Article

Not peer-reviewed version

Cyber-Physical Systems: The Last Defense

Frank J Furrer *

Posted Date: 16 September 2025

doi: 10.20944/preprints202509.1380.v1

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

 doi:10.20944/preprints202509.1380.v1

Article

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 a cross-domain and holistic process. The process encompasses all elements of a cyber-physical system, including computation infrastructure, software, interfaces to the physical world, human interactions, as well as safety and security engineering domains. The process is holistic because it must assure conceptual integrity and correct interoperability across all elements of the CPS. Unfortunately, in all stages of this process, vulnerabilities can be introduced into the system (Because of 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 anomaly detection during runtime, using artificial intelligence/machine learning as the detection mechanism. This 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.

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

Introduction and Context

Cyber-physical systems (Rajeev, 2015/Möller, 2016) are a double-edged sword. On the one hand, they are extremely useful and enable a large number of 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 that encompasses the development, evolution, and operation of the CPS. This process is a cross-domain and holistic process. The process is cross-domain because it covers all elements of the cyber-physical system, i.e., computation infrastructure, software, interfaces to the physical world, and human interactions, including the 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 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 (Griffor, 2016). Vulnerabilities generated during the development process are transferred into the runtime system! During operation, additional vulnerabilities appear, such as emergent behavior and emergent properties (Rainey, 2018), autonomous decisions, malicious attacks, errors or faults in an element of the runtime system, failures in the execution platform, systems software weaknesses

 doi:10.20944/preprints202509.1380.v1

(Operating system, database, browser, etc), sensor and actuator malfunctions, and uncertain or unpredicted operating conditions.

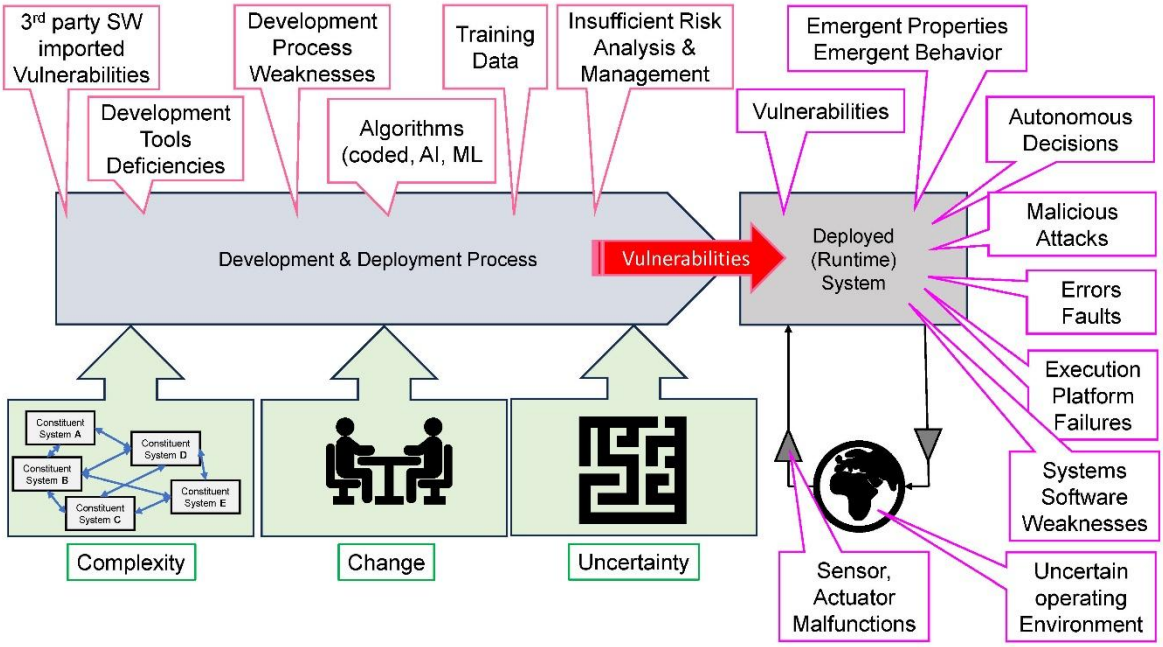

Figure 1. Cyber-Physical System Threat Context.

In many cases, the cause of a safety accident or security incident is software-related (Papow, 2010; Stewart, 2021). Fortunately, a vast and valuable literature exists for building safe and secure systems (e.g., Axelrod, 2012; Bahr, 2017; Furrer, 2022; Knight, 2012). However, the undeniable fact remains that cyber-physical runtime systems contain hidden vulnerabilities, which may become the source of 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" (Furrer, 2019). The three devils often force damaging compromises on design and implementation decisions!

Cyber-Physical Systems Behavior

The cyber-physical runtime system has precisely two categories of behavior (Figure 2):

(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, trustworthy in all situations and operating conditions;

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

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.

 doi:10.20944/preprints202509.1380.v1

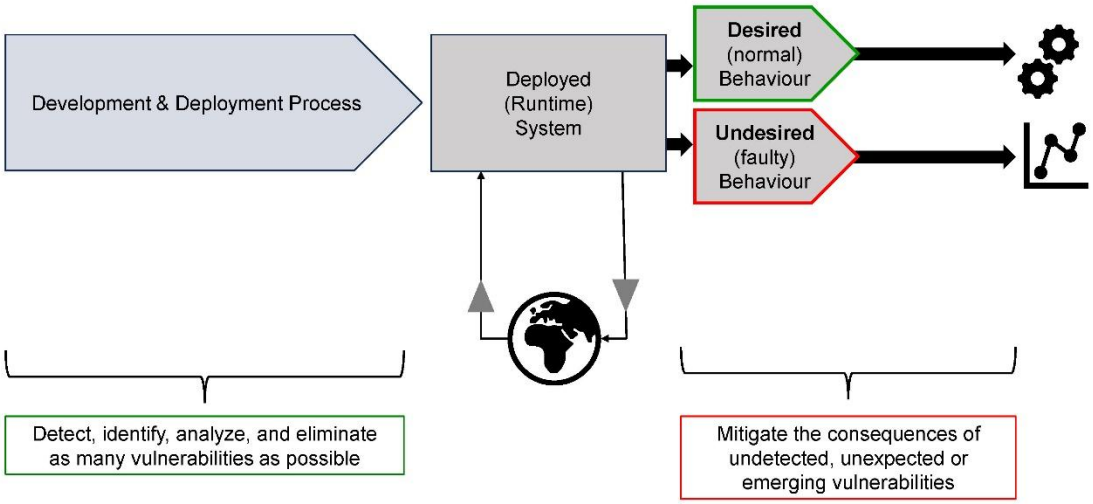

Figure 2. Desired and Undesired Behavior of a Cyber-Physical System.

