UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Tecnica e Gestione dei Sistemi Industriali

Corso di Laurea in Ingegneria Meccatronica

L'Intelligenza Artificiale nelle Funzioni di Sicurezza delle Macchine

Relatore: Prof. Dainese Diego

Laureando: Batoul Kalout

Indice

Per prima cosa, vorrei ringraziare il mio relatore Prof. DAINESE DIEGO, per i suoi preziosi consigli e per la sua disponibilità. Grazie per avermi fornito spunti fondamentali nella stesura di questo lavoro e per avermi indirizzato nei momenti di indecisione.

ABSTRACT

Non si può negare che nel corso degli ultimi anni l'intelligenza artificiale abbia ampliato il proprio ambito per coprire numerosi settori, dai motori di ricerca sul web alle automobili automatizzate. Tuttavia, e come tutte le tecnologie sviluppate dagli umani è suscettibile a errori, rendendo necessaria l'adozione di uno standard per guidare lo sviluppo del software IA nell'ambito delle funzioni di sicurezza, come modo per governare tale integrazione. In questo documento analizzeremo i punti discussi dallo standard ISO/IEC 5469: Functional Safety and AI systems, in cui affronteremo i rischi associati ai sistemi IA e come affrontarli per garantire un adeguato livello di sicurezza che consenta loro di essere implementati nelle funzioni di sicurezza delle macchine. Inoltre, sarà analizzata la fase di verifica e validazione di tali sistemi, fornendo indicazioni agli sviluppatori su come raggiungere sistemi sicuri e affidabili.

Introduzione

Secondo la definizione fornita dal Parlamento europeo [4], l'intelligenza artificiale (AI) è quella forma di intelligenza che una macchina o un sistema può possedere, consentendole di prendere decisioni al fine di raggiungere obiettivi specifici. A differenza dei sistemi basati su software tradizionali, che operano esclusivamente sotto condizioni predefinite, i sistemi fondati sull'intelligenza artificiale sono in grado di affrontare situazioni al di fuori del set di condizioni per cui sono stati progettati o programmati.
La forma conosciuta oggi dell'intelligenza artificiale è il risultato di un lungo percorso di esperienze, e ricerca, nonché da un periodo di regressione. Infatti, dal 1974 al 1980, l'intelligenza artificiale ha attraversato una fase in cui l'interesse per questo campo è diminuito a causa della mancanza di progressi tecnologici nell'epoca. Tuttavia, successivamente a quel periodo e fino ai giorni nostri, l'intelligenza artificiale ha conosciuto una crescita significativa, raggiungendo ora un livello tale da poter essere integrata per garantire la sicurezza dei macchinari.
Un esempio significativo di questa integrazione è rappresentato *dall'analisi predittiva*, la quale si basa principalmente sulla capacità di anticipare le tendenze future attraverso l'analisi dei dati derivanti da osservazioni e problematiche passate. Tale approccio consente l'adozione di misure preventive, contribuendo in tal modo a mitigare i potenziali esiti avversi. In effetti, quando un sistema predittivo acquisisce informazioni relative a incidenti precedenti avvenuti nel contesto lavorativo, è in grado di analizzarli e di stabilire in "che modo" e "per quali motivi" si siano verificati. Questo approccio consente una gestione del rischio più sofisticata e sicura. Ad esempio, la "General Motors" ha adottato un programma avanzato di intelligenza artificiale che facilita l'analisi in tempo reale dei dati provenienti dai veicoli, con l'obiettivo di anticipare potenziali pericoli prima che si manifestino; le statistiche indicano che, nell'arco di un anno, gli incidenti sul posto di lavoro sono diminuiti di circa il 25% [60]. Alla luce di quanto esposto, l'intelligenza artificiale si profila come una risorsa promettente nel campo della sicurezza.
Ulteriori applicazioni dell'intelligenza artificiale nelle funzioni di sicurezza comprendono la *manutenzione predittiva e l'ispezione automatizzata,* che consentono il rilevamento di segni di danno, perdite d'olio, rischi elettrici o altre tipologie di malfunzionamenti. Tale approccio offre ai professionisti della manutenzione l'opportunità di affrontare tempestivamente le problematiche prima che queste si manifestino.

Però è necessario sottolineare, che i sistemi di intelligenza artificiale non sono ancora completamente affidabili; in effetti, si sono verificati incidenti in cui questa tecnologia altamente sofisticata ha manifestato dei fallimenti, talvolta con esiti fatali. Un esempio significativo è rappresentato dall'incidente del 2018, quando un veicolo a guida autonoma sviluppato da Uber investì e uccise un pedone in Arizona [13]. Tale evento ha sollevato interrogativi riguardo alla trasparenza della tecnologia di guida autonoma di Uber e alla capacità dell'automobile di percepire adeguatamente l'ambiente circostante.

Alla luce di quanto esposto, l'intelligenza artificiale si profila come una risorsa promettente nel campo della sicurezza. Infatti, nel nuovo regolamento macchine UE 2023/1230, e che andrà in vigore nel 2027, è stato aggiunto nell'allegato I di esso, tra i componenti per cui va applicata una delle procedure di valutazione di conformità, i componenti di sicurezza che adottano un natura parzialmente o completamente autoevolutivo e che garantiscono funzioni di sicurezza. E riguardo questo punto, è necessario avere una linea guida standardizzata per verificare e convalidare la conformità del sistema ai requisiti di sicurezza.

Tutto ciò sottolinea l'urgenza di stabilire una normativa che consenta di valutare l'integrazione di questa tecnologia, attualmente poco diffusa, nelle funzioni di sicurezza delle macchine. A tal proposito, è stata adottata la norma ISO/IEC 5469 [37], che costituirà il nostro punto di discussione nella presente tesi.

1 L’ISO/IEC 5469 e Funzioni di sicurezza

1.1 ISO/IEC 5469: CENNI STORICI E NECCESSITA DI TALE Norma:

La presente norma internazionale è stata pubblicata il **08 gennaio 2024**, a seguito di una proposta approvata il **09 giugno 2020**. Prima della pubblicazione di tale norma, l'integrazione dei sistemi di intelligenza artificiale nelle funzioni di sicurezza dei macchinari si basava su un insieme di norme generali non specifiche [37]. Esempi di tali norme di riferimento includono:

• **EN ISO 13849-1:2023(en) Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design;** applicabile alle parti dei sistemi di comando destinate a fornire funzioni di sicurezza [21];

• **IEC 61508: Functional Safety of E/E/PE;** che tratta i requisiti da seguire per l'intero ciclo di vita dei sistemi elettrici, elettronici ed elettronici programmabili relativi alla sicurezza;

• **EN ISO 10218-1: Requisiti di Sicurezza per Robot Industriali**; che fornisce linee guida per la progettazione e l'utilizzo sicuro dei robot industriali;

• **EN ISO 12100: Sicurezza del macchinario - Principi generali di progettazione - Valutazione del rischio e riduzione del rischio**; che definisce le linee guida per l'analisi e la gestione dei rischi associati ai macchinari.

Pertanto, prima dell'entrata in vigore della presente norma, non esisteva alcuna regolazione che tenesse conto delle differenze e delle caratteristiche peculiari dei sistemi di intelligenza artificiale rispetto ai sistemi software tradizionali. La principale distinzione risiede nel fatto che i sistemi di intelligenza artificiale si fondano su algoritmi di apprendimento automatico, piuttosto che su un insieme di istruzioni predefinite; essi sono in grado di operare e prendere decisioni autonomamente, anche quando si trovano a lavorare in condizioni a loro sconosciute.

È da notare anche che nel nuovo regolamento macchine UE 2023/1230 che andrà in vigore il 2027, i componenti di sicurezza che adottano tecnologie AI, parzialmente o interamente, sono pertinenti a tale regolazione; in particolare

Detto ciò, è fondamentale chiarire che esistono tre categorie di algoritmi impiegati dai sistemi di apprendimento automatico [3]:

- Algoritmi di Apprendimento Supervisionato. Gli algoritmi in questione sono essenzialmente progettati per stabilire una relazione tra l'input e l'output, basandosi sulle informazioni acquisite durante la fase di addestramento. Questi strumenti trovano ampia applicazione nei sistemi di previsione. A tal fine, è necessaria l'acquisizione di dati di apprendimento etichettati, ovvero dati che includono esempi corretti di input e output.

- Algoritmi di Apprendimento Non Supervisionato .Questi algoritmi sono impiegati in assenza di dati etichettati e vengono comunemente utilizzati per identificare pattern e raggruppamenti.

- Algoritmi di Apprendimento Rinforzato. Si tratta di algoritmi che apprendono attraverso l'interazione con l'ambiente circostante, utilizzando un approccio fondato su un sistema di ricompense e punizioni, con l'obiettivo di massimizzare le ricompense ottenute.

Dalle considerazioni esposte, è possibile dedurre che i sistemi di intelligenza artificiale si distinguono per le loro caratteristiche in funzione dell'algoritmo adottato, il quale rappresenta una peculiarità rilevante di tali tecnologie. Ciò mette in evidenza l'importanza della presente normativa e la sua necessità, soprattutto in un contesto attuale in cui i progettisti mirano a implementarla in diversi settori, data la sua notevole potenza.

Nella presente normativa hanno cercato di discutere la sicurezza funzionale, l'integrazione dell'AI nei sistemi di sicurezza elettroniche, gli elementi delle tecnologie AI , le loro caratteristiche e rischi relativi, la fase di verificazione e validazione e sue requisite, misure per la riduzione dei rischi derivanti da loro, e alcune metodologie da seguire.

1.2 Sicurezza Funzionale e Funzioni di Sicurezza

Come stabilito dalla norma I*EC61508-4:2010: Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 4: Definitions and abbreviations,* [32]la funzione di sicurezza è definita come una funzione progettata per garantire il mantenimento di una condizione sicura in presenza di un rischio specifico che comporta danni alla salute dei lavoratori. Tale funzione è realizzata mediante un sistema elettrico, elettronico o programmabile, oppure attraverso altri sistemi di mitigazione del rischio. È opportuno sottolineare che una funzione di sicurezza può essere implementata con l'intento di eliminare un rischio che possa avere conseguenze economiche significative.

Un sistema che svolge una funzione di sicurezza deve possedere un elevato grado di affidabilità, poiché un malfunzionamento potrebbe condurre a una situazione pericolosa. Vi sono diverse tipologie di tali funzioni, tra cui l'arresto di emergenza, il monitoraggio della velocità sicura e la disattivazione sicura della coppia. L'implementazione di queste funzioni dipende in larga misura dall'analisi del rischio, perché ciascuna funzione è progettata per ridurre e gestire un rischio specifico. È importante sottolineare che l'obiettivo di tali sistemi non consiste nell'eliminare completamente i rischi, poiché tale ideale è irrealizzabile; piuttosto, lo scopo è raggiungere un livello di rischio che possa essere considerato tollerabile [31].

Segue questo, la sicurezza funzionale e quella disciplina concerna a verificare la giusta progettazione dei sistemi che svolgano funzioni di sicurezza; i requisiti di tali sistemi è definito quindi dalla sicurezza funzionale.

Nell'ambito della sicurezza funzionale, si distingue tra due categorie di guasti: quelli *hardware randomici* e quelli *sistematici*. I guasti sistematici sono principalmente attribuibili a errori nelle fasi di assemblaggio, installazione e produzione, nonché all'uso improprio del sistema. Inoltre, l'impiego di un sistema in un contesto inadeguato può anch'esso determinare un guasto sistematico [32]. È fondamentale sottolineare che tali guasti possono essere rimossi

esclusivamente attraverso una revisione delle procedure di progettazione, produzione, o altri fattori significativi.

Al contrario, guasti hardware randomici risultano più complessi da eliminare o persino da identificare, poiché possono manifestarsi improvvisamente e in qualsiasi momento, e quando si manifestano, comportano uno o più degradi della capacità del componente di svolgere il suo scopo [32].

Prima dell'emergere dei sistemi software, quando le funzioni di sicurezza erano prevalentemente associate all'hardware, l'obiettivo della sicurezza funzionale era quello di contenere gli effetti dei guasti hardware casuali. Con l'avvento del software, è emersa la necessità di ridurre gli impatti derivanti dai guasti sistematici che possono verificarsi durante lo sviluppo del sistema, divenendo così un obiettivo fondamentale della sicurezza funzionale.

Poiché i sistemi di intelligenza artificiale sono maggiormente influenzati dall'analisi dei dati piuttosto che dalle specifiche, uno degli aspetti trattati da tale norma riguarda la descrizione delle misure per la gestione dei guasti sistematici, tenendo conto delle peculiarità di questi sistemi. In effetti, **l'appendice A** esamina l'applicabilità della norma IEC 61508-3:2010, che tratta la sicurezza funzionale e le modalità di gestione dei guasti sistematici nei sistemi non basati sull'intelligenza artificiale. Tra i metodi adottati dalla norma ISO/IEC 5469 si annoverano: metodi formali, metodi semi-formali e strumenti di specifica assistita da computer, con l'obiettivo generale di ridurre l'ambiguità del sistema e aumentare la sua trasparenza, facilitando così l'interpretazione e il rilevamento degli errori presenti [37].

2 AI NEI SISTEMI DI SICUREZZA E/E/PE

Definizione : E/E/EP "electrical / electronical / programmable electronical": Il termine, come stabilito dalle normative internazionali tecniche, si riferisce a tutti i sistemi che operano secondo principi elettrici*[32]*.

È fondamentale sottolineare che tutti i sistemi E/E/EP di sicurezza devono conformarsi a un insieme di proprietà generali al fine di garantire l'implementazione delle misure destinate alla gestione e riduzione dei rischi. Tali caratteristiche sono state delineate in numerose norme tecniche antecedenti alla norma *ISO/IEC 5469*. Tra queste, si annoverano *IEC 61508-3:2010[31], ISO 26262[35]*, *IEC 62061[33]* e *ISO 13849[21]*. È rilevante osservare che tali proprietà vengono selezionate in relazione alle specifiche applicazioni e tecnologie impiegate nella progettazione di questi sistemi. Questo aspetto sarà ulteriormente analizzato considerando le peculiarità dell'intelligenza artificiale.

2.1 Classificazione dei Sistemi AI

I sistemi AI seguono tecnologie diversi da questi adottati dagli soggetti delle norme elencate, per la quale la norma in studio *(ISO/IEC 5469)* ha fornito una schema di classificazione dell'applicabilità delle technologie in intelligenza artificiale nei sistemi E/E/PE relativi alla sicurezza. Tale classificazione si basa principalmente su due concetti fondamentali : **l'Applicazione del AI** ed il **Livello di Utilizzo**, e **il Classe del AI.**

2.1.1 l'Applicazione del AI ed il Livello di Utilizzo**:**

Questa classificazione ha identificato sei livelli di utilizzo, da A a B, in funzione dell'impiego della tecnologia AI nel sistema di sicurezza E/E/PE e della sua applicazione nella fase decisionale.

I primi livelli di utilizzo **A1** e **A2** si riferiscono ai sistemi di sicurezza elettrici in cui vengono integrate tecnologie di intelligenza artificiale [37]. La distinzione principale risiede nel fatto che nel sistema di livello A1 è prevista la possibilità di prendere decisioni in modo autonomo, mentre nel sistema di livello A2 tale possibilità non è presente. Un esempio pertinente è rappresentato dal "Hazard detector", un sistema di monitoraggio basato su algoritmi di Machine Learning, dedicato all'analisi delle variazioni nell'ambiente circostante della macchina, come le fluttuazioni della temperatura o la rilevazione di gas tossici, tra gli altri aspetti [57].
I livelli **B1** e **B2** vengono attribuiti quando la tecnologia di autoapprendimento è impiegata esclusivamente a livello della produzione del sistema e non è quindi integrata nel sistema di sicurezza [37]. L'unica differenza tra questi due livelli intermedi risiede nel fatto che il primo livello si riferisce alla possibilità che il sistema possa prendere decisioni in modo autonomo, mentre nel secondo livello tale possibilità non sussiste. Un sistema di validazione che utilizza la tecnologia AI ha un livello di utilizzo B, con eventuale classificazione in un livello B1 o B2 a seconda della caratteristica decisionale [57].
Si stabilisce poi un livello di utilizzo **C** quando la tecnologia dell'intelligenza artificiale non è integrata nella funzione di sicurezza, ma può esercitare un impatto indiretto sul sistema E/E/PE [37]. Un esempio pertinente è l'integrazione di sistemi basati su tecnologie AI per ottimizzare il consumo energetico; nel caso in cui tali sistemi siano impiegati in macchinari dedicati a processi chimici caratterizzati da un elevato consumo energetico durante le ore di punta, una riduzione aggressiva della domanda energetica potrebbe influenzare il sistema di

raffreddamento [57]. Ciò comporterebbe l'attivazione dei protocolli del sistema di sicurezza volti a mitigare l'elevata temperatura generata.
In conclusione, la norma ISO/IEC 5469 ha definito il livello di utilizzo **D**, che si applica ai sistemi nei quali l'intelligenza artificiale non è integrata nel sistema di sicurezza e non influisce in alcun modo, nemmeno indirettamente, sulla funzione di sicurezza stessa, poiché è opportunamente isolata da essa [37]. Un esempio di tale situazione è rappresentato dall'impiego dell'AI per la fornitura e generazione di disegni destinati alle stampanti 3D [57].

2.1.2 CLASSE DELLA TECHNOLOGIA AI

Questa classificazione si fonda sulla verifica della conformità del sistema di intelligenza artificiale rispetto al set di caratteristiche stabilite dalle normative tecniche in materia di sicurezza, al fine di accertarne l'integrazione nei sistemi di sicurezza E/E/PE.
Secondo la norma in questione, sono stati identificati tre classi : I, II, III [37].
È fondamentale osservare che la classe della tecnologia si sviluppa in concomitanza con l'aumento della sua complessità; infatti, maggiore è l'articolazione della tecnologia, più arduo risulta il processo di validazione.
Quando la tecnologia dell'intelligenza artificiale viene sviluppata, testata e convalidata in conformità alle misure e ai metodi delineati dalle normative tecniche relative alle funzioni di sicurezza, essa è classificabile come **Classe I**. I sistemi appartenenti a questa categoria sono generalmente fondati su "*Narrow AI*" o sono progettati per eseguire un compito specifico, solitamente semplice, alla volta; esempi rappresentativi di questa classe includono "Siri" e tutti gli assistenti vocali, nonché i chatbot.
Le tecnologie classificate nella **Classe II** sono quelle per le quali, nel processo di sviluppo, non è sufficiente fare riferimento unicamente alle norme tecniche pertinenti per convalidare la conformità. È necessaria l'implementazione di ulteriori misure per raggiungere tale obiettivo, poiché queste tecnologie si fondano sull'intelligenza artificiale generale (*General AI*), ovvero su algoritmi che consentono alla macchina di emulare un'ampia gamma di funzioni cognitive umane, affrontando compiti complessi e articolati. La capacità del sistema di prendere decisioni in autonomia rappresenta, ad esempio, una caratteristica distintiva di tali sistemi.
Infine, viene attribuita la **Classe III** alle tecnologie che non possono essere associate alle normative tecniche di sicurezza e che non soddisfano le proprietà necessarie attraverso le misure e le tecniche in esse indicate. Tali sistemi sono comunemente definiti come *"Strong AI"* o come sistemi caratterizzati da un elevato grado di autonomia, come nel caso dei "veicoli completamente autonomi", i quali possono esercitare un impatto significativo in termini di sicurezza, poiché presentano un alto livello di imprevedibilità.

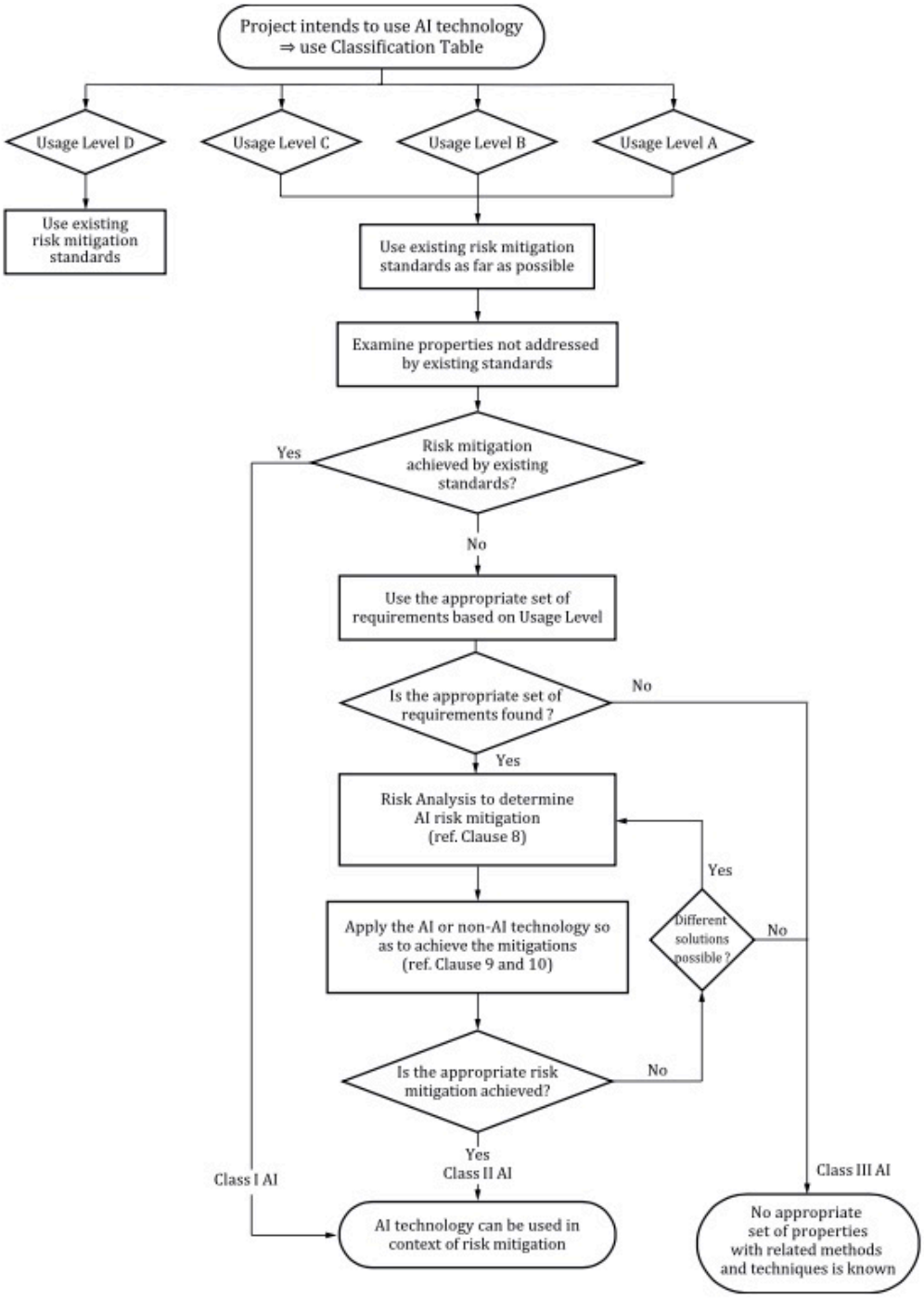


Figura1 Diagramma di Flusso per la classificazione delle tecnologie AI

Per facilitare la selezione e la classificazione delle tecnologie da integrare nei sistemi di sicurezza, la norma ISO/IEC 5469[37] ha fornito una mappa generale, come illustrato nella Figura 1. Dalla suddetta mappa si evince che la classificazione primaria è basata sul livello di utilizzo; nel caso in cui il livello di utilizzo sia D, poiché non esercita un effetto significativo sulla sicurezza, è sufficiente fare riferimento alle norme tecniche esistenti riguardo alle misure di mitigazione e gestione del rischio, senza necessità di considerare la classe a cui appartiene tale tecnologia. Al contrario, qualora il livello di utilizzo rientri in uno degli altri livelli (A, B o

C), diventa imprescindibile comprendere a quale classe appartenga questa tecnologia. In primo luogo, è essenziale identificare quali proprietà non siano soddisfatte in base alle norme tecniche vigenti. Una volta effettuata tale analisi, si procederà a valutare se sia possibile raggiungere uno stato sicuro applicando le tecniche e le misure delineate dalle normative esistenti, condizione che rappresenta un requisito sufficiente per classificare la tecnologia come *Classe I*. Qualora tale requisito non venga soddisfatto, è necessario procedere all'analisi del set di requisiti in relazione al livello di utilizzo. Se questo set non viene raggiunto, considerando le misure e le tecniche disponibili, la tecnologia verrà classificata nella *Classe III*. Pertanto, si può concludere che, per quanto concerne le tecnologie di *Classe III* e fino al momento della pubblicazione della norma in questione, non sussistono informazioni adeguate a conseguire uno stato accettabile in termini di riduzione del rischio. Qualora il set indicato sia stato definito, si procede con l'analisi del rischio e l'applicazione delle misure specifiche destinate alla loro mitigazione, che costituiscono un elemento fondamentale di tale normativa e su cui concentriamo progressivamente la nostra attenzione. Attraverso l'impiego delle tecniche necessarie per la riduzione del rischio, si valuta se è stato raggiunto un livello di sicurezza accettabile; qualora ciò venga confermato, la tecnologia viene classificata sotto la *Classe II.* Qualora le misure adottate non risultassero adeguate alla mitigazione dei rischi identificati, è necessario esaminare se l'implementazione di ulteriori soluzioni possa condurre al raggiungimento di tale obiettivo. In tal caso, una volta verificata, la tecnologia si classificherebbe come Classe II anziché Classe III.

È fondamentale sottolineare che, sulla base di questa analisi, le tecnologie che vengono classificate come Classe III non possono essere impiegate nel contesto della riduzione del rischio in quanto mancano le informazioni necessarie.

3 GLI ELEMENTI DELLE TECNOLOGIE IN AI E 3 STAGE PRINCIPLE

Per una comprensione esaustiva del concetto di intelligenza artificiale e della sua integrazione nelle funzioni di sicurezza, è fondamentale analizzare gli elementi costitutivi delle tecnologie in questo settore. In effetti, per garantire un sistema completo e assolutamente sicuro, tali elementi devono conformarsi a un insieme di requisiti specifici. A tal riguardo, la norma ISO/IEC 5469 ha fatto riferimento alla norma pertinente che è rappresentata dalla ISO/IEC 22989:2022 [36], che fornisce un'analisi approfondita dei concetti legati all'intelligenza artificiale.

