applied sciences

MDPI

Tutorial

Cyber–Physical Systems: The Last Defense

Frank J. Furrer

Faculty for Computer Science, Technical University of Dresden, DE-01062 Dresden, Germany; frank.j.furrer@bluewin.ch

Abstract

The development, evolution, and operation of a cyber–physical system are cross-domain, holistic processes. The process encompasses all elements of a cyber–physical system, including computation infrastructure, software, interfaces to the physical world, human interactions, and safety and security engineering. The process is holistic because it must assure conceptual integrity and correct interoperability across all elements of the CPS. Unfortunately, at every stage of this process, vulnerabilities can be introduced into the system (due to negligence, mistakes, lack of skills, malicious activities, etc.). These dormant vulnerabilities can cause failures of the runtime system, possibly resulting in damage, loss of property or life, safety accidents, or security incidents. A promising approach to mitigate such risks is runtime anomaly detection using artificial intelligence/machine learning. This tutorial paper introduces the fundamental concepts of AI/ML anomaly detection and describes the corresponding intervention mechanisms. Automated intervention mechanisms are the last line of defense against failures, faults, malfunctions, and malicious activities—and their unfortunate consequences. The paper remains at the conceptual level and defers implementation details to subsequent publications. The content addresses advanced students (at the master's level) and researchers entering this fascinating field.

Keywords: cyber-physical system; vulnerabilities; safety accident; security incidents; runtime monitoring; anomaly detection; real-time intervention; protective shell

1. Introduction and Context

Cyber–physical systems [1,2] are a double-edged sword. On the one hand, they are extremely useful and enable numerous indispensable applications. On the other hand, they pose a considerable risk and may lead to safety accidents or security incidents. Therefore, safety and security are the critical quality properties of a trustworthy cyber–physical system. Unfortunately, every day, safety accidents and security incidents—some with grave damages—are reported. Safety accidents and security incidents are always the consequence of a vulnerability in the cyber–physical system.

Cyber–physical systems (CPS) result from a process encompassing their development, evolution, and operation. The process is cross-domain because it encompasses all elements of the cyber–physical system, including computation infrastructure, software, interfaces to the physical world, and human interactions, as well as safety and security engineering domains. The process is holistic because it must assure complete conceptual integrity and correct interoperability across all elements of the CPS.

Figure 1 shows the threat context of a cyber–physical system. Vulnerabilities can be introduced during the development process or appear in the elements of the runtime system. Sources of vulnerabilities during the development process include weaknesses

Check for updates

Academic Editor: Yutaka Ishibashi

Received: 12 September 2025
Revised: 10 December 2025
Accepted: 10 March 2026
Published: 2 April 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

in the development process, deficiencies in development tools, vulnerabilities imported through third-party software, deficits in algorithms or training data, and, last but not least, insufficient risk analysis and risk management [3]. Vulnerabilities introduced during development are transferred to the runtime system! During operation, additional vulnerabilities appear, such as emergent behavior and emergent properties [4,5], autonomous decisions, malicious attacks, errors or faults in an element of the runtime system, failures in the execution platform, systems software weaknesses (Operating system, database, browser, etc.), sensor and actuator malfunctions, and uncertain or unpredicted operating conditions.

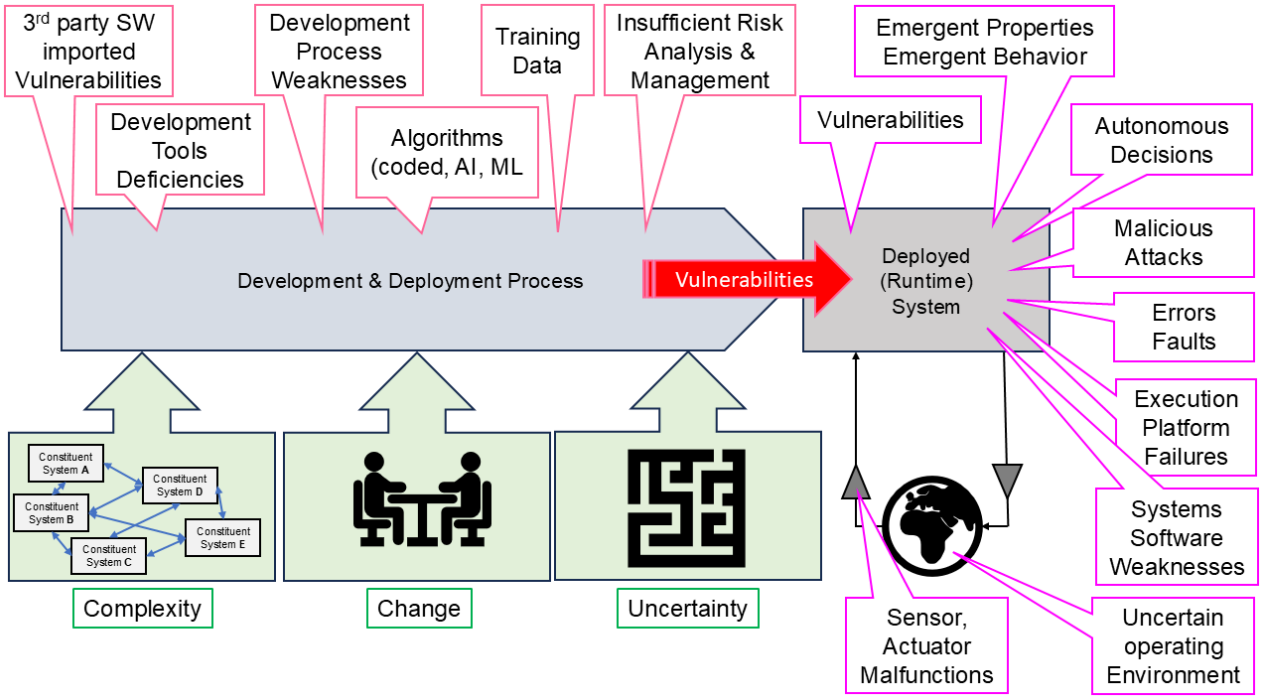


Figure 1. Cyber–Physical System Threat Context showing the primary sources of vulnerabilities.

In many cases, the cause of a safety accident or security incident is software-related [6,7]. Fortunately, a vast and valuable literature exists for building safe and secure systems [8–11]. However, the undeniable fact remains that cyber–physical runtime systems contain hidden vulnerabilities that may lead to safety accidents and security incidents.

In addition, today's software development suffers from three challenges—complexity, change, and uncertainty—known as the "Three devils of systems engineering" [12]. The three devils often force damaging compromises on design and implementation decisions!

2. Impossibility Statement

The storyline of safety and security in cyber–physical systems begins with the "Impossibility Statement" (Figure 2, adapted from [13]). The consequence of the impossibility statement is that the deployed runtime cyber–physical system contains unmitigated vulnerabilities that may lead to safety accidents or security incidents. The last defense against such hidden vulnerabilities is runtime monitoring with automatic, preferably real-time intervention. A powerful technique for this task is anomaly detection, particularly when combined with artificial intelligence/machine learning. An anomaly detected by a machine learning algorithm is an observation, event, or pattern in the real data that deviates significantly from the learned notion of normal behavior and is therefore flagged as unusual or unexpected relative to the majority of the data, thus requiring intervention (https://en.wikipedia.org/wiki/Anomaly_detection, accessed 10 November 2025). The

real-time intervention mechanism encompasses all the elements for runtime protection of the CPS and thus constitutes the last line of defense before a safety accident or security incident occurs.

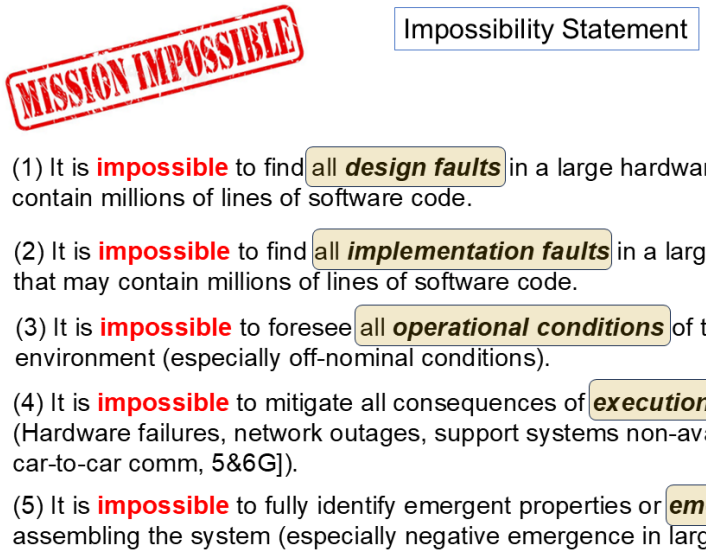


Figure 2. Impossibility statement declaring the fundamental situation of developing and operating a cyber–physical system.

3. Cyber–Physical Systems Behavior

The cyber–physical runtime system has two categories of behavior (Figure 3):

(1) Desired behavior: Safe and secure operation, complete adherence to specifications, full conformance with laws and regulations, respecting all timing conditions, satisfactory handling of all errors, faults, and failures, comprehensive interaction with the user, and trustworthy in all situations and operating conditions;

(2) Undesired behavior: Unsafe, insecure, dangerous, incomprehensible, untrustworthy, or erratic behavior.