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 (Banish, 2023). An extensive literature on runtime monitoring and its applications exists (e.g., Bartocci et al., 2018; Harrison, 2020; Haupt et al., 2019; Kane, 2015; Khan et al., 2016; Janicke et al., 2015).

The key idea of runtime monitoring is shown in Figure 3: 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.

"Classical" runtime monitoring is based on programmed algorithms, i.e., faulty conditions trigger a preprogrammed intervention (Bartocci et al., 2018). 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.

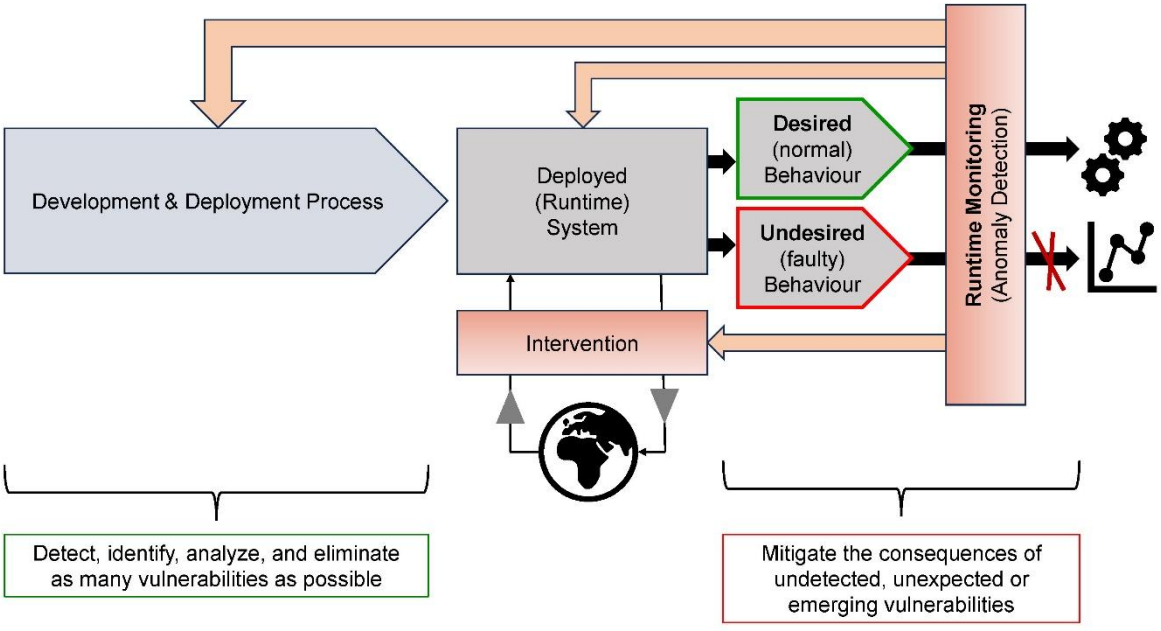

Figure 3. CPS last Defense – Runtime Anomaly Detection and Real-time Intervention.

 doi:10.20944/preprints202509.1380.v1

Monitor – Analyze – Intervene

Figure 4 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.

Monitor

The anomaly detection element (e.g., Mehrotra, 2019; Dunning, 2014; Bhuyan, 2018) 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 (such as sensor and actuator values and internal variables, such as speed or position). As reference inputs for the desired behavior, the static part utilizes functional specifications (including quality property requirements), a set of rules (such as bracket values for output signals), policies, models, log files, and operational data (e.g., Bartocci et al., 2018). Anomalies are detected if discrepancies between the inputs and the reference information are discovered.

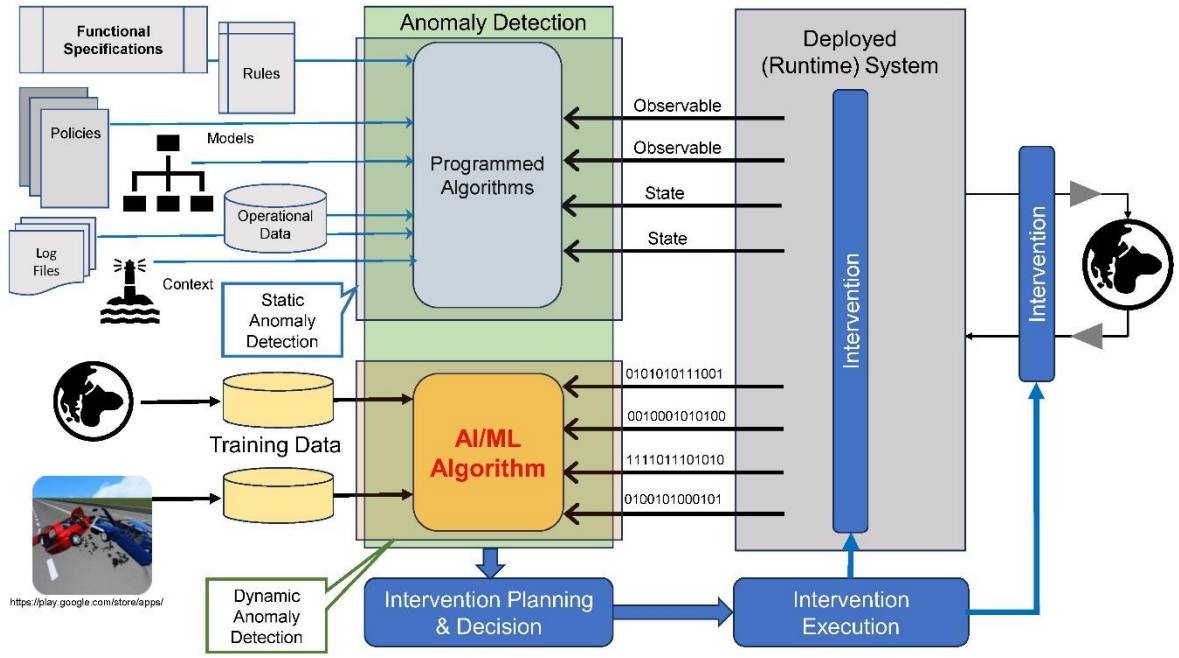

Figure 4. Real-Time Intervention.

The dynamic part contains AI/Machine learning. Its inputs are information streams (Time series; Kuo, 2022) 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 (Sarkis, 2023). 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 (e.g., Nikolenko, 2021 / Nassif, 2024).

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 offers excellent support for generating synthetic training data effectively (Buttfield-Addison et al., 2022).

 doi:10.20944/preprints202509.1380.v1

Analyze

During the analysis phase, the ML algorithm compares the actual behavioral data of the cyber-physical system with a previously learned "normal" behavior (e.g., Adari, 2024). A conceptual diagram is shown in Figure 5. 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 model for normal behavior using machine learning, trained on real data, simulated data, or both. As a third step, anomaly criteria must be defined as rules or parameters for the model. Now, the system is ready for operation and can be deployed.