3.1 I componenti dei modelli in AI

In particolare, una tecnologia di intelligenza artificiale si compone essenzialmente di sei elementi, i quali sono dettagliati nella tabella sottostante, derivante dalla norma ISO/IEC 5469:

AI technology element
AI services
Machine learning — Model development and use — Tools — Data for machine learning
Engineering — Knowledge based on domain experience — Tools
Cloud and edge computing and big data and data sources
Resource pool-compute, storage, network
Resource management-resource provisioning

Figura 2 Elementi di una Tecnologia AI

1. **Servizi AI**

   Un servizio di intelligenza artificiale è concepito come un'entità o un'organizzazione che offre servizi di AI, i quali possono essere utilizzati direttamente o integrati in sistemi che includono anche componenti non-AI [36]. Esempi rilevanti comprendono i chatbot che utilizzano i Natural Language Processing (NLP) il che evidenzia la capacità della tecnologia AI di analizzare e comprendere il linguaggio umano, sia esso scritto che parlato [80]. Questi servizi devono dimostrare un elevato grado di affidabilità quando impiegati nelle funzioni di sicurezza, le quali dipendono dai dati utilizzati nei modelli, aspetto che sarà oggetto di un'analisi approfondita in seguito.

2. **Machine Learning**

L'Apprendimento Automatico (ML) rappresenta un ambito di studio scientifico dedicato agli algoritmi e ai modelli statistici utilizzati nei sistemi informatici, consentendo loro di eseguire compiti senza necessità di programmazione esplicita per tali operazioni [15]. I modelli si fondano su quattro categorie di dati, ovvero: dati di addestramento, dati di validazione, dati di test e dati di produzione; in effetti, tali modelli sono tenuti a elaborare i dati al fine di identificare i corretti "schemi" necessari per formulare soluzioni adeguate. Questi dati vengono raccolti e analizzati attraverso strumenti software quali il "Data pre-processing", metodi di ottimizzazione e matrici di valutazione. È importante sottolineare che l'affidabilità della tecnologia AI è fortemente influenzata dalla robustezza del ML all'interno del sistema. Infatti, i sistemi ML presentano un insieme significativo di rischi che devono essere gestiti affinché si possa giungere a un sistema stabile ed efficiente. Tra tali rischi si annoverano: la mancanza di trasparenza e chiarezza, bias e azioni non intenzionali, overfitting e fallimenti generalizzati, oltre alla vulnerabilità rispetto ai rischi informatici... Nella sezione successiva verranno trattati in maniera approfondita tutti questi concetti e le relative modalità di gestione.

3. **Engineering**

Questo elemento è sostanzialmente ancorato alle competenze degli ingegneri dei sistemi software, la cui valutazione avviene in base alle conoscenze ed esperienze accumulate in un settore specifico [36]. Pertanto, si può dedurre che l'abilità del progettista del sistema gioca un ruolo cruciale nella completezza e nella conformità del progetto stesso.

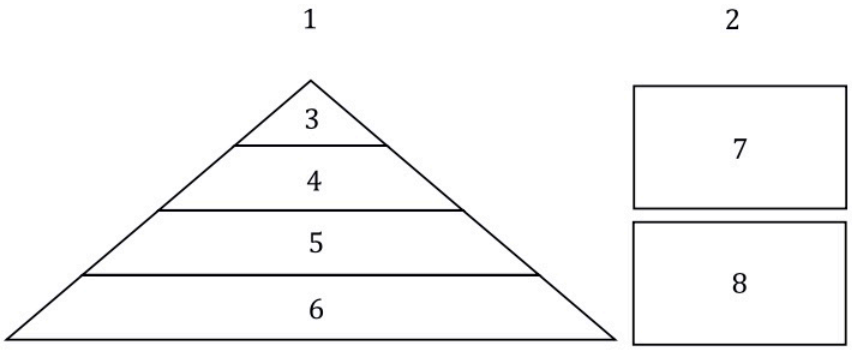
4. Il cloud computing, l'edge computing e i big data come fonti di dati

I dati rappresentano un elemento cruciale nelle tecnologie di intelligenza artificiale. I big data si configurano come un ampio insieme di informazioni caratterizzato da volume, varietà, velocità e variabilità, consentendo ai progettisti di disporre di un vasto repertorio per il training e l'ottimizzazione del sistema [36]. Le fonti dei dati possono includere sondaggi, osservazioni registrate, statistiche derivanti da ricerche, sensori, tra le altre. Come già evidenziato in precedenza, il dataset riveste un ruolo significativo in relazione alla prestanza e alla potenza di un sistema; tale aspetto sarà ulteriormente approfondito nell'analisi che segue.

5. Resource pool

Un sistema di intelligenza artificiale (AI) si fonda su un insieme diversificato di risorse, comprendenti dati e risorse computazionali come CPU, GPU e TPU, che rivestono un'importanza fondamentale per l'elaborazione del sistema hardware e software [36]. Tali risorse sono caratterizzate in base ai requisiti specifici del sistema di apprendimento automatico (ML). Inoltre, vi sono risorse dedicate alla memoria e al networking. Tutte queste risorse eterogenee devono essere gestite e coordinate autonomamente attraverso meccanismi di provisioning. La completezza e l'affidabilità del sistema AI dipendono significativamente dalla robustezza di tali risorse; in particolare, le risorse dati devono garantire un dataset completo e rappresentativo della realtà, con particolare attenzione alla qualità elevata. L'inadeguatezza nel soddisfare uno qualsiasi di questi requisiti può condurre a situazioni insicure; ad esempio, un dataset non sufficientemente rappresentativo potrebbe comportare interpretazioni distorte da parte del sistema, generando così previsioni e soluzioni errate; questo aspetto verrà esaminato più dettagliatamente in seguito. Inoltre, le risorse computazionali, essendo integrate in sistemi real-time, devono possedere caratteristiche di elevata velocità e latenza minima.

Per l'utilizzo di tali elementi si applica una certa gerarchia definita dalla norma ISO/IEC 5469[37]; lo scopo di tale classificazione e quello di definire il set d proprietá che deve essere a ciascun elemento in base alla posizione che si occupa. L'esempio allegato sotto è fornito dalla norma.



Key

1 code/models
2 tools/libraries
3 application graph
4 ML model
5 executable machine code
6 computational hardware
7 ML framework, code/graph/ML compilers
8 libraries

Figura 3La gerarchia degli elementi di una tecnologia AI

Gli elementi che si trovano nei livelli più alti come, ad esempio, i modelli e strumenti devono essere conformi ad un set di proprietà definita in tale norma come la robustezza, trasparenza, chiarezza del software…, mentre gli elementi come il code od il hardware sono oggetti della norma ISO/IEC 61508: 2022 in quanto è applicabili a tutti sistemi E/E/PE.

3.2 3 stage realization principle and the acceptance criteria

I sistemi AI si realizzano seguendo un criterio composta di tre passi:

- L'acquisizione dei data
- Induzione della conoscenza

- Elaborazione e generazione di output

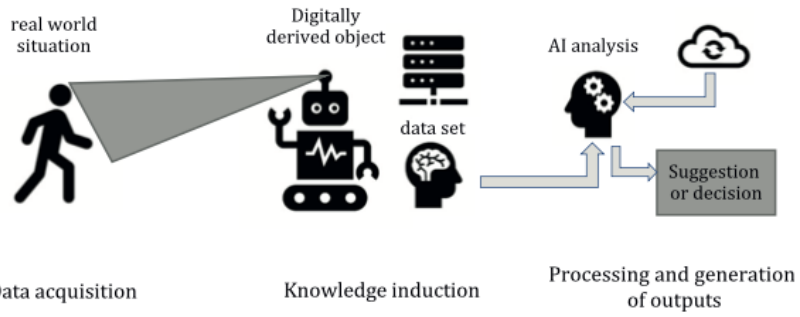


Figura 4 Three stage realization principle

Ogni singolo passaggio deve raggiungere uno stato di accettabilità al fine di garantire la conformità totale del sistema di intelligenza artificiale. Pertanto, viene stabilito un criterio di accettabilità specifico per ciascuno dei tre passaggi. Tale criterio si basa fondamentalmente sulla definizione di un insieme di proprietà che devono essere soddisfatte al termine di ogni fase del processo di sviluppo del sistema. Queste caratteristiche non sono generali, ma piuttosto dipendono dalla funzionalità del sistema e derivano dalle normative tecniche esistenti, in funzione del livello dell'elemento, come già delineato nella gerarchia del sistema AI [37].

Consideriamo, a titolo esemplificativo, il caso dello strumento Monitor e controllore di velocità basato su un sistema AI all'interno di un robot industriale: applichiamo il principio dei tre passaggi e il criterio di accettabilità a questo sistema:

3.2.1 Acquisizione dei dati

Il sistema impiega sensori, telecamere, GPS e radar per raccogliere informazioni sufficienti riguardo alle condizioni della strada, alla velocità del robot e all'ambiente circostante. Questi dati vengono acquisiti in tempo reale, il che richiede, ad esempio, sensori capaci di elaborare le informazioni con alta precisione e tempi di latenza estremamente ridotti [42]. Inoltre, i dati

raccolti devono essere rappresentativi in modo tale da garantire che il sistema operi correttamente anche nelle situazioni più critiche (worst-case scenario). Questo punto è un oggetto della norma in esame, nonché della norma ISO/IEC 22989:2022 [36], poiché entrambe delineano le caratteristiche dei sistemi di intelligenza artificiale e dei dati su cui si fondano. Attraverso l'implementazione dei metodi e delle tecniche specificate in tali norme, è imperativo raggiungere uno stato accettabile riguardo all'acquisizione dei dati.

3.2.2 Induzione delle informazioni

Una volta raccolti i dati, i modelli statistici e gli algoritmi del sistema devono stabilire una relazione significativa per essere in grado di identificare condizioni di rischio e, conseguentemente, determinare quali livelli di velocità possano condurre a situazioni insicure in contesti estremi. In questo contesto, gli algoritmi devono dimostrare la capacità di identificare tali situazioni con elevata precisione e gestire adeguatamente il contesto. Un valido riferimento normativo per questa materia è rappresentato dalla norma ISO/IEC 22989:2022 [36].

3.2.3 Elaborazione dei dati e generazione di output

Con i dati acquisiti e interpretati, il sistema deve produrre una relazione che consenta di determinare in tempo reale la velocità appropriata in qualsiasi condizione operativa. Pertanto, il sistema deve garantire affidabilità e rapidità, includendo anche la capacità di generare output d'emergenza, come ad esempio un arresto immediato nel caso venga rilevata una condizione anomala. Le norme pertinenti per soddisfare tali requisiti sono ISO/IEC 5469 [37] e ISO/IEC 22989 [36].

4 LE CARATTERISTICHE E I RISCHI DEI SISTEMI DI INTELLIGENZA ARTIFICIALE

Come evidenziato nel paragrafo precedente, per progettare un sistema accettabile è imprescindibile definire il set di proprietà che deve possedere. Questo aspetto risulta essere tra i più approfonditi nella normativa vigente [37], poiché analizza anche i rischi associati ai sistemi di intelligenza artificiale e le metodologie per la loro mitigazione.

4.1 Modelli e Algoritmi

Il primo argomento trattato concerne gli algoritmi e i modelli, in quanto, all'interno della gerarchia tecnologica, rappresentano l'elemento più critico sia dal punto di vista della sicurezza che della funzionalità del sistema. Infatti, l'affidabilità del sistema è significativamente influenzata da questo aspetto, poiché determina la capacità dello stesso di convertire gli input acquisiti e analizzati negli output desiderati. Tali modelli sono caratterizzati da un insieme di parametri definiti durante la fase di addestramento del sistema; questi parametri influiscono sul comportamento del modello e, conseguentemente, sulle decisioni e previsioni effettuate dal sistema. In effetti, un modello si configura tipicamente come un'equazione matematica e può assumere forme diverse quali funzioni lineari o relazioni logiche; le reti neurali artificiali (ANN) sono esempi in cui i parametri, come ad esempio i coefficienti, possono essere generati dagli algoritmi di machine learning oppure determinati da ingegneri. Basandosi su tali parametri e sui dati d'input forniti, il modello genera una relazione che determina il comportamento finale. Da quanto esposto, si può dedurre che la sicurezza funzionale del sistema è intrinsecamente legata a due elementi fondamentali: i modelli e i parametri. È essenziale sottolineare che i modelli del sistema privi di parametri sono classificati come modelli software non-AI, evidenziando così l'importanza cruciale di tali elementi.

La norma in studio afferma che per realizzare la sicurezza funzionale dei modelli e parametri nei sistemi AI si fa riferimento alla norma IEC 61508. Tale norma essenzialmente nella parte 3[31] definisce un set di requisisti che devono avere i sistemi software in termini della sicurezza.

Al fine della progettazione di tali modelli i progettisti e ingegneri devono validare e verificare che essi soddisfano un gruppo di requisiti che sono elencati in tanti norme tecniche e sono:

- Completezza
- Correttezza rispetto agli requisiti di sicurezza
- Assenza di ambiguità ed errori intrinseci

- Corretta conoscenza dei requisiti di sicurezza che vengono determinati in base all'analisi del rischio
- Assenza di interferenze avversale di funzioni normali su quelli di sicurezza

Il mancato conseguimento di uno o più requisiti fondamentali può condurre l'intero sistema a una condizione di instabilità. Per analizzare questa problematica, consideriamo il *"safe position monitor",* un sistema dedicato al monitoraggio della posizione delle parti mobili di un robot industriale o di una macchina operativa, il cui scopo è garantire che tali componenti non effettuino movimenti indesiderati né raggiungano posizioni suscettibili di generare situazioni pericolose nell'ambiente lavorativo o di entrare in collisione con altri elementi. Un modello incompleto, in particolari condizioni critiche, può comportarsi in modo errato e non conforme alle aspettative desiderate.

Nel nostro caso studio, un modello **incompleto** potrebbe generare "blind spots" o aree in cui il monitoraggio risulta impossibile a causa dell'incompletezza del modello e dei dati acquisiti; ciò potrebbe portare, ad esempio, a un robot collaborativo all'interno di un'industria a continuare a muoversi verso una posizione potenzialmente conflittuale con il personale umano o altre macchine. Un altro rischio che può essere generato è l'interpretazione non giusta dei data che è stata ricevuta dai sensori di posizione. Questo può anche portare ad una sopravalutazione, o sottovalutazione degli zoni interpretati dal sistema sicuri. Pensiamo di un braccio robotico dedicato a fare una chirurgia, un'estimazione sbagliato della zona sicura può portare a impatti gravi sulla salute della paziente e quindi la completezza di un modello di un sistema di sicurezza critico è un aspetto fondamentale in termini della sicurezza.

Consideriamo ora un sistema che ha degli errori rispetto ai requisiti di sicurezza; un modello **scorretto**. Infatti un modello si considera scorretto rispetto ai requisiti della sicurezza quando si verifica almeno uno di questi "rischi" :

- Bias
- Overfitting
- Mancanza di Robustezza

Per capire tali concetti, e importante parlare dalla data di input, inquanto la correttezza di un modello si basa essenzialmente sulla data che si acquisisce durante la fase di training.

Un modello o sistema è definito "biased" o prevenuto quando nel dataset di input si riscontrano gruppi che sono sovra-rappresentati rispetto ad altri [7,82]. Questa distorsione si traduce in un modello che può funzionare efficacemente nelle condizioni più comuni e frequentemente incontrate, mentre in situazioni nuove, o in quelle precedentemente emarginate durante la fase di addestramento, il comportamento del modello risulta errato e instabile. È opportuno considerare che il sistema oggetto di studio (monitoraggio della posizione sicura) è stato addestrato in ambienti lavorativi con una bassa densità di lavoratori o dove non esiste un significativo contatto con oggetti mobili. Tale sistema, se impiegato in un robot industriale

collaborativo all'interno di un'industria di immagazzinamento, potrebbe generare numerosi problemi e costituire una fonte di rischio sia per il personale che per le altre macchine presenti sul luogo di lavoro, poiché non sarà in grado di analizzare adeguatamente le condizioni circostanti, portando quindi a decisioni errate. Pertanto, i dati devono possedere una diversità tale da rappresentare quanto più possibile lo stato reale. Una strategia efficace anche è quella di implementare delle metriche per la quantificazione di data bias nel dataset [82].

L'over-fitting rappresenta un problema significativo che deve essere evitato. Esso si riferisce alla capacità di un modello di fornire previsioni accurate e corrette esclusivamente per un insieme di dati simile a quello utilizzato durante la fase di addestramento; al contrario, quando il modello si confronta con un nuovo set di dati, non riesce a operare in modo adeguato, risultando così in decisioni errate e inaccurate [6]. La causa principale di questo fenomeno è l'impiego, durante la fase di allenamento, di un insieme di dati eccessivamente limitato, che non contempla tutte le possibili variabili che il sistema potrebbe incontrare [6]. Ad esempio, un sistema di monitoraggio della posizione progettato per operare all'interno di un intervallo specifico di velocità può fallire nel suo funzionamento qualora tale intervallo venga superato, creando pertanto una condizione insicura nell'ambiente lavorativo. Il sistema potrebbe quindi accedere a zone pericolose sulla base delle condizioni ambientali circostanti. Oltre all'importanza dell'utilizzo di dati rappresentativi che coprano ampiamente tutte le eventualità previste, esiste una metodologia nota come "validazione incrociata" [26]. Questa tecnica si fonda sulla suddivisione del dataset disponibile in sottoinsiemi o pieghe, dove uno dei pieghi viene impiegato come set di convalida mentre gli altri vengono utilizzati nella fase di addestramento. La suddetta procedura viene reiterata al fine di consentire la convalida del sistema su tutte le pieghe disponibili; successivamente, i risultati di ciascuna fase di convalida vengono analizzati per conseguire, infine, un sistema in grado di generare output nel modo più preciso possibile e su un ampio insieme di dati.

Un concetto di rilevante importanza è la **robustezza**. Si definisce un sistema come robusto quando possiede la capacità di operare in sicurezza sotto qualsiasi condizione, più precisamente in presenza di variazioni delle circostanze ambientali, anche se tali condizioni non erano state previste durante la fase di addestramento. Inoltre, il sistema dimostra una certa tolleranza verso guasti nel sistema sensoriale o input contraddittori e perturbazioni esterne [19]. È evidente, quindi, che tale concetto o proprietà risulta essenziale nei sistemi di intelligenza artificiale quando sono implementati nelle funzioni di sicurezza. La robustezza del sistema dipende significativamente dalla progettazione del modello. Un aspetto cruciale riguarda i dati utilizzati per l'addestramento: questi devono essere di alta qualità e sufficienti a garantire l'evitamento dell'overfitting; devono essere raccolti in modo da consentire al sistema una forma di generalizzazione rispetto alle situazioni che si presenteranno nel mondo reale. La regolarizzazione rappresenta un'altra tecnica pertinente a questo concetto, poiché implica la generalizzazione del modello attraverso una riduzione della sua complessità; l'obiettivo principale di questa strategia è quello di permettere al modello di apprendere piuttosto che semplicemente memorizzare [1]. Ciò può essere realizzato mediante tecniche che riducono il peso di alcuni parametri o neuroni (nel caso degli algoritmi delle reti neurali) del modello, con

l'intento di rendere gli algoritmi meno dipendenti da tali componenti nella generazione dell'output. Non entriamo in dettaglio in questa tecnica pero elenco sotto alcuni metodi :

- Regolarizzazione L2 e L1
- Dropout
- Arresto anticipato
- Aumento del dataset

Un'altra metodologia, tuttavia, non è correlata al modello, ed è quella di implementare una certa ridondanza nei sistemi di rilevamento e nell'hardware, che risulta fondamentale per il corretto funzionamento del sistema. Ciò è particolarmente essenziale nel contesto dei sistemi di sicurezza critici.

Passiamo ora a un altro requisito cruciale: **l'assenza di ambiguità e di errori intrinseci**. In primo luogo, l'ambiguità si riferisce alla condizione in cui il sistema inizia a formulare previsioni e prendere decisioni instabili e incerte. Questo può derivare da ambiguità nei dati acquisiti durante la fase di addestramento; quando i dati contengono informazioni poco chiare o contraddittorie, come nel caso in cui il sistema di monitoraggio delle posizioni di sicurezza venga alimentato con dati che non considerano le dimensioni degli ostacoli nella loro determinazione, ciò può generare confusione per il sistema durante l'analisi della situazione attuale [16]. Pertanto, il primo aspetto da garantire è la disponibilità costante di un dataset caratterizzato da alta qualità e chiarezza elevata. Un'altra causa dell'ambiguità all'interno di un sistema può essere rappresentata dall'ambiguità del modello stesso, ovvero quando il modello risulta troppo complesso per essere interpretato facilmente o, viceversa, se è troppo semplice per affrontare situazioni relativamente difficili. Infatti, un modello molto complesso può risultare vulnerabile a qualsiasi errore, anche minimo, mentre un sistema estremamente semplice potrebbe non comportarsi adeguatamente a livello della sua predizione. Pertanto, è fondamentale regolare adeguatamente il modello in relazione alla sua complessità. In merito all'overfitting, abbiamo già discusso delle tecniche pertinenti. Inoltre, l'ambiguità si manifesta anche attraverso una certa incertezza nelle predizioni e nelle decisioni assunte dal modello. Ad esempio, un modello applicato a un sistema può formulare previsioni riguardo alla posizione di un veicolo con una certa incertezza circa la sua sicurezza. Questa forma di ambiguità può essere mitigata tramite metodi di quantificazione dell'ambiguità, come il metodo bayesiano, che è sostanzialmente un approccio statistico fondato sulla probabilità legata alle informazioni disponibili; pertanto, aggiorna i propri parametri in base ai dati ricevuti. Di conseguenza, la distribuzione a priori combinata con i dati aggiornati consente di determinare la distribuzione posteriore. L'ambiguità può inoltre derivare dalla mancanza di chiarezza nella definizione degli obiettivi del modello [51]. Si prenda ad esempio un sistema di allerta antincendio in un impianto industriale: tale sistema deve inviare un avviso ai lavoratori ma nel contempo deve attivare un arresto d'emergenza del macchinario. In questo caso, il modello deve stabilire una priorità chiara; prima deve interrompere il funzionamento e successivamente inviare l'allerta. È cruciale che il sistema non privilegi la trasmissione del messaggio di allerta rispetto all'arresto della macchina, poiché ciò potrebbe comportare gravi rischi per la sicurezza sul posto di lavoro. Una delle strategie per attenuare gli effetti delle ambiguità non eliminabili consiste nel garantire la presenza di risorse umane dedicate alla gestione della situazione e al controllo, una volta

identificati eventuali errori nel funzionamento del sistema [78]. Tali sistemi, nel contesto dell'automazione, sono noti come "human-in-loop systems". In questi sistemi, il lavoratore collabora con il sistema per prendere decisioni corrette basate sulle informazioni analizzate dal modello.

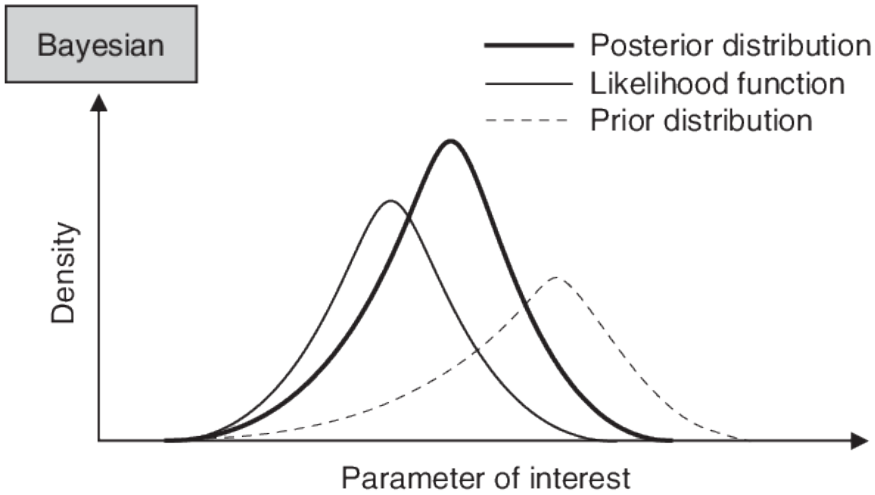


Figura 5 La distribuzione statistica del metodo di Bayes

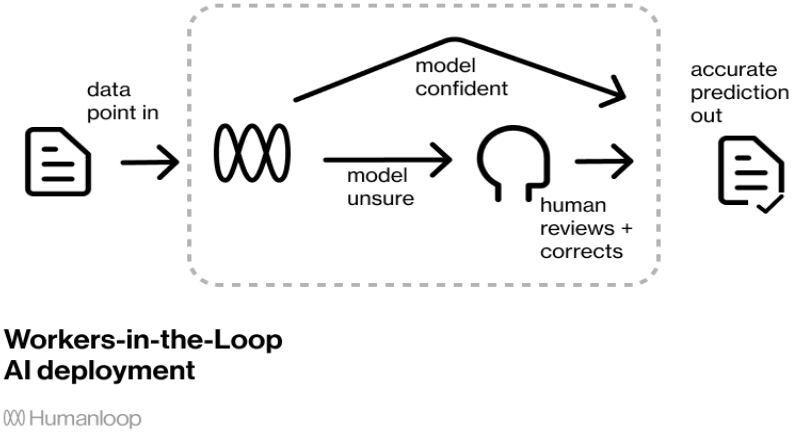


Figura 6 Human-in-the-loop system

Tutti i problemi menzionati nelle sezioni precedenti rientrano nella categoria degli errori intrinseci; a questi si aggiungono l'instabilità e la non convergenza, che si manifestano

attraverso decisioni e previsioni errate e inconsistenti derivanti dall'interpretazione fornita dal modello. Questo fenomeno è essenzialmente attribuibile a parametri inadeguati o algoritmi di ottimizzazione imperfetti. La regolarizzazione rappresenta un'altra possibile soluzione per mitigare tali problematiche.