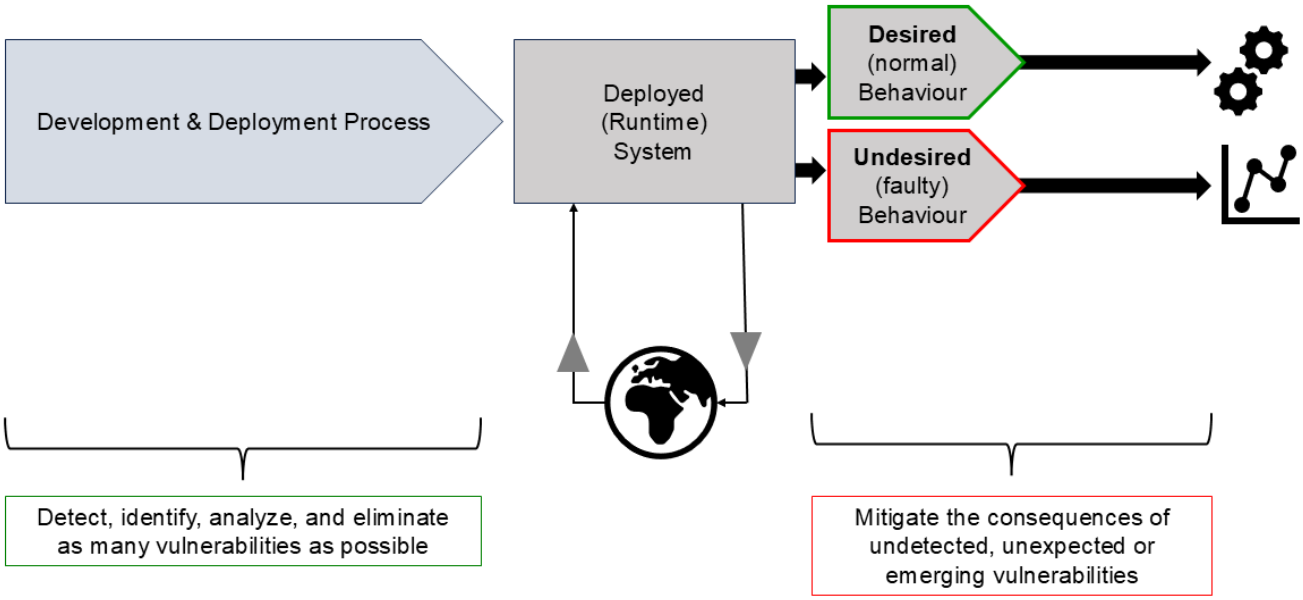


Figure 3. Desired and Undesired Behavior of a Cyber–physical System.

Once every reasonable effort has been made during development and evolution to detect, identify, analyze, and eliminate as many vulnerabilities as possible, the runtime system must be protected against undesired behavior.

4. Cyber–Physical Systems Runtime Monitoring

Runtime monitoring is not a new technology. It has, for example, been introduced as Onboard Diagnostics (OBD) as an integral part of vehicle design for decades, and OBD-II became mandatory in the USA in 1996 [14]. An extensive literature on runtime monitoring and its applications exists [15–20].

The key idea of runtime monitoring is shown in Figure 4: The behavior of the runtime system is supervised, and additional hardware and software attempt to detect and identify undesired (anomalous) behavior. If such behavior is detected, the analysis starts, and a suitable response is launched.

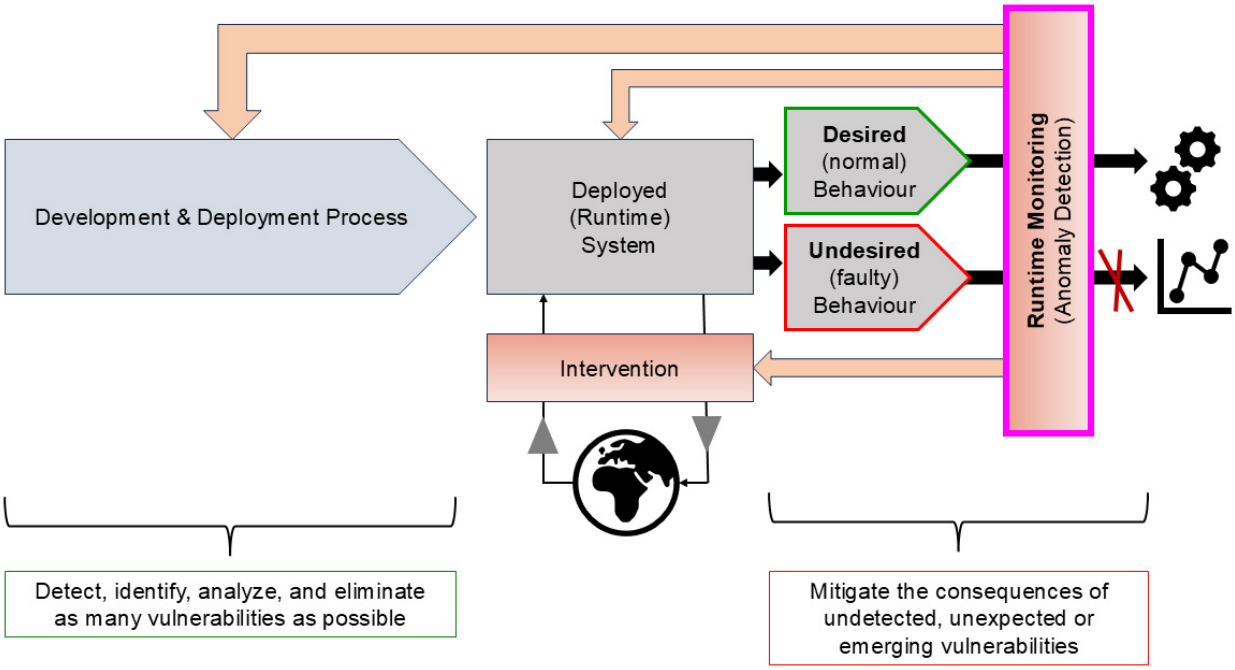


Figure 4. CPS Last Defense—Runtime Anomaly Detection and Real-time Intervention.

"Classical" runtime monitoring is based on preprogrammed algorithms; i.e., when faulty conditions are detected, a preprogrammed intervention is triggered [15]. This type of runtime monitoring is static. Modern runtime monitoring utilizes machine learning and is expected to detect and identify many more, as well as unforeseen, abnormalities; thus, this type of runtime monitoring is dynamic.

5. Monitor–Analyze–Intervene

Figure 5 introduces the four elements of a machine learning last defense architecture:

(1) The deployed cyber–physical runtime system in its operating environment;
(2) The anomaly detection (symbolizing the necessary hardware and software for the application at hand);
(3) The intervention planning element;
(4) The intervention execution blocks.

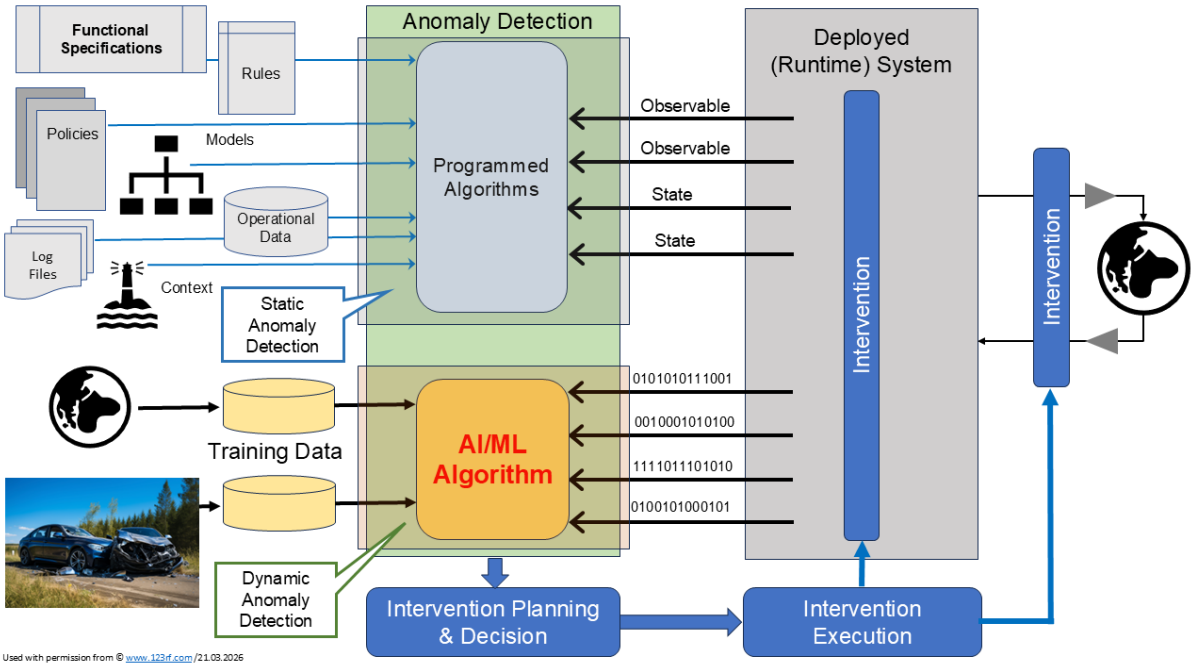


Figure 5. Real-Time Intervention as a Last Defense to Protect the CPS.

5.1. Monitor

The anomaly detection element [21–23] consists of two parts: a static and a dynamic part. The static part consists of programmed algorithms. The inputs from the runtime system are observables (e.g., sensor and actuator values, and internal variables such as speed or position). As reference inputs for the desired behavior, the static part uses functional specifications (including quality property requirements), a set of rules (e.g., bracket values for output signals), policies, models, log files, and operational data [15]. Anomalies are detected if discrepancies between the inputs and the reference information (model) are discovered.

The dynamic part contains AI/Machine learning. Its inputs are information streams [24] from the cyber–physical runtime system, such as video, LIDAR, Radar, trajectory information, and sensor data value sequences. Of most significant importance for the dynamic part is the training data [25]. Two sources of training data are shown in Figure 4:

- Data obtained from the real world;
- Synthetic data: data generated by algorithms, artificial intelligence, or simulations [26,27].