In many cases, incoming data streams are subjected to a noise reduction or removal algorithm (e.g., Vaseghi, 2000/Burger, 2022). 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 (Adari, 2024). After an anomaly has been identified, an intervention is triggered. Many systems continue to use machine learning during operation to enhance their detection capabilities.

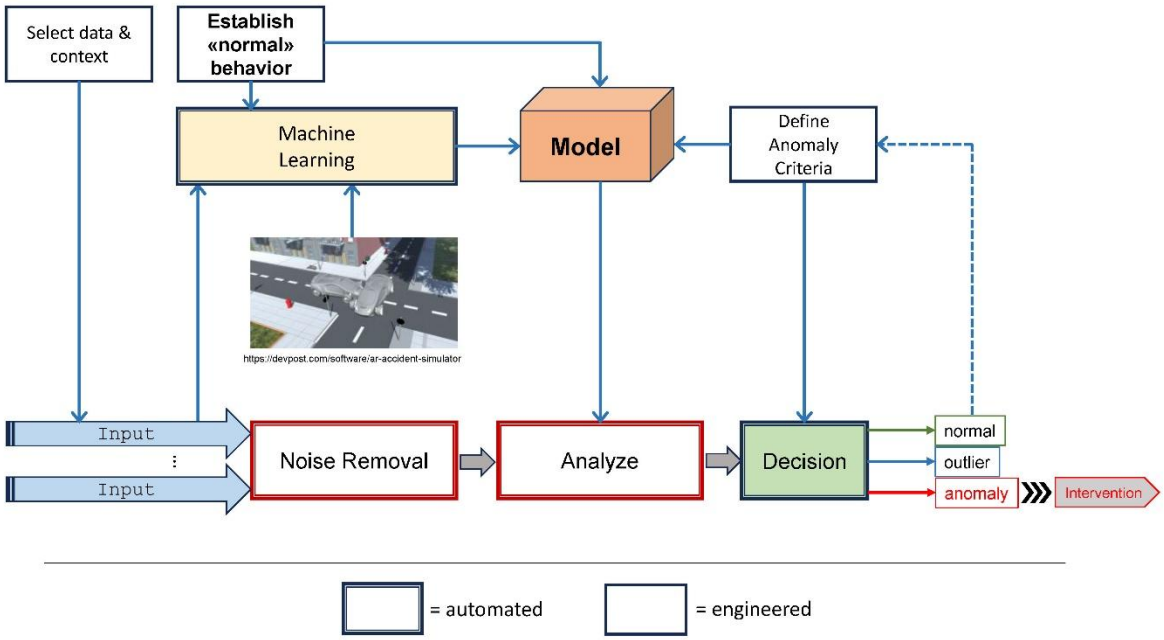

Figure 5. ML Analysis Phase.

Intervene

Intervention is the most risky phase of the last defense: Because autonomous, machine decisions directly impact the physical world, they can become the reason for safety accidents or security incidents. Figure 6 shows the autonomous intervention concept. Once the decision is made 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 of 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;

 doi:10.20944/preprints202509.1380.v1

- Shutting down a TCP/IP port when excessive inor 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.

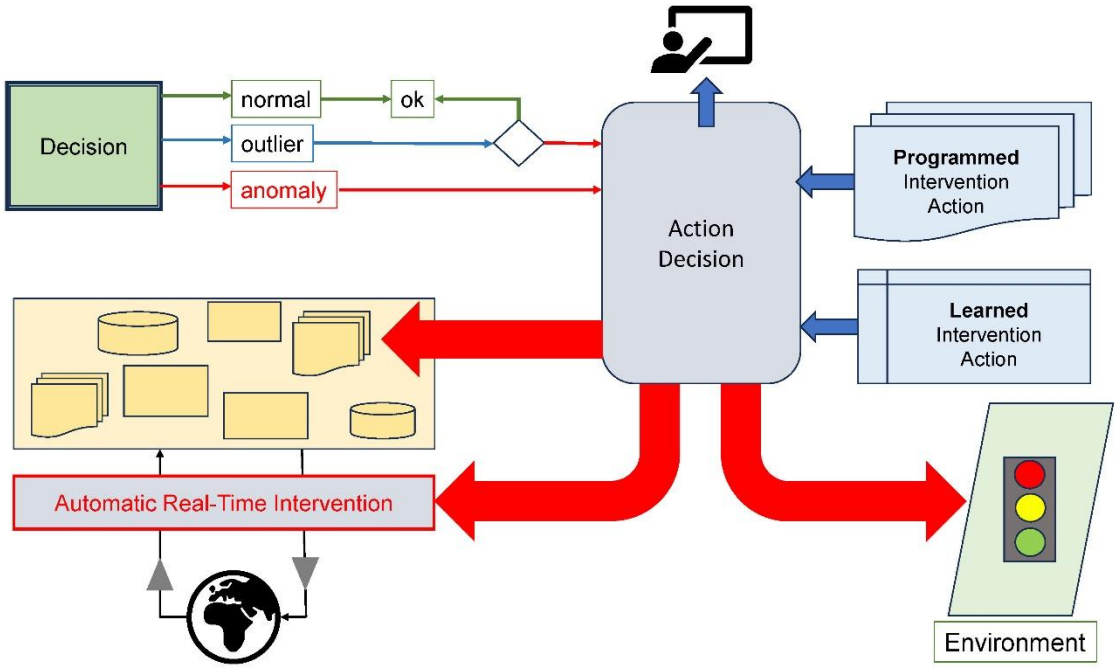

Figure 6. Autonomous Intervention Concept.