Inoltre, la comprensione dei requisiti di sicurezza del sistema software costituisce un ulteriore aspetto fondamentale che deve essere soddisfatto da un modello di intelligenza artificiale integrato in una funzione di sicurezza. Un requisito imprescindibile riguarda l'immunità del sistema agli **attacchi avversari**, ovvero quelli perpetrati da criminali informatici con l'intento di manipolare e alterare il funzionamento del modello di intelligenza artificiale. Uno dei metodi per proteggere il sistema da tali attacchi consiste nell'addestrare il modello utilizzando input contraddittori, affinché il modello acquisisca familiarità con questo tipo di dati e possa quindi classificare correttamente le informazioni in fase operativa; questa tecnica è conosciuta come "adevrsarial training" [61]. Un altro approccio descritto dalla norma in studio riguarda la estrazione dei disturbi che sono stati introdotti nel sistema in modo artificiale come ad esempio l'High-*Level representation Denoiser*; è essenzialmente basata sull'implementazione di un sistema che elimina e migliora le prestazione di un input che contiene delle disturbanze e quindi può creare confusione per il modello a livello dell'interpretazione [37,48]. Tale metodo è stato classificato come il migliore tra altre tecniche visto che è applicabile a sistemi white-box è black-box , alcune esempi sono *MagNet* e *Defense-GAN [61, 48].* La randomizzazione della data è un altro approccio che può essere implementata per ridurre l'effetto di tali disturbi e aumentare la robustezza del sistema contro gli attacchi dannosi. Questo metodo si manifesta nell'aggiungere alcuna perturbazione all'input in modo casuale che riduce la possibilità che l'attacco riesce a modificare facilmente il modello. Un'idea importante da dire è che la non linearità del modello serve ad aumentare il livello di immunità del sistema a tali attacchi. Un gesto consigliabile per verificare che il nostro sistema è robusto sufficientemente contro tali attacchi è quello di provare a rispondere a due domande: [49]

- Come possiamo definire un esempio avversario tale da manipolare il modello efficacemente ma andando a creare poco perturbazione?
- Come possiamo allenare il modello in modo tale da ridurre al più possibile la probabilità che un attacco la può rovinare?

Rispondendo a queste due domande permette a livello di progettazione di far pensare gli ingegneri da metodi per aumentare al più possibile l'immunità del sistema a tali attacchi.

Un ulteriore requisito per un sistema software è la capacità di operare in tempo reale, garantendo così un adeguato grado di reattività; infatti, i sistemi sottoposti a funzioni di sicurezza devono gestire rischi potenziali senza alcuna latenza, rendendo pertanto necessario considerare la latenza generata dal sistema software stesso. Inoltre, è fondamentale il determinismo, inteso come la capacità del sistema di produrre risultati identici quando viene fornito lo stesso input; ciò implica una certa stabilità nelle sue analisi e interpretazioni. Infine,

l'adattamento del software ai mutamenti delle condizioni operative rappresenta un ulteriore requisito essenziale, poiché il sistema deve sempre essere in grado di funzionare in sicurezza sotto qualsiasi circostanza.

Un altro requisito fondamentale è quello di garantire l'assenza di interferenze involontarie e potenzialmente pericolose tra le funzioni di sicurezza e quelle non di sicurezza del sistema. Questo principio, come delineato dalla norma ISO 26262 [35], si traduce nell'assenza di guasti a cascata che possano condurre a situazioni pericolose tra i vari sistemi, in particolare tra quelli critici per la sicurezza (che gestiscono funzioni di sicurezza) e quelli dedicati a funzioni normali. La figura 6, estratta dalla stessa normativa, illustra che i guasti dell'elemento 2 non devono comportare il guasto dell'elemento 1 [24].

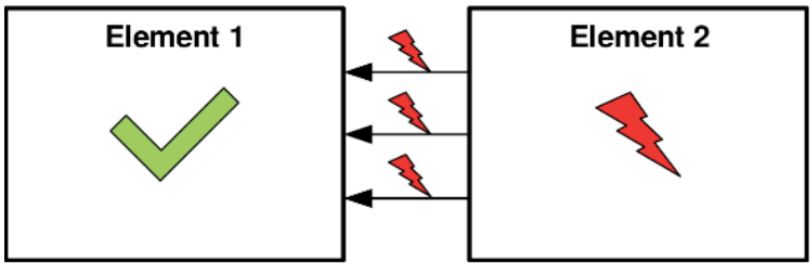


Illustration 1: Element 1 is Free from Interference. Copyright © ISO.

Figura 7 sistema libera da interferenza

Si identificano tre tipologie di interferenza che possono manifestarsi e devono essere ridotte al minimo: interferenze temporali ed esecutive, interferenze sulla memoria e scambi informativi.

La prima categoria si riferisce alla situazione in cui il funzionamento del software responsabile di una funzione di sicurezza viene influenzato da un altro sistema software associato a una funzione normale. Una strategia per affrontare i problemi legati al blocco o ai ritardi derivanti è l'implementazione di un watch-dog all'interno del sistema operativo; tale approccio consente il monitoraggio e la ripresa delle funzioni bloccate [24].

La seconda tipologia d'interferenza concerne la memoria e rappresenta la situazione in cui un componente software altera il codice memorizzato da un altro elemento. Questo problema può essere affrontato mediante l'implementazione a livello hardware di un'unità di protezione della memoria, la quale è incaricata di identificare gli errori che si manifestano a seguito delle modifiche apportate. Tale unità emette una richiesta di interruzione che costringe il sistema di gestione degli errori a riavviare tutti gli elementi software coinvolti nell'interruzione [24].

L'ultima categoria di problemi è associata allo scambio di dati tra due sistemi e può derivare dalla ripetizione delle informazioni, dalla perdita parziale durante la fase di comunicazione o da ritardi. Una possibile soluzione per affrontare questa problematica consiste nell'impiego di

algoritmi block-free, i quali presentano il vantaggio di minimizzare i tempi di latenza durante il trasferimento dei dati [24].

Tuttavia, la norma ISO/IEC 5469[37] ha stabilito una classificazione riguardante i parametri del modello. È opportuno sottolineare che, quando si trattano parametri di modelli complessi, come ad esempio i modelli di reti neurali, la loro interpretazione risulta particolarmente difficile; pertanto, è necessario adottare misure supplementari per garantire la sicurezza del sistema. Un approccio possibile consiste nell'effettuare numerose simulazioni al fine di assicurare il raggiungimento degli obiettivi prefissati. Un ulteriore metodo prevede l'implementazione di strumenti ausiliari per accrescere la trasparenza e il livello di interpretabilità del sistema. Al contrario, nel caso in cui si utilizzino modelli più semplici, è sufficiente fare riferimento ai punti analizzati dalle norme pertinenti già esaminate in precedenza. Con ciò si conclude una descrizione generale delle proprietà e dei requisiti che devono caratterizzare i **modelli e gli algoritmi**, nonché dei potenziali rischi da mitigare quanto più possibile.

4.2 Trasparenza e spiegazione

La **trasparenza e la spiegazione** di un sistema rappresentano aspetti fondamentali nel contesto delle proprietà dei sistemi di intelligenza artificiale. Questi concetti sono particolarmente rilevanti quando si discute della sicurezza funzionale del sistema, poiché influenzano il suo operato. La spiegazione (explainability) si riferisce alla capacità del sistema AI di illustrare in modo comprensibile per l'intelligenza umana i concetti e i fattori che determinano i risultati generati dal sistema [37,45]. Al contrario, la trasparenza è legata alla disponibilità delle informazioni riguardanti ciò che avviene all'interno del sistema AI per le parti interessate; in altre parole, la trasparenza chiarisce il “come” mentre la chiarezza spiega il “perché”[45]. È opportuno notare che i comitati tecnici tendono a raggruppare entrambi i concetti sotto l'egida della "trasparenza"[37]. È fondamentale raggiungere un livello adeguato di trasparenza, evitando estremi sia alti che bassi. Infatti, un modello caratterizzato da un elevato grado di trasparenza risulta particolarmente vulnerabile ad attacchi malevoli, in quanto tutte le informazioni relative al funzionamento del sistema diventano facilmente accessibili agli hacker [72]. Un ulteriore rischio associato a un alto livello di trasparenza è la potenziale confusione che potrebbe sorgere durante l'esecuzione delle operazioni del sistema; infatti, perseguire una notevole trasparenza comporta un aumento della complessità del sistema stesso, risultando in un sovraccarico informativo [72]. Però l’importanza della trasparenza è che prima di tutto permette il sistema di vincere la fiducia dei lavoratori e utilizzatori in quanto chiarisce per essi perché sono presi tali decisioni e come; ulteriormente rende più facile la gestione e il rilevamento degli errori che possano verificarne. Secondo la norma ISO/IEC TR 24028: 2020 la “explainability” non è sufficiente per realizzare un modello trasparente, pero ogni modello trasparente deve essere spiegabile [40]. L'importanza di comprendere le ragioni alla base delle decisioni adottate dal sistema è fondamentale per prevenire azioni potenzialmente dannose ed errate, poiché consente all'utente di discernere il motivo per cui una determinata scelta è stata effettuata, permettendo così un intervento volto a

risolvere e correggere tali errori. La spiegazione fornita dal sistema può essere di due tipi: Ex-ante (prima dell'utilizzo del sistema) ed Ex-post (dopo l'impiego del sistema); la prima modalità, Ex-ante, offre all'utente una comprensione delle proprietà generali che influenzano la generazione degli output, garantendo così che il sistema funzioni come previsto e instillando un senso di fiducia nell'efficienza del suo operato [40]. Al contrario, l'Ex-post serve a chiarire le motivazioni e le condizioni sotto le quali sono state prese determinate decisioni; ciò è utile per analizzare le performance e valutare se gli obiettivi siano stati raggiunti in modo adeguato [40]. Un livello adeguato di spiegazione offerta dal sistema contribuisce significativamente ai processi di verifica e validazione, i quali saranno trattati successivamente.

Da quanto esposto sorge la questione su come individuare il giusto livello di trasparenza. In primo luogo, è necessario osservare che il grado di spiegazione fornito dipende in gran parte da chi si intende definire come "pubblico mirato"; questo perché, come indicato da [22], possono essere identificati otto tipologie di trasparenza, elencate nella tabella 1.

L'applicazione di tali concetti a sistemi destinati ad essere integrati nelle funzioni di sicurezza consente di comprendere il grado di trasparenza richiesto dal modello. Poiché questi sistemi devono necessariamente essere validati e verificati, in quanto requisito imprescindibile per garantire la conformità a questa norma, risulta evidente che sarà necessario un livello adeguato di trasparenza per i progettisti. Questo è fondamentale al fine di identificare i punti deboli del modello e definire le modalità per gestirli. Un livello di trasparenza tale da consentire all'utente di comprendere il funzionamento generale del sistema è essenziale affinché possa rilevare eventuali errori quando si presentano [22]. Analogamente, il manutentore deve possedere la competenza necessaria per individuare l'origine del guasto e comprenderne le cause. Inoltre, nel caso in cui la macchina debba interagire frequentemente con gli esseri umani, è imperativo stabilire un certo grado di fiducia; ad esempio, in un robot dotato di un sistema AI per il monitoraggio della posizione sicura, le persone presenti nello stesso ambiente lavorativo devono sentirsi a proprio agio durante l'interazione con tali sistemi, certi che questi siano sicuri e in grado di gestire autonomamente qualsiasi situazione potenzialmente pericolosa [22]. Pertanto, è cruciale considerare tutti questi aspetti durante la progettazione del modello al fine di raggiungere un livello di trasparenza capace di soddisfare tali requisiti durante le fasi di validazione e verifica. Non approfondiremo il processo di realizzazione della trasparenza, in quanto si tratta di una questione altamente tecnica di competenza degli ingegneri dell'informazione e strettamente legata alle normative che delineano i concetti associati all'intelligenza artificiale.

Transparency in the context of robotics and AI	
For a...	**To...**
Developer	Understand whether their system is working properly in order to identify and remove errors from the system or improve it
User	Provide a sense for what the system is doing and why, to enable intelligibility of future unpredicted actions circumstances and build a sense of trust in the technology
	Understand why one particular decision was reached
	Allow a check that the system worked appropriately
	Enable meaningful challenge (e.g. credit approval or criminal sentencing)
Society broadly	Understand and become comfortable with the strengths and limitations of the system
	Overcome a reasonable fear of the unknown
Expert/Regulator	Provide the ability to audit a prediction or decision trail in detail, particularly (un)intended harmful actions, e.g. a crash by an autonomous car
Deployer	Make a user feel comfortable with a prediction or decision, so that they keep using the system

Tabella 1livello di trasparenza richiesto in base al personale interessato

4.3 Livello di automazione e controllo

In questa sezione, la norma analizza due aspetti fondamentali: la prontezza del sistema di automazione e l'esigenza di una risorsa per la supervisione, che può consistere in un operatore umano o in un ulteriore sistema automatizzato, al fine di garantire la sicurezza funzionale del sistema stesso. Questi due elementi sono intrinsecamente interconnessi al livello di automazione del sistema, definito come il grado in cui il sistema è capace di operare autonomamente e senza intervento umano. È importante notare che i sistemi dotati di un elevato grado di autonomia possono manifestare comportamenti complessi e imprevedibili, potenzialmente conducendo a situazioni pericolose. Pertanto, comprendere il livello di automazione del sistema implica l'adozione di un meccanismo per la supervisione del corretto funzionamento dello stesso. La norma ISO/IEC 22989:2022 fornisce un elenco di definizioni e classificazioni relativi a questi livelli, come evidenziato nella Tabella 3 [36].

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

Tabella 2 relazione tra autonomia, eteronomia, e sistemi automatici

In primo luogo, è fondamentale evidenziare la differenza principale tra i concetti di autonomia ed eteronomia. L'autonomia si riferisce alla capacità di un sistema di operare in modo indipendente, prendendo decisioni che ritiene siano le più appropriate in base alle circostanze attuali, senza necessità di intervento da parte di un operatore né di un sistema di monitoraggio esterno [36]. Al contrario, i sistemi eteronomi, caratterizzati da vari livelli, sono in grado di svolgere la missione assegnata solo a seguito di una richiesta, iniziando da un livello completamente automatizzato e quindi senza alcun intervento esterno; ciò può gradualmente evolversi verso situazioni in cui è richiesto l'intervento umano nel caso in cui non vi sia alcuna automatizzazione e l'operatore gestisca l'intero sistema [36].

4.3.1 Necessita di Supervisione

È importante notare che l'autonomia di un sistema non è sempre vantaggiosa; pertanto, in molte circostanze si rende necessaria una supervisione poiché nel contesto elettronico e informatico l'idealità non viene mai raggiunta. Quando si considerano le funzioni legate alla sicurezza, tale necessità diventa particolarmente pressante poiché è coinvolta la salute del personale. Prima di approfondire il tema della supervisione, intendo elencare alcuni casi in cui l'intelligenza artificiale autonoma ha manifestato insuccessi causando problematiche significative.

- Il 10 marzo 2019, sei minuti dopo il decollo, si è verificato un incidente riguardante il volo Ethiopian Airlines 302, operato da un Boeing 737 MAX 8 [34]. Questo evento seguiva un incidente simile occorso cinque mesi prima con il volo Lion Air 610, anch'esso dello stesso modello. Le indagini condotte hanno rivelato che vi era un errore nella stima fornita dal sensore che monitorava “angle of attack", il quale misura essenzialmente la probabilità di uno stallo del velivolo qualora si trovi in una condizione in cui il muso dell'aeromobile è eccessivamente inclinato verso l'alto rispetto alle ali, compromettendo così la generazione di portanza necessaria a velocità elevate. Il sensore riportava una stima eccessiva della possibilità di uno stallo, attivando un sistema autonomo che inclinava progressivamente il muso dell'aereo verso il basso, rendendo estremamente difficile per il pilota gestire la situazione [34]. Questo esempio evidenzia le vulnerabilità intrinseche dei sistemi automatizzati nel rilevare problematiche relative agli input, le quali devono essere affrontate prima di prendere decisioni critiche.

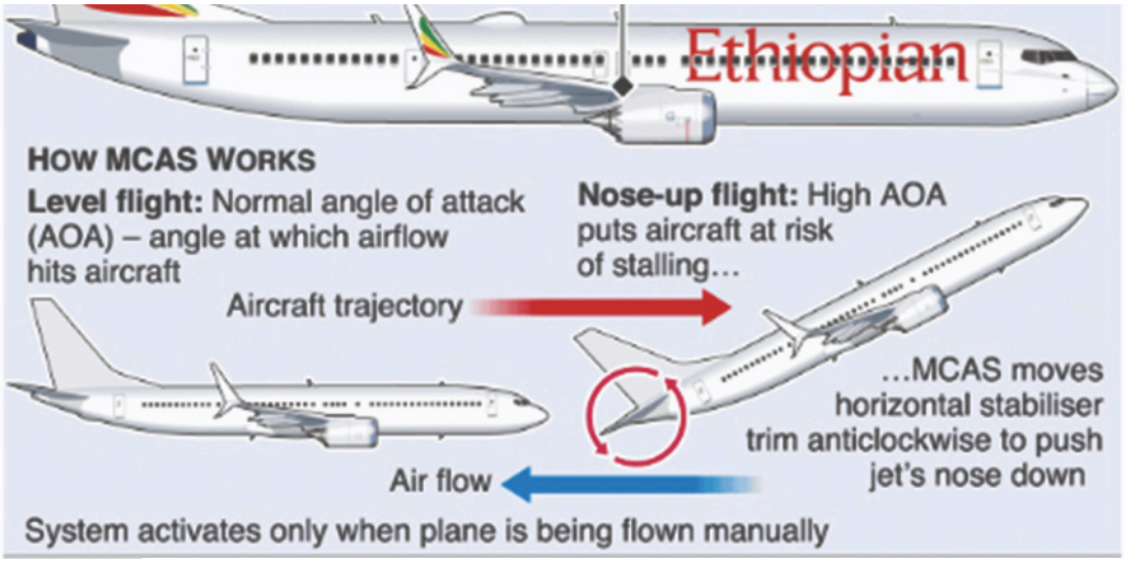


Figura 8 infografico dell'incidente BOEING 737 MAX 8

- Il 19 luglio 2022, un robot progettato per giocare a scacchi ha provocato la frattura di un dito di un ragazzo di sette anni [18]. Durante il gioco, il robot ha manifestato confusione e non è riuscito a seguire il ritmo delle azioni del bambino; mentre questo stava posizionando il dito per effettuare la mossa successiva, il robot ha erroneamente interpretato il dito come un pezzo degli scacchi e lo ha afferrato per diversi secondi prima che un intervento esterno potesse rimuovere il dito del ragazzo. Pertanto, una disfunzione nell'analisi e nella comprensione della situazione ha causato questo

incidente, che avrebbe potuto essere evitato mediante l'integrazione di un elemento di supervisione in grado di valutare correttamente se l'input ricevuto fosse appropriato.

Figura 9 Robot che gioca a scacchi rompe la dita di un ragazzo

- L'8 novembre 2023, in Corea del Sud, un robot operante su una linea di distribuzione ha causato la morte di un manutentore durante l'ispezione del suo sensore [71]. Il robot ha erroneamente identificato l'uomo come un ostacolo da rimuovere, schiacciandolo violentemente contro la linea e provocandogli gravi lesioni al collo che ne hanno determinato la morte. Questo episodio rappresenta un ulteriore esempio delle potenziali calamità derivanti dalla completa dipendenza e fiducia nei sistemi automatizzati.

Figura 10 Linea di distribuzione con bracci robotiche

Come precedentemente evidenziato, al fine di mitigare gli aspetti imprevedibili associati ai sistemi automatizzati, e in particolare quelli relativi alle funzioni di sicurezza, esistono diverse tecniche. Tra queste, la "supervisione" emerge come l'approccio predominante.

4.3.2 Supervisione

Questo tema è stato trattato nella norma al Paragrafo 10, dove vengono discusse le metodologie per la mitigazione e riduzione dei rischi connessi ai sistemi di intelligenza artificiale (AI).

In linea generale, la supervisione si manifesta attraverso due modalità principali: la prima implica l'implementazione di un sistema di controllo e supervisione nominati come sistemi SCADA volto alla convalida del corretto funzionamento del sistema AI principale, nel nostro caso di studio integrato in una funzione di sicurezza, e prendere le giuste azioni una volta verificato un errore; la seconda modalità è caratterizzata dalla sorveglianza da parte di un operatore e dalla gestione della situazione nel momento in cui venga identificato un errore o un malfunzionamento [23].

Secondo quanto riportato in [73] la supervisione può essere classificata secondo il modello rappresentato nella figura 11.

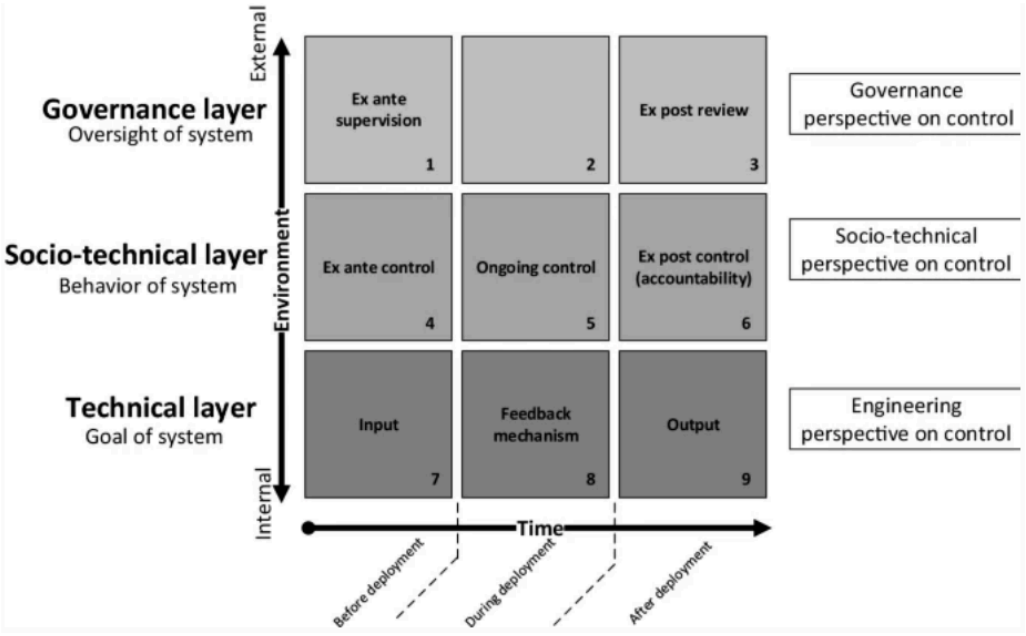


Figura 11 diverse gradi di supervisione

Da tale figura emerge una classificazione delle tipologie di supervisione e controllo in relazione al tempo di intervento e agli strati coinvolti nelle azioni intraprese. Infatti, questa supervisione può essere attuata per verificare il corretto funzionamento a livello dell'aspetto esterno del sistema, oppure riguardo alle azioni eseguite o agli obiettivi prefissati, in tre distinte fasi: quella

antecedente all'attivazione del sistema, durante il suo funzionamento e successivamente al raggiungimento dell'obiettivo iniziale.

La scelta della tecnica più appropriata per la supervisione implica, come primo passo, la definizione del cosiddetto “safe envelope”, ovvero i limiti entro i quali il sistema intelligente è considerato sicuro e non deve oltrepassare. Tali limiti vengono stabiliti attraverso un'analisi approfondita del rischio.

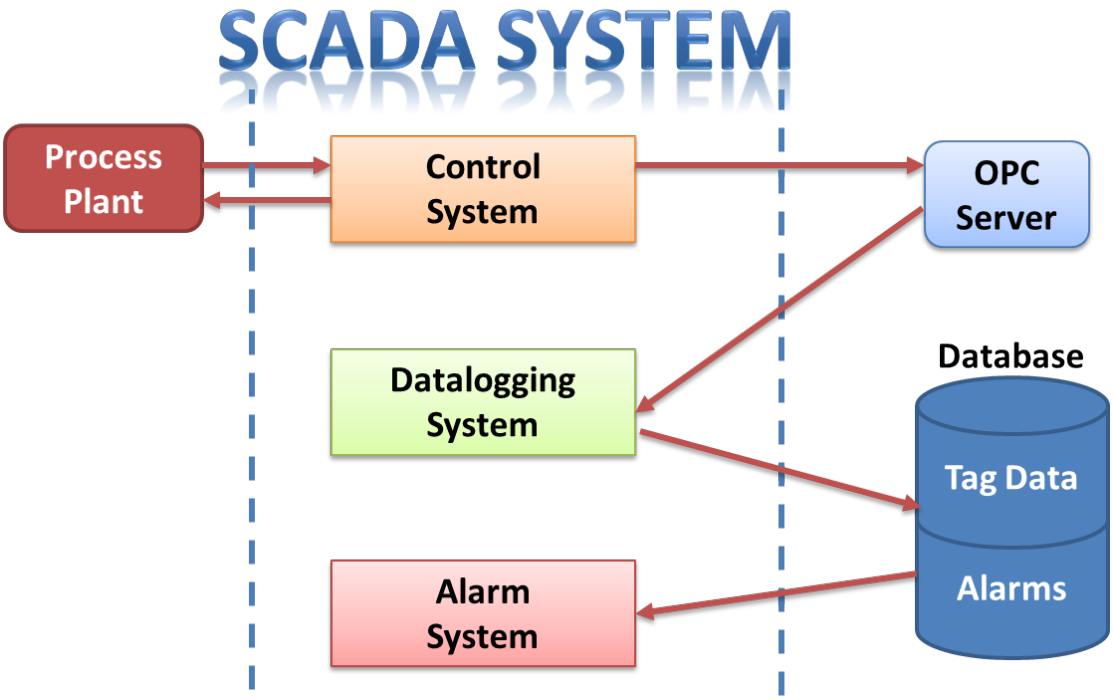


Figura 12 Struttura di sistemi di supervisione SCADA

Nel contesto di una funzione di sicurezza, è fondamentale discutere l'input e la sua analisi. Infatti, per attivare una funzione di emergenza, l'input deve essere analizzato e acquisito con precisione; ad esempio, nel caso di un sistema di allerta antincendio integrato in un robot o in una macchina, i sensori termici devono funzionare correttamente affinché si possa generare un arresto e inviare un’allerta in seguito a un aumento anomalo della temperatura. Pertanto, in tali sistemi, è essenziale implementare un elemento di supervisione dei sensori che garantisca una valutazione accurata dell'input; questi devono essere integrati per operare efficacemente. Consideriamo il caso di un drone impiegato nell'ispezione di perdite di gas in un impianto del settore Oil & Gas: se a causa di problemi hardware si verifica un incendio all'interno di questo sistema automatizzato e il sistema AI non riesce ad attivare la funzione di sicurezza a causa di malfunzionamenti dei sensori, ciò potrebbe portare a conseguenze catastrofiche. La criticità associata alla sicurezza di tali sistemi obbliga i progettisti a integrare meccanismi adeguati di

supervisione. Come delineato dalle normative vigenti, tali sistemi di controllo e supervisione devono essere di livello di utilizzo C o D [37].