In many modern applications, there is a lack of real-world data, particularly for critical scenarios like serious safety accidents. In such cases, simulations provide precious and broad training data, such as car accidents and their successful evasive maneuvers. Interestingly, video game technology provides effective support for generating synthetic training data [28].

5.2. Analyze

During the analysis phase, the ML algorithm compares the actual behavioral data of the cyber–physical system with a previously learned “normal” behavior [29]. A conceptual diagram is shown in Figure 6. The first engineering step is to select the data and context to be used in the actual application. The second step is to construct a machine learning model for normal behavior, trained on real or simulated data, or both. As a third step, anomaly criteria must be defined as rules or parameters for the model. The system is now ready for operation and can be deployed.

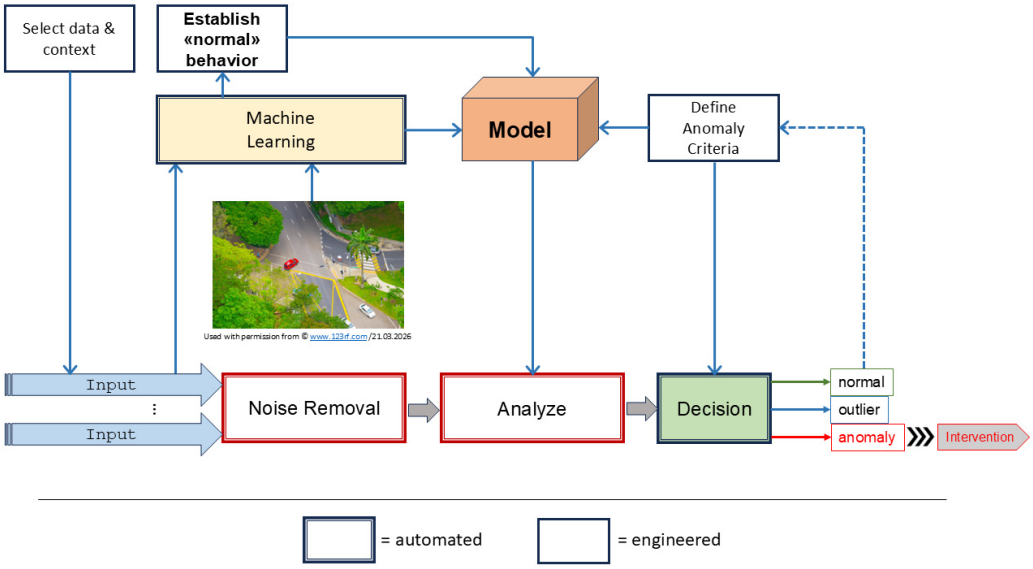


Figure 6. ML Analysis Phase to Detect Anomalies in the Behavior of the Runtime System.

In many cases, incoming data streams are subjected to a noise reduction or removal algorithm [30,31]. The ML algorithm then analyzes the data stream(s) and detects irregularities. Lastly, the decision gate classifies into <normal>, <outlier>, or <anomaly>. Note that outliers and anomalies are not the same: An outlier is a statistically rare but possible point. An anomaly is a value that is not supposed to exist [29]. Once an anomaly is identified, an intervention is triggered. Many systems continue to use machine learning during operation to enhance their detection capabilities.

5.3. Intervene

The intervention phase is the most risky phase of the last defense: because autonomous machine decisions directly impact the physical world, they can cause or even enable safety accidents or security incidents. Figure 7 shows the autonomous intervention concept. Once it is determined that a situation is anomalous, the algorithm must decide which action to take. The actions can be preprogrammed, such as limiting the range of actuator outputs or monitoring the rate of change in sensor inputs. The more advanced forms are learned actions, such as ship collision avoidance.

The action decision algorithm can (Figure 6):

- Directly influence the behavior of the control system, i.e., block, change, or override functionality;
- Override or correct sensor or actuator values, i.e., protect the system from harmful input or output values;
- Manipulate the operating environment of the CPS.

Examples of actions are:

- Isolating a network node immediately after an unknown thread in the operating system is detected;
- Shutting down a TCP/IP port when excessive in- or outbound traffic is measured;
- Taking over a ship's control when a threatening collision course is computed;
- Setting all traffic lights by the involved car to red after an accident at the intersection;
- Automatically take evasive action when a car recognizes an imminent hit of a pedestrian, animal, or cyclist;
- Force the CPS into a safe state, e.g., trigger an emergency stop of the train;

- Switch to a safe, degraded mode of operation, e.g., limit the maximum speed of a car to 80 km/h after a failure of the antiskid control;
- Close the system access for an employee after unauthorized data access attempts;
- Contact a cardholder after untypical payment behavior is detected from their card.

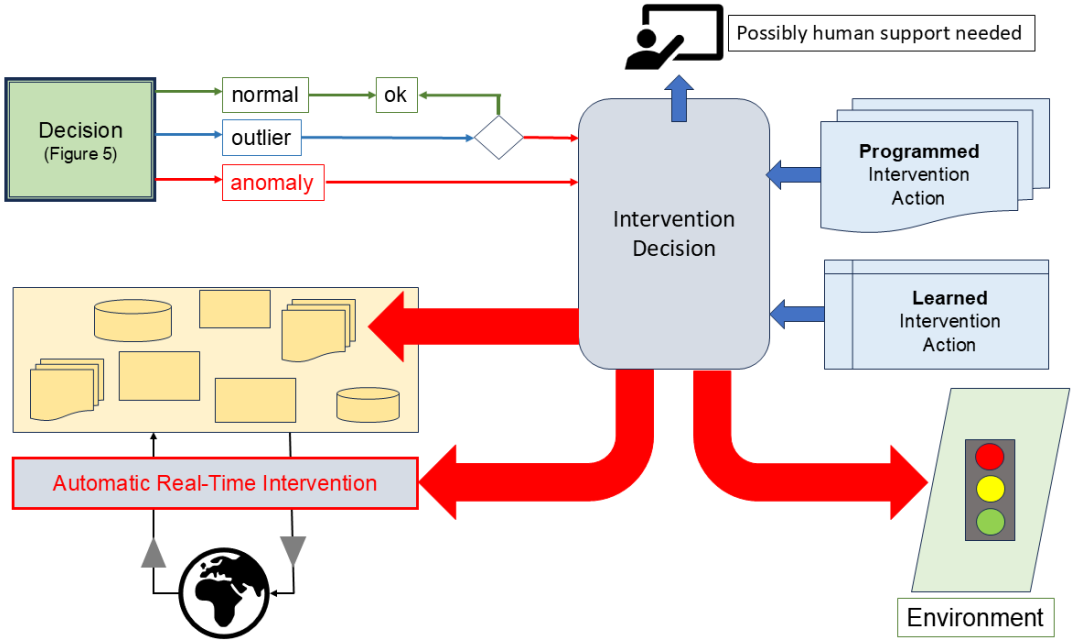


Figure 7. Autonomous Intervention Concept as the Last Defense after Anomaly Detection (Details from Figure 5).

A possible autonomous intervention mechanism is shown in Figure 8. The training part is as described above, usually using real, simulated, or hybrid data. Here, multiple anomaly detection algorithms are used in specific applications, and a voting system decides, thereby reducing false positives and false negatives [32].

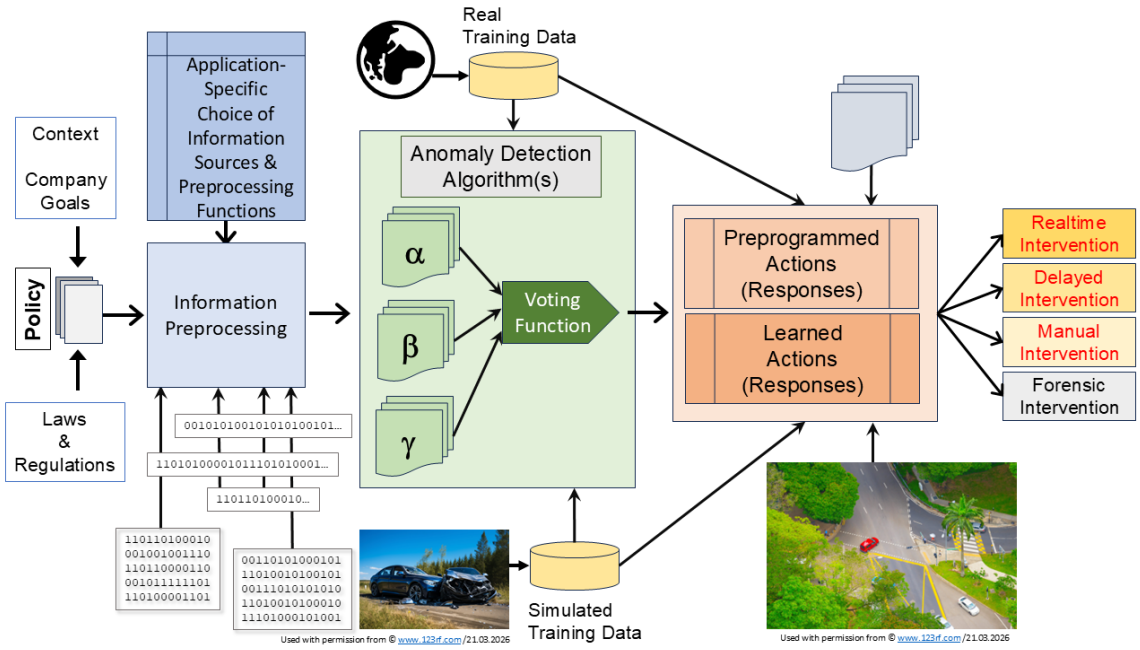


Figure 8. Autonomous Intervention Implementation for the Protection of the Runtime System.