A possible autonomous intervention mechanism is shown in Figure 7. The training part is as described above, usually using real, simulated, or hybrid data. Here, more than one anomaly detection algorithm is used in specific applications, and a voting system decides, thus reducing false positives and false negatives (Bilgin, 2020).

 doi:10.20944/preprints202509.1380.v1

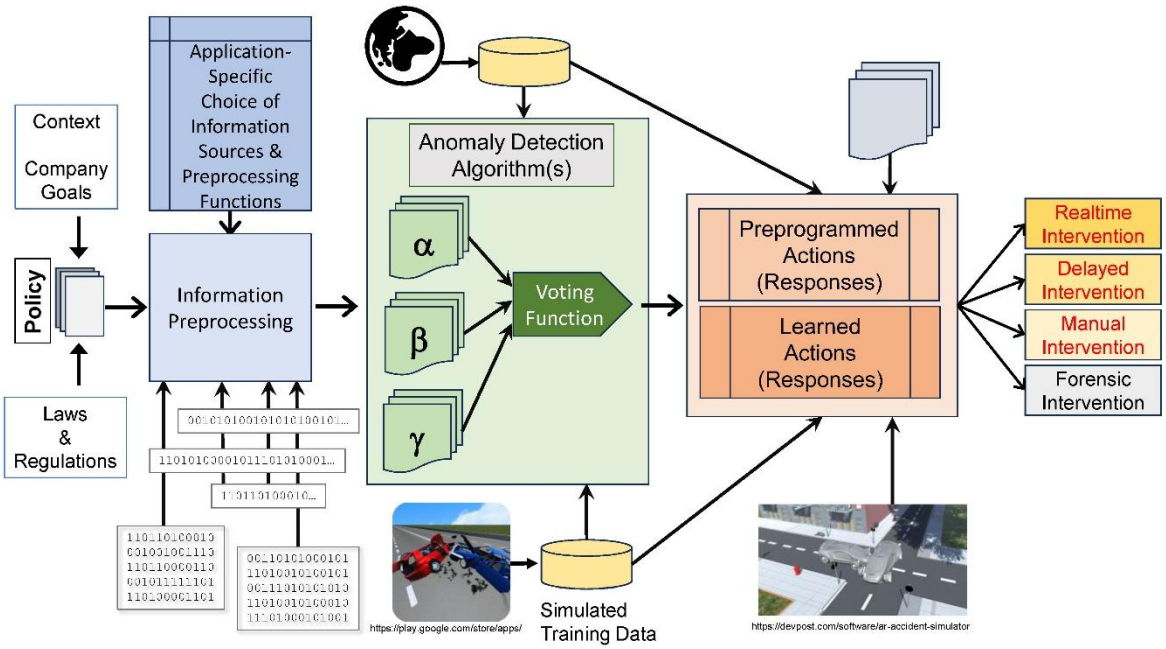

Figure 7. Autonomous Intervention Implementation.

Unfortunately, today's most significant obstacle to automatic, real-time intervention is the lack of binding, comprehensive, actionable laws, regulations, and liability provisions (e.g., Beckers et al., 2023).

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. The example chosen is autonomous collision avoidance of ships (Source: Hashimoto et al., 2021 / Shen et al., 2019). 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 (⇒ Protective shell) the mariners or guides the ship autonomously.

This example uses constrained Deep Reinforcement Learning as AI technology (RL, e.g., Bilgin 2020 / Francois-Lavet, 2018). Deep reinforcement learning AI is distinguished from supervised or unsupervised learning because it aims at long-term goals or policies (Winder, 2020). 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 (IMO, 1972 / Cockcroft 2012).

The setup of the example is shown in Figure 8. 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.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 September 2025 doi:10.20944/preprints202509.1380.v1

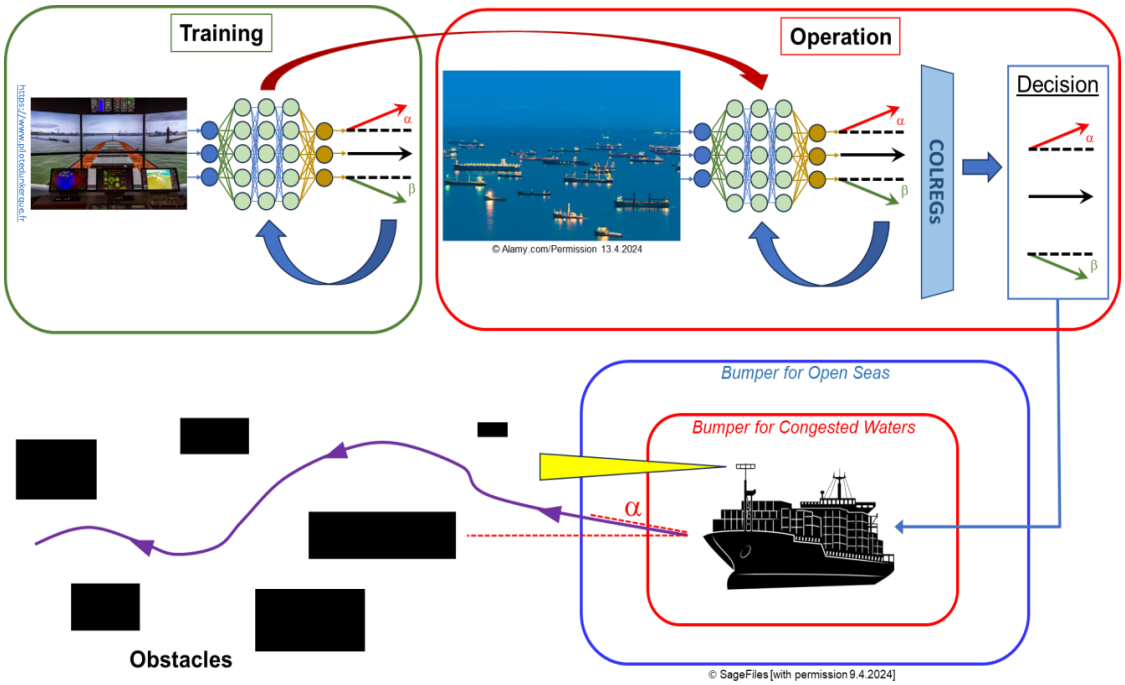

Figure 8. AI/ML Safety Example – Ship Collision Avoidance.

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 the collision avoidance capability.

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. The example is unsupervised learning for intrusion detection for the MIL-STD-1553 communication bus (Source: Hadeer et al., 2023 / Sachdev et al., 2023).

The MIL-STD-1553 communication bus (https://www.milstd1553.com/) 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, the MIL-STD-1553 is a critical component of countless military and civilian platforms, a protocol change is no option. Therefore, a robust defense in the form of intrusion detection is required!