Un ulteriore aspetto di un sistema di supervisione riguarda la verifica del corretto funzionamento e l'analisi del sistema stesso. Consideriamo un sistema di monitoraggio della posizione di sicurezza, il quale è fortemente influenzato dalla capacità del sistema di stimare e analizzare la situazione e l'ambiente circostante. Come evidenziato negli esempi precedentemente menzionati, nel caso specifico di un robot operante su una linea di distribuzione, la difficoltà nel discernere se l'oggetto presente davanti a esso fosse una persona ha comportato un grave incidente. Pertanto, l'integrazione di un sistema di supervisione in grado di garantire una corretta interpretazione avrebbe potuto prevenire tale evento.

Un'altra modalità per attuare la supervisione è quella denominata "Supervision Human Control" (SHC), che si basa essenzialmente sull'intervento di un operatore designato a riprendere il controllo nel caso in cui venga identificato un errore nel funzionamento del sistema [73]. È fondamentale, tuttavia, evidenziare che questo metodo presenta delle limitazioni che ne compromettono l'affidabilità. Secondo quanto documentato nella letteratura [73], questo approccio è afflitto da aspetti problematici: il primo riguarda la perdita della consapevolezza situazionale dell'operatore, mentre il secondo concerne le circostanze in cui le condizioni vengono alterate, generando confusione nell'operatore stesso, il quale potrebbe non essere in grado di gestire autonomamente tali situazioni. Un ulteriore svantaggio è quello riferito all'eccessiva fiducia da parte dell'operatore sulle decisioni presi dalla macchina, e questo aspetto cresce di più di quanto stato funzionato bene il sistema. Però l'intervento umano risulta efficiente per gestire situazione dove possano verificare degli attacchi avversarie. Infatti, la disponibilità di una persona dedicata alla rilevazione di qualsiasi tentativo da parte dei hacker.

Pertanto, da quanto esposto, è possibile comprendere che tale funzione di supervisione è implicata in relazione all'applicazione del sistema e alle funzioni che deve svolgere e ovviamente in base all'analisi del rischio del sistema.

4.4 Problemi correlati all'ambiente circostante

4.4.1 Complessità dell'ambiente circostante

Un aspetto cruciale nella discussione dei rischi associati ai sistemi di intelligenza artificiale, e che deve essere considerato durante la progettazione di tali sistemi da integrare nelle funzioni di sicurezza, è la complessità dell'ambiente circostante in cui l'automa opererà. Nei sistemi di autoapprendimento, la capacità del sistema di identificare e adottare la decisione più appropriata porta i progettisti a definire un insieme di obiettivi specifici che rimangono volutamente generali, come indicato nella normativa attraverso il termine "specificazioni vaghe" [37]. Questa limitazione, intrinseca alla consapevolezza umana nel tentativo di delineare tutti i possibili casi e condizioni che il sistema potrebbe affrontare, costringe i progettisti a sfruttare questo particolare aspetto dei sistemi intelligenti.

Tuttavia, tale caratteristica rappresenta una vulnerabilità significativa in termini di sicurezza, poiché in tali contesti risulta arduo convalidare la conformità del sistema ai requisiti della sicurezza funzionale; le conseguenze di questa difficoltà si manifestano maggiormente in ambienti lavorativi complessi [64]. Si pensi, ad esempio, al classico caso del monitoraggio della distanza di sicurezza: se al sistema viene impartito un comando generale quale "mantenere una distanza sufficiente di sicurezza dagli oggetti circostanti", senza però specificare cosa si intenda esattamente per "sufficienza", allora in un robot operante all'interno di un'industria logistica o distributiva — caratterizzata da un elevato numero di interazioni con vari oggetti — potrebbero sorgere problematiche significative e quindi complessità elevata; il sistema può trovarsi in uno stato di confusione, risultando incapace di definire una posizione di sicurezza, il che può condurre a situazioni pericolose.

Un ulteriore effetto di tale situazione è la difficoltà nel convalidare la sicurezza del sistema durante le fasi di verifica e validazione, un aspetto fondamentale per i sistemi che svolgono funzioni di sicurezza. Pertanto, considerando la criticità intrinseca a tali sistemi, la normativa stabilisce il concetto di completezza funzionale, un approccio da adottare in maniera non necessariamente eccessiva al fine di conseguire un livello accettabile di sicurezza funzionale [37].

La completezza funzionale si manifesta attraverso una descrizione estremamente dettagliata delle specifiche oppure mediante l'addestramento del sistema per coprire quanto più possibile tutte le condizioni ambientali che potrebbe incontrare; è anche possibile realizzarla combinando entrambe le modalità. Nei paragrafi precedenti abbiamo esaminato i rischi associati all'incompletezza del dataset e le relative soluzioni. Qui viene proposto un ulteriore metodo, ritenuto piuttosto efficace nelle fasi di verifica e validazione; tale approccio è noto come "test e design iterativo" del sistema [52]. Questo si fonda essenzialmente, come descrive il nome stesso, su un ciclo progettuale caratterizzato da feedback: dopo aver effettuato l'analisi dei rischi e definito gli obiettivi richiesti dal sistema, viene realizzato un primo prototipo. Il presente prototipo è sottoposto a un gruppo di test al fine di valutare la sua efficienza in diverse situazioni, evidenziando le problematiche riscontrate in ciascuna fase. In funzione dei punti deboli identificati, si procede a perfezionare il sistema mediante l'adozione di approcci differenti per eliminarli. Questo processo di affinamento viene ripetuto ciclicamente fino a conseguire un sistema che soddisfi integralmente i requisiti di sicurezza e funzionalità.

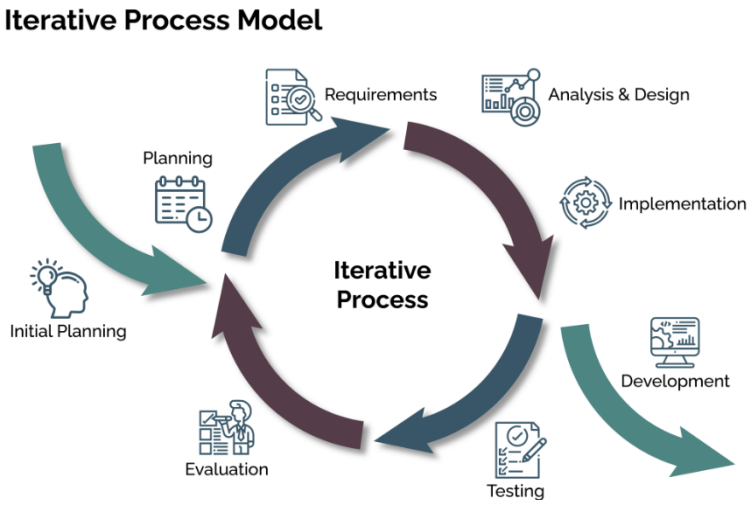


Figura 13 Processo iterativa di un modello

Attraverso tali metodologie è possibile affrontare in modo significativo le problematiche legate alle specifiche vaghe in contesti complessi.

Tuttavia, un ulteriore aspetto che può risultare rischioso negli ambienti complessi, come evidenziato dalla normativa, riguarda il fatto che i sistemi di intelligenza artificiale deterministici tendono a generare output probabilistici sotto l'influenza della complessità ambientale [37]. Ciò avviene poiché si trovano ad affrontare situazioni mai precedentemente incontrate o per le quali non dispongono di informazioni adeguate nel loro dataset. Questo elemento suscita generalmente critiche nei confronti del modello implementato nella realizzazione del sistema, poiché l'incertezza può derivare da una mancanza di chiarezza nei parametri del modello stesso.

La norma in questione propone un approccio volto a mitigare l'impatto di tale problematica, attraverso l'implementazione dei cosiddetti modelli stocastici [37]. Secondo [17] tali modelli hanno dimostrato una notevole capacità nel migliorare l'adattabilità dei sistemi di intelligenza artificiale, in particolare nelle condizioni impreviste che caratterizzano gli ambienti complessi. Di conseguenza, essi contribuiscono ad accrescere la robustezza, l'efficienza e l'affidabilità del sistema, soddisfacendo così gran parte dei requisiti di sicurezza precedentemente delineati. Per evitare di addentrarmi in dettagli eccessivamente tecnici, desidero evidenziare brevemente alcune delle caratteristiche che rendono questi modelli particolarmente efficaci per i sistemi operanti in contesti complessi [17].

• La prima proprietà riguarda la loro dinamica temporale; infatti, tali modelli sono connotati da evoluzioni discrete o continue nel tempo.

• La seconda è rappresentata dai valori casuali associati al modello in ogni istante, descritti mediante una variabile casuale che evolve secondo una determinata distribuzione probabilistica.

• Infine, il terzo aspetto concerne la definizione di uno spazio degli eventi per tali modelli, all'interno del quale vengono specificati tutti i possibili valori assunti dalla variabile.

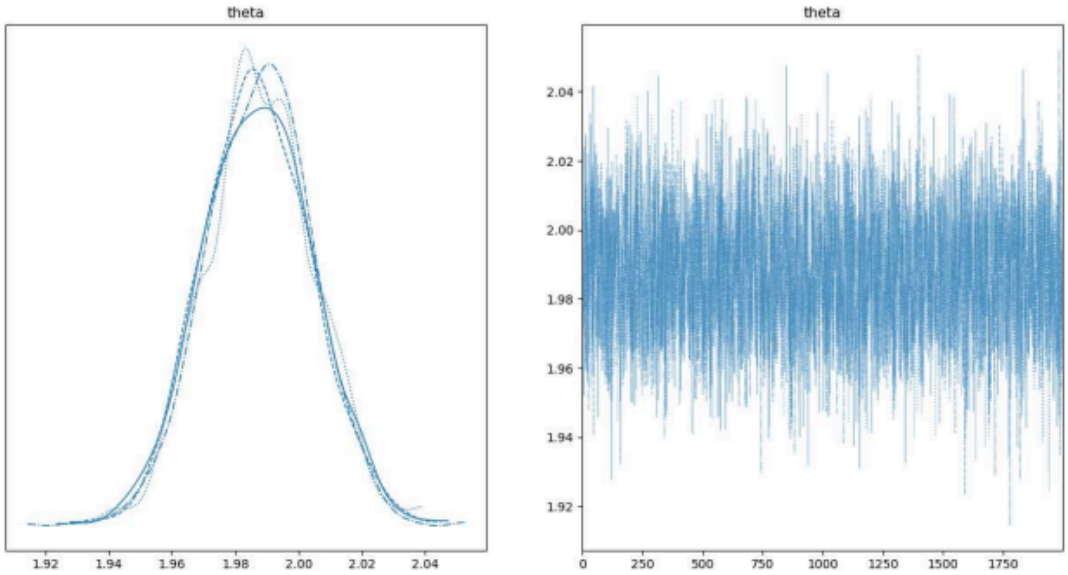


Figura 14 distribuzione stocastica di un parametro teta

Alcuni esempi di metodologie associate a questi approcci includono: le simulazioni Monte Carlo, il Mini-batch Gradient Descent e Adam (Adaptive Moment Estimation), tra gli altri.

Utilizzo l'esempio illustrato nella figura.14 fornito da [17] che si riferisce a un'analisi dei valori ottenuti di una variabile casuale "teta" all'interno di un modello stocastico. La distribuzione di tali valori, rappresentata nei grafici, evidenzia come il 95% degli stessi si collochi in un determinato intervallo; questa informazione non solo contribuisce alla stima del valore ma fornisce anche la percentuale di incertezza associata al calcolo di tale parametro. Qui si manifesta l'efficienza di tali modelli, i quali sono in grado di offrire una distribuzione di valori che indica le probabilità più elevate e la percentuale di incertezza ad esse correlata.

4.4.2 Problemi riferiti al cambio di condizioni dell'ambiente circostante del sistema

Model Drift

Come accennato nel paragrafo introduttivo riguardante l'intelligenza artificiale, questi sistemi fondano la loro operatività su modelli statistici e matematici che stabiliscono una relazione specifica tra un output y e un input x. In alcune circostanze, la complessità del modello o dell'ambiente operativo può generare una deriva del modello stesso; ciò implica che il modello perde la sua "espressività", significando che la relazione tra y e x non risulta più precisa, compromettendo significativamente la sua efficienza [8].

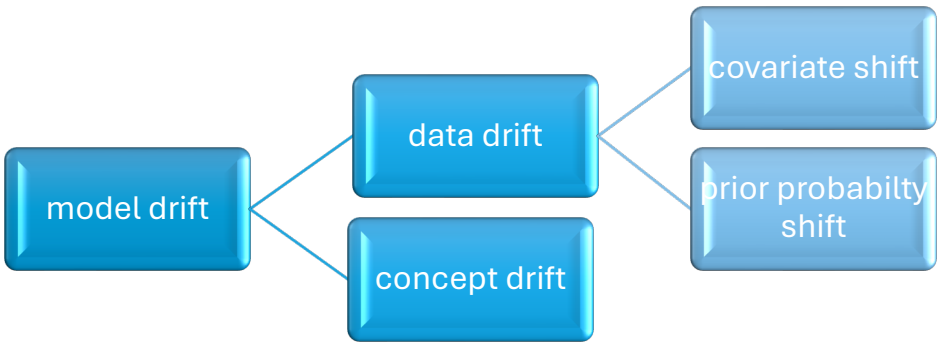


Figura 15 Drift nei modelli ML

La deriva del modello, come evidenziato nella figura 3 e citato in [67], si riferisce essenzialmente a due concetti distinti: il data drift e il concept drift. Questi due fenomeni non sono strettamente correlati; infatti, il data drift si verifica quando il modello, una volta implementato, incontra un insieme di input significativamente diverso da quello su cui è stato addestrato, manifestando pertanto una deriva a livello degli input x. Tale variazione avviene esclusivamente a livello di x, poiché tutti gli altri fattori rimangono invariati. Essa può essere rappresentata attraverso una modifica nella distribuzione degli input, nota come covariate shift, oppure può manifestarsi come un cambiamento della prior probability shift, indicando una modifica della distribuzione associata all'output y [69]. D'altro canto, il concept drift comporta una variazione nella relazione tra y e x, implicando quindi una modifica sostanziale nell'obiettivo del modello.

Le cause del data drift possono derivare da diversi fattori; oltre alla limitatezza del dataset utilizzato per l'addestramento del modello, come indicato in ,un esempio significativo è l'adozione di nuovi sensori durante la fase di training che presentano caratteristiche modificate o imprecise [8]. Consideriamo ad esempio un sistema di monitoraggio della posizione di sicurezza: questo sistema deve necessariamente essere alimentato da un sensore di posizionamento che traduce il movimento dell'oggetto in un segnale utilizzabile per l'analisi del modello. Qualora un sensore venga impiegato nel sistema e restituito da un altro dispositivo con una precisione superiore rispetto a quello utilizzato durante la fase di addestramento, il modello potrebbe non essere in grado di interpretare adeguatamente i dati che verranno acquisiti. Questo si verifica poiché un sensore più preciso possiede la capacità di raccogliere informazioni più raffinate e dettagliate, conducendo così a una situazione di data drift. Una soluzione efficace in questo contesto specifico è il cosiddetto "re-training" del modello, che consiste essenzialmente nell'addestrare nuovamente il modello utilizzando un dataset che somigli quanto più possibile ai dati che saranno incontrati durante la fase di testing del sistema. Tuttavia, in determinate circostanze, il sistema può sperimentare data drift al momento dell'applicazione, rendendo impraticabile il re-training; in tali situazioni, è necessario apportare modifiche al modello per renderlo più adattabile e robusto.

D'altra parte, come stabilito dalla norma EN ISO/IEC 22989 [36], quando si manifesta il problema del concept drift, le variabili target del dataset di addestramento devono essere etichettate e il sistema deve subire un re-training. Le variabili target sono essenzialmente quelle su cui il modello si propone di effettuare calcoli e costituiscono la base della mappatura tra i dati e gli output.

Considerando la rilevanza delle funzioni di sicurezza nella salvaguardia della salute, la normativa stabilisce che per i sistemi di intelligenza artificiale da integrare in tali funzioni, è necessario individuare i drift, il che comporta generalmente l'impiego di metodi specifici per il loro rilevamento. Questi metodi non solo segnalano l'insorgenza di un drift, ma stimolano anche l'adozione di misure correttive per affrontare la situazione. Uno degli approcci comunemente utilizzati è quello noto in inglese come EDDM (Early Drift Detection Method); come evidenziato da [11], questo metodo si basa sull'idea di valutare la distanza tra due errori consecutivi anziché limitarsi a contare il numero complessivo di errori riscontrati. Implementando tale metodo durante la fase di apprendimento del modello, si osserva che la distanza tra due errori tende ad aumentare fino a raggiungere un valore massimo quando il modello riesce ad approssimare nel modo più accurato possibile i concetti rappresentati nel dataset. Successivamente, il processo di rilevamento richiede la definizione di una soglia per gli errori oltre la quale il sistema deve attivarsi per identificare un "concept drift".

Altri metodi impiegati nel rilevamento del drift includono le SVM (Support Vector Machines), modelli supervisionati finalizzati a trovare un'ipotesi h che consenta di garantire una probabilità minima d'errore [44]. In conclusione, indipendentemente dalla tecnologia adottata per mitigare questo aspetto, è imperativo che il modello venga infine validato come sicuro e affidabile.

4.5 Rischi associati all'apprendimento dall'ambiente

4.5.1 L'apprendimento rinforzato

L'apprendimento rinforzato rappresenta una delle metodologie del machine learning (ML) che ottimizza le decisioni assunte dai modelli attraverso un approccio basato su ricompense e penalità durante l'elaborazione dei dati. In tal modo, il sistema si impegna a massimizzare il feedback positivo ricevuto. Tuttavia, sebbene questa strategia appaia logicamente valida per minimizzare gli errori, nella realtà si manifesta una problematica nota come "reward hacking", la quale comporta un deterioramento degli obiettivi sistemici; in altre parole, anziché concentrarsi sul compito primario assegnatogli, il sistema tende a privilegiare attività che garantiscono ricompense. Un esempio esplicativo è fornito da un sistema di antincendio integrato in un veicolo: qualora il sistema riceva una ricompensa ogni volta che segnala un potenziale incendio, potrebbe incorrere nel rischio di generare falsi allarmi frequenti al fine di incrementare il numero di riconoscimenti ottenuti, compromettendo così gravemente l'efficacia e l'affidabilità del sistema stesso.

Una delle soluzioni proposte per affrontare tale dilemma è rappresentata dalle "Adversarial Reward Functions", una tecnica che consente al sistema di analizzare e identificare le problematiche associate alla modalità con cui vengono accumulate le ricompense [62]. È fondamentale sottolineare che il meccanismo di verifica delle ricompense deve prevalere su quello dedicato alla raccolta delle stesse .

Un'altra strategia consiste nel cosiddetto "reward pretraining", la quale si fonda sull'addestramento del modello a un sistema di ricompensa fisso e predefinito durante la fase di formazione. Questo approccio mira a limitare l'autoapprendimento del modello dall'ambiente una volta implementato. Le tecniche menzionate sono solo due delle numerose modalità che possono essere adottate per affrontare tale problematica [62].

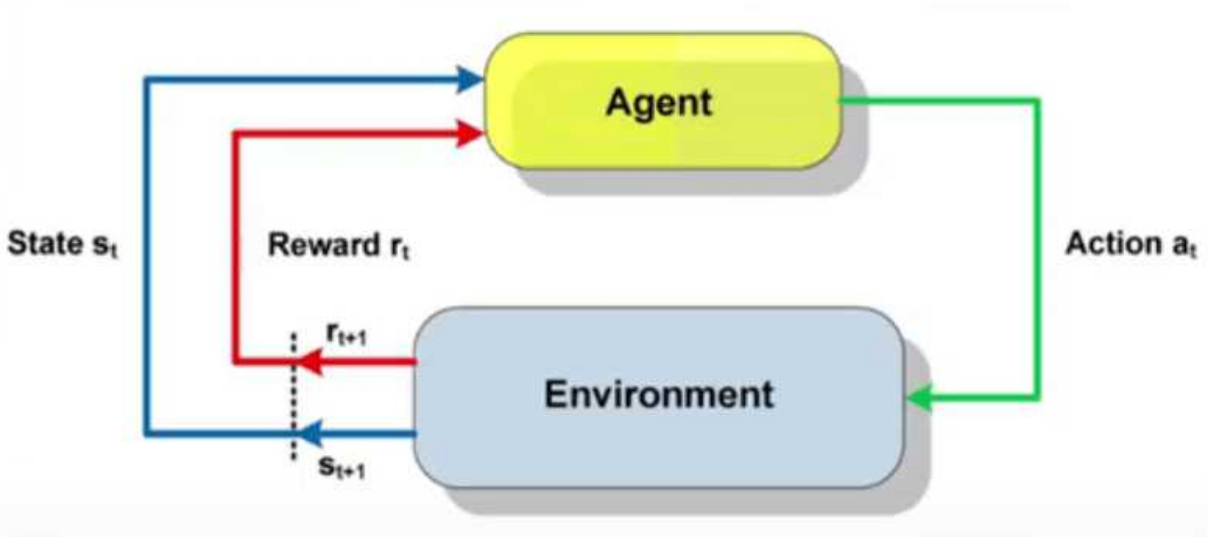


Figura 16 Apprendimento Rinforzata

4.5.2 Esplorazione sicura

L'esplorazione rappresenta un'attività normalmente intrapresa dai sistemi autonomi per raccogliere informazioni sull'ambiente, cercando così di apprendere in modo indipendente. Tuttavia, il sistema generalmente non dispone di una base dati sufficiente che gli consenta di valutare adeguatamente tali azioni, esponendosi pertanto a situazioni potenzialmente pericolose. Quando i possibili errori e rischi sono limitati e possono essere previsti a priori, il progettista è in grado di implementare a livello codificato un modello atto ad evitare comportamenti rischiosi, riducendo in tal modo il rischio associato all'esplorazione [62]. Tuttavia, come già accennato, nel caso in cui il sistema raggiunga elevati livelli di automazione o operi in ambienti complessi, questa tecnica risulta evidentemente inefficace, rendendo necessario l'impiego di altri metodi [62].

Una delle diverse strategie per gestire tale rischio consiste nell'addestrare il modello a esplorare all'interno di un ambiente simulato prima della sua applicazione nel mondo reale [62]. È indubbio che un ambiente simulato non possa replicare tutte le diverse condizioni che possono manifestarsi nel mondo reale; tuttavia, l'efficacia di tale approccio risiede nella sua capacità di consentire al sistema di acquisire una comprensione preliminare dei concetti legati ai rischi, prima di affrontare la realtà. Ciò consente l'adozione di una metodologia di "esplorazione sicura" più cauta. Un'alternativa per mitigare i pericoli associati all'esplorazione del sistema è rappresentata dall'approccio della "esplorazione limitata". Questa tecnica implica che il modello si concentri sull'esplorazione esclusivamente in aree predefinite come sicure, considerando anche lo scenario peggiore possibile.

Sebbene esistano ulteriori metodologie che i progettisti possono adottare, è fondamentale affrontare questa problematica attraverso il metodo più appropriato nel contesto della sicurezza funzionale del sistema.

4.6 Hardware issues

Un sistema di intelligenza artificiale è, per definizione, un insieme complesso di componenti software, algoritmi e componenti hardware; ciò implica che tutti questi elementi devono essere sicuri affinché si giunga a un sistema completamente affidabile. I componenti hardware di un sistema AI comprendono processori, sensori, semiconduttori, memoria e acceleratori AI.

Per chiarire questo concetto, i requisiti relativi ai componenti hardware per sistemi AI integrati nelle funzioni di sicurezza sono delineati dalla norma ISO/IEC 5469, la quale fa riferimento alla norma IEC 61508-2:2010: (*Functional Safety of electrical, electronic and programmable electronic (E/E/PE) safety-related systems- Requirements for electrical/electronic/programmable electronic safety- related systems)* poiché tali requisiti sono equivalenti a quelli applicabili ai sistemi non-AI.

La norma IEC 61508-2 [30] stabilisce che è necessario effettuare una valutazione delle misure di gestione del rischio associate all'hardware, relative a ciascuna funzione di sicurezza, seguendo un insieme di verifiche dettagliate qui di seguito esposte.

• L'architettura del sistema E/E/PE, comprendente la determinazione dei sottosistemi (sistemi incaricati della gestione della distribuzione di potenza e dell'elaborazione) configurati in serie o in parallelo.

• L'architettura dei sottoinsiemi dei sistemi E/E/PE a livello dei loro componenti (semiconduttori, transistor, ecc.).

• Una stima del tasso di guasto per ogni sottoinsieme e i relativi componenti che possono presentare rischi, accompagnata da giustificazioni basate su dati e statistiche.

• La valutazione della suscettibilità dei sistemi E/E/PE che svolgono funzioni di sicurezza a errori e guasti hardware casuali, considerando tutti i tipi di errori che potrebbero causare il fallimento di tali sistemi, anche se non direttamente correlati a difetti nei componenti hardware.

• I punti coperti dai test diagnostici e il tasso di pericolo e rischio derivante da guasti casuali nel hardware dei sottoinsiemi non identificati dalla diagnostica. In questa sede si devono considerare anche i tempi medi per il ripristino (MTTR) e i tempi medi per la riparazione (MRT).