Unfortunately, today's most significant obstacle to automatic, real-time intervention is the lack of binding, comprehensive, actionable laws, regulations, and liability provisions [33].

6. Machine Learning Algorithms

Various machine learning algorithms exist, and new ones are continuously under development [34]. In this tutorial paper, two algorithms are used:

- Deep reinforcement learning [35] for the AI safety example. Reinforcement learning is a machine learning paradigm in which the machine learns by interacting with the context/environment and collects rewards or penalties according to a gain function, to maximize cumulative reward over time;
- Unsupervised learning [36] for the AI security example. Unsupervised machine learning is a machine learning paradigm in which algorithms learn patterns and structure directly from unlabeled data, without using explicit target outputs or example answers to guide learning.

7. Safety: Last Defense

AI/ML for runtime safety in CPSs is a relatively new, exciting engineering discipline. Therefore, no consolidated body of knowledge or accepted development methodology is yet available. Due to the limited space in this publication, an illustrative example is used to demonstrate the power of AI/ML in safety. Note that this is not an experiment, but a conceptual sample demonstrating the power of AI/ML for anomaly detection. The example chosen is autonomous collision avoidance for ships [37,38]. Ocean-going ships face ever-increasing traffic, larger and faster vessels, and a shortage of experienced mariners. Therefore, ship automation, up to fully autonomous ships, is important for the future of sea traffic.

A significant step towards autonomous sailing is fully autonomous collision avoidance. Autonomous collision avoidance for a ship either supports or overrides ($\Longrightarrow$Protective shell) the mariners or guides the ship autonomously.

This example uses constrained Deep Reinforcement Learning as AI technology (RL) [32,39]. Deep reinforcement learning AI is distinguished from supervised or unsupervised learning because it aims at long-term goals or policies [40]. The reinforcement learning AI determines the collision-free trajectory. However, the trajectory is constrained by the IMO's (International Maritime Organization) "COLREG"—the mandatory set of rules preventing collisions at sea [41,42].

The example setup is shown in Figure 9. The ship ("Own Ship") is enclosed in two "bumper zones": An inner bumper for congested waters and an outer zone for open seas. If an obstacle, such as another vessel, a buoy, or a stationary obstacle, enters the bumper zone, a negative reward for intrusion is assigned to the RL. This value function guides the learning process.

Due to the scarcity of real-world training data, the training process utilizes simulation. This is achieved by providing a large number of simulated traffic scenarios during the training phase. After the successful training, the trained neural network is transferred to the operational RL. Note that the training continues during actual sailing, continuously and consistently improving collision-avoidance capability.

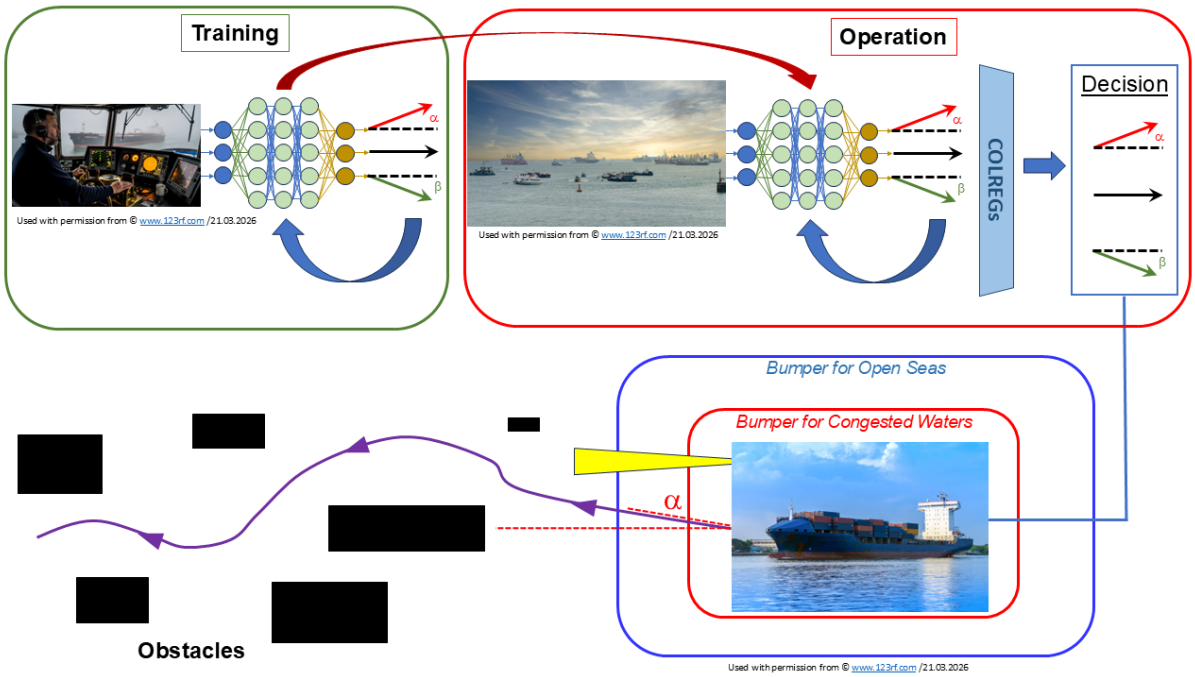


Figure 9. AI/ML Safety Example—Ship Collision Avoidance.

8. Security: Last Defense

AI/ML for runtime security in CPSs is a relatively new, promising engineering discipline. Therefore, no consolidated body of knowledge or accepted development methodology is yet available. Because of the limited space of this publication, again (as in safety above), an illustrative example is used to demonstrate the power of AI/ML in security. Note that this is not an experiment, but a conceptual sample demonstrating the power of AI/ML detection. The example is unsupervised learning for intrusion detection on the MIL-STD-1553 communication bus [43,44].

The MIL-STD-1553 communication bus (https://www.milstd1553.com/, accessed on 10 November 2025) is a message-based local communication system widely accepted for over five decades. Many legacy and modern systems (e.g., the F-35) are based on the MIL-STD-1553. Unfortunately, research has uncovered a number of vulnerabilities that may allow a range of attacks. The MIL-STD-1553 standard was introduced at a time (1973) when cybersecurity was no primary concern. Because today MIL-STD-1553 is a critical component of countless military and civilian platforms, a protocol change is not an option. Therefore, a robust defense in the form of intrusion detection is required!

Figure 10 shows the architecture of the example: it represents an intrusion detection system based on unsupervised machine learning that uses anomalies in bus traffic. The MIL-STD-1553 follows a strict, deterministic message exchange protocol. This time-based regularity yields a set of relevant features that define the model for regular (“normal”) operation.

The example has three components (Figure 9):

(1) The MIL-STD-1553 bus with the bus controller, the bus monitor, and 1…31 bus participants (= remote terminals);

(2) The AI/ML learning component. The first function is a traffic-generation simulator that generates malicious traffic patterns based on a list of attack vectors. The second function is the feature extraction parameters based on a predefined feature list. The relevant feature parameters and their anomaly decision parameters, together with the decision criteria (thresholds), are then provided to the anomaly detection mechanism;

(3) The anomaly detection mechanism. This mechanism monitors bus traffic and detects anomalies using the CUSUM (Cumulative Sum) algorithm [45,46]. This statistical change-detection algorithm [47] identifies when a time-series feature deviates from a reference one. Finally, the decision mechanism—usually based on thresholds—raises an alarm if an anomaly is suspected and triggers defense mechanisms, such as isolating suspicious nodes.

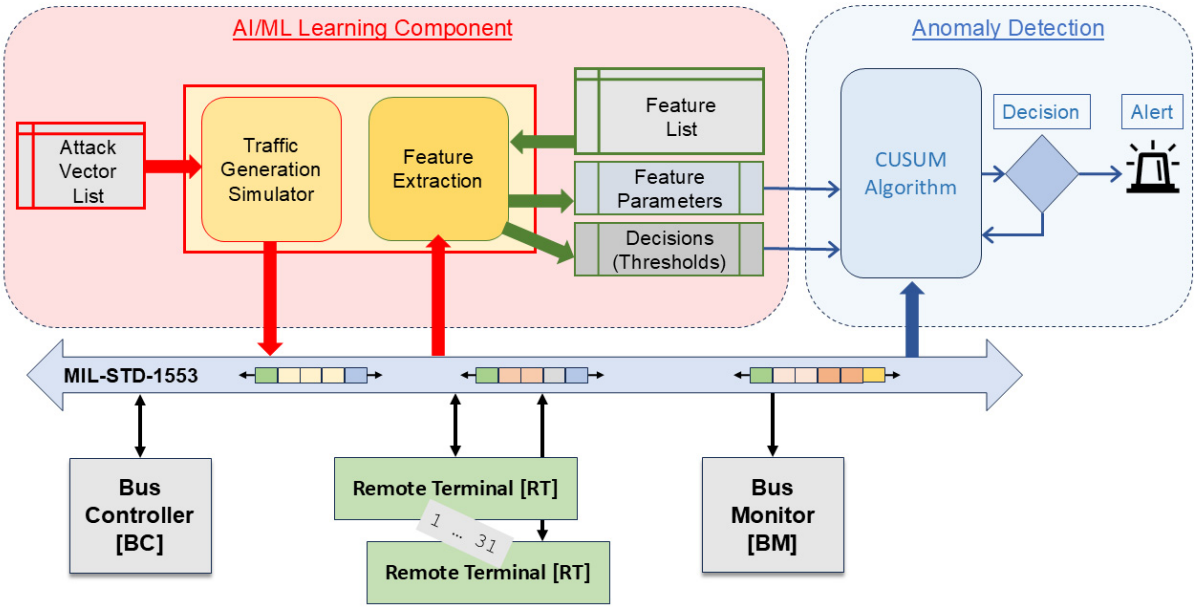


Figure 10. AI/ML security example using anomaly detection in MIL-STD-1553.