Figure 9 shows the architecture of the example: It represents an intrusion detection system based on unsupervised machine learning using anomalies in the bus traffic. The MIL-STD-1553 follows a strict, deterministic message exchange protocol. This time-based regularity leads to a set of relevant features that generate the model for regular (“normal”) operation.

 doi:10.20944/preprints202509.1380.v1

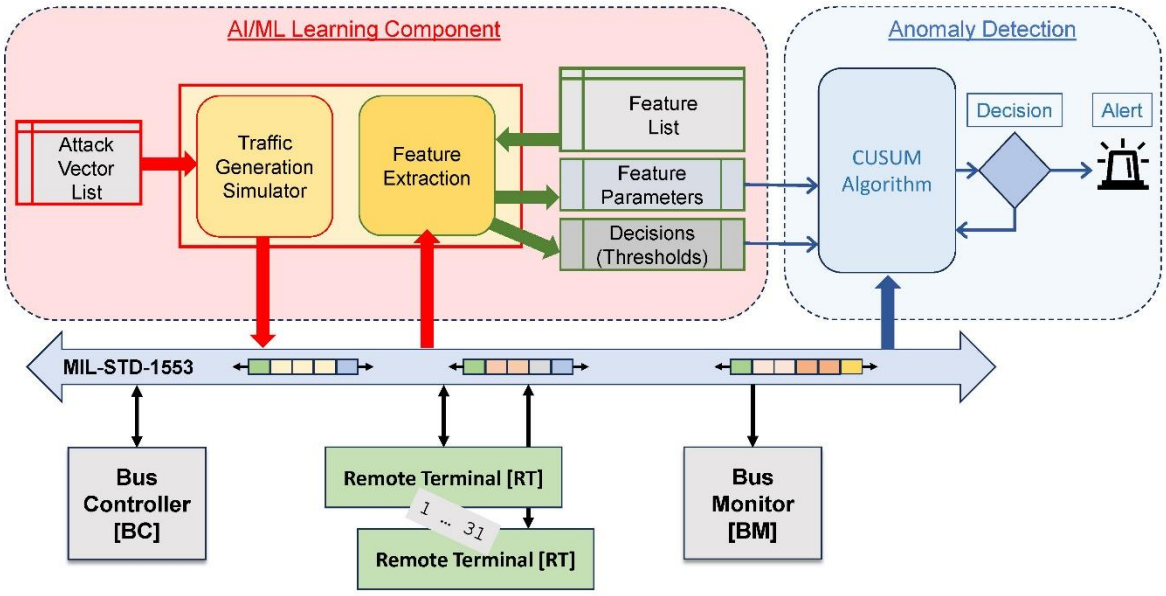

Figure 9. AI/ML Security Example - MIL-STD-1553.

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, which creates 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 the bus traffic and detects anomalies using the CUSUM algorithm (Cumulative Sum, e.g., Granjon, 2014 / Koshti, 2011). This statistical change detection algorithm (Page, 1954) 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.

Protective Shell

The idea of protecting a cyber-physical system against the misbehavior of the artificial/machine learning algorithm was presumably first presented by Lance Eliot (Eliot, 2016). He proposed it under the name of «AI Guardian Angel Bot» to improve the trust-worthiness of machine-learning systems. Here, the more technically-oriented (and less esoteric) term «Protective Shell» is used.

Figure 10 shows the protective shell (Eliot, 2016/Furrer, 2023). 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 contains additional hardware and software to detect, analyze, and react to anomalies, thus aiming to protect the system from safety accidents and security incidents during runtime.

 doi:10.20944/preprints202509.1380.v1

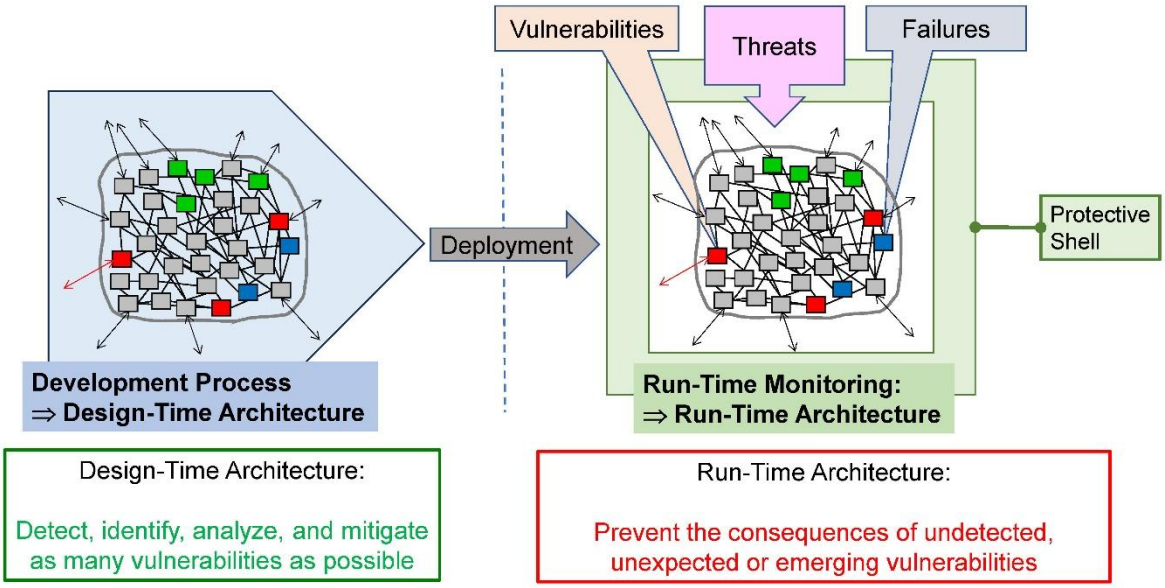

Figure 10. Protective Shell.

Extended Aflack Surface, XAI, and Robust AI

Implementing AI/ML-based runtime monitoring (i.e., a protective shell) introduces three 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 (e.g., Cybellium Ltd., 2023), i.e., generates new failure modes (Stamatis, 2019) and enables new attack vectors (Haber et al., 2018). Unfortunately, effective kill chains, i.e., specific attack methods and tools, are readily available to strike AI systems (e.g., Gibian, 2022). 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 solely use explainable artificial intelligence (XAI) mechanisms! (e.g., Simon, 2023 / Huang, 2023).

(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 (e.g., Guerraoui, 2024/Huang, 2023/Yampolskiy, 2024/ Hall et al., 2023).

Forensics and The Law

The task of the protective shell is not completed while guarding the safe and secure operation of the CPS. It must also protect the CPS against unjustified legal action and liability claims after a safety accident or a security incident. The protective shell – in addition to the already installed data logging activities of the CPS – must collect, store, and protect sufficient information (e.g., Winkle, 2022 / Marcella, 2021 / Kävrestad, 2020 / Oettinger, 2022) to:

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

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

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

 doi:10.20944/preprints202509.1380.v1

Open Challenges

Large-scale application of artificial intelligence/machine learning to cyber-physical systems' safety and security is a relatively young research discipline, although with its roots far back in computing history (e.g., VC3 Corporation, 2023). Therefore, open challenges exist, such as:

(1) 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;
(2) Dependable, transparent machine learning algorithms (Explainable AI, e.g., Mehta, 2022);
(3) Reliable, consistent, and fair training data (either from the real world or simulated), including their provisioning methods;
(4) Correct, timely, and effective intervention mechanisms, including their planning and decision mechanisms;
(5) Collection of sufficient, meaningful, and legally acceptable forensic data (e.g., Marcella, 2021);
(6) Metrics for the evaluation of the efficiency and effectiveness of AI/ML usage;
(7) Defense against attacks on the AI/ML algorithms, their implementations, and training data;

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 responsibility, liability, and ethical issues handled in autonomous AI/ML decision systems? This is not the place to discuss the emerging literature on AI ethics (e.g., Coeckelbergh, 2021/EU, 2021/Press, 2024), but it will significantly influence AI/ML research and application in the future.