• Gli intervalli temporali durante i quali il sistema è soggetto a test di verifica per l'identificazione di errori potenzialmente pericolosi.

• La valutazione dell'efficienza al 100% delle prove di verifica effettuate, corredata da eventuali giustificazioni.

• La stima del tempo necessario per la riparazione può essere considerata istantanea solo nel caso in cui l'intervento avvenga quando il sistema è disattivato e in uno stato di sicurezza.

• Qualora il sistema richieda l'intervento di un operatore umano per gestire una situazione pericolosa segnalata, deve essere valutato l'impatto di un errore casuale da parte di tale operatore.

Questo insieme di verifiche e prove deve essere applicato a tutti i componenti hardware che costituiscono la funzione di sicurezza al fine di garantire un sistema completamente conforme e sicuro..

4.7 Maturità della tecnologia adottata

Una tecnologia è definita matura se è stata impiegata per un periodo sufficientemente lungo da aver ridotto o eliminato gli errori iniziali riscontrati nei primi modelli. Altre definizioni collegano le tecnologie mature a quelle che possiedono una base tecnica solida e adeguata, anche se non ampiamente adottate. Tale contesto è menzionato nella normativa poiché si deve considerare che la maturità o meno di una tecnologia influisce sulla sicurezza; infatti, una tecnologia poco matura o emergente presenta inevitabilmente delle ambiguità, in quanto le tecniche per la riduzione dei rischi o la gestione dei pericoli non sono ancora chiaramente definite [37]. Tuttavia, ciò non esclude il fatto che anche i sistemi basati su tecnologie relativamente mature possano essere vulnerabili a rischi ed errori. Pertanto, il conseguimento

della maturità non esime dalla necessità di implementare tutte le prove di verifica e le misure di riduzione del rischio, frequentemente sottovalutate, che possono condurre il sistema a situazioni pericolose e rischiose.

5 VERIFICAZIONE E VALIDAZIONE

Una fase fondamentale al termine del ciclo produttivo di qualsiasi sistema, anche quelli privi di funzioni di sicurezza, è rappresentata dalla verifica e validazione (V&V). La verifica implica l'assicurazione che il prodotto sia stato realizzato in conformità ai requisiti stabiliti, mentre la validazione consiste nel processo attraverso cui si accerta che il prodotto finale soddisfi effettivamente le necessità previste; in sintesi, la verifica risponde alla domanda “il prodotto è stato progettato correttamente?” mentre la validazione si riferisce a “il prodotto è quello giusto?”.

I sistemi di intelligenza artificiale, come può essere dedotto dai punti trattati nei paragrafi precedenti, presentano peculiarità tali da rendere necessario un approfondimento sui processi di verifica e validazione quando sono integrati nelle funzioni di sicurezza.

Questa sezione della norma analizza la fase di V&V per i sistemi di intelligenza artificiale basati su Machine Learning; pertanto, sono definiti data-driven, ovvero sistemi il cui apprendimento è fondato sui dati acquisiti.

5.1 le sfide nella verificazione e validazione

5.1.1 Problemi relativi alla tracciabilità

Tale problematica concerne i sistemi AI basati su tecnologie di Machine Learning e pertanto classificabili come data-driven; in altre parole, i modelli e gli obiettivi del sistema sono rappresentati da un dataset sul quale il sistema viene addestrato. L'assenza di un insieme definito di specifiche e regole chiare per il raggiungimento degli obiettivi del sistema costituisce frequentemente un ostacolo significativo nella fase di verifica e validazione. Tale mancanza genera ambiguità per i progettisti che devono valutare l'efficacia e la qualità delle decisioni del sistema.

Un requisito fondamentale per agevolare il processo di verifica e validazione è la tracciabilità, in quanto essa consente una comprensione chiara dei meccanismi operativi, dall'acquisizione dei dati fino alla decisione finale. Alcuni modelli di machine learning, come le reti neurali profonde (DNN), possono manifestare problematiche in questo contesto a causa della loro complessità intrinseca [37]. Pertanto, è imperativo considerare attentamente il tipo di modello da implementare durante la progettazione del sistema; se si opta per modelli complessi, devono essere adottate strategie volte a mitigare gli effetti associati alla loro natura "black box", caratterizzata dalla disponibilità di input e output per i progettisti, mentre ciò che accade internamente rimane oscuro.

5.1.2 Interferenza tra le misure di mitigazione e gestione del rischio

In merito all'interferenza tra le misure di mitigazione e gestione del rischio, la difficoltà risiede nel fatto che molti sistemi di intelligenza artificiale presentano una natura "black box". Garantire un certo grado di indipendenza tra i dati utilizzati per addestrare un determinato modello volto alla mitigazione del rischio e altri modelli rappresenta quindi una sfida significativa. In effetti, la verifica e la validazione devono essere applicate a tutti i sistemi o modelli adottati per la mitigazione e la gestione dei rischi identificati durante la fase di analisi del rischio, in modo indipendente [5]. Inoltre, qualora durante il processo di verifica e validazione del sistema emergesse un nuovo rischio, essendo basato su dati, risulta necessario introdurre un nuovo set di dati per riaddestrare il modello, tenendo conto che questo nuovo dataset potrebbe generare ulteriori fonti di rischio poiché potrebbe influenzare il funzionamento dell'intero sistema [5]. Ciò impone ai progettisti l'obbligo di effettuare una nuova verifica e convalida del sistema. Da ciò si evince l'importanza di disporre di un sistema adeguatamente trasparente e spiegabile.

5.1.3 **V&V del software**

Considerata la particolare natura dei sistemi di intelligenza artificiale, la verifica del software presenta maggiori complessità rispetto ai sistemi non-AI, introducendo così una sfida ulteriore nella verifica e convalida di tali sistemi. Uno degli aspetti cruciali su cui si fonda la verifica del software in un sistema basato su dati è la stima della qualità dei dati; tuttavia, lo stesso riferimento evidenzia ostacoli che complicano significativamente questo aspetto, come ad esempio i costi elevati [75]. La carenza e le limitazioni negli strumenti per la validazione della qualità dei dati rappresentano un altro serio problema capace di rallentare le fasi di verifica e convalida. Tuttavia, tale limitazione si manifesta anche nei modelli e nei parametri utilizzati per la valutazione della qualità dei dati.

Pertanto, tutti questi aspetti rappresentano delle vulnerabilità intrinseche dei sistemi di intelligenza artificiale che devono essere considerati nel contesto della verifica e convalida (V&V), particolarmente quando tali sistemi sono integrati nelle funzioni di sicurezza.

5.1.4 La natura probabilistica dei sistemi AI

Come evidenziato nei paragrafi precedenti, la natura probabilistica dei modelli di apprendimento automatico costituisce un elemento significativamente problematico. Infatti, essendo fondati su principi statistici, questi modelli non presentano determinismo elevato, il che introduce ulteriori complicazioni. Più specificamente, nell'ambito della verifica e convalida, i progettisti devono garantire che le decisioni e gli output generati dal sistema siano conformi ai requisiti stabiliti e operino senza errori; tuttavia, a causa della loro intrinseca incertezza, risulta difficile affermare con certezza che il sistema prenderà sempre la decisione corretta. Naturalmente, questo aspetto nei sistemi di intelligenza artificiale può essere mitigato attraverso l'adozione di numerosi metodi e tecniche.

Tra le prove applicabili ai sistemi di intelligenza artificiale figurano quelle delineate nella norma ISO/IEC TR 29119-11:2020: Software and System Engineering - Software Testing [41]. Tecniche come DeepCover e DeepTest sono specificamente riferite a sistemi di tipo "white box", ma possono essere adattate anche per l'applicazione a sistemi di intelligenza artificiale.

5.1.5 Drift

Il fenomeno del model drift e del data drift è stato oggetto di ampie descrizioni e analisi. È emerso chiaramente che tale concetto è intrinsecamente legato all'ambiente circostante in cui opera il sistema. Pertanto, risulta fondamentale garantire che il sistema progettato mantenga la sua funzionalità anche mesi o anni dopo la sua implementazione, specialmente quando si opera in contesti complessi e dinamici. In fase di verifica e convalida del sistema, è cruciale riconoscere che tali attività non possono attestare la conformità del sistema per utilizzi a lungo termine; ciò implica l'adozione frequente di tecniche di ri-addestramento o ri-convalida, necessarie per raggiungere elevati standard di affidabilità e prestazioni [5].

5.2 Soluzioni

Di fronte ad ogni problematica, la scienza ha sempre cercato soluzioni capaci di eliminare completamente o parzialmente le difficoltà. Anche nel contesto della verifica e convalida (V&V) dei sistemi basati su modelli di machine learning, sono state sviluppate metodologie in grado di soddisfare tali obiettivi. La normativa vigente propone due approcci generali: il primo concerne la valutazione dei livelli di sicurezza ottenuti attraverso metodi di gestione e mitigazione del rischio mediante un'analisi dei processi produttivi del sistema; il secondo si fonda sul principio dell'intelligenza artificiale spiegabile [37].

5.2.1 Primo approccio: Analisi delle fasi di progettazione del sistema

Questo approccio si fonda principalmente sulle definizioni di un insieme di parametri (metriche) quali accuratezza, affidabilità e specificamente in relazione ai dati di addestramento. In particolare, la valutazione e l'analisi in questo metodo vengono applicate in parallelo o in sequenza con le tecniche di mitigazione, specialmente nella fase riguardante i dati di addestramento.

L'efficacia di tale approccio è quella di attenuare l'impatto derivante dal fatto che i pericoli e i rischi possono manifestarsi in ogni fase del processo produttivo del sistema, interessando diversi componenti. Il bias dei dati può rappresentare un esempio di tali rischi, il quale può verificarsi a livello della preparazione e dell'acquisizione dei dati [83].

5.2.1.1 analisi di rischio e distribuzione di data

Come sottolineato più volte, il funzionamento dei sistemi basati sui dati si fonda essenzialmente sui dati utilizzati per addestrare il modello, consentendogli così di apprendere i metodi e i compiti necessari per prendere autonomamente decisioni corrette. È pertanto fondamentale

valutare se la fase di addestramento sia stata adeguata a permettere al sistema di generare output corretti e sicuri; ciò costituisce un elemento centrale nella verifica e validazione del sistema.

Dopo aver stabilito il dataset di allenamento, è fondamentale delineare un dominio operativo ben definito; tali limitazioni si applicano ai dati di input o vengono stabilite attraverso l'identificazione di un insieme di esempi pratici nel mondo reale [5]. Questa delimitazione è utile per determinare i parametri su cui basare le verifiche e le valutazioni, poiché definisce in modo preciso e rigoroso quali siano gli obiettivi richiesti dal sistema. Infatti, tale restrizione contribuisce anche a mitigare l'impatto delle azioni impreviste che il sistema potrebbe intraprendere. Tuttavia, è necessario prestare attenzione a non imporre queste limitazioni in maniera arbitraria o non sufficientemente ponderata, al fine di evitare l'insorgere di ulteriori rischi, come bias o overfitting; ogni rischio identificato deve essere accompagnato da un set di dati adeguato per garantire che il sistema sia in grado di gestire tutte le criticità emerse durante la fase di analisi del rischio. Inoltre, il livello di acquisizione dei dati deve essere affiancato da una fase di analisi e valutazione dei risultati durante la fase di training, consentendo così la risoluzione tempestiva degli errori che possono ostacolare il modello mentre apprende e struttura il suo funzionamento. Questo aspetto risulta imprescindibile, poiché, sebbene si possa ritenere che la distribuzione dei dati fornita per l'allenamento copra tutte le azioni necessarie per gestire e mitigare i rischi, rimane incerto se il modello interpreti tali dati come previsto e atteso.

Pertanto, si evince una certa coerenza nel parallelismo tra l'attuazione delle due fasi di acquisizione dei dati e la verifica e convalida del modello.

5.2.1.2 V&V a livello del modello e preparazione di data

Nei paragrafi precedenti dedicati alle proprietà e ai rischi associati ai sistemi di intelligenza artificiale, abbiamo trattato i potenziali pericoli derivanti dalla natura dei dati impiegati nella fase di addestramento, con particolare attenzione al "data-bias" e all'overfitting. Questi due rischi, come già menzionato, sono fortemente influenzati dalla rappresentatività e dalla qualità del dataset. Per sottolineare l'importanza di tali concetti, la norma ha delineato un insieme di requisiti che possono supportare il processo di verifica del sistema [37]. Tali requisiti si fondano sui principi esposti nella norma ISO/IEC TR 29119-11: 2021 [41], che evidenzia la necessità di testare il modello mediante l'applicazione di tutte le possibili combinazioni di input per garantire la funzionalità e la sicurezza del sistema in ogni scenario contemplato; questo approccio è definito come "dynamic testing".

Per approfondire questi aspetti, la norma ISO/IEC 5469 ha formulato quattro "domande" relative ai dati di prova, le quali devono essere giustificate e soddisfatte per ogni test a cui il sistema deve essere sottoposto; esse sono le seguenti [37]:

- Se, in base all'analisi del rischio, tutti gli scenari pertinenti alla sicurezza funzionale identificati presentano dati rappresentativi nel dataset.
- Per tutti i rischi identificati durante la fase di analisi del rischio, il dataset di prova considera tutte le possibili situazioni e variazioni che potrebbero verificarsi nel mondo reale e generare tali rischi.

- Il dataset di prova è sufficientemente ampio e diversificato per rappresentare tutte le potenziali situazioni associate alle fonti di rischio emerse dall'analisi del rischio, coprendo tutti gli stati possibili del sistema.
- I risultati ottenuti dai test, per ogni situazione di rischio possibile identificata dall'analisi del rischio, sono stabili rispetto alle variazioni prevedibili degli input e, secondo l'analisi umana, appartengono allo stesso gruppo e presentano una natura simile.

Per chiarire ulteriormente come debbano essere soddisfatti questi requisiti, la norma ha fornito esempi di risposte e giustificazioni appropriate.

Ad esempio, relativamente al primo requisito, è fondamentale che il progettista assicuri l'identificazione di un insieme di attributi di dati per ciascun rischio individuato. Questi attributi possono includere condizioni ambientali, parametri operativi e persino comportamenti dell'operatore, al fine di garantire che il modello venga testato in relazione a tutti i possibili scenari che potrebbero manifestarsi nel contesto reale. Questo aspetto sottolinea l'importanza della rappresentatività del dataset di prova.

In riferimento al secondo punto, è imperativo che la progettista del modello assicuri l'inclusione di tutti gli attributi identificati durante la fase preliminare nel dataset di prova, il quale deve risultare rappresentativo per ciascuno di essi. È altresì essenziale individuare tutti i potenziali casi di bias nei dati attraverso un'analisi approfondita della distribuzione dei dati relativi agli altri attributi. Tuttavia, la definizione di tutti questi casi di test e prove necessari per simulare il modello al fine di rilevare possibili situazioni di fallimento ed errori si configura come un compito gravoso sia per gli sperimentatori che per il sistema stesso. Come riportato nella [46] e supportato da numerosi studi, è emerso che il 70% dei sistemi software analizzati, sui quali si fondano i risultati della ricerca, presenta un numero massimo di sei condizioni in grado di generare determinate tipologie di fallimento. Di conseguenza, è stato proposto il concetto di "testing pseudoesaustivo", che implica che ogni tipo di fallimento potenzialmente riscontrabile possa essere "attivato" da un insieme finito e limitato di condizioni [47].

Il "testing combinatorio" suggerito dalla norma ISO/IEC TR 29119-11:2021 costituisce una soluzione praticabile come previsto dalla normativa in oggetto. Il contesto del "combinatorial testing" è stato illustrato dalla norma ISO/IEC IEEE 29119-4:2021 [39] e consiste principalmente nell'identificazione delle coppie di parametri (P) e valori specifici (V), le quali rappresentano queste coppie (P-V). Questi parametri possono rappresentare degli input di prova ,come ad esempio condizioni ambientali, e possiedono certi valori che devono essere gestibili e limitati. Questi parametri possono fungere da input per le prove, come nel caso delle condizioni ambientali, e presentano valori che devono essere gestiti e limitati. Tale tipologia di prova si esprime attraverso diversi metodi e tecniche, tra cui il "combinational testing" e il "pair-wise testing". La distinzione tra le varie tecniche risiede negli elementi di copertura della prova; nel primo metodo, essi sono rappresentati da un insieme appartenente al set di coppie P-V definiti, in modo tale che ogni parametro sia incluso almeno una volta nel gruppo. Al contrario, nella seconda metodologia, è necessario selezionare coppie P-V dal set di coppie definite affinché ciascuna coppia faccia riferimento a un parametro di prova distinto. Pertanto, attraverso la seconda tecnica è possibile testare il modello con un numero inferiore di prove

poiché essa copre, anziché tutte le combinazioni possibili, tutte le coppie. Ulteriori metodologie comprendono il “each choice testing” e il “base choice testing”, anch'esse analizzate secondo la medesima norma.

Riguardo al bias nei dati che potrebbe emergere, qualora si ritenga che il dataset acquisito dal mondo reale non sia sufficientemente rappresentativo o differente, è consentito l'impiego di simulazioni come dati sperimentali. Inoltre, è permesso escludere determinati dati provenienti dal mondo reale se necessario per bilanciare adeguatamente i dati al fine di ridurre qualsiasi effetto di bias o overfitting.

Proseguendo con il terzo requisito, è fondamentale che le tecniche di mitigazione siano in grado di operare efficacemente sotto qualsiasi variazione che il sistema possa incontrare. Pertanto, risulta essenziale verificare la diversità necessaria per garantire che il modello sia adeguatamente predisposto a elaborare differenti input. La fase di acquisizione e preparazione dei dati deve essere accompagnata da una fase di verifica volta ad accertare l'assenza di bias nel dataset di prova. Il bias può manifestarsi in diverse forme, come discusso nella norma ISO/IEC TR 24027:2021[39] Bias nei sistemi AI e nelle decisioni assistite dall’ AI. Ad esempio, un tipo di bias può derivare da un dataset che non riflette accuratamente la realtà, noto come "selection bias". Questo fenomeno può essere causato da una selettività involontaria da parte dell'operatore responsabile della scelta del dataset di prova, un aspetto definito "human cognitive bias"; una soluzione per prevenire tale problematica consiste nell'applicare una certa randomizzazione nel processo di selezione dei dati. Un ulteriore tipo di bias è il "sampling bias", che si origina da un'acquisizione non randomizzata e non rappresentativa della popolazione considerata. Questo tipo di bias si manifesta in modo particolarmente evidente nei sistemi di riconoscimento, come quelli implementati nelle macchine progettate per mantenere una posizione di sicurezza. Se i dati utilizzati per testare il modello non comprendono tutti gli oggetti potenzialmente presenti nell'ambiente in cui la macchina opererà, il progettista non sarà in grado di valutare con precisione la capacità del sistema di riconoscere ogni ostacolo con l'accuratezza necessaria. In questo contesto, la randomizzazione nella selezione degli esemplari di dati rappresenta una strategia efficace per limitare o persino prevenire tale problematica.

Un ulteriore aspetto da considerare, relativo al terzo requisito, concerne l'ampiezza dei dati associati a ciascun rischio. La densità di questo dataset è infatti determinata principalmente dalla probabilità di mitigazione del rischio stesso; se durante l'analisi del rischio si interpreta che un certo rischio possieda una elevata probabilità di verificarsi, risulta evidente che deve essere rappresentato da un insieme di dati sufficientemente ampio.

Un ulteriore elemento che incide sulla larghezza del dataset è rappresentato dalla quantità di dati necessari durante la fase di addestramento. In questo contesto, è fondamentale riferirsi a specifici indicatori di accuratezza, le metriche attraverso le quali si valuta l'efficacia e la qualità delle previsioni generate dal sistema. Queste metriche possono includere la precisione delle previsioni corrette, la sensibilità del sistema e il rapporto tra le previsioni formulate come

corrette e quelle realmente esatte, oltre ad altre misure come l'accuratezza generale, il numero di false previsioni e quelle corrette. L'importanza di tali indicatori risiede nella loro capacità di indirizzare i progettisti verso gli errori che possono manifestarsi quando il sistema viene esposto a dati differenti rispetto a quelli utilizzati per l'addestramento; pertanto, potrebbe rendersi necessaria un'espansione del volume dei dati impiegati per testare il modello [68].

In precedenza, è stato analizzato il rischio associato alla complessità ambientale in relazione al funzionamento sicuro del sistema; per quanto concerne i dati, è stata trattata la questione del data shift. Questo aspetto deve anch'esso essere identificato e mitigato. Infatti, un punto cruciale legato al terzo requisito riguarda la capacità della prova di considerare tutti gli stati del sistema successivamente all'addestramento; pertanto, il data shift, inteso come deviazione dei dati rispetto a quanto osservato nella fase di addestramento, deve essere attentamente preso in considerazione. Pertanto, il set di dati di prova deve essere in grado di riflettere la complessità dell'ambiente e delle situazioni che il modello potrebbe affrontare.

Passando ora all'ultimo punto, relativo alla verifica del funzionamento del sistema sotto variazioni degli input, si fa riferimento, in altre parole, alla gestione del fenomeno dell'overfitting. I metodi per affrontare tale problematica durante la progettazione del sistema sono stati discussi nei paragrafi precedenti. Inoltre, la robustezza rappresenta un ulteriore aspetto legato alla stabilità del sistema in presenza delle variazioni previste degli input. Anche questo concetto è stato analizzato in precedenza nei paragrafi dedicati alle proprietà dei sistemi di intelligenza artificiale; tuttavia, desidero aggiungere un elemento che è stato evidenziato nella normativa per garantire che il sistema raggiunga un livello adeguato di robustezza: la valutazione numerica e diretta di tale proprietà. Questa "quantificazione" della robustezza è riferita a ciò che viene definito "raggio massimo sicuro"; la tecnica si basa sulla valutazione della distanza esistente tra gli input e il cosiddetto "confine decisionale", che rappresenta essenzialmente la distanza nello spazio incorporato tra diverse categorie di dati. Maggiore è la distanza tra questi due elementi, più robusto sarà il modello e quindi sarà in grado di operare come previsto anche quando si confronta con piccole variazioni nei dati. Garantire che i dati di prova siano sicuri e privi di contenuti dannosi dal punto di vista della sicurezza informatica costituisce un ulteriore aspetto rilevante da considerare per soddisfare il quarto requisito.

Tutti i rischi associati all'acquisizione dei dati rappresentano una componente, se così possiamo definirla, di fondamentale importanza nel contesto della produzione e progettazione del sistema.

5.2.1.3 Scelta di metriche di performance

L'integrazione dei sistemi di intelligenza artificiale in diverse tecniche, specificamente nelle funzioni di sicurezza, impone ai progettisti la necessità di valutare le prestazioni del sistema nel raggiungimento degli obiettivi prefissati. Per facilitare tale valutazione, sono state introdotte le metriche e i cosiddetti KPI (Key Performance Indicators). Questi ultimi, secondo la definizione fornita da [54], sono indicatori logici o matematici utilizzati per misurare la prossimità tra i risultati ottenuti e quelli attesi. Esempi significativi di tali metriche includono l'accuratezza, la precisione, la diversità e la rappresentatività dei dati, ecc.

La normativa esamina alcune considerazioni riguardanti queste metriche. In primo luogo, discute il legame tra il significato e l'affidabilità delle metriche da un lato e il numero di prove effettuate nonché l'ampiezza dei dati utilizzati in tutte le fasi di produzione del sistema (allenamento, validazione e test) dall'altro [11]. Infatti, qualora i dati impiegati risultassero limitati, basandosi su principi statistici, le indicazioni fornite da queste metriche non sarebbero né affidabili né accurate. Pertanto, si considera il livello di confidenza della metrica stessa, che è fortemente influenzato dal numero di casi esaminati durante le prove e dalla quantità complessiva di dati disponibili. Il livello di confidenza di una metrica rappresenta un valore numerico che indica la percentuale di accuratezza associata a tale metrica. Ad esempio, nel caso della metrica di precisione, se questa mantiene un'accuratezza dell'80% durante la fase di testing, si può affermare che possiede un livello di confidenza pari all'80%. È evidente che, con un numero limitato di prove e dati, questo livello perde significato, costringendo il progettista a considerare questi due aspetti nella fase di progettazione del sistema, poiché rivestono un'importanza fondamentale per la validazione della conformità del sistema.

Un ulteriore aspetto inerente alle metriche riguarda gli svantaggi potenziali legati al loro utilizzo; infatti, l'impiego delle metriche può offuscare alcuni elementi cruciali relativi alla sicurezza. Ciò avviene perché tendono a sintetizzare le prestazioni del sistema in valori specifici, riducendo così la quantità di informazioni necessarie a descrivere il funzionamento del sistema in tutte le condizioni possibili. Consideriamo ad esempio un sistema anti-incendio installato su un veicolo: sebbene venga rilevata un'accuratezza complessiva del 95%, è preoccupante che il restante 5% possa riguardare l'affidabilità dell'hardware sotto condizioni di vibrazione. Per affrontare questa problematica, è possibile utilizzare metriche supplementari per individuare dettagli precedentemente oscurati, come le classificazioni errate associate alla sicurezza. Questi fenomeni possono includere la frequenza dei cosiddetti "falsi negativi", che rappresentano situazioni in cui il sistema non riesce a identificare una condizione critica per la sicurezza, come nel caso di un sistema anti-incendio che fallisce nel rilevare un incendio. Un'altra categoria è quella dei "falsi positivi", in cui il sistema interpreta erroneamente una situazione sicura come pericolosa; ad esempio, lo stesso sistema anti-incendio potrebbe attivare l'arresto d'emergenza della macchina senza che vi sia realmente una condizione di incendio, il che può risultare dannoso per la produzione poiché interrompe il processo produttivo a causa di segnali errati. Pertanto, le prestazioni del sistema possono essere valutate sulla base della sua capacità di rilevare correttamente incendi effettivi piuttosto che falsi allarmi.