9. Protective Shell

The idea of protecting a cyber–physical system against the misbehavior of the artificial/machine learning algorithm during runtime was presumably first presented by Lance Eliot [48]. He proposed it under the name of «AI Guardian Angel Bot» to improve the trustworthiness of machine learning systems. Here, the more technically oriented (and less esoteric) term «Protective Shell» is used.

Figure 11 shows the protective shell [48,49]. During development, evolution, and deployment of the system, all reasonable care is applied to detect, identify, analyze, and mitigate as many vulnerabilities as possible. However, some vulnerabilities inevitably find their way into the runtime system. To mitigate these hidden vulnerabilities, the protective shell includes additional hardware and software to detect, analyze, and respond to anomalies, thereby protecting the system from safety accidents and security incidents during runtime.

In modern CPS systems engineering, the need to supervise an AI/ML-based mission-critical cyber–physical system at runtime has become an accepted best practice [50–52]. Protective shells are an active field of research. Figure 12 presents a conceptual architecture for a protective shell: it is based on the well-known and proven MAPE-K concept (Monitor-Analyze-Plan-Execute with Knowledge [53,54].

The monitoring part reads, preprocesses, and fuses all the available information from the runtime system. The required information is defined during the development process in the safety assurance and security assessment cycles.

The analyze part examines monitoring information and detects, identifies, and classifies anomalies using both preprogrammed scenarios and AI/ML-trained models.

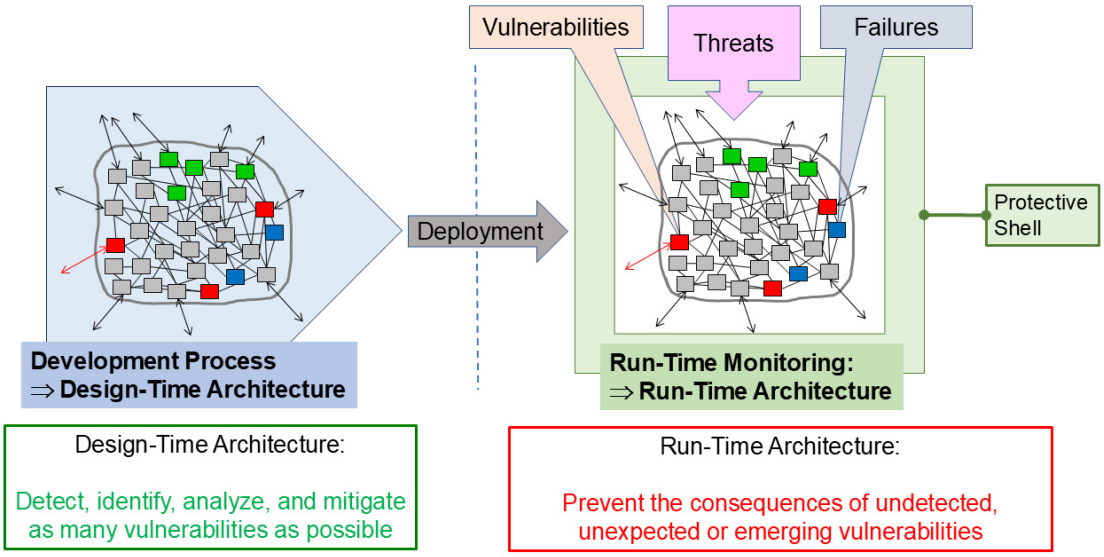


Figure 11. Protective shell including all the elements for anomaly detection and mitigation.

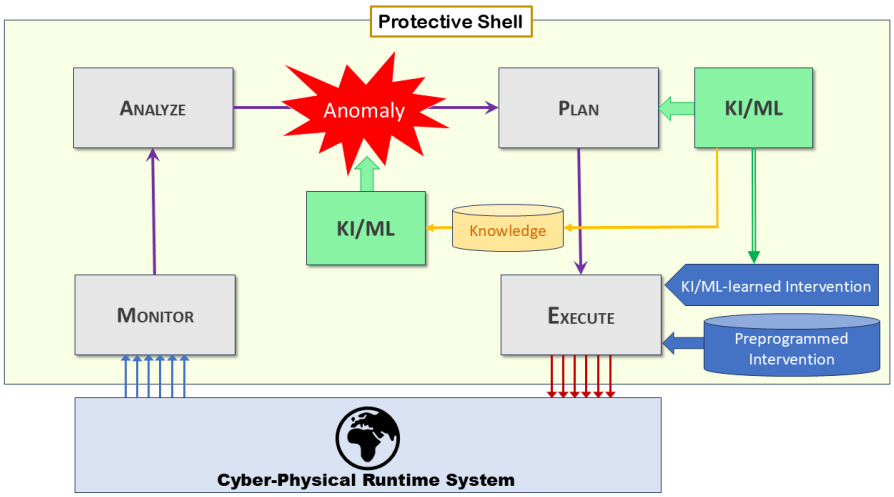


Figure 12. Autonomic Computing with the MAPE-K Architecture Applied to the Protective Shell.

The plan module searches for an appropriate response to protect the application, preferably in real time. Here, both preprogrammed and learned information are used.

The execute module transforms the reaction decided by the plan module into an actual intervention to the runtime system. Note that possible intervention mechanisms must also be defined and implemented during the development cycle. This requires special attention and care.

10. Extended Attack Surface, XAI, and Robust AI

Implementing AI/ML-based runtime monitoring (i.e., a protective shell) introduces four new risks:

(1) The protective shell increases the system's complexity by introducing additional hardware, software, and processes. As experience shows, additional complexity also enlarges the attack surface [55], i.e., generates new failure modes [56] and enables new attack vectors [57]. Unfortunately, effective kill chains, i.e., specific attack methods and tools, are readily available to target AI systems [58]. A meticulous and thorough risk analysis for the protective shell is therefore required.

(2) Artificial intelligence and machine learning may execute autonomous, inexplicable (although possibly correct) decisions and actions. This autonomous behavior may

cause legal, audit, and regulatory issues, but it also poses the danger of harmful decisions. Therefore, safety- or security-critical AI applications must rely solely on explainable artificial intelligence (XAI) mechanisms [59,60] and safety-certified algorithms [52,61].

(3) In AI-based safety- or security-critical CPS, severe damage can result from their malfunction. Failures, malicious activities, deficient learning, and other issues can cause such malfunctions. To guard against these, the AI must be safe and robust. Fortunately, the field of safe and robust AI is developing rapidly, and sources are available [60,62–64].

(4) Unethical use of AI/ML: AI/ML may be used in applications that ignore or violate ethical standards, thereby causing incidental or accidental harm to people, organizations, social structures, or the environment [65,66].

11. AI/ML Laws and Regulations

As with any technology, AI/ML must be embedded in the legal and sociological systems of Nations. As a first step, regulations must be proposed and then transformed into local or international law. Many efforts are underway or completed, e.g., for AI/ML regulations [67–69]. Unfortunately, the individual regulations in different Nations seem to diverge strongly.

Also, great efforts are made in most Nations to develop laws governing the development, governance, use, and evolution of AI/ML [70,71]. Complying with existing law and adapting to evolving law is becoming a significant challenge and effort for businesses using AI/ML, and it also poses a significant commercial risk.

12. Forensics

The protective shell's task is not complete while ensuring the safe and secure operation of the CPS. It must also protect the CPS against unjustified legal action and liability claims arising from a safety accident or security incident. The protective shell—in addition to the already installed data logging activities of the CPS—must collect, store, and protect sufficient information [72–75] to:

(1) Defend against post-accident and post-incident unwarranted accusations;

(2) Allow for authoritative discovery of the causes/sources of accidents and incidents, and use them to improve the CPS.

Note that in many cases, the information gathered during operations must be secured during transfer to storage to prevent accidental or malicious modifications. Here, a digital signature may be required.

13. The Value of Machine Learning for Safety and Security

Figure 13 shows a classification of the paths from the cause of a vulnerability to its exploitation and, finally, to the resulting safety accident or security incident. The cause may either be known (identified) or unknown (hidden). Two types of propagation paths from the cause to the resulting vulnerability of the system exist: deterministic (fully understood propagation path) or statistical (uncertainty in mechanism, timing, and effect). If the vulnerability is exploited, again, two propagation paths to the safety accident or security incident are possible: deterministic and statistical.

For each system, process, or product, the state-of-the-art provides a number of protection mechanisms, such as:

❶ = Generic protection mechanisms (technologies and methods applicable to all systems, processes, and products, such as a formalized, strict, and safety- or security-aware development process [10,76];

❷ = Best practices (technologies and methods applicable to specific systems, processes, and products, such as a firewall to protect the perimeter of an Internet-banking system [77,78];

❸ = Risk Assessment and Management Process [79] aiming to identify and mitigate as many risks as possible;

❹ = Machine Learning [80,81].

