risk mitigation strategies for automated current and future mine hoists. Saf Sci 2023; **167**: 106267
[81] Ferrucci F. Design and implementation of the safety system of a solar-driven smart micro-grid comprising hydrogen production for electricity & cooling co-generation. Int J Hydrogen Energy 2024; **51**: 1096–1119.
[82] Shapley LS. A value for n-person games. In: Contributions to the Theory of Games (AM-28), Princeton University Press, 1953, Vol. 2, pp. 307–318.

**Christina Klüver** is a private Lecturer in Soft Computing at the University of Duisburg-Essen, and CEO of the own consulting company for Artificial Intelligence and Artificial Life. As a member of the CoBASC Research Group, she has developed new algorithms in these areas together with Jürgen Klüver, such as the Self-Enforcing Network (SEN), the Regulatory Algorithm (RGA), and the Algorithm for Neighborhood Generating (ANG). The main research is the computer-based analysis of technical, social and cognitive complexity. In addition, she is an active member of the VDI/VDE-GMA FA 1.13 Neural Networks in Sensor Data Processing.

**Anneliesa Greisbach** is part of the CoBASC Research Group and a Senior Consultant at Campana & Schott focused on IT Strategy, Data & AI Strategy and Management. Her research interests include Explainable AI and the Self-Enforcing Network (SEN). She has developed the Explainable AI component and various visualizations for the Self-Enforcing Network. The Explainable AI component with SEN has been included in a research project for recommending the runway selection at Frankfurt Airport.

**Michael Kindermann** is currently *Head of Functional Safety* at Pepperl+Fuchs SE and as such responsible for Functional Safety Management and related Processes and Standardization. He is an active Member of the National and International Standardization Committees for Functional Safety and as such leading the German Mirror Committees for Safe Software GAK 914.0.3 and Functional Safety and AI in AK 914.0.11 for the DKE in Offenbach, Germany.

**Bernd Püttmann** is currently Deputy Specialist Manager for Cybersecurity Assessments and Senior Assessor for Functional Safety and Cybersecurity at TÜV NORD CERT GmbH. His research interests include the use of artificial intelligence algorithms in the area of functional safety-critical and cybersecurity-related systems.