Per quanto concerne tali metriche relative alla sicurezza, esse devono essere progettate affinché possano rilevare condizioni critiche, ovvero quelle più gravi, garantendo così che il modello venga valutato anche in situazioni estreme. È opportuno monitorare queste metriche durante l'implementazione del sistema; ciò consente un intervento tempestivo qualora si riscontri che le metriche non indicano prestazioni adeguate del sistema, in particolare quelle associate alla sicurezza. Questo processo è noto come "monitoraggio sul campo". L'importanza di questo aspetto risiede nella sua capacità di segnalare un potenziale rischio prima che si manifesti, basandosi esclusivamente su un certo degrado degli indicatori di performance. Ciò consente di riportare il sistema a una condizione sicura prima dell'insorgere del pericolo. Pertanto, una volta

identificata una degradazione delle prestazioni, il progettista può intervenire riallenando il modello con dati aggiornati o eseguendo il “tuning del modello”. Questa tecnica si riferisce alla regolarizzazione del modello in modo tale da soddisfare al meglio i requisiti di sicurezza del sistema. Un metodo per conseguire ciò consiste nell'ottimizzare gli "iper-parametri", ovvero quei parametri i cui valori vengono definiti prima della fase di allenamento con l'obiettivo di massimizzare le performance del modello. L'intento è quello di individuare la combinazione ottimale di iper-parametri che consenta al sistema di prendere decisioni e fare previsioni più accurate.

Pertanto, come evidenziato riguardo alle metriche, il loro utilizzo riveste un'importanza fondamentale e risulta estremamente utile nella valutazione del sistema, ma solo quando il progettista considera tutte le variabili che possono costituire punti deboli di tali elementi.

5.2.2 Back to back testing

Per valutare l'esito delle prove effettuate al fine di verificare e convalidare il sistema, è necessario definire un opportuno oracolo di prova. Gli oracoli di prova rappresentano un insieme di meccanismi utilizzati per valutare l'output di un sistema sottoposto a test, attraverso un confronto tra l'output ottenuto e quello atteso [2]. Tuttavia, considerando la natura probabilistica dei sistemi di intelligenza artificiale, la definizione di tali oracoli può risultare complessa e talvolta ardua. Per questo motivo, è stata sviluppata una metodologia di testing nota come "back-to-back testing"[76].

Questa tecnica si fonda sul confronto degli output generati da versioni varianti e indipendenti del sistema, alimentate dallo stesso input [76]. Qualora si riscontrino differenze nel comportamento, si cerca di modificare il modello del sistema al fine di ottimizzarlo. La principale distinzione tra questa metodologia e quelle tradizionali consiste nel fatto che la sua efficienza non è influenzata dalle limitazioni degli oracoli di prova; essa dipende infatti dalla coerenza tra le due versioni indipendenti piuttosto che dalla corretta definizione dell'output target. Tuttavia, come per tutte le tecniche, esistono alcune considerazioni fondamentali da tenere in conto affinché risulti efficace. Uno degli aspetti più rilevanti riguarda la necessità che i vari sistemi siano sufficientemente indipendenti per evitare la condivisione degli stessi errori, il che potrebbe compromettere la validità della verifica. Tale problematica può essere affrontata mediante l'impiego di sistemi dotati di algoritmi, configurazioni o parametri differenti.

Successivamente, gli scenari di prova devono riflettere in modo adeguato l'ambiente operativo del sistema, consentendo così ai progettisti di testare il sistema in relazione a diverse situazioni probabili. Anche in questo contesto, è fondamentale considerare le assunzioni e le varie limitazioni dei sistemi comparati per garantire un'analisi esaustiva e precisa.

Tuttavia, talvolta i metodi precedentemente esaminati si rivelano insufficienti, data l'elevata complessità che caratterizza il sistema; pertanto, la norma suggerisce di apportare alcune modifiche al sistema stesso per passare a un livello di utilizzo di tipo C, limitando così l'intervento dell'intelligenza artificiale nel sistema [37].

5.2.3 *System-Level Testing*

Un elemento essenziale nel processo di validazione e verifica è rappresentato dal testing a livello di sistema, il quale si basa principalmente sulla valutazione delle interazioni tra i vari componenti software e hardware del sistema complesso, con l'obiettivo di identificare il funzionamento globale dello stesso. Questo tipo di prova si fonda prevalentemente su simulazioni che possono essere realizzate in forma virtuale piuttosto che nel mondo reale. È opportuno osservare che le prove condotte nell'ambito reale sono generalmente svantaggiose da un punto di vista economico, poiché risultano costose; inoltre, talvolta si dimostrano pericolose, il che ne limita l'utilizzo. Infine, un sistema valutato esclusivamente nel contesto reale non garantisce che tutte le condizioni siano sempre soddisfatte. e quindi non sicuramente conforme [50]. Tuttavia, ciò non esclude la necessità di ricorrere a determinati strumenti in specifiche circostanze, soprattutto quando il sistema deve essere configurato in un contesto complesso. Infatti, in tali situazioni, il simulatore potrebbe non riuscire a rappresentare con precisione tutte le variabili e gli elementi pertinenti. Pertanto, risulta vantaggioso applicare entrambe le tecniche per sfruttarne i benefici reciproci.

Passando all'analisi dei simulatori virtuali, si evidenzia come essi assistano i progettisti nell'identificazione di rischi e pericoli che potrebbero non emergere durante le fasi di produzione del sistema. Ciò è possibile poiché è possibile creare diversi scenari che emulano la realtà, inclusi eventi che si verificano raramente nel mondo reale. Di conseguenza, tra i molteplici vantaggi offerti da queste tecniche, il più significativo risiede nella capacità di valutare la sicurezza funzionale del sistema in scenari critici che possono manifestarsi con bassa frequenza nella realtà.

5.2.3.1 Considerazioni sulle prove virtuale (simulazioni)

Nell'introduzione precedentemente menzionata è stato sottolineato che le simulazioni virtuali presentano una serie di vantaggi per quanto concerne la valutazione e la verifica del sistema; tuttavia, tali benefici si manifestano in modo particolare sotto determinate condizioni rispetto alle prove condotte nel mondo reale.

Riferendo alla norma in studio, risulta opportuno sfruttare la simulazione perché:

- Talvolta, le prove nel mondo reale si rivelano costose e presentano rischi significativi in termini di sicurezza. Un esempio dell'impiego della simulazione per tali motivi è rappresentato dal programma “Gazebo” [25]. Questo simulatore consente agli ingegneri di valutare le prestazioni di un robot industriale o di un cobot, analizzando come questi interagiscono con ambienti dinamici nei quali saranno collocati. Questo aspetto, ad esempio, non può essere realizzato attraverso test nel mondo reale, poiché risulterebbe estremamente pericoloso collocare un robot non ancora adeguatamente sicuro in un

ambiente condiviso con operatori umani al fine di verificare la sua capacità di evitare ostacoli mobili e identificare i percorsi ottimali in tali situazioni.

- Il modello è contraddistinto da un ampio dataset di input e, mediante l'uso di un simulatore, è possibile iterare su vari input, consentendo così una valutazione del sistema sotto diverse condizioni; ciò risulta quasi inefficiente utilizzando tecniche tradizionali di prova. Un esempio di un sistema che richiede un vasto dataset di input è il veicolo a guida autonoma, il quale deve essere in grado di elaborare numerose informazioni in tempo reale, incluse le condizioni ambientali, il traffico e il comportamento dei pedoni. A tal fine, è stato sviluppato il programma di simulazione "CARLA" [70], capace di generare ambienti con varie variazioni delle condizioni precedentemente menzionate, permettendo così la validazione del comportamento del sistema sotto diverse circostanze tenendo conto della diversità degli input che potrebbe incontrare.
- L'impiego di simulatori virtuali consente di accelerare il ciclo produttivo del sistema, poiché offre ai progettisti la possibilità di modificare rapidamente le condizioni di prova in pochi secondi, a fronte dei prolungati tempi richiesti per la configurazione delle prove nel mondo reale [20]. Tale approccio facilita un più tempestivo rilevamento degli errori e dei rischi precedentemente non identificati.
- Una tipologia di simulazione particolarmente diffusa è quella relativa all'iniezione di errori [81]. Questo metodo si fonda sull'introduzione intenzionale di diverse tipologie di errori nel modello al fine di valutare la capacità del sistema di gestirli e rispondere in modo sicuro e accettabile. L'uso della simulazione per emulare questa metodologia consente agli operatori di introdurre errori che risulterebbero complessi e rischiosi da implementare attraverso test nel mondo reale. Un esempio pertinente riguarda l'introduzione di un ritardo in un sensore termico all'interno di un sistema antincendio, permettendo così l'osservazione della risposta del sistema a un incremento della temperatura nella macchina; tale esperimento sarebbe laborioso se condotto nel contesto reale.
- Le simulazioni, essenzialmente ancorate a algoritmi basati su regole predefinite e chiaramente delineate per i progettisti, non evidenziano gli effetti degli errori sistematici. Tali errori, riconducibili a misurazioni ripetibili nel contesto del mondo reale, si manifestano a causa delle imperfezioni intrinseche agli strumenti di misura e ai sensori, nonché della gestione delle condizioni ambientali e dei disturbi; in altre parole, rappresentano la non idealità insita nella realtà [63]. L'effetto di tali errori sulla valutazione consiste nell'influenzare l'accuratezza delle stime relative alla performance del sistema. Poiché le simulazioni sono effettuate in ambienti virtuali dove è possibile esercitare un controllo totale e le stime fornite dai sensori non dipendono più dall'hardware e dalle sue imperfezioni, la valutazione risulta pertanto accurata e stabile.

Tuttavia, nessun sistema può essere considerato ideale; pertanto, vi sono alcune considerazioni da tenere in debito conto quando si utilizzano tali sistemi di simulazione. Il primo aspetto concerne la fedeltà del simulatore, definita come il grado di accuratezza con cui la simulazione riflette e rappresenta i vari aspetti del sistema reale, costituendo un elemento cruciale per

l’efficacia della valutazione condotta mediante essa [12]. Il simulatore deve essere dotato di un livello di fedeltà adeguato e ben equilibrato, variando in relazione al sistema oggetto di test. In effetti, è evidente che un simulatore con elevati standard di fedeltà offre vantaggi significativi in termini di accuratezza e stabilità del test; tuttavia, tale configurazione richiede una considerevole potenza computazionale, limitandone così l'applicabilità, soprattutto in situazioni estreme. D'altro canto, un simulatore caratterizzato da un basso livello di fedeltà presenta svantaggi legati a una ridotta precisione e accuratezza, ma risulta meno oneroso dal punto di vista computazionale [12]. Pertanto, si evidenzia una certa trade-off tra la complessità elaborativa e l'accuratezza del sistema. Di conseguenza, il raggiungimento di un livello di fedeltà adeguato all'applicazione da valutare richiede una comprensione approfondita del sistema sottostante, dei suoi requisiti e delle sue caratteristiche in termini di sicurezza, nonché dell'impatto negativo derivante da una scarsa accuratezza sulla valutazione complessiva del sistema. Una possibile soluzione a questa problematica è rappresentata dalla "Multi-fidelity Optimization with Ordinal Transformation & Optimal Sampling" [77], tecnica che mira a sfruttare la rapidità computazionale dei simulatori a bassa fedeltà insieme all'accuratezza offerta da quelli ad alta fedeltà. L'idea fondamentale di questa metodologia consiste nell'applicare inizialmente una simulazione a bassa fedeltà, al fine di valutare l'efficienza in diverse dimensioni, per poi, laddove necessario e sulla base dei risultati ottenuti nella fase preliminare, implementare una simulazione ad alta fedeltà. Questo approccio consente di raggiungere una simulazione ottimizzata dal punto di vista della fedeltà.

È stato precedentemente sottolineato che il testing a livello di sistema concerne la valutazione delle prestazioni complessive del sistema stesso e le modalità con cui i vari componenti interagiscono tra loro sotto differenti condizioni. Ciò spiega perché i tipi di simulatori utilizzati nei sistemi di intelligenza artificiale per valutare la loro sicurezza funzionale si suddividono essenzialmente in tre categorie principali: model in loop (MIL), software in loop (SIL) e hardware in loop (HIL). Di seguito, fornirò una breve definizione di ciascun tipo di simulazione.

- MIL: la prima fase nella realizzazione di un sistema AI, dopo aver fatto un certo prototipo del hardware che sarà essenziale per capire come controllare il sistema, è quello di realizzare il modello di controllo teorico. Una volta fatto, il modello deve essere testato per valutare la sua logica e vedere se fa quello da cui richiesto in modo corretto e sicuro, e per tale ragione è stato introdotto il MIL che permette tramite un certo software di simulazione, ad esempio Simulink, di emulare il sistema fisico e quello di controllo in modo completamente virtuale [58]. Questo tipo di testing è accompagnato da una riduzione del costo della prova in quanto limita la richiesta della prova fisica solo ai componenti essenziali di dinamicità incognita. Esso permette anche di giocare sulla configurazione del modello simulato per vedere come raggiungere al modello più ottimizzato senza necessità di cambiare o introdurre nuovi componenti sul sistema fisico attuale.
- SIL: il modello di controllo realizzato alla prima fase viene rappresentato da un certo codice per essere implementato attualmente nel sistema attuale [9]. Quindi, prima di passare alla realizzazione del sistema a livello di hardware viene verificato il software e viene valutato nelle prime fasi della sua progettazione. Tale tecnica permette alla

progettista di gestire e testare varie configurazioni del software e dell'algoritmo con flessibilità, in quanto si può veder l'interazione del software con le varie componenti del sistema emulati col simulatore dove l'hardware attuale non viene considerato.

- HIL: L'idea base di questo tipo di simulazione, e che normalmente risulta necessario, è quello di includere all'interno del loop di simulazione le parti del hardware che risulta complesso emularli con i software di simulazione, come ad esempio gli attuatori che sono estremamente difficile da modellare [10]. A questo livello, la simulazione è una simulazione a tempo reale e questo ha il vantaggio di verificare che il sistema di controllo embedded riesce ad eseguire il suo lavoro entro il periodo di tempo richiesto.

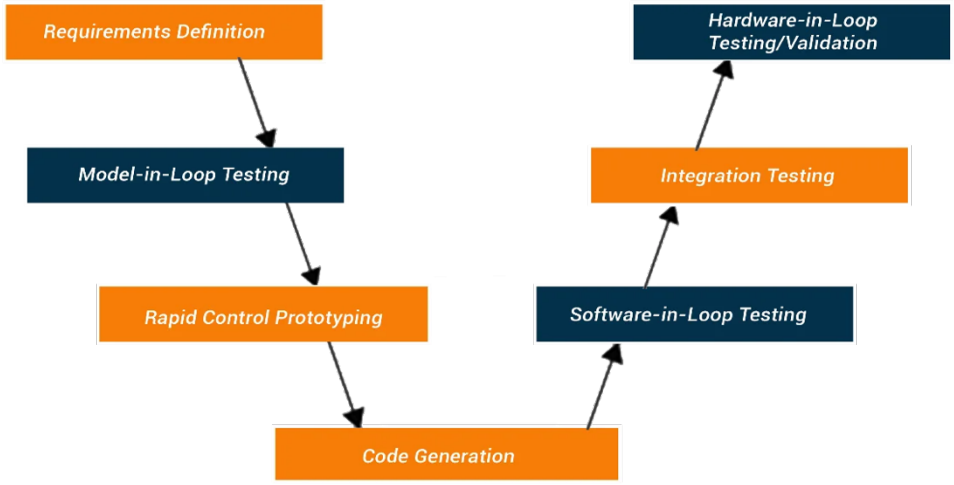


Figura 17 V-model per lo sviluppo e la verificazione dei sistemi di controllo

Un altro aspetto da notare quando si considera la validità delle simulazione è quello nominato come "test-coverage approach" è che si concentra sul garantire che il sistema di intelligenza artificiale sia testato a fondo nelle diverse dimensioni del suo spazio di input [65]. Per tale ragione sono già trovate degli approcci che tendono a soddisfare tale requisito che risulta fondamentale vedendo la criticità dei sistemi in studio in termine della sicurezza. La norma ha fatto un elenco di tale diverse tecniche che definisco bravamente in seguito [37]:

- Random test sampling : dal nome, è una tecnica in cui l'operatore sceglie un set di data dal input in modo randomico, permettendo cosí di valutare il performance del sistema su varie scenari e set di input probabili.

- Constrained test sampling: è l'opposito del del approccio randmico, dove l'operatore esegue il test su specifiche input ma giustificando tale scelte.
- Distribution-based test samplig : approccio basandosi sul profilo dell'utilizzatore riflette come il sistema sarà applicato e utilizzato.
- Criticality sampling : è questo approccio considera tutti gli input legati alla verifica dell asicurezza del sistema e tutte le funzioni critiche in termini della sicurezza.
- Stress-based sampling: tecniche per lo studio del comportamento del sistema nei casi limiti ed estremi.

E quindi, in relazione a tale concetto viene esposta la domanda: come capire se le simulazioni fatte ed eseguite risultano sufficiente per garantire la sicurezza funzionale del sistema?

Infatti, il numero delle simulazioni da fare è influenzata da varie fattori, come ad esempio la complessità del sistema in studio; è ovvio che più complesso è il sistema più sono le simulazione che si devono subire per assicurare la copertura delle sue varie aspetti. Poi, e basandosi sull'analisi del rischio e la possibile situazione di pericolo che può scontrare o introdurre il sistema, viene definita se si serve un numero elevato o limitato di simulazioni. La complessità dell'ambiente, e le possibili scenari che possano verificarsi; la dimensione del dataset di input, la criticità delle diverse funzioni, tutti questi aspetti influenzano la quantità e le tipologie delle simulazioni su cui deve essere sottoposto il sistema al fine di verificare la sua conformità in termine di sicurezza.

L'efficienza della simulazione nella verificazione e validazione del comportamento di un sistema AI, dipende molto sulla scelta degli instrumenti di simulazione, e che devono che si basa su 4 concetti fondamentali: fit-for purpose, capacità, correttezza, e l'accuratezza.

Innanzitutto, il primo punto riguarda l'adattabilità dello strumento con gli obiettivi generali e le caratteristiche del sistema da testare, e questo è stato riferito a come il "fit-for purpose"[27]. Il criterio per analizzare tale idoneità del simulatore si basa essenzialmente su diversi punti che riguardano aspetti diversi del sistema da simulare e testare. Come punto principale da riguardare è la fonda conoscenza dell'obiettivo principale del sistema ed il suo scopo. Ad esempio, se consideriamo un sistema di antincendio integrato in una macchina industriale, il suo scopo principale è quello di rilevare possibili incendi che si verifichino internamente dalla macchina, e quindi il simulatore deve testare se il sistema AI riesce a distinguere le varie tipi di incendi che possono verificarsi ed arrestare rispettando le limiti di tempo definiti e indicati per esso. Il secondo punto riguarda le definizioni delle boundary conditions di funzionamento del sistema; un sistema di antincendio implementato per una macchina industriale normalmente, ad eccetto di casi particolari, funziona quando la macchina sia in operazione, e quindi testarlo in condizioni di macchina fuori servizio o in arresto risulta irrilevante, e questo deve essere preso in considerazione quando si considera il modello di simulazione. L'ambiente di operazione è un altro fattore che ovviamente influenza la scelta degli strumenti di simulazione; il simulatore deve rappresentare bene le condizioni di temperatura in cui viene implementato il sistema e messa ad operazione, così per permettere una verifica efficiente della sua interpretazione della situazione. Poi, il livello di complessità del modello di simulazione dipende sulla funzionalità

del sistema; un sistema di antincendio deve rilevare i surriscaldamenti ed i scintilli che possono ad esempio generarsi dal motore della macchina, e quindi il modello del simulatore deve essere progettato in modo tale da permettere emulare questi effetti accuratamente per vedere come reagisca il sistema. E l'ultimo punto, che è quasi ovvia, è che il simulatore non deve essere utilizzato per testare il sistema in condizioni che risultano irrilevanti per il sistema, e questo ovviamente per velocizzare il processo di verificazione e validazione del sistema e ridurre i costi relativi.

Il secondo punto da notare è legato alla capacità e l'efficienza del simulatore stesso a permettere la rilevazione accurata dei possibili errori del sistema e i rischi associati ad essi. A tale riguardo vengono definiti le assunzioni e le limitazioni degli strumenti del simulatore, e sono un set di considerazioni che riguardano le proprietà generali che assumono tali strumenti quando replicano il mondo-reale. Tra questi c'è ad esempio l'ipotesi dell'idealità delle varie componenti del sistema da emulare come i sensori, i componenti di comunicazione, l'assenza di latenza nelle varie componenti del hardware, oppure la stabilità nelle condizioni ambientali. Tali "difetti" nella rappresentazione del mondo reale sono un risultato della complessità di emulare esattamente il mondo reale. Questo aspetto risulta importante in quanto rappresenta un punto fondamentale per passare dalla valutazione ottenuta dalla simulazione al mondo reale, che alla fine è l'obiettivo principale della simulazione. La fidelità anche e che è stata descritta in dettaglio in precedenza, è un altro aspetto che deve essere considerato nella scelta degli strumenti del simulatore e per le stesse ragioni che sono stati spiegati per quanto riguarda tutto il simulatore; è da notare che la fidelità di tali strumenti serve ad assistere anche la fidelità delle metriche e degli indicatori di valutazione della performance del sistema. E vedendo tale "deviazione" tra il simulatore e il mondo reale, i progettisti devono definire un certo livello di tolleranza di questa deviazione e deve essere accompagnato da una certa giustificazione perché sia sufficiente questo livello di tolleranza.

La correttezza degli strumenti di simulazione è un concetto che riguarda la verificazione di essi, e quindi valutare la loro robustezza ed efficienza nel realizzare il loro lavoro. Questo concetto riguarda gli algoritmi ed i modelli matematici su cui sono basati i strumenti del simulatore. Infatti, la loro verificazione non differisce tanto dalla verificazione di qualsiasi sistema o modello di software. E questo deve garantire che tali strumenti non subiscano azioni che non possano verificare nel mondo-reale per certi input su cui non sono stati convalidati. Quindi deve assicurare che hanno un livello di robustezza tale per cui riescono a comportare in modo realistico qualsiasi era l'input o il dataset che devono elaborare. E come qualsiasi modello o sistema di software la verificazione deve essere fatta a livello di codice, di calcolo e sensibilità a qualsiasi variazione nei parametri che descrivono il modello stesso.

L'ultimo aspetto è quello che riguarda l'accuratezza del test virtuale a riprodurre il data d target e che basa essenzialmente sulla capacità degli strumenti a fare il loro compito, e quindi quello richiesto da essi , e questo è ovviamente il concetto della validazione di tali strumenti. Vedendo che un simulatore deve replicare il mondo reale e il sistema da testare esso si compone da vari modelli dove ciascuna rappresenta un componente essenziale del mondo reale; infatti, c'è i

modelli che replicano i sensori, quelli che riguardano le condizioni ambientali altri per modellare l'hardware del sistema attuale. E quindi tutti i modelli costituenti di un certo simulatore devono essere validati per assicurare la convalidata del simulatore completo.

Ma come accennato prima, esistono 3 categorie di simulatori e ciascuna è caratterizzata dai suoi strumenti specifici, e quindi per arrivare ad un sistema di simulazione e di prova virtuale convalidato e efficiente tutte le varie punti discussi sopra devono essere applicati a tutti i tre categorie MIL, SIL, e HIL.

Questo per quanto riguarda il virtual testing (simulazione) e la sua utilità in termini della V&V dei sistemi AI. Però alcune volte, e abbiamo accennato questo fatto precedentemente, l'applicazione di tale tipo di testing risulta insufficiente per garantire la conformità del sistema, il che richiede l'assistenza del physical testing e che sarà tra breve discusso.

5.2.3.2 Physical testing

Quindi, come già accennato prima, a volte la modalità del virtual testing non risulta sufficiente ed efficiente da sola, il che serve l'assistenza dal physical testing. Il physical testing, è la tecnica di valutazione della performance del sistema ed il suo comportamento nel mondo-reale, e quindi in un ambiente reale anziché virtuale, e questo lo interpreta la progettista qualora vede che la simulazione non riesce a dare risultati chiari, oppure il simulatore ha tante limitazioni tale da non fornire risultati fedeli.

Ma ovviamente, c'è un paio di considerazioni da notare quando si utilizza questa metodologia di prova per la convalidazione del sistema. Tali considerazioni ovviamente sono dichiarate dalla norma, vedendo la criticità dei sistemi AI in studio.

Innanzitutto, le prove da subire devono essere ben strutturate, cioè ben progettati e basati su condizioni di controllo predefiniti, e questo ovviamente richiede una conoscenza profonda del sistema e il suo scopo finale. Infatti, Gli input delle prove di questa natura si basano su requisiti di sicurezza, tecnologici come ad esempio le limitazioni dei sensori o altre considerazioni che riguardano i componenti di hardware, e l'obiettivo principale del sistema. Ad esempio, se consideriamo un robot industriale esso deve essere alimentato da un sistema di monitoraggio della posizione di sicurezza e che può essere basato su un sistema AI; allora, se vogliamo verificare l'efficienza di tale sistema tramite un test fisico, le progettiste devono genrare un ambiente di prova ben controllata e quindi una traccia di prova su cui vengono posizionati degli ostacoli predefiniti e di diversi dimensioni e forme tale da valutare l'efficienza del sistema a rilevarli ed evitarli basandosi sull'analisi fatta dal monitor. E quindi l'ambiente di controllo deve essere ben controllato e chiara.

Un altro punto riguarda il fatto che, normalmente, il physical testing, non è considerato opportuno per la verificazione e validazione di un sistema AI che è integrato in una funzione di sicurezza, e quindi deve essere considerato come un passo assistente al testing virtuale tramite simulazioni. Questo perché sono costosi, e complessi da gestire e da sistemare e consumano

tempo anche. E quindi risulta opportuno sfruttare i vantaggi di entrambi metodologie. In effetti, con un test combinato (virtual & physical) anziché testare il sistema in un ambiente ben conosciuta e ben controllata come nel caso del structured testing, con l'utilizzo di entrambe le tecniche si possono testare il sistema su condizioni non predefiniti; prendiamo l'esempio di prendiamo l'esempio di un sistema di antiincendio implementato in una macchina industriale che deve essere testato in corrispondenza di una condizione di surriscaldamento della macchina.