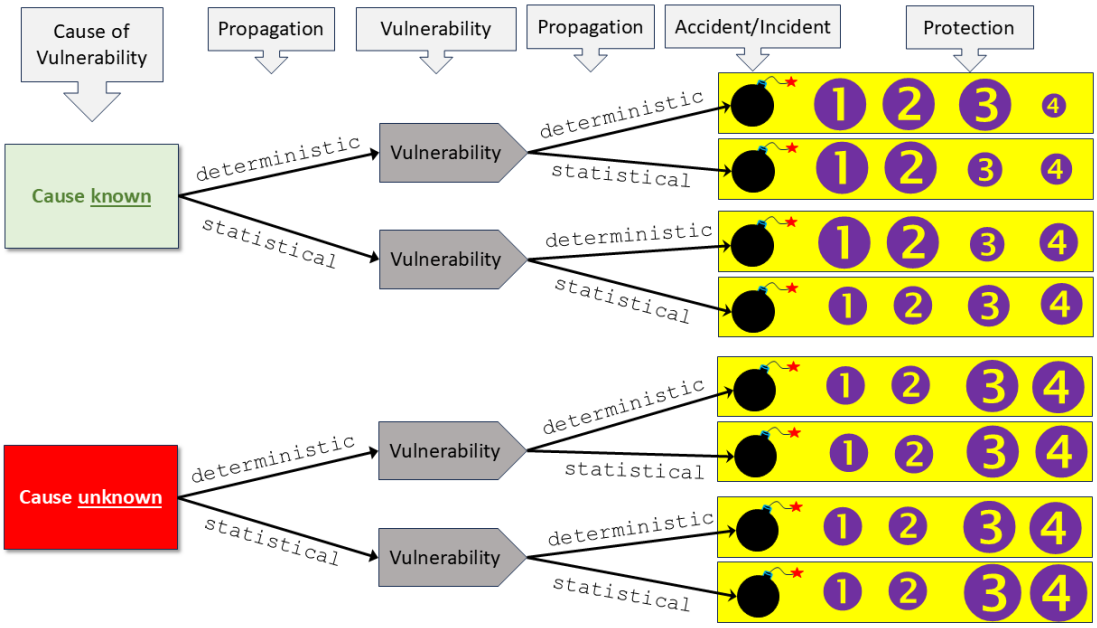


Figure 13. Protection Classification. Legend: ❶ = Generic protection mechanisms; ❷ = Best practices; ❸ = Risk Assessment and Management Process; ❹ = Machine Learning.

Figure 13 shows the relative importance of the four protection mechanisms for the eight cases, represented by the diameter of the circled numbers (size = relative importance for the case in question). The more uncertainty in the system exists, the higher the protection value of AI/ML becomes. Because modern systems, especially systems-of-systems, have tremendous complexity, they contain a mix of all the cases in Figure 13 and include all the causes of the "impossibility statement" (Figure 2)! Therefore, the adequate use of AI/ML is indispensable.

14. Open Challenges

Large-scale application of artificial intelligence/machine learning to the safety and security of cyber–physical systems is a relatively young research discipline. However, it has roots deep in computing history [82]. Therefore, open challenges exist, such as:

- Authoritative and law-conforming policies covering the application of artificial intelligence/machine learning for safety and security in different fields of application and diverse organizations;
- Dependable, transparent machine learning algorithms (Explainable AI) [83];
- Reliable, consistent, and fair training data (either from the real world or simulated), including their provisioning methods;
- Correct, timely, and effective intervention mechanisms, including their planning and decision mechanisms;
- Collection of sufficient, meaningful, and legally acceptable forensic data [73];
- Metrics for the evaluation of the efficiency and effectiveness of AI/ML usage;

- Defense against attacks on the AI/ML algorithms, their implementations, and training data;
- Development and enforcement of ethical principles, guidelines, and standards for AI/ML applications.

15. The Last Question: AI Ethics

This paper is about machines making autonomous decisions that impact the physical world. Erroneous, incomplete, late, or dangerous decisions can cause severe damage to people, organizations, property, or the environment. This fact raises the last question: How are governance, responsibility, liability, and ethical issues handled in autonomous AI/ML decision systems?

An interesting variety of emerging literature on AI ethics exists [84–88]. Ethical challenges will significantly influence AI/ML research and applications in the future.

Fortunately, valuable guidelines for AI/ML systems that respect ethical principles are available, foremost from the United Nations Educational, Scientific, and Cultural Organization [89]. This work defines four values, 10 core principles, and 11 actionable policies for the design, implementation, use, and evolution of AI/ML systems.

At the heart of AI-ethical organizations, services, processes, and products are an adequate ethics policy and assertive governance! A common misunderstanding is that an organization needs only one ethics policy to cover all activities. Unfortunately, ethical issues depend strongly on the application, e.g., fairness in an AI application evaluating job candidates or provable safety in an autonomous vehicle. Therefore, ethics is more of an organizational issue than a technical issue. As a consequence, each organization should implement an ethics framework [69,90,91]. Figure 14 shows the main elements of such a framework:

- A complete, actionable, enforceable ethics policy, aligned to laws, regulations, and general principles, and focused on the specific application;
- A strong, dedicated, and responsible governance (governance board, assigned responsibilities;
- Monitoring and auditing facilities (partially automated) to detect and correct ethics violations.

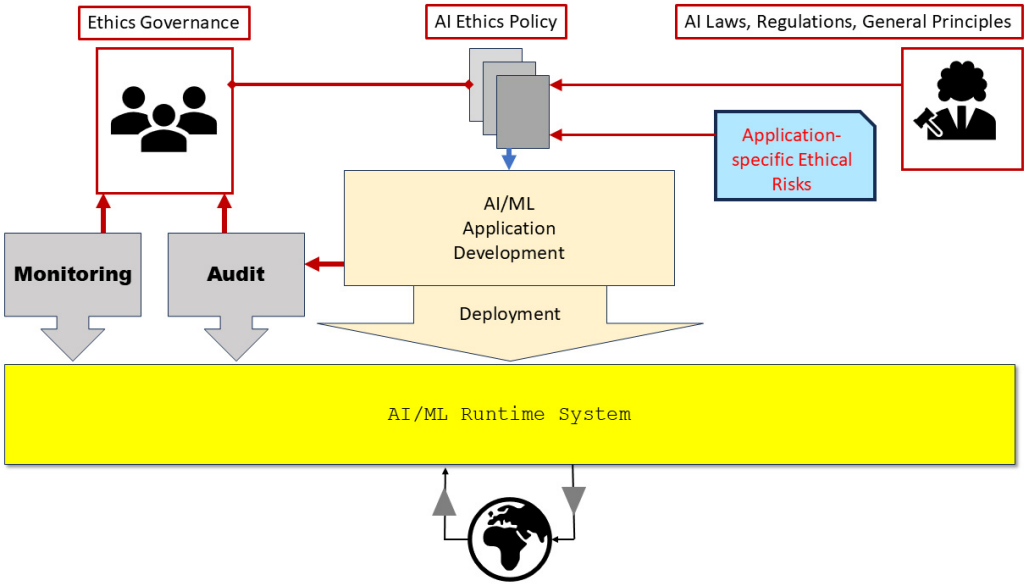


Figure 14. Elements of an Ethics Framework.

16. Conclusions

Many of today's cyber–physical systems control mission-critical applications. Because they can directly influence the physical environment, they can enable or even cause safety accidents or security incidents. Best practices increasingly demand reducing these risks through runtime monitoring and real-time intervention to prevent or mitigate safety accidents or security incidents.

One promising mechanism for detecting imminent misbehavior and mitigating potential consequences is anomaly detection. The runtime system's operation is monitored, and anomalous conditions are identified. To enable timely detection of anomalous behavior, not only preprogrammed scenarios are used, but also artificial intelligence/machine learning is applied.

The methods and mechanisms for anomaly detection, classification, and mitigation are the elements of the protective shell. The protective shell includes specific hardware, software, sensors, and processes for preventing and mitigating runtime system safety accidents or security incidents through anomaly-based detection.

Adding a protective shell to the runtime system increases the system's complexity. The protective shell must therefore be developed with utmost care with respect to safety and security, including emerging behaviors.

Anomaly detection and mitigation of the possible consequences with AI/ML is a fascinating, active field of research. This tutorial paper presents an introduction to the relevant concepts.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data availability does not apply to this article as no new data were created or analyzed in this study.

Acknowledgments: The author thanks the Technical University of Dresden, its Computer Science Faculty, his colleagues, and the numerous students with whom he enjoyed working on CPS safety and security. Also, thanks go to the three (anonymous) reviewers who improved both the content and the quality of the paper.

Conflicts of Interest: The author has no conflicts of interest.

References

1. Alur, R. *Principles of Cyber-Physical Systems*; MIT Press: Cambridge, MA, USA, 2015; ISBN 978-0-262-02911-7.
2. Möller, D. *Guide to Computing Fundamentals in Cyber-Physical Systems—Concepts, Design Methods, and Applications*; Springer International Publishing: Cham, Switzerland, 2016; ISBN 978-3-319-79747-2.
3. Griffor, E. *Handbook of System Safety and Security—Cyber Risk and Risk Management, Cyber Security, Threat Analysis, Functional Safety, Software Systems, and Cyber-Physical Systems*; Elsevier: Amsterdam, The Netherlands, 2016; ISBN 978-0-128-03773-7.
4. Rainey, L.B.; Jamshidi, M. *Engineering Emergence—A Modeling and Simulation Approach*; CRC Press: Boca Raton, FL, USA, 2018; ISBN 978-1-138-04616-0.
5. Fitzgerald, J.; Morisset, C. Can We Develop Holistic Approaches to Delivering Cyber-Physical Systems Security? *Res. Dir. Cyber-Phys. Syst.* **2024**, *2*, e2. [CrossRef]
6. Papow, J. *Glitch—The Hidden Impact of Faulty Software*; Prentice Hall: Upper Saddle River, NJ, USA, 2010; ISBN 978-0-132-16063-6.
7. Stewart, A.J. *A Vulnerable System—The History of Information Security in the Computer Age*; Cornell University Press: Ithaca, NY, USA, 2021; ISBN 978-1-501-75894-2.
8. Axelrod, C.W. *Engineering Safe and Secure Software Systems*; Artech House Books: Norwood, MA, USA, 2012; ISBN 978-1-608-07472-3.
9. Bahr, N.J. *System Safety Engineering and Risk Assessment: A Practical Approach*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2017; ISBN 978-1-138-89336-8.