Conclusions

The storyline of safety and security of cyber-physical systems starts with the "Impossibility Statement" (Figure 11, Koped, 2022). The consequence of the impossibility statement is that the deployed runtime cyber-physical system contains unmitigated vulnerabilities, which may become the source of 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, especially when using artificial intelligence/machine learning. The protective shell contains all the mechanisms for the runtime protection of the CPS and thus constitutes the last defense before a safety accident or a security incident occurs.

 doi:10.20944/preprints202509.1380.v1

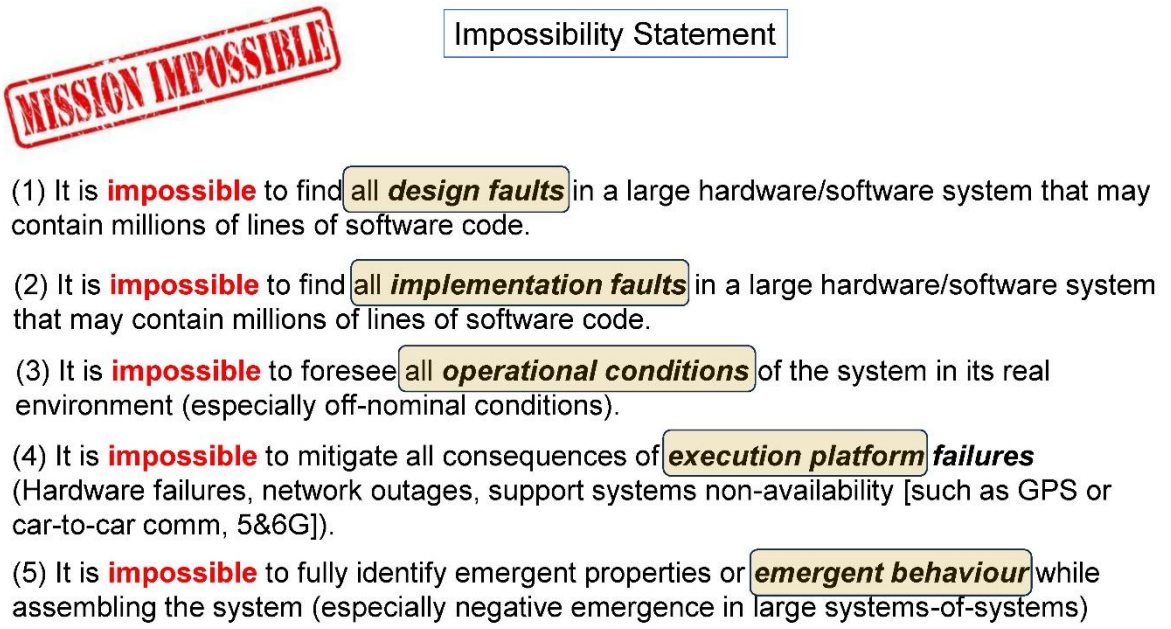

Figure 11. Impossibility Statement.

Author Contributions: The author is the only contributor to this paper.

Conflicts of Interest: The authors declare no conflict of interest.

Funding: This research received no specific grant from any funding agency, commercial or not-for-profit sectors.

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.

References

1. Adari S.K., Alla S. (2024): Beginning Anomaly Detection Using Python-Based Deep Learning - Implement Anomaly Detec- tion Applications with Keras and PyTorch. *Apress Media, New York, NY, USA*. ISBN 979-8-8688-0007-8. https://doi.org/10.1007/979-8-8688-0008-5
2. Alur Rajeev (2015): Principles of Cyber-Physical Systems. *MIT Press, Cambridge, MA, USA*. ISBN 978-0-262-02911-7
3. Axelrod C.W. (2012): Engineering Safe and Secure Software Systems. *Artech House Books, Norwood, MA, USA*. ISBN 978-1- 608-07472-3
4. Bahr N.J. (2017): System Safety Engineering and Risk Assessment: A Practical Approach. *CRC Press (Taylor & Francis Group), Boca Raton, FL, USA*. 2nd edition. ISBN 978-1-138-89336-8
5. Banish G. (2023): OBD-I & OBD-II - A Complete Guide to Diagnosis, Repair, and Emissions Compliance. *CarTech Books, North Branch, MN, USA*. ISBN 978-1-61325-752-4
6. Bartocci E., Falcone Y., Editors (2018): Lectures on Runtime Verification - Introductory and Advanced Topics. *Springer Na- ture, Cham, Switzerland*. ISBN 978-3-319-75631-8 (LNCS 10457)
7. Beckers A., Teubner G. (2023): Three Liability Regimes for Artificial Intelligence - Algorithmic Actants, Hybrids, Crowds. *HART Publishing, Oxford, UK*. ISBN 978-1-5099-4937-3