- Nel caso di un test strutturale il sistema viene sottoposto a condizioni ben definiti e sotto controllo; quindi, oltre alle condizioni che ha generato e definito la progettista il sistema non si trova a gestire una situazione incognita e randomica, e quindi la progettista conosce bene come deve rispondere il sistema.
- Nel caso invece di una metodologia combinata:
  - Virtual testing: tramite il simulatore si possono introdurre dei malfunzionamenti al sistema di sensing vedendo come risponde il sistema a tale situazione, e che risulta impossibile rilevare con un test fisico in quanto risulta rischioso e complesso. E quindi si possono verificare se il sistema riesce a gestire la situazione una volta rilevato un errore in un componente di hardware.
  - Physical testing: si possono studiare come influenzano ad esempio le altre macchine il comportamento del sistema che non sono state considerate nelle simulazioni; se ci sono interferenze elettromagnetiche, vibrazioni introdotti o disturbi e questo non si può capire se non viene testato attualmente nel mondo-reale.

Quindi la combinazione di entrambe le tecniche risulta in un sistema più robusto e fedele in quanto i risultati delle prove si basano su una varietà importante degli scenari e situazioni che può incontrare il sistema.

Oltre a questo, a volte le prove fisiche vengono applicate per verificare i risultati ottenuti dalle simulazioni, e quindi ovviamente ci deve essere verificata una certa corrispondenza tra entrambi test.

Ulteriormente, i test reale devono essere sempre accompagnate con dei feedback su qualsiasi incidente che può verificare durante una certa prova, anche se esso non risiede negli interessi della prova stessa. Questo aspetto risulta importante in quanto permette una rilevazione di errori incognite al progettista sfruttando così la randomica dell'ambiente reale e valutando come risponde il sistema a tali condizioni variabili e randomici.

Un altro aspetto che deve essere riguardato sono i domini di prova e sono dei limiti entro quale il sistema deve comportare efficacemente e in sicurezza. Tali limiti sono quelli operativi-come, ad esempio, prendendo il sistema di antincendio, tali sistemi di solito operano entro certi limiti di temperature che possono supportare i sensori, le vibrazioni sotto quali possono comportare bene, oppure anche i tipi di incedi che possono rilevare- e quelli dell'ambiente operativa, come i polveri, umidità, il flusso d'aria che potrebbero influenzare gli strumenti di sensing del sistema. Questo è importante per fornire una valida convalidazione del sistema che sarà messo ad operazione sotto gli stessi condizioni. Poi anche l'interazione con altri sistemi del sistema

sotto studio deve essere testata; nel caso del sistema antincendio esso deve attivare l'arresto di emergenza ma anche un sistema di segnalazione. E come già enunciato la valutazione della validità o meno del test viene definita con l'utilizzo di metriche di performance di cui ho fatto un discorso nei paragrafi sopra precedenti.

Ultimatamente, e come passo per garantire che le prove sono convalide e affidabile, il test deve avere una certa significazione statistica, cioè una distribuzione statistica per chiarire che i risultati di prova su cui è stata verificata la conformità del sistema non applica solo per i scenari su cui è stata testato il sistema, ma anche per altri scenari che possano accadere durante l'operazione effettiva del sistema. Questa natura statistica si può realizzare in base alla natura del test e la funzionalità che deve convalidare. Esistono infatti diverse modalità di quello descrivo alcuni. Una delle tecniche che riguarda la verificazione del software è la replicazione, e si basa sulla ripetizione del test un paio di volte per vedere se i risultati si possono essere riprodotti [43]. Se simili risultati escono da diversi test allora questo punta il fatto che essi non sono valori randomici ma anzi sono delle proprietà intrinseche del sistema. Questa modalità aiuta a rilevare errori randomici che possano verificare nel sistema, oltre alla definizione di una certa distribuzione entro la quale il sistema funziona come aspettato da esso. Questo per quanto riguarda la validazione, e cioè verificare che il sistema esegue i compiti richiesti da esso in modo giusto e conforme. Per la fase di verificazione invece dove la progettista deve valutare la sicurezza funzionale del sistema, si tende a realizzare il test basandosi su un dataset di prova e questo perché si deve valutare il comportamento del sistema in vari scenari e condizioni; la natura statistica della prova si manifesta qua in base all'ampiezza del dataset e la sua copertura e rappresentabilità del mondo-reale. Un esempio dove si può applicare tale concetto e nella valutazione della sensibilità del sistema nella rilevazione di operatori umani.

Alla fine di questa panoramica del testing virtuale e quello nel mondo-reale, si vede che con le giuste misure e tecniche ogni metodo di testing può essere ottimizzato, e la norma fornisce una buona visione a riguardo di tali metodi.

Vulnerabilità agli Errori Hardware Randomici

La norma IEC 61511-1: "Functional safety - Safety instrumented systems for the process industry sector" definisce i guasti randomici a livello di hardware come I guasti che possano accadere in modo randomico in qualsiasi momento a causa di una degradazione dei meccanismi a livello di Hardware del sistema. esempi di tali guasti possono essere bit-flip nella memoria RAM, errori a livello del processore (ALU), errori randomici a livello dei sensori, oppure anche a livello dei circuiti di alimentazioni del sistema ..etc.

Quasi tutte le ricerche hanno proposto la metodologia del fault injection per gestire e mitigare tali errori [28,14]. Tale approccio rileva I punti deboli del sistema prima che essi diventano

degli guasti serie e critiche, tramite l'iniezione volontaria di errori per vedere il comportamento del sistema quando si scontra situazioni anormali nel sistema.

Una tecnica può essere quella di introdurre errori su diversi aspetti del sistema per capire chi dei componenti risponde in modo sicuro ad un errore di una certa natura e chi invece porta il sistema ad una situazione pericolosa. Dando a tale procedura un aspetto statistico, come quello già descritto, fornisce una certa stima di un livello di confidenza che il sistema sia resiliente e robusta contro tali aspetti.

5.2.4 Monitoraggio e feedback

In base alle varie punti discussi prima e che riguardano le diverse aspetti particolari dei sistemi intelligineti, e specialmente quelli basati su modelli ML, e vedendo la criticità dei sistemi che realizzano funzioni di sicurezza, la norma suggerisce un approccio fondamentale per garantire che il sistema sia verificato sicuro durante la sua messa in operazione dopo la dichiarazione della sua conformità. Questo si realizza tramite tramite un continuo monitoraggio delle varie componenti del sistema e riportando tutte gli incidenti che possano verificarsi durante l'operazione el sistema. Applicando tale tecnica permette di migliorare l'affidabilità del sistema in termini di sicurezza, in quanto un feedback sugli incidenti che si verifichino danno i progettisti la possibilità di migliorare gli aspetti che non potevano rilevare nella fase di testing del sistema. Poi un altro vantaggio è relativo al miglioramento della fase di verificazione e validazione del sistema, e questo perché le varie feedback e l'analisi delle condizioni sotto quali sono stati verificati tali incidenti dare ai progettisti l'opportunità di generali nuovi scenari di prova che non sono stati considerate prima, e quindi aumentare la robustezza del sistema e resilienza contro ampie situazioni.

Per alcune funzionalità del sistema, e possono essere quelli relativi alla sicurezza, risulta difficile garantire tramite i modelli matematici o osservazioni e tecniche empirci che esse si operano sempre in modo sicuro durante la sua operazione. Per tale ragione, la norma ha suggerito di definire un certo "tasso di fallimento tollerabile di target" di tale funzione giustificandolo, in modo empirico ad esempio. Questo concetto è legato fortemente al livello di sicurezza integrato (SIL) definito per un certo sistema, ed il che la norma IEC 61508-4 l'ha dato valori da 1 a 4 con livello di sicurezza garantita crescente. Sotto inserisco le tabelle prese dalla parte 1 della norma IEC 61508 e che elencano la media per ora della frequenza di un guasto pericoloso nella funzione di sicurezza in funzione del SIL determinato.

Safety integrity level (SIL)	Average frequency of a dangerous failure of the safety function [h^{-1}] (PFH)
4	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-6}$ to $< 10^{-5}$

Tabella 3 Safety integrity levels – target failure measures for a safety function operating in high demand mode of operation or continuous mode of operation

Alla luce di quanto esposto, si vede che la norma da un riferimento anche per garantire la sicurezza del sistema anche dopo la sua messa in operazione, il che può essere utile nel realizzare un'applicazione affidabile e robusta, e più che altro fiduciosa per un utilizzatore finale.

5.2.5 SECONDO APPROCCIO: EXPLAINABLE AI

Abbiamo accennato nell'introduzione delle tecniche per verificare e convalidare un sistema AI che una delle due metodologie si basa sul concetto dell'AI spiegabile (explainable AI). L'explainability come definita dalla norma EN ISO/IEC 22989:2023 è la proprietà che può avere un sistema AI e si manifesta nel fatto che i fattori che influenza in gran modo le decisioni presi dal sistema sono espressi in modalità chiara per l'intelligenza umana. E quindi risulta opportuno sfruttare tale aspetto per validare il sistema, in quanto è reso disponibile alla progettista il ragionamento che sta facendo il sistema e quindi può verificare la sua giustezza o meno. Però questo si può considerare solo se il sistema è sufficientemente spiegabile, e questo dipende in gran parte sul tipo di intelligenza che assume il sistema; infatti, la stessa norma dichiara che i sistemi AI di tipo DDN (deep neural networks) sono problematiche da spiegare e questo perché rappresentano dei modelli molto complessi.

Una possibile soluzione quando risulta inconveniente o complesso provare a realizzare un sistema spiegabile, è come la norma propone, è quello di implementare il modello logico del sistema AI utilizzando algoritmi di software tradizionale, e quindi semplificare la natura del modello permettendo così la referenza per quanto riguarda la sicurezza funzionale del sistema alle norme tecniche già presenti. Questa implementazione si basa sull'interpretazione della progettista osservando il comportamento del sistema durante la sua operazione.

Un altro punto che la norma discute per quanto riguarda l'utilità dell' AI spiegabile, è quello che anche se il livello di interpretazione del sistema non risulta sufficiente, e per ora non sono ancora trovati delle tecniche garantite per realizzarlo, si può riferire ad alcune tecniche in tale contesto per capire un po' la logica con cui evolve il sistema; una delle varie tecniche è quella del "heatmap" [74]. Heatmap è una rappresentazione grafica che evidenzia gli elementi importanti in base a cui vengono presi le decisioni del sistema. Questa tecnica ha dimostrato una capacità notevole per chiarire un po' i sistemi complessi come NN (neural networks). Altri metodi che sono stati sviluppati: LIME(Local Interpretabile Model-agnostic Explanations), Silency Maps, SHAP(Shapley Additive exPlanations)…etc. Analizzando le informazioni acquisite dalle varie interpretazione fornite da tale tacniche permete alla progettista di valutare se essi siano conforme con le requisti della sicurezza funzionale del sistema.

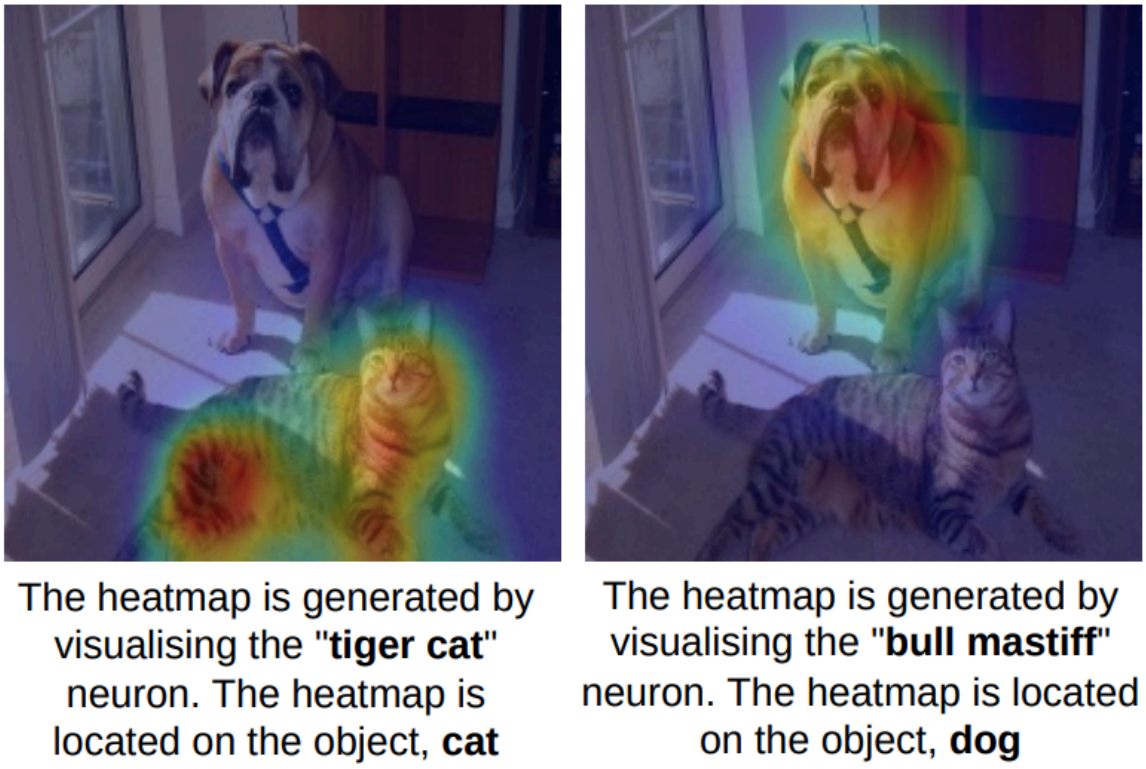


Figura 18 Heat map

6 VERSO ARCHITETTURE AI ROBUSTE

Dopo questa panoramica fatta che riguarda tutte le sfide incontrati in tutte le fasi del ciclo di progettazione di un sistema AI, e vedendo la criticità dei sistemi che vanno ad implementare funzioni di sicurezza, la norma ha fornito negli ultimi 2 parti una visioni su come raggiungere sistemi sicuri e robusti, ma questa volta giocando sull'architettura del sistema. E quindi in questa sezione discuteremo alcuni metodi accennati dalla norma su come migliorare i modelli ML e come le varie sottosistemi del sistema influenzano le proprietà non-funzionale del sistema (qualità, affidabilità, …etc.).

La norma considera due approcci generali a tale riguardo, una che riguarda il miglioramente dell'architettura dei sottoinsiemi, tramite varie tecniche, e l'altra si basa sull'affidabilità del sistema.

6.1 ARCHITETTURA DI SOTTOSISTEMI

Per elaborare tale metodo, la norma considera 4 varie tecniche, tra cui la supervisione di cui abbiamo parlato in una delle paragrafi che parlava del livello di automazione del sistema.

Le altre tecniche sono: meccanismi di rilevamento, Ridondanza, e la valutazione statistica.

6.1.1 Meccanismi di rilevamento

Un possibile metodo in questo contesto, si manifesta nel implementazione di un sistema di monitoraggio, capace di intervenire per mantenere una situazione di sicurezza una volta rilevato qualsiasi azioni insicura che il sistema sta prendendo [37]. Tale sistema di monitoraggio puo essere implementato con tecnologie non-AI oppure AI, ma nel caso laterale la norma proppone di fornire giustificazione per tale scelta in quanto deve essere realizzato una certa indipendenza tra tale sistema ed il sistema principale.

Una possibile architettura dei sistemi AI è quella nominata come il "passive redundancy", fig.19, il concetto qua è quello di implementare diversi architetture dove solo una opera in tutte le situazioni, e gli altri vengono messi in standby finche non si verifica qualche errore nel architettura principale [37].

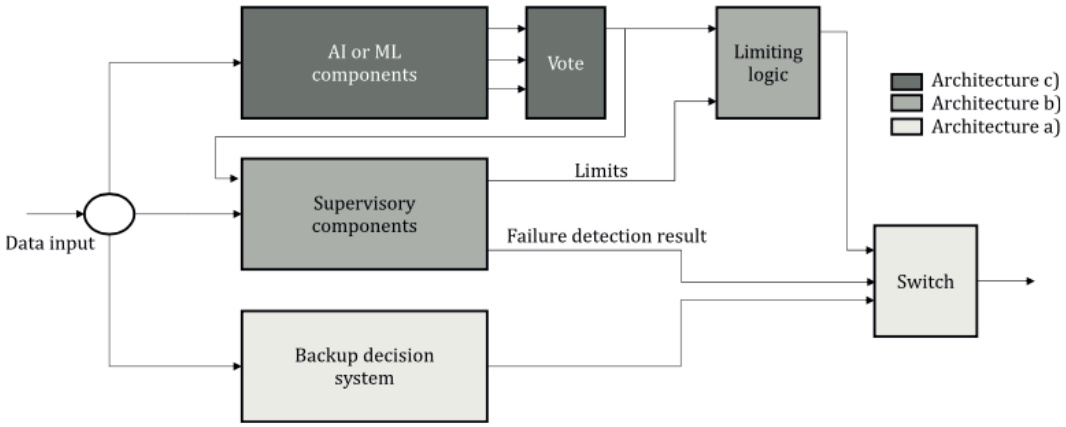


Figura 19 Strutture architetturale per sistemi in cui sono integrati tecnologie AI

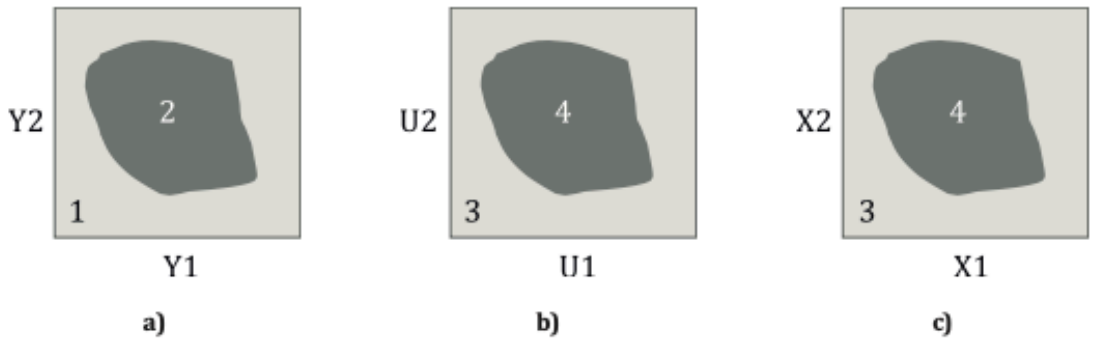
In questa architettura, come si vede nella figura, sono messi a lavorare in parallelo tre sottosistemi:

- Archiettura a) : Backup decision system : è un sistema di backup non-AI che serve a mantenere la funzionalità del sistema quando entra in errore il sistema AI.
- Architettura b) : Supervisory components: si puo notare qua che il sistema di supervisione ha come input, oltre al daat input, l'output del sistema AI. Tale sistema quindi serve a valutare le decisioni presi dal sistema AI ed intervenire, entro i limit del sistema AI per mantenere uno stato sicura.
- Architettura c) : AI or ML components : sistema di decisione complesso monitorato e giudicato dagli altri due sistemi.

Si nota anche un commutatore da cui esce l'output finale del sistema, il che lascia uscire l'output del sistema che ha preso controllo sul sistema.

Questa modalità permette di garantire uno stato sicuro del sistema anche nel caso in cui si verifichi un guasto o errore su una linea di controllo, è quindi presente un sistema resiliente e robusto in termini di sicurezza.

Oltre a questo, la norma fornisce una descrizione che riguarda l'utilità del monitoraggio nel valutare l'accettabilità del sistema AI, e per elaborare questo concetto faccio uso della figura che la norma fornisce descrivendo la valutazione del comportamento accettabile di un sistema AI.



Key

X1	input (Y1, Y2, ...)
X2	output (U1, U2, ...)
Y1	input Y1
Y2	input Y2
U1	output U1
U2	output U2
1	unseen
2	tested
3	unsafe
4	safe

Figura 20 Comportamento accettabile di un sistema AI

Per quanto riguarda il diagramma a), esso riferisce al monitoraggio dell'input. L'importanza si tale aspetto risiede nella sua utilità nel rilevamento di input che risiedono fuori dalla distribuzione del data su cui è stato allenato il sistema dove diventa complicato verificare la correttezza del comportamento del sistema. Infatti, quello che caratterizza i sistemi AI rispetto agli altri sistemi è la loro capacità a generalizzare quello su cui sono stati allenati a nuovi input, però non è detto che questa generalizzazione riesce sempre a dare output giusti, e quindi rilevare tali input per cui il sistema prende decisioni errati risulta utile per migliorare il set di data su cui allenare il modello e di conseguenza realizzare un sistema più robusto. Indicare i limiti in base al quale si classifica l'input come accettabile o meno può essere utile solo quando il sistema sia abbastanza semplice; per i sistemi complessi invece questo di solito è inefficiente in quanto tali sistemi si caratterizzano da tante variabili. A tale regione sono state definiti un paio di metodologie di monitoraggio per il rilevamento delle anomalie dei modelli ML. Una delle seguenti tecniche è "uncertainty estimation" oppure la stima delle incertezze che espone quello che il modello non conosce; la seguente metodologia ha mostrato la sua efficienza per varie modelli, e anche a livello del rilevamento degli attacchi avversarie, però non rappresenta una buona scelta per le stime a tempo-reale per la sua latenza e costi computazionali [53]. Altro

metodo è quello del “In-distribution Error detector” e questa tecnica tende a rilevare errori che sono sviluppati a causa di una interpretazione errata del modello di data che risiede nel suo set di distribuzione; e quindi un sistema di monitoraggio di tale tipo aiuta a rilevare gli aspetti deboli nella rappresentazione della data. Esistono altri tanti metodi e sono descritti bene in [53].

Passando ora al monitoraggio del output del sistema, diagramma b), esso deve classificare l’output come sicuro o meno. Il sistema esegue questa classificazione in base ad un set di limiti predefiniti. Tali limiti devono essere adattivi, in modo tale da permetter il mantenimento di uno stato sicura anche con la variazione degli input del sistema e questo è mostrato dal diagramma c). infatti, per rispondere alla dinamicità dei sistemi tali tecniche di monitoraggio sono di solito basati su modelli ML. Alcune tecniche rivolte a tale scopo sono: student-teacher architecture, autoencoders, Bayesian Neural Networks …,etc.

Per quanto riguarda l’implementazione del monitore la norma definisce 4 punti su cui basare la scelta e sono :

- Il tipo di errore rilevato nella tecnologia AI.
- La modalità di rilevamento degli errori durante l’operazione del sistema.
- Benchmarks del performance dei diversi sistemi di monitoraggio.
- Modalità di intervento per aggirare l’errore e il rischio relativo.

In base a quanto detto, l’adozione dei sistemi di monitoraggio del sistema principale è una tecnica che tende ad aumentare notevolmente la robustezza e l’affidabilità del sistema, ovviamente quando viene scelto il sistema adatto all’applicazione.

6.1.2 Ridondanza

Un’altra tecnica rivolta ad aumentare la robustezza del sistema è la ridondanza che può essere realizzata in diversi modi; c’è la ridondanza strutturale che si manifesta nella duplicazione di componenti critici del sistema, la ridondanza temporale e che si basa sulla ripetizione delle operazioni eseguite dal sistema siano corretti, e l’ultimo tipo è quello funzionale ed è essenzialmente basato sull’utilizzo di diversi algoritmi per realizzare un certo funzione ed applicare un sistema che va a scegliere l’output più adatto. E per sistemi complessi a volte risulta opportuno combinare più di una tecnica per raggiungere un adeguato livello di robustezza.

Per tale sezione la norma ha mostrato un paio di tecniche che possono essere utilizzati nel caso di Neural Networks di cui elenco alcuni.

La ridondanza analitica è un tipo della ridondanza funzionale e si basa essenzialmente su modelli di rilevamento ed isolazione di errori tramite il confronto del comportamento delle varie misure adottati dal sistema [59]. Il termine analitico spiega la natura di tale metodo, in quanto

esso tende a descrivere l'interconnessione dinamica tra i vari componenti del sistema tramite relazioni analitiche.

L'approccio più utilizzato per realizzare la ridondanza temporale [29]. Si basa sulla ripetizione della stessa computazione utilizzando i componenti del sistema diversamente per ciascuna ripetizione. Poi, i risultati vengono confronti per la deduzione degli errori, e se possibile li correggono.

La ridondanza può essere realizzata anche con l'adozione di architetture deep neural netwroks ridondante, oppure tramite algoritmi che sono tollerabili rispetto agli errori, e tante altre metodi elencati dalla norma.

Un'altra possibile tecnica che risiede sotto il concetto di ridondanza è quello di implementare metodi di autocontrollo del sistema. Questi sono di solito rivolti al monitoraggio tramite sistemi di sensing; alcuni esempi che sono normalmente utilizzati e descritte da NE 107 Self-Monitoring and Diagnosis of Field Equipment [55] sono: Internal signal monitoring, utilizzo di segnale di riferimento, signal processing test..., etc. Tali metodi che sono stati elencati sono utilizzati per l'auto-monitoraggio dei dispositivi di campo , e che sono I dispositvi che influenzano in gran modo il controllo del sistema.

La norma suggerisce, come metodo per aumentare la robustezza del sistema, di combinare la ridondanza con diversità. Questo significa di considerare delle variazioni per la progettazione delle diverse tecnologie che devono eseguire la stessa funzionalità nel sistema. E per valutare la differenza tra le varie sistemi risulta opportuno far uso delle metriche su cui basare la valutazione delle scelte e della diversità che stata adottata ed il suo effetto sulla robustezza del sistema. La diversità si può realizzare tramite le seguente: utilizzando diversi dataset di training, o diversi hardware per ogni sistema, oppure ogni sistema ha il suo particolare modalità di autocontrollo e automonitoraggio, o ogni sistema ha il suo sistema di sensing diverso dall'altro e cosi via.

È essenziale dopo l'applicazione della ridondanza convalidare tramite metodi analitici che i sistemi ridondanti sono sufficientemente indipendenti in modo tale da assicurare che essi non condividono gli stessi cause di errori o guasti, e quindi la ridondanza risulta efficiente nel caso in cui in uno dei sistemi si verifica uno guasto.