10. Furrer, F.J. *Safety and Security of Cyber-Physical Systems*; Springer Vieweg: Wiesbaden, Germany, 2022; ISBN 978-3-658-37181-4.
11. Knight, J. *Fundamentals of Dependable Computing for Software Engineers*; CRC Press: Boca Raton, FL, USA, 2012; ISBN 978-1-439-86255-1.
12. Furrer, F.J. *Future-Proof Software-Systems—A Sustainable Evolution Strategy*; Springer Vieweg: Wiesbaden, Germany, 2019; ISBN 978-3-658-19937-1.
13. Kopetz, H. An Architecture for Safe Driving Automation. In *Principles of Systems Design*; Raskin, J.F., Chatterjee, K., Doyen, L., Majumdar, R., Eds.; Lecture Notes in Computer Science; Springer Nature: Cham, Switzerland, 2022; Volume 13660, ISBN 978-3-031-22336-5. [CrossRef]
14. Banish, G. *OBD-I & OBD-II—A Complete Guide to Diagnosis, Repair, and Emissions Compliance*; CarTech Books: North Branch, MN, USA, 2023; ISBN 978-1-61325-752-4.
15. Bartocci, E.; Falcone, Y. (Eds.) *Lectures on Runtime Verification—Introductory and Advanced Topics*; Springer Nature: Cham, Switzerland, 2018; ISBN 978-3-319-75631-8 (LNCS 10457).
16. Harrison, L. How to Use Runtime Monitoring for Automotive Functional Safety. TechDesignForum White Paper 2020. Available online: https://www.ti.com/lit/wp/snla457/snla457.pdf?ts=1773218575155&ref_url=https%253A%252F%252Fwww.bing.com%252F (accessed on 9 December 2025).
17. Haupt, N.B.; Liggesmeyer, P. A Runtime Safety Monitoring Approach for Adaptable Autonomous Systems. In *Computer Safety, Reliability, and Security*; Romanovsky, A., Troubitsyna, E., Gashi, I., Schoitsch, E., Bitsch, F., Eds.; SAFECOMP 2019. Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2019; Volume 11699. Available online: https://www.researchgate.net/publication/335557336_A_Runtime_Safety_Monitoring_Approach_for_Adaptable_Autonomous_Systems/link/5dd2b7b0299bf1b74b4e14f7/download (accessed on 10 November 2025).
18. Kane, A. Runtime Monitoring for Safety-Critical Embedded Systems. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2015. Available online: https://users.ece.cmu.edu/ koopman/thesis/kane.pdf (accessed on 10 November 2025).
19. Khan, M.T.; Serpanos, D.; Shrobe, H. Sound and Complete Runtime Security Monitor for Application Software. *arXiv* **2016**, arXiv:1601.04263v1. Available online: https://www.researchgate.net/publication/291229626_Sound_and_Complete_Runtime_Security_Monitor_for_Application_Software (accessed on 10 November 2025).
20. Janicke, H.; Nicholson, A.; Webber, S.; Cau, A. Run-Time-Monitoring for Industrial Control Systems. *Electronics* **2015**, *4*, 995–1017. [CrossRef]
21. Mehrotra, K.G.; Mohan, C.K.; Huang, H. *Anomaly Detection Principles and Algorithms (Terrorism, Security, and Computation)*; Springer International Publishing: Cham, Switzerland, 2019; ISBN 978-3-319-88445-5.
22. Dunning, T.; Friedman, E. *Practical Machine Learning—A New Look at Anomaly Detection*; O'Reilly and Associates: Sebastopol, CA, USA, 2014; ISBN 978-1-491-91160-0.
23. Bhuyan, H. *Network Traffic Anomaly Detection and Prevention—Concepts, Techniques, and Tools*; Springer International Publishing: Cham, Switzerland, 2018; ISBN 978-3-319-87968-0.
24. Kuo, C. *Modern Time Series Anomaly Detection—With Python & R Code Examples*; Packt Publishing: Birmingham, UK, 2022; ISBN 979-8-363-29575-1.
25. Sarkis. *Training Data for Machine Learning—Human Supervision from Annotation to Data*; O'Reilly and Associates: Sebastopol, CA, USA, 2023; ISBN 978-1-492-09452-4.
26. Nikolenko, S.I. *Synthetic Data for Deep Learning*; Springer Nature Switzerland: Cham, Switzerland, 2021; ISBN 978-3-030-75177-7.
27. Nassif, J.; Tekli, J.; Kamradt, M. *Synthetic Data—Revolutionizing the Industrial Metaverse*; Springer Nature: Cham, Switzerland, 2024; ISBN 978-3-031-47559-7.
28. Buttfield-Addison, P.; Buttfield-Addison, M.; Nugent, T.; Manning, J. *Practical Simulations for Machine Learning—Using Synthetic Data for AI*; O'Reilly Media Inc.: Sebastopol, CA, USA, 2022; ISBN 978-1-492-08992-6.
29. Adari, S.K.; Alla, S. *Beginning Anomaly Detection Using Python-Based Deep Learning—Implement Anomaly Detection Applications with Keras and PyTorch*; Apress Media: New York, NY, USA, 2024; ISBN 979-8-8688-0007-8. [CrossRef]
30. Vaseghi, S.V. *Advanced Digital Signal Processing and Noise Reduction*, 2nd ed.; John Wiley & Sons: Chichester, UK, 2000; ISBN 978-0-4716-2692-3.
31. Burger, W.; Burge, M.J. *Digital Image Processing—An Algorithmic Introduction*, 3rd ed.; Springer Nature Switzerland: Cham, Switzerland, 2022; ISBN 978-3-031-05743-4. [CrossRef]
32. Bilgin, E. *Mastering Reinforcement Learning with Python—Build Next-Generation, Self-Learning Models Using Reinforcement Learning Techniques and Best Practices*; Packt Publishing: Birmingham, UK, 2020; ISBN 978-1-838-64414-7.
33. Beckers, A.; Teubner, G. *Three Liability Regimes for Artificial Intelligence—Algorithmic Actants, Hybrids, Crowds*; HART Publishing: Oxford, UK, 2023; ISBN 978-1-5099-4937-3.
34. Prince, S.J.D. *Understanding Deep Learning*; The MIT Press: Cambridge, MA, USA, 2023; ISBN 978-0-262-04864-4.
35. Oh, J.; Farquhar, G.; Kemaev, I.; Calian, D.A.; Hessel, M.; Zintgraf, L.; Singh, S.; van Hasselt, H.; Silver, D. Discovering state-of-the-art reinforcement learning algorithms. *Nature* **2025**, *648*, 312. [CrossRef] [PubMed]

36. Koul, N. *Kickstart Unsupervised Machine Learning*; Orange Education Pvt Ltd.: Delhi, India, 2025; ISBN -978-9-349-88743-5.
37. Hashimoto, H.; Nishimura, H.; Nishiyama, H.; Higuchi, G. Development of AI-based Automatic Collision Avoidance System and Evaluation by Actual Ship Experiment. *ClassNK Tech. J.* **2021**, *3*, 41–51. Available online: https://www.classnk.or.jp/hp/pdf/research/rd/2021/03_e05.pdf (accessed on 10 November 2025).
38. Shen, H.; Hashimoto, H.; Matsuda, A.; Taniguchi, Y.; Terada, D.; Guo, C. Automatic collision avoidance of multiple ships based on deep Q-learning. *Appl. Ocean. Res.* **2019**, *86*, 268–288. [CrossRef]
39. Francois-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An Introduction to Deep Reinforcement Learning. *Found. Trends Mach. Learn.* **2018**, *11*, 219–354. [CrossRef]
40. Winder, P. *Reinforcement Learning—Industrial Applications of Intelligent Agents*; O'Reilly Media: Sebastopol, CA, USA, 2020; ISBN 978-1-098-11483-1.
41. IMO. *COLREG—Preventing Collisions at Sea*; IMO (The International Maritime Organization): London, UK, 1972. Available online: https://www.imo.org/en/OurWork/Safety/Pages/Preventing-Collisions.aspx (accessed on 10 November 2025).
42. Cockcroft, A.N.; Lameijer, J.N.F. *A Guide to the Collision Avoidance Rules*, 7th ed.; Butterworth-Heinemann: Kidlington, UK, 2012.
43. Hadeer, A.; Traore, I.; Quinan, P.; Ganame, K.; Boudar, O. A Collection of Datasets for Intrusion Detection in MIL-STD-1553. In *Artificial Intelligence for Cyber-Physical Systems Hardening*; Springer: Cham, Switzerland, 2022; Chapter 4. [CrossRef]
44. Sachdev, K.; Saad, H.S.; Traore, I.; Ganame, K.; Boudar, O. *Unsupervised Anomaly Detection for MIL-STD-1553 Avionic Platforms Using CUSUM*; Chapter 5 in Artificial Intelligence for Cyber-Physical Systems Hardening; Springer: Berlin/Heidelberg, Germany, 2023. [CrossRef]
45. Granjon, P. *The CUSUM Algorithm*; GIPSA-lab, Grenoble Campus: Saint Martin d'Hères, France, 2014. Available online: https://hal.science/hal-00914697/document (accessed on 10 November 2025).
46. Koshti, V.V. Cumulative Sum Control Chart. *Int. J. Phys. Math. Sci.* **2011**, *1*, 28–32. Available online: https://www.researchgate.net/publication/230888065_Cumulative_sum_control_chart (accessed on 10 November 2025).
47. Page, E.S. Continuous Inspection Schemes. *Biometrika* **1954**, *41*, 100–115. [CrossRef]
48. Eliot, L. *AI Guardian Angel Bots for Deep AI Trustworthiness—Practical Advances in Artificial Intelligence*; LBE Press Publishing: Tucker, GA, USA, 2016; ISBN 978-0-6928-0061-4.
49. Furrer, F.J. Safe and secure system architectures for cyber-physical systems. *Inform. Spektrum* **2023**, *46*, 96–103. [CrossRef]
50. Diver, R. *Guardians of AI—Building Innovation and Safety with AI*; Amazon Fulfillment: Wroclaw, Poland, 2024; ISBN 979-8-32210-964-4.
51. Voulgaris, Z.; Engelfriet, A. *AI Safety—Strategies for Ensuring Responsible, Ethical, and Reliable AI Systems*; Technics Publications: Sedona, AZ, USA, 2025; ISBN 978-1-63462-734-4.
52. Landon, E.V. *Engineering Safety for Embodied AI (Building Trustworthy AI in Robotics, Vehicles, and Beyond)*; Amazon Italia Logistics S.r.l.: Torrazza Piemonte, TO, Italy, 2025; ISBN 979-8-264618-43-7.
53. Lalanda, P.; McCann, J.A.; Diaconescu, A. *Autonomic Computing—Principles, Design and Implementation*; Springer: London, UK, 2013; ISBN 978-1-4471-5006-0.
54. IBM. *Business Consulting Services an Architectural Blueprint for Autonomic Computing*, 4th ed.; IBM Autonomic Computing: Yorktown Heights, NY, USA, 2006. Available online: http://www-01.ibm.com/software/tivoli/autonomic (accessed on 10 November 2025).
55. Cybellium Ltd. *Mastering Attack Surface Management—A Comprehensive Guide to Learn Attack Surface Management*; Cybellium Ltd.: Tel Aviv, Israel, 2023; ISBN 979-8-8591-4008-4.
56. Stamatis, D.H. *Risk Management Using Failure Mode and Effect Analysis (FMEA)*; Quality Press: Milwaukee, MI, USA, 2019; ISBN 978-0-8738-9978-9.
57. Haber, M.J.; Hibbert, B. *Asset Attack Vectors—Building Effective Vulnerability Management Strategies to Protect Organizations*; Apress Media LLC.: New York, NY, USA, 2018; ISBN 978-1-4842-3626-0. [CrossRef]
58. Gibian, D. *Hacking Artificial Intelligence—A Leader's Guide from Deepfakes to Breaking Deep Learning*; Rowman & Littlefield: London, UK, 2022; ISBN 978-1-5381-5508-0.
59. Simon, C. *Deep Learning and XAI Techniques for Anomaly Detection—Integrate the Theory and Practice of Deep Anomaly Explainability*; Packt Publishing: Birmingham, UK, 2023; ISBN 978-1-804-61775-5.
60. Huang, X.; Jin, G.; Ruan, W. *Machine Learning Safety*; Springer Nature Singapore Pte. Ltd.: Singapore, 2023; ISBN 978-981-19-6813-6. [CrossRef]
61. Koopman, P. *Embodied AI Safety (Reimagining Safety Engineering for Artificial Intelligence in Physical Systems)*; Amazon Fulfillment: Wroclaw, Poland, 2025; ISBN 979-8-29238-441-0.
62. Guerraoui, R.; Gupta, N.; Pinot, R. *Robust Machine Learning—Distributed Methods for Safe AI*; Springer Nature Singapore: Singapore, 2024; ISBN 978-981-97-0687-7. [CrossRef]
63. Yampolskiy, R.V. *AI—Unexplainable, Unpredictable, Uncontrollable*; Chapman & Hall/CRC: Boca Raton, CA, USA, 2024; ISBN 978-1-032-57626-8. [CrossRef]