 doi:10.20944/preprints202509.1380.v1

8. Bhuyan H. (2018): Network Traffic Anomaly Detection and Prevention - Concepts, Techniques, and Tools. *Springer Interna- tional Publishing, Cham, Switzerland*. ISBN 978-3-319-87968-0
9. Bilgin E. (2020): Mastering Reinforcement Learning with Python - Build next-generation, self-learning models using rein- forcement learning techniques and best practices. *Packt Publishing, Birmingham, UK.* ISBN 978-1-838-64414-7
10. Burger W., Burge M., J. (2022): Digital Image Processing - An Algorithmic Introduction. *3rd edition, Springer Nature Switzerland, Cham, Switzerland*. ISBN 978-3-031-05743-4. https://doi.org/10.1007/978-3-031-05744-1
11. Buttfield-Addison P., Buttfield-Addison M., Nugent T., Manning J. (2022): Practical Simulations for Machine Learning- Using Synthetic Data for AI. *O'Reilly Media Inc., Sebastopol, CA, USA*. ISBN 978-1-492-08992-6
12. Cockcroft A.N, Lameijer J. N. F. (2012): A Guide to the Collision Avoidance Rules. *Butterworth-Heinemann, Kidlington, UK, 7th edition*
13. Coeckelbergh M., 2020: AI Ethics. *The MIT Press, Cambridge, MA, USA*. ISBN 978-0-262-53819-0
14. Cybellium Ltd., 2023: Mastering Attack Surface Management - A Comprehensive Guide To Learn Attack Surface Manage- ment. *Cybellium Ltd., Tel Aviv, Israel*. ISBN 979-8-8591-4008-4
15. Dunning T., Friedman E. (2014): Practical Machine Learning - A New Look at Anomaly Detection. *O'Reilly and Associates, Sebastopol, CA, USA*. ISBN 978-1-491-91160-0
16. Eliot L. (2016): AI Guardian Angel Bots for Deep AI Trustworthiness - Practical Advances in Artificial Intelligence. *LBE Press Publishing*. ISBN 978-0-6928-0061-4
17. EU, (2021): Regulation of the European Parliament and of the Council laying down harmonised rules on Artificial Intelli- gence (Artificial Intelligence Act). *European Commission, Brussels, Belgium, COM(2021) 206 final, 21.4.2021*. Downloadable from: https://eur-lex.europa.eu/resource.html?uri=cellar:e0649735-a372-11eb-9585-01aa75ed71a1.0001.02/DOC_1&for- mat=PDF
18. Fidgerald J., Morisset C., (2024): Can We Develop Holistic Approaches to Delivering Cyber-Physical Systems Security? *Cambridge University Press, Research Directions: Cyber-Physical Systems*. https://doi:10.1017/cbp.2024.1
19. Francois-Lavet V., Henderson P., Islam R., Bellemare M.G., Pineau J. (2018): An Introduction to Deep Reinforcement Learning. *Foundations and Trends in Machine Learning: Vol. 11, No. 3-4, 2018.* https://doi.org/10.1561/2200000071. Down- loadable from: https://arxiv.org/abs/1811.12560
20. Furrer F.J. (2019): Future-Proof Software-Systems – A Sustainable Evolution Strategy. *Springer Vieweg Verlag, Wiesbaden, Germany*. ISBN 978-3-658-19937-1
21. Furrer F.J. (2022): Safety and Security of Cyber-Physical Systems. *Springer Vieweg Verlag, Wiesbaden, Germany*. ISBN 978-3- 658-37181-4
22. Furrer, F.J. (2023). Safe and secure system architectures for cyber-physical systems. *Informatik Spektrum*, 46, 96–103. https://doi.org/10.1007/s00287-023-01533-z
23. Gibian D. (2022): Hacking Artificial Intelligence - A Leader's Guide from Deepfakes to Breaking Deep Learning. *Rowman & Littlefield, London, UK*. ISBN ISBN 978-1-5381-5508-0
24. Granjon P. (2014): The CUSUM algorithm. *GIPSA-lab, Grenoble Campus, Saint Martin d'Hères, France*. Downloadable from: https://hal.science/hal-00914697/document
25. Griffor Edward, Editor (2016): Handbook of System Safety and Security - Cyber Risk and Risk Management, Cyber Secu- rity, Threat Analysis, Functional Safety, Software Systems, and Cyber-Physical Systems. *Syngress (Elsevier), Amsterdam, Netherlands*. ISBN 978-0-128-03773-7
26. Guerraoui R., Gupta N., Pinot R. (2024): Robust Machine Learning - Distributed Methods for Safe AI. *Springer Nature Sin- gapore, Singapore*. ISBN 978-981-97-0687-7. https://doi.org/10.1007/978-981-97-0688-4.
27. Haber M.J., Hibbert B., 2018: Asset Attack Vectors - Building Effective Vulnerability Management Strategies to Protect Or- ganizations. *Apress Media LLC, New York, NY, USA*. ISBN 978-1-4842-3626-0. https://doi.org/10.1007/978-1-4842-3627-7
28. Hadeer A., Traore I., Quinan P., Ganame K., Boudar O. (2023): A Collection of Datasets for Intrusion Detection in MIL- STD-1553. {Chapter 4 in Traore et al., 2023}. https://doi.org/10.1007/978-3-031-16237-4_4
29. Hall P., Curtis J., Pandey P. (2023): Machine Learning for High-Risk Applications - Techniques for Responsible AI. *O'Reilly Media Inc., Sebastopol, CA, USA*. ISBN 978-1-098-10243-2.

30. Harrison L. (2020): How to use Runtime Monitoring for Automotive Functional Safety. *TechDesignForum White Paper*. Downloadable from: https://www.techdesignforums.com/practice/technique/how-to-use-runtime-monitoring-for-automo- tive-functional-safety
31. Hashimoto H., Nishimura H., Nishiyama H., Higuchi G. (2021): Development of AI-based Automatic Collision Avoidance System and Evaluation by Actual Ship Experiment. *ClassNK Technical Journal, No. 3, 2021. Chiba, Japan.* Downloadable from: https://www.classnk.or.jp/hp/pdf/research/rd/2021/03_e05.pdf
32. Haupt N.B, Liggesmeyer P. (2019): A Runtime Safety Monitoring Approach for Adaptable Autonomous Systems. In: Ro- manovsky A., Troubitsyna E., Gashi I., Schoitsch E., Bitsch F. (Editors): Computer Safety, Reliability, and Security. SAFECOMP 2019. Lecture Notes in Computer Science, Vol 11699. *Springer International Publishing, Cham, Switzerland*. Downloadable from: https://www.researchgate.net/publication/335557336_A_Runtime_Safety_Monitoring_Ap-proach_for_Adaptable_Autonomous_Systems/link/5dd2b7b0299bf1b74b4e14f7/download
33. Huang X., Jin G., Ruan W. (2023): Machine Learning Safety. Springer Nature Singapore Pte. Ltd., Singapore. ISBN 978- 981-19-6813-6. https://doi.org/10.1007/978-981-19-6814-3
34. IMO (1972): COLREG - Preventing Collisions at Sea. *IMO (The International Maritime Organization), London, UK.* https://www.imo.org/en/OurWork/Safety/Pages/Preventing-Collisions.aspx
35. Janicke H., Nicholson A., Webber S., Cau A. (2015): Run-Time-Monitoring for Industrial Control Systems. *Electronics 2015, 4(4), 995-1017*; https://doi.org/10.3390/electronics4040995 Downloadable from: https://www.mdpi.com/2079-9292/4/4/995
36. Johannesson P., Perjons E., 2021: An Introduction to Design Science. *Springer Nature, Cham, Switzerland*. ISBN 978-3-030- 78131-6. https://doi.org/10.1007/978-3-030-78132-3
37. Kane A. (2015): Runtime Monitoring for Safety-Critical Embedded Systems. Ph.D.Thesis, *Carnegie Mellon University, Pitts- burgh, PA, USA*. Downloadable from: https://users.ece.cmu.edu/~koopman/thesis/kane.pdf
38. Kävrestad J. (2020): Fundamentals of Digital Forensics - Theory, Methods, and Real-Life Applications. *Springer Nature Switzerland, Cham, Switzerland*. 2nd edition. ISBN 978-3-030-38953-6. https://doi.org/10.1007/978-3-030-38954-3.
39. Khan M.T., Serpanos D., Shrobe H. (2016): Sound and Complete Runtime Security Monitor for Application Software. *Pre- print, arXiv:1601.04263v1 [cs.CR]*. Downloadable from: https://www.researchgate.net/publica-tion/291229626_Sound_and_Complete_Runtime_Security_Monitor_for_Application_Software
40. Knight J. (2012): Fundamentals of Dependable Computing for Software Engineers. *CRC Press (Taylor & Francis), Boca Raton, FL, USA*. ISBN 978-1-439-86255-1
41. Koped H. (2022): An Architecture for Safe Driving Automation. *In: Raskin, JF., Chatterjee, K., Doyen, L., Majumdar, R. (Edi- tors): Principles of Systems Design. Lecture Notes in Computer Science, Volume 13660. Springer Nature, Cham, Switzerland*. ISBN 978-3-031-22336-5. https://doi.org/10.1007/978-3-031-22337-2_4
42. Koshti V.V. (2011): Cumulative Sum Control Chart. *International Journal of Physics and Mathematical Sciences. 2011, Vol. 1, October-December, pp.28-32. ISSN: 2277-2111*. Downloadable from: https://www.researchgate.net/publica-tion/230888065_Cumulative_sum_control_chart
43. Kuo C. (2022): Modern Time Series Anomaly Detection - With Python & R Code Examples. *Packt Publishing, Birmingham, UK*. ISBN 979-8-363-29575-1
44. Marcella A.J., Editor (2021): Cyber Forensics - Examining Emerging and Hybrid Technologies. *CRC Press, Boca Raton, FL, USA*. ISBN 978-0-367-52424-1
45. Mehrotra K.G., Mohan C.K., Huang H. (2019): Anomaly Detection Principles and Algorithms (Terrorism, Security, and Computation). *Springer International Publishing, Cham, Switzerland*. ISBN 978-3-319-88445-5
46. Mehta M., Palade V., Chatterjee I. (2022): Explainable AI - Foundations, Methodologies and Applications. *Springer Nature, Cham, Switzerland*. ISBN 978-3-031-12806-6. https://doi.org/10.1007/978-3-031-12807-3
47. Möller Dietmar (2016): Guide to Computing Fundamentals in Cyber-Physical Systems - Concepts, Design Methods, and Applications. Springer International Publishing, Cham, Swiderland. ISBN 978-3-319-79747-2