6.1.3 valutazione statistica

Questa modalità si utilizza quando il sistema è caratterizzato da una distribuzione di input ampio. Infatti, la valutazione della sicurezza funzionale di un sistema AI si basa sull'analisi della distribuzione del suo output. Per alcune applicazioni, quelli che normalmente richiedono qualche flessibilità, si considera la distribuzione statistica di tale output per una certa distribuzione di input, cioè anziché determinare un livello massimo di errori fisso si va a determinare un intervallo di confidanza statistico per una distribuzione di input. Con questo si tende a riconoscere la natura probabilistica del sistema AI.

Alcune volte, si incontra situazioni dove i scenari meno frequenti sono quelli più influenzanti sul comportamento del sistema, e per risolvere tale aspetto si tende ad allenare il sistema in modo tale da avere una relazione proporzionale tra il rischio di uno certo evento e la sua frequenza nel data di training anziché avere una relazione proporzionale con l'attuale occorrenza di essa. Con questo si può garantire un aumento nella robustezza del sistema contro gli eventi più pericolosi.

La modalità con cui si valuta un sistema che adotta la seguente tecnica è quello di analizzare il modello del sistema considerando solo la sua natura probabilistica, quindi trattenendolo come un modello matematico normale.

Nella giustificazione della classificazione del sistema come classe II o III, di solito si devono considerare i risultati dell'analisi statistica fatta e questo perché tali informazioni influenzano la valutazione del performance del sistema e di conseguenza il livello di rischio associato al sistema.

6.2 MIGLIORAMENTO DELL'AFFIDABILITA DEI COMPONENTI

Oltre alle considerazioni che riguardano l'architettura del sistema per realizzare un sistema più robusto ed affidabile, esistono altre tecniche che tendono ad aumentare tale due proprietà del sistema; la norma per tale ragione ha discusso 4 metodologie di queste tecniche di supporto e sono: l'utilizzo del robust learning, tecnologie di ottimizzazione e compressione, meccanismi di attenzione e la protezione di data e parametri.

Per quanto riguarda il primo punto, robust learning, essa include le tecniche che possono essere applicate per ridurre gli effetti dei rischi associati alle tecnologie AI come l'overfitting, data drift, attacchi avversarie, .., etc. quindi per tale argomento riferirsi a (risks of ai)

6.2.1 tecnologie di ottimizzazione e compressione

Tali tecnologie tendono essenzialmente a ridurre il costo computazionale del sistema, e quindi ottimizzare il sistema di elaborazione. Esempi di essi possono essere la quantizzazione e che si manifesta nella riduzione della dimensione binaria dei parametri e di conseguenza la dimensione del modello, aumentando in questo modo la velocità di elaborazione del sistema ed il che ha benefici sui sistemi che devono lavorare in tempo reale tra cui sistemi che implementano funzioni di sicurezza. Pruning è un'altra tecnica e che tende ad eleminare dal modello tutti i parametri meno significativi. La replicazione di un modello complesso con un modello più semplice può essere un altro approccio per ottimizzare il sistema. Ma tali tecniche, visto che eliminano qualche fattori dal modello devono essere accompagnati da un analisi del rischio che possono indurre al sistema. A volte anche vengono presi approcci per ridurre la dimensione dell'input per semplificare la fase di training con il costo di perdere informazioni importanti. Per quello la norma suggerisce un alternativo come l'utilizzo di embedding layers.

Tale tecnica si manifesta nella trasformazione della data di input in vettori densi di dimensione fissa su cui è piu facile agire, e quindi migliorando la performance del sistema.

Oltre alla semplificazione della fase di training, la semplificazione del modello del sistema tramite le varie tecniche migliora l'interpretabilità del modello visto che c'è meno parametri da studiare per analizzare la relazioni tra input ed output, e di conseguenza migliora la trasparenza del modello, che risulta molto utile per la fase di validazione e verificazione. Però si deve tener conto qua che diventa complessa la tracciabilità dei parametri, e quindi capire come ogni parametro influenza il sistema diventa complicato, rendendo difficile la verifica della affidabilità è la sicurezza del sistema stesso. In questo caso risulta opportuno fare uso delle tecniche del AI spiegabile.

Abbiamo visto l'effetto di alcune tecniche di semplificazione del modello all'interoperabilità del sistema, ma altre tecniche tendono ad aumentare la robustezza contro gli attacchi avversarie. E questa è la tecnica di distillazione che. Infatti, [56] hanno trovato che questa tecnica aumenta il numero di caratteristiche su cui deve agevolare l'attacco avversario per influenzare il modello di 800%. La distillazione si basa sull'allenamento di un modello semplice sulla data di output di un sistema più complesso.

Detto ciò, le tecniche di ottimizzazione e compressione del sistema incidono su due aspetti essenziali del sistema, quello computazionale e quello difensivo contro gli attacchi avversarie.

6.2.2 meccanismi di attenzione

I meccanismi di attenzioni sono dei Neural Network modules, che basandosi su calcoli per determinare gli elementi più significativi in un certo input, permettono al modello principale di concentrarsi sui aspetti più importanti nel input, cioè quelli che influenzano di più i decisioni presi dal sistema [66].

L'adozione di tali meccanismi nei sistemi che implementano funzioni di sicurezza può avere tanti vantaggi. Se prendiamo ad esempio il caso di un cobot, i meccanismi di attenzione permettono al sistema di dare priorità agli ostacoli che si trovano sul suo sentiero, migliorando così il dataset su cui deve basare il sistema di posizionamento le sue decisioni. Un altro aspetto che si può sfruttare di tale meccanismi è quello che essi aiutano il modello a trattare la data presa da diversi risorsi (sensori, fotocamere, ..) in modo completo anziché analizzarli indipendentemente, e questo ovviamente aumenta l'affidabilità del sistema in quanto prende in considerazioni diverse aspetti della situazioni in cui si trova e come le varie aspetti influenzano le decisioni che deve prendere.

Un altro aspetto che l'utilizzo dei meccanismi di attenzione può migliorare, e questa volta i mappi di attenzione, è l'exblainability del sistema; e abbiamo mostrato nei paragrafi precedenti come questa proprietà migliora tanti altri aspetti del sistema. I mappi di attenzione evidenziano quali sono gli aspetti più influenzanti per la generazione di un certo output, e quindi rendendo disponibile ai progettisti informazioni cruciali per convalidare il comportamento di progetto, o anche rilevare i punti deboli del sistema.

L’attenzione può essere integrata anche nella fase di training. Il concetto base del attentino training è quello di fare il modello apprendere da solo durante la fase di training i pesi di attenzione degli elementi presenti nel data su cui viene allenato.

6.2.3 Protezione di data e parametri

Abbiamo accennato tanto nelle sezioni precedenti che i dati e parametri sono due aspetti fondamentali per la giusta comportamento di un sistema AI, ed infatti questi due elementi sono vulnerabili a diversi aspetti che possono metter in rischio il sistema come errori randomici, ed attacchi avversarie. Quindi risulta importante per garantire la loro sicurezza, e di conseguenza la sicurezza del sistema, proteggerli.

Per tale ragione si possono riferirsi alle norme relativi alla sicurezza dell’informazioni che descrivano i requisiti che devono avere tali sistemi e come garantire la loro sicurezza. Riferendo ad esempio al NTS (National Institute of Standards and Technology) più in specifico nella Special Publication 800-30 una fase di analisi del rischio relativo al sistema di informazione e data, determinando tutte le vulnerabilità che possono influenzare il sistema. Poi si devono prendere le mitigazioni in corrispondenza della frequenza di occorrunza del rischio, il grado di impatto sul sistema e il grado di mitigazione che puo introddure un elemento per gestirlo.

Esistono varie tecniche con cui si può gestire i rischi associati ai sistemi di informazione. Una delle tecniche che si può applicare quando il data deve essere condivisa con altri sistemi [79], e quindi con un rischio di perdere la privacy del sistema quando la data può contenere informazioni sensibili, è il cosi detto “collaborative learning” e che si basa essenzialmente di condividere informazioni a livello di solo parametri e non data. La supervisione della data durante la fase di training e di operazione del sistema, per rilevare qualsiasi cambiamento o manipolazione avversaria del modello. Si possono anche adottare sistemi di filtraggio della data di input per eliminare qualsiasi disturbo che può influenzare il comportamento del modello.

È opportuno dire che la norma, nella sua annex A informativo discute l’applicabilità della parte 3 della norma IEC 61508:2020 sulle tecnologie AI, e quindi come si possono mappare tutte le tecniche che riguardano la mitigazione dei rischi associati ai sistemi software.

7 CONCLUSIONE

In conclusione, dopo aver esaminato l'integrazione delle tecnologie di intelligenza artificiale (AI) nelle funzioni di sicurezza, abbiamo identificato i punti chiave e gli aspetti trattati dalla norma ISO/IEC 5469. Abbiamo analizzato la classificazione delle tecnologie AI, essenziale per valutare la loro potenziale integrazione nelle funzioni di sicurezza, oltre ai rischi associati a tali sistemi e alle varie tecniche e metodologie per la loro mitigazione. Inoltre, è stata posta particolare attenzione alla criticità delle fasi di verifica e validazione, evidenziando le diverse tecniche applicabili in questo ambito. Infine, sono state presentate alcune soluzioni volte a gestire le problematiche riscontrate, migliorare le prestazioni del sistema e garantire un adeguato livello di sicurezza.

Tuttavia, sulla base di quanto emerso dalla norma e dalle ricerche condotte, risulta evidente che l'adozione delle tecnologie AI nelle funzioni di sicurezza rimane limitata. Questo fenomeno può essere attribuito al fatto che l'intelligenza artificiale è ancora in fase evolutiva; considerando la delicatezza delle funzioni di sicurezza, i vari rischi e le sfide connesse alle fasi di sviluppo (come ad esempio la validazione e la verifica dei sistemi AI) possono ostacolare la sua applicazione nel settore della sicurezza delle macchine.

Ritengo tuttavia che ulteriori ricerche mirate a chiarire gli aspetti ancora poco definiti di queste tecnologie possano consentire lo sviluppo di sistemi avanzati di sicurezza basati sull'AI, capaci di incrementare significativamente il livello di protezione.

Bibliografia

1. ---. “Regularization in Machine Learning.” *GeeksforGeeks*, 5 Aug. 2024, www.geeksforgeeks.org/regularization-in-machine-learning
2. ---. *“Test Oracles.” GeeksforGeeks, 3 Oct. 2022, www.geeksforgeeks.org/test-oracles*.
3. “Che Cos’è L’apprendimento Supervisionato? | Google Cloud.” *Google Cloud*, cloud.google.com/discover/what-is-supervised-learning?hl=it.
4. “Che Cos’è L’intelligenza Artificiale? | Tematiche | Parlamento Europeo.” *Tematiche | Parlamento Europeo*, 9 Mar. 2020, www.europarl.europa.eu/topics/it/article/20200827STO85804/che-cos-e-l-intelligenza-artificiale-e-come-viene-usata.
5. *“The Application of Artificial Intelligence in Functional Safety.” Home, 2024, electrical.theiet.org/guidance-and-codes-of-practice/publications-by-category/artificial-intelligence/the-application-of-artificial-intelligence-in-functional-safety/*.
6. “What Is Overfitting? - Overfitting in Machine Learning Explained - AWS.” *Amazon Web Services, Inc.*, aws.amazon.com/what-is/overfitting.
7. Aghemo, Raffaella. “Robustezza E Spiegabilità Dell’Intelligenza Artificiale.” *Medium*, 7 Jan. 2022, raffa-aghemo.medium.com/robustezza-e-spiegabilit%C3%A0-dellintelligenza-artificiale-7dfd625646c1.
8. Ali, Moez. “Understanding Data Drift and Model Drift: Drift Detection in Python.” *DataCamp*, DataCamp, 11 Jan. 2023, www.datacamp.com/tutorial/understanding-data-drift-model-drift.
9. *Ayed, M. Ben, Lilia Zouari, and Mohamed Abid. "Software in the loop simulation for robot manipulators." Engineering, Technology & Applied Science Research 7.5 (2017)*.
10. *Bacic, Marko. "On hardware-in-the-loop simulation." Proceedings of the 44th IEEE Conference on Decision and Control. IEEE, 2005*.
11. Baena-Garcıa, Manuel, et al. "Early drift detection method." *Fourth international workshop on knowledge discovery from data streams*. Vol. 6. 2006.
12. *Baheri, Ali. "Exploring the role of simulator fidelity in the safety validation of learning-enabled autonomous systems." AI Magazine 44.4 (2023): 453-459*.
13. BBC News. *Uber’s Self-driving Operator Charged Over Fatal Crash*. 16 Sept. 2020, www.bbc.com/news/technology-54175359.
14. *Beyer, Michael, et al. "Quantification of the impact of random hardware faults on safety-critical ai applications: Cnn-based traffic sign recognition case study." 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). IEEE, 2019*.
15. Brown, Sara. “Machine Learning, Explained.” *MIT Sloan*, 21 Apr. 2021, mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained.
16. Brownlee, Jason. *A Gentle Introduction to Uncertainty in Machine Learning*. 25 Sept. 2019.
17. Cajic, Elvir, and Zvezdan Stovajonić. *Stochastic Methods in Artificial Intelligence*, 14 Nov. 2023, www.researchgate.net/publication/375612252_Stochastic_Methods_in_Artificial_Intelligence.

18. Chadwick, Jonathan, and Fiona Jackson. “Worst Robotic Accidents in History - After Chess Robot Breaks Seven-year-old Boy’s Finger in Russia.” *Mail Online*, 26 July 2022, www.dailymail.co.uk/sciencetech/article-11046663/Worst-robotic-accidents-history-chess-robot-breaks-seven-year-old-boys-finger-Russia.html.
19. CyberLaws Europe and By CyberLaws Europe. “What Does ‘Robustness (Robust AI)’ Mean? – Legal Definition.” *CyberLaws*, 28 Aug. 2023, www.cyberlaws.it/en/2023/what-does-robustness-mean-legal-definition.
20. *Donà, Riccardo, and Biagio Ciuffo. "Virtual testing of automated driving systems. A survey on validation methods." IEEE Access 10 (2022): 24349-24367.*
21. EN ISO 13849:2023: *Safety of machinery–Safety-related parts of control systems*
22. Felzmann, Heike, et al. “Transparency You Can Trust: Transparency Requirements for Artificial Intelligence Between Legal Norms and Contextual Concerns.” *Big Data & Society*, vol. 6, no. 1, Jan. 2019, p. 205395171986054, doi:10.1177/2053951719860542
23. Folgado, Francisco Javier, et al. “Review of Industry 4.0 From the Perspective of Automation and Supervision Systems: Definitions, Architectures and Recent Trends.” *Electronics*, vol. 13, no. 4, Feb. 2024, p. 782, doi:10.3390/electronics13040782.
24. *Freedom from Interference (FFI) - Tasking*, resources.tasking.com/sites/default/files/2021-02/TASKING-Whitepaper-Freedom - from-Interference-Pt-1_WEB.pdf. Accessed 30 Oct. 2024.
25. *Gazebo, gazebosim.org/home.*
26. GeeksforGeeks. “Cross Validation in Machine Learning.” *GeeksforGeeks*, 7 Aug. 2024, www.geeksforgeeks.org/cross-validation-machine-learning.
27. *Hamilton, Serena H., et al. “Fit-for-purpose Environmental Modeling: Targeting the Intersection of Usability, Reliability and Feasibility.” Environmental Modelling & Software, vol. 148, Dec. 2021, p. 105278, doi:10.1016/j.envsoft.2021.105278.*
28. *He, Yi, et al. "Understanding and mitigating hardware failures in deep learning training systems." Proceedings of the 50th Annual International Symposium on Computer Architecture. 2023.*
29. *Hsu, Yuang-Ming, Vincenzo Piuri, and E. E. Swartzlander. "Time-redundant multiple computation for fault-tolerant digital neural networks." 1995 IEEE International Symposium on Circuits and Systems (ISCAS). Vol. 2. IEEE, 1995.*
30. IEC 61508-2:2010: *Functional Safety of electrical, electronic and programmable electronic (E/E/PE) safety-related systems- Requirements for electrical/electronic/programmable electronic safety- related systems*
31. IEC 61508-3: 2010: *Functional Safety of electrical, electronic and programmable electronic (E/E/PE) safety-related systems- Software requirements*
32. IEC 61508-4: 2010: *Functional Safety of electrical, electronic and programmable electronic (E/E/PE) safety-related systems-Definitions and abbreviations*
33. *IEC 62061: Safety of machinery – Functional safety of safety-related control systems*
34. *Infographics: China Grounds Boeing 737 MAX 8 Jets*. myrepublica.nagariknetwork.com/news/infographics-china-grounds-boeing-737-max-8-jets.
35. ISO 26262-2:2018: *Road vehicles – Functional safety*
36. ISO/IEC 22989:2022 : *Information technology — Artificial intelligence — Artificial intelligence concepts and terminology*
37. ISO/IEC 5469:2024: *Functional Safety and AI systems*

38. ISO/IEC IEEE 29119-4:2021: *Software and systems engineering — Software testing-Test Techniques*
39. ISO/IEC TR 24027:2021*: Information technology — Artificial intelligence (AI) — Bias in AI systems and AI aided decision making*
40. ISO/IEC TR 24028:2020: *Information technology — Artificial intelligence — Overview of trustworthiness in artificial intelligence*
41. ISO/IEC TR 29119-11:2020: *Software and System Engineering - Software Testing*
42. Jameel Aliyu, Muhammed, et al. "Design of an AI Base Framework for Speed Limit Prediction and Control Using GPS Data." *International Journal of Advance Research and Innovative Ideas in Education*, ijariie.com/AdminUploadPdf/Design_of_an_AI_Base_Framework_for_Speed_Limit_Prediction_and_Control_Using_GPS_Data_ijariie21302.pdf.
43. *Juristo, Natalia, and Omar S. Gómez. "Replication of Software Engineering Experiments." SpringerLink, Springer Berlin Heidelberg, 1 Jan. 1970, link.springer.com/chapter/10.1007/978-3-642-25231-0_2.*
44. Klinkenberg, Ralf, and Thorsten Joachims. "Detecting concept drift with support vector machines." *ICML*. 2000.
45. Kourinian, Arsen, and Mayer Brown. *Addressing Transparency & Explainability When Using AI ...*, www.mayerbrown.com/-/media/files/perspectives-events/publications/2024/01/addressing-transparency-and-explainability-when-using-ai-under-global-standards.pdf?rev=8f001eca513240968f1aea81b4516757. Accessed Oct. 2024.
46. *Kuhn, et al. "Software Fault Interactions and Implications for Software Testing." IEEE Transactions on Software Engineering, vol. 30, no. 6, June 2004, pp. 418–21, doi:10.1109/tse.2004.24.*
47. *Kuhn, Richard, and Vadim Okun. Pseudo-Exhaustive Testing for Software - TSAPPS at NIST, tsapps.nist.gov/publication/get_pdf.cfm?pub_id=917175.*
48. Liao, Fangzhou, et al. "Defense Against Adversarial Attacks Using High-Level Representation Guided Denoiser." *arXiv.org*, 8 Dec. 2017, arxiv.org/abs/1712.02976.
49. Madry, Aleksander, et al. "Towards Deep Learning Models Resistant to Adversarial Attacks." *arXiv.org*, 19 June 2017, arxiv.org/abs/1706.06083.
50. *Majzik, István, et al. "Towards system-level testing with coverage guarantees for autonomous vehicles." 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS). IEEE, 2019.*
51. Massed Compute. "FAQ Answers - Massed Compute." *Massed Compute*, 27 Sept. 2024, massedcompute.com/faq-answers/?question=How+does+machine+learning+handle+uncertainty+and+ambiguity%3F.
52. Massobrio, Anthony. *Iterative Design Process: A Guide &Amp; the Role of Deep Learning*. 2 Feb. 2024, www.neuralconcept.com/post/the-iterative-design-process-a-step-by-step-guide-the-role-of-deep-learning#:~:text=The%20way%20the%20iterative%20design,prototypes%20either%20in%20the%20physical.
53. *Mohseni, Sina, et al. "Practical solutions for machine learning safety in autonomous vehicles." arXiv preprint arXiv:1912.09630 (2019).*
54. *Naser, M. Z., and Amir H. Alavi. "Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences - Architecture, Structures and Construction." SpringerLink, Springer International Publishing, 24 Nov. 2021, link.springer.com/article/10.1007/s44150-021-00015-8.*

55. NE 107 *'Self-Monitoring and Diagnosis of Field Devices'*
56. *Papernot, Nicolas, et al. "Distillation as a defense to adversarial perturbations against deep neural networks." 2016 IEEE symposium on security and privacy (SP). IEEE, 2016.*
57. Perez-Cerrolaza, Jon, et al. "Artificial Intelligence for Safety-Critical Systems in Industrial and Transportation Domains: A Survey." *ACM Computing Surveys*, vol. 56, no. 7, Apr. 2024, pp. 1–40, doi:10.1145/3626314.
58. *Plummer, Andrew R. "Model-in-the-loop testing." Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 220.3 (2006): 183-199.*
59. *Pouliezos, A. D., and George S. Stavrakakis. Real time fault monitoring of industrial processes. Vol. 12. Springer Science & Business Media, 2013.*
60. psico-smart.com. *Future Trends in Workplace Safety: Predictive Analytics and AI Applications*. psico-smart.com/en/blogs/blog-future-trends-in-workplace-safety-predictive-analytics-and-ai-applications-162719#:~:text=The%20Rise%20of%20Predictive%20Analytics%20in%20Workplace%20Safety,-In%20a%20groundbreaking&text=By%20implementing%20targeted%20safety%20measures,promoting%20a%20safer%20workplace%20environment.
61. Qiu, Shilin, et al. "Review of Artificial Intelligence Adversarial Attack and Defense Technologies." *Applied Sciences*, vol. 9, no. 5, Mar. 2019, p. 909, doi:10.3390/app9050909
62. Raji, Inioluwa Deborah, and Roel Dobbe. "Concrete problems in AI safety, revisited." *arXiv preprint arXiv:2401.10899* (2023).
63. *Random Vs. Systematic Error. www.physics.umd.edu/courses/Phys276/Hill/Information/Notes/ErrorAnalysis.html.*
64. Rao, Vasu. *The Limits of AI in Dynamic and Unstructured Environments*. 31 May 2024, www.linkedin.com/pulse/limits-ai-dynamic-unstructured-environments-vasu-rao-jmysc
65. *Shah, Hardik. "A Detailed Guide on Test Coverage." Simform - Product Engineering Company, 18 Sept. 2023, www.simform.com/blog/test-coverage/#:~:text=Test%20coverage%20is%20defined%20as,then%20test%20coverage%20is%2090%25.*
66. *Soydaner, Derya. "Attention mechanism in neural networks: where it comes and where it goes." Neural Computing and Applications 34.16 (2022): 13371-13385.*
67. Steimers, André, and Moritz Schneider. "Sources of Risk of AI Systems." *International Journal of Environmental Research and Public Health*, vol. 19, no. 6, Mar. 2022, p. 3641, doi:10.3390/ijerph19063641.
68. *Stocco, Andrea, et al. "Model VS System Level Testing of Autonomous Driving Systems: A Replication and Extension Study - Empirical Software Engineering." SpringerLink, Springer US, 2 May 2023, link.springer.com/article/10.1007/s10664-023-10306-x.*
69. *Storkey, Amos J. When Training and Test Sets Are Different: Characterising ..., 8 Mar. 2008, homepages.inf.ed.ac.uk/amos/publications/Storkey2009TrainingTestDifferent.pdf.*
70. *Team, CARLA. "Carla." CARLA Simulator, carla.org/.*

71. The Peninsula Newspaper. *The Peninsula Qatar*. 8 Nov. 2023, thepeninsulaqatar.com/article/08/11/2023/south-korean-man-killed-by-industrial-robot.
72. *Transparency and Explainability Risks | Digital Ethics for Tech Professionals*. www.ethics-for-tech.org/part-3-knowledge-bank-for-the-identification-and-mitigation-of-risk/transparency-and-explainability-risks.
73. Tsamados, Andreas, et al. "Human Control of AI Systems: From Supervision to Teaming." *AI And Ethics*, May 2024, doi:10.1007/s43681-024-00489-4.
74. *Tursun, Osman, et al. "Towards self-explainability of deep neural networks with heatmap captioning and large-language models." arXiv preprint arXiv:2304.02202 (2023).*
75. *Vogel, Thomas, et al. Challenges for Verifying and Validating Scientific Software in Computational Materials Science | Ieee Conference Publication | IEEE Xplore, 21 June 2019, ieeexplore.ieee.org/document/8823757/.*
76. Vouk, Mladen A. "On back-to-back testing." *Computer Assurance, 1988. COMPASS'88*. IEEE, 1988.
77. *Xu, sJie, et al. "Efficient multi-fidelity simulation optimization." Proceedings of the Winter Simulation Conference 2014. IEEE, 2014.*
78. Yadav, Amit. "Human-in-the-Loop Systems in Machine Learning - Biased-Algorithms - Medium." *Medium*, 12 Oct. 2024, medium.com/biased-algorithms/human-in-the-loop-systems-in-machine-learning-ca8b96a511ef.
79. *Yan, Hongyang, et al. "PPCL: Privacy-preserving collaborative learning for mitigating indirect information leakage." Information Sciences 548 (2021): 423-437.*
80. Yasar, Kinza. "Artificial Intelligence as a Service (AIaaS)." *Enterprise AI*, 14 July 2023, www.techtarget.com/searchenterpriseai/definition/Artificial-Intelligence-as-a-Service-AIaaS#:~:text=Artificial%20Intelligence%20as%20a%20Service%20(AIaaS)%20is%20the%20third%2D,investment%20and%20with%20lower%20risk.
81. *Yu, Guangba, et al. "A Survey on Failure Analysis and Fault Injection in AI Systems." arXiv preprint arXiv:2407.00125 (2024).*
82. Zhang, Xiaoge, et al. "Towards risk-aware artificial intelligence and machine learning systems: An overview." *Decision Support Systems* 159 (2022): 113800.
83. *Zillner 1122, Ronald Schnitzer 1122 Andreas Hapfelmeier 11 Sven Gaube 11 Sonja, et al. Ai Hazard Management: A Framework for the Systematic Management of Root Causes for AI Risks, 7 Mar. 2024, arxiv.org/html/2310.16727v2.*