64. Hall, P.; Curtis, J.; Pandey, P. *Machine Learning for High-Risk Applications—Techniques for Responsible AI*; O'Reilly Media Inc.: Sebastopol, CA, USA, 2023; ISBN 978-1-098-10243-2.
65. King, A. *AI, Automation and War—The Rise of a Military-Tech Complex*; Princeton University Press: Princeton, NJ, USA, 2025; ISBN 978-0-691-26514-8.
66. Blessed Editions. *The Silent Threat of Artificial Intelligence—Geoffrey Hinton's Urgent Warning*; Springer: Singapore, 2025; ISBN 979-8-2922-7287-8.
67. Windholz, N. *The EU AI Act Handbook: Compliant Usage of Artificial Intelligence in the Private and Public Sectors*; Carl Hanser Verlag GmbH & Co. KG: Munich, Germany, 2025; ISBN 978-1-56990-314-8.
68. EUR-Lex Regulation (EU). *2024/1689 of the European Parliament and of the Council of 13 June 2024 Laying Down Harmonised Rules on Artificial Intelligence*; Document 32024R1689; Official Journal of the European Union: Brussels, Belgium, 2025. Available online: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689 (accessed on 10 November 2025).
69. European Union Artificial Intelligence Act. *Regulation (EU) 2024/1689 of the European Parliament and of the Council*; Official Journal of the European Union: Brussels, Belgium, 2024. Available online: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L_202401689 (accessed on 10 November 2025).
70. Lee, J. *Artificial Intelligence and International Law*; Springer Nature Singapore Pte Ltd.: Singapore, 2023; ISBN 978-981-191498-0. [CrossRef]
71. Tzimas, T. *Legal and Ethical Challenges of Artificial Intelligence from an International Law Perspective*; Springer Nature: Cham, Switzerland, 2022; ISBN 978-3-030-78587-1. [CrossRef]
72. Winkle, T. *Product Development within Artificial Intelligence, Ethics, and Legal Risk—Exemplary for Safe Autonomous Vehicles*; Springer Fachmedien Vieweg: Wiesbaden, Germany, 2022; ISBN 978-3-658-34292-0. [CrossRef]
73. Marcella, A.J. *Editor Cyber Forensics—Examining Emerging and Hybrid Technologies*; CRC Press: Boca Raton, FL, USA, 2021; ISBN 978-0-367-52424-1.
74. Kävrestad, J. *Fundamentals of Digital Forensics—Theory, Methods, and Real-Life Applications*, 2nd ed.; Springer Nature: Cham, Switzerland, 2020; ISBN 978-3-030-38953-6. [CrossRef]
75. Oettinger, W. *Learn Computer Forensics—Your One-Stop Guide to Searching, Analyzing, Acquiring, and Securing Digital Evidence*, 2nd ed.; Packt Publishing: Birmingham, UK, 2022; ISBN 978-1-803-23830-2.
76. Karmakar, G.; Wakankar, A.; Kabra, A.; Pandya, P. *Development of Safety-Critical Systems—Architecture and Software*; Springer Nature Switzerland: Cham, Switzerland, 2023; ISBN 978-3-031-27900-3.
77. Leveson, N.G. *An Introduction to System Safety Engineering*; The MIT Press: Cambridge, MA, USA, 2023; ISBN 978-0-262-54688-1.
78. Mudgal, S. *Managing the Cyber Risk—A CISO's Practical Guide to Threat and Vulnerability Management*; BPB Publications: New Delhi, India, 2025; ISBN 978-93-65892-918.
79. Landoll, D. *The Security Risk Assessment Handbook—A Complete Guide for Performing Security Risk Assessments*; CRC Press: Boca Raton, FL, USA, 2021; ISBN 978-1-032-04165-0.
80. Musser, M.; Garriott, A. *Machine Learning and Cybersecurity—Hype and Reality*; CSET Center for Security and Emerging Technology, Georgetown Walsh School of Foreign Services; Georgetown University: Washington, DC, USA, 2021. [CrossRef]
81. NAP (National Academies of Sciences, Engineering, and Medicine). *Machine Learning for Safety-Critical Applications: Opportunities, Challenges, and a Research Agenda*; The National Academies Press: Washington, DC, USA, 2025. [CrossRef]
82. VC3. *Corporation the Evolution of Artificial Intelligence in Cybersecurity*; VC3 Corporation: Columbia, SC, USA, 2023. Available online: https://www.vc3.com/blog/the-evolution-of-artificial-intelligence-in-cybersecurity (accessed on 10 November 2025).
83. Mehta, M.; Palade, V.; Chatterjee, I. *Explainable AI—Foundations, Methodologies and Applications*; Springer Nature: Cham, Switzerland, 2022; ISBN 978-3-031-12806-6. [CrossRef]
84. Coeckelbergh, M. *AI Ethics*; The MIT Press: Cambridge, MA, USA, 2020; ISBN 978-0-262-53819-0.
85. EU. *Regulation of the European Parliament and of the Council laying Down Harmonized Rules on Artificial Intelligence (Artificial Intelligence Act)*; 21.4.2021 COM(2021) 206 Final; European Commission: Brussels, Belgium, 2021. Available online: https://eur-lex.europa.eu/resource.html?uri=cellar:e0649735-a372-11eb-9585-01aa75ed71a1.0001.02/DOC_1&format=PDF (accessed on 10 November 2025).
86. Press, L. *EU Artificial Intelligence Act—The Essential Reference*; Lex Press: Paris, France, 2024; ISBN 979-8-8827-1805-2.
87. Huang, C.; Zhang, Z.; Mao, B.; Yao, X. An Overview of Artificial Intelligence Ethics. *IEEE Trans. Artif. Intell.* **2023**, *4*, 799–819. [CrossRef]
88. Boddington, P. *AI Ethics: Foundations, Theory, and Algorithms*; Springer Nature Singapore Pte Ltd.: Singapore, 2024; ISBN 978-981-19-9384-8. [CrossRef]
89. UNESCO (United Nations Educational, Scientific and Cultural Organization). *Recommendation on the Ethics of Artificial Intelligence*; UNESCO: Paris, France, 2023. Available online: https://www.unesco.org/en/artificial-intelligence/recommendation-ethics (accessed on 10 November 2025).

90. Davis, J.G. *The Path Forward—Restructuring AI for Safety, Security & Sovereign Accountability. A Modern Approach to Modern Autonomous Systems*; ALETHEIA Protocol: Acapulco, Mexico, 2026; ISBN 979-8242379497.
91. Spark, C. *Ethical AI Blueprint—Frameworks for Fairness, Safety, and Trust in Machine Learning*; Published Independently by Charles Spark; 2025; ISBN 979-8268446548.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.