48. Nassif J., Tekli J., Kamradt M. (2024): Synthetic Data - Revolutionizing the Industrial Metaverse. *Springer Nature, Cham, Switzerland*. ISBN 978-3-031-47559-7
49. Nikolenko S.I. (2021): Synthetic Data for Deep Learning. *Springer Nature Switzerland, Cham, Switzerland*. ISBN 978-3 030 75177 7
50. Oettinger W. (2022): Learn Computer Forensics - Your one-stop guide to searching, analyzing, acquiring, and securing digital evidence. *2nd edition. Packt Publishing, Birmingham, UK*. ISBN 978-1-803-23830-2
51. Page E.S. (1954): Continuous Inspection Schemes. *Biometrika, Vol. 41, No. 1-2, June 1954, pp. 100-115*. https://doi.org/10.2307/2333009. Available at: https://security.cybellum.com/the-state-of-automotive-security-2023
52. Papow, J. (2010): Glitch - The Hidden Impact of Faulty Software. *Prentice Hall, Upper Saddle River, NJ, USA*. ISBN 978-0- 132-16063-6.
53. Press L. (2024): EU Artificial Intelligence Act - The Essential Reference. *Lex Press, Independently published*. ISBN 979-8-8827- 1805-2
54. Rainey L.B., Jamshidi M., Editors (2018): Engineering Emergence - A Modeling and Simulation Approach. *CRC Press, Boca Raton, FL, USA*. ISBN 978-1-138-04616-0
55. Sachdev K., Saad H.S., Traore I., Ganame K., Boudar O. (2023): Unsupervised Anomaly Detection for MIL-STD-1553 Avi- onic Platforms using CUSUM. {Chapter 5 in Traore et al., 2023}. https://doi.org/10.1007/978-3-031-16237-4_5
56. Sarkis (2023): Training Data for Machine Learning - Human Supervision from Annotation to Data. *O'Reilly and Associates, Sebastopol, CA, USA*. ISBN 978-1-492-09452-4
57. Shen H., Hashimoto H., Matsuda A., Taniguchi Y., Terada D., Guo C. (2019): Automatic collision avoidance of multiple ships based on deep Q-learning. *Applied Ocean Research, Volume 86, May 2019, Pages 268-288*. https://doi.org/10.1016/j.apor.2019.02.020
58. Simon C. (2023): Deep Learning and XAI Techniques for Anomaly Detection - Integrate the theory and practice of deep anomaly explainability. *Packt Publishing, Birmingham, UK*. ISBN 978-1-804-61775-5
59. Stamatis D.H., 2019: Risk Management Using Failure Mode And Effect Analysis (FMEA). *Quality Press, Milwaukee, MI, USA*. ISBN 978-0-8738-9978-9
60. Stewart A., J. (2021): A Vulnerable System - The History of Information Security in the Computer Age. *Cornell University Press, Ithaca, USA*. ISBN 978-1-501-75894-2.
61. Traore I., Woungang I., Saad S., Editors (2023): Artificial Intelligence for Cyber-Physical Systems Hardening. *Springer Na- ture, Cham, Switzerland*. ISBN 978-3-031-16239-8. https://doi.org/10.1007/978-3-031-16237-4
62. Vaseghi S., V. (2000): Advanced Digital Signal Processing and Noise Reduction. *2nd edition, John Wiley & Sons, Chichester, UK*. ISBN 978-0-4716-2692-3
63. VC3 Corporation (2023): The Evolution of Artificial Intelligence in Cybersecurity. *VC3 Corporation, Columbia, SC, USA*. Downloadable from: https://www.vc3.com/blog/the-evolution-of-artificial-intelligence-in-cybersecurity
64. Winder P. (2020): Reinforcement Learning - Industrial Applications of Intelligent Agents. *O'Reilly Media, Sebastopol, CA, USA*. ISBN 978-1-098-11483-1
65. Winkle T. (2022): Product Development within Artificial Intelligence, Ethics, and Legal Risk - Exemplary for Safe Autono- mous Vehicles. *Springer Fachmedien Vieweg, Wiesbaden, Germany*. ISBN 978-3-658-34292-0. https://doi.org/10.1007/978-3-658- 34293-7
66. Yampolskiy R.V. (2024): AI - Unexplainable, Unpredictable, Uncontrollable. *Chapman & Hall/CRC, Boca Raton, CA, USA*. ISBN 978-1-032-57626-8. https://doi.org/10.1201/9781003440260