This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

Intent-Based Security for Functional Safety in cyber-physical Systems

Emrah Tomur, Zeki Bilgin, Utku Gülen, Elif Ustundag Soykan, Leyli Karaçay and Ferhat Karakoç

Abstract—We propose a novel intent-based method to prevent security attacks on the safety functions in cyber-physical systems used in smart manufacturing. Context information about a cyber-physical system is collected from various sensors including non-safety sensors measuring device temperature, motor rotation speed, or instantaneous power consumption of machines. Such contextual information along with operational and business intent of the system under consideration are then used to check whether the current situation is indeed an emergency situation or a normal situation. Unlike the conventional safety systems that only rely on raw sensor data and safety protocol status packets from safety sensors, which might be spoofed and/or modified, decision on the safety situation in our method is intelligently made by comparing aggregated sensor information from the cyber-physical system and its environment for compliance with pre-configured operational intents that define the normal safe and secure operation of the system. We also show how to integrate Machine Learning (ML) and Artificial Intelligence (AI) into the proposed method for efficient and automated analysis of both intents and aggregated context information to make more intelligent decisions in execution of functional safety protocols. Our proposed AI/ML integration approach also enables the prediction of safety critical situations before they occur. The proposed method aims to prevent unnecessary switching to fail-safe mode causing insecure system states such as emergency doors opening or system halting in normal situations. In addition, it prevents sticking in non-safe states and not switching to fail-safe mode in real emergencies which could cause hazard on device and/or people.

Index Terms—functional safety, intent-based security, cyber-physical systems, smart manufacturing, industrial automation, industrial internet of things, machine learning

1 Introduction

Deployment of industrial automation systems in various verticals ranging from manufacturing to mining, automotive to smart grids has been increasing in the last decade, particularly with proliferation of the Industry 4.0. Modern communication technologies allow merging of the IT (Information Technology) network with the OT (Operational Technology) network, which had been mostly isolated from external access. This change has made attacks on industrial systems easier since they can be remotely initiated by malicious parties. These attacks could cause harm not only on the equipment in the automation network but also on people in the environment.

Such security attacks could be more devastating for society in the near future due to foreseen seamless connectivity of critical infrastructures with proliferating mobile wireless communication technologies. One important point related to security within the scope of industrial automation is different prioritization in preventing unwanted consequences impacting the valuable assets. In the IT domain, security is mostly concerned with protection of the information and related systems (cyber assets). On the other hand, one of the most important requirements in automation systems is protecting the well-being of people, equipment and environment (physical assets) from any potential harmful situation, which is safety. In other words, safety is concerned with reducing risk of danger to people and machinery caused by accidents or errors whereas security is mainly concerned with protecting confidentiality, integrity, and availability of information systems and data from malicious intentional misuse [1]. Systems having both cyber and physical parts are called cyber-physical systems. Since such systems are used in various industrial applications, it is vital to provide security and safety simultaneously, managing these two in a balanced way despite potentially conflicting requirements.

Security violations originating from intentional acts of malicious parties might affect the safety of the industrial automation system by fooling the system into thinking that there is no emergency and therefore safety function does not need to be triggered. This malfunction, in turn, could cause harm to people and equipment. In addition, such attacks could also deteriorate availability by causing frequent and/or long interruptions to the operation of the automation system. This can be due to intentional malicious acts of someone tampering with the safety function causing it to switch to fail-safe operation when there is no real emergency or danger.

Current functional safety systems operate as simple rule-based control systems based on conditional statements, which fully rely on raw sensor readings (if temperature is above threshold, if a sensor's value is in a specific range, if light intensity is below a threshold, etc.) to trigger safety functions (stop the system, open the doors, cut power). However, the main objective should ideally be to trigger safety function if there is a real emergency with certain potential of causing hazard to environment, people, or equipment.

Current safety systems are not designed to deal with intelligent malicious parties; therefore they rely on overly simplistic approaches, which can be fooled by capable at-

- *All authors except for the second are with Ericsson Research, Istanbul. E-mail: emrah.tomur@ericsson.com*
- *Second Author is with Arcelik.*

Manuscript received January xx, 20xx; revised January xx, 20xx.

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

tackers. Thus, intelligence should be added to safety systems to protect them from malicious parties. We embed intelligence into our proposed solution by jointly utilizing context information and operational intent. Context provides a rich information content about the safety situation, and intent is the true decision parameter for checking a current situation's conformity with normal operation. Use of intent as the main decision baseline in a safety program to defeat malicious actors is a novel approach for providing security for safety functions of cyber-physical systems.

In this paper, we propose an intelligent, context-aware, intent-based solution to detect and prevent security attacks targeting critical safety functions in cyber-physical systems. Our solution scope is limited to the safety software running at the application layer, and we do not include here potential control-theoretical solutions utilizing physical dynamics of the cyber-physical system. Still, it should be noted that control theory can be utilized to tackle the safety aspects of the problem that we address. An exemplary proposal on how we can adapt a control-based approach in our proposed solution is given in the following paragraph.

As widely employed in many industrial applications, we can assume that a DC motor is used for controlling rotation of the robotic arm in our use case. A DC motor works with electric power i.e., voltage and current — such that inverting their polarity results in changing the rotation direction, while varying their strengths allows us to adjust the motor's rotation speed. In particular, the speed of the motor is proportional to the voltage applied at the input terminals of the motor, which means the more voltage we provide at the input, the faster the motor will rotate at the output, and vice versa. Thus, it is possible to slow down or completely stop the motor rotation by adjusting the input voltage. This can be achieved in an effective and automatic manner by integrating our sensor signals or safety signal with PWM (Pulse Width Modulation) that controls the motor speed. Specifically, the output of the safety signal can set the duty cycle, which is the proportion of 'ON' time to the regular interval or period of time, in PWM that adjusts the average power delivered to the motor and thus the rotation speed. Moreover, for a rapid stop in an emergent safety situation, an electrical braking mechanism in the motor can be activated based on our sensor signals.

Control-theoretical solutions take a model-driven approach to provide guarantees with respect to known operating conditions. On the other hand, machine-learning based approach, as we propose in this paper, takes the data-driven approach to make safety decisions more adaptable to new contexts. This comes with the expense of providing formal guarantees which is possible with control theory. Using a hybrid approach where both model-driven and data-driven approaches have the potential to benefit from the positive aspects of both worlds and it is a promising research direction, as explained in [32]. The contributions of this paper are listed as:

- We improve the current operation of functional safety protocols that only rely on raw sensor data transmitted from safety sensors to safety controllers. This is achieved by utilizing a collective set of sensor data received from all relevant sensors to make intelligent evaluations and decisions on the current safety situation. We derive aggregated context information from data collected by all relevant sensors on and around the cyber-physical system. This aggregated context information is used to check whether there is indeed an actual emergency or if the current situation is normal and aligned with the system's operational and business intents.
- We propose a complete method describing how to utilize intent mechanisms on securing safety functions for various cyber-physical systems including smart manufacturing. Although there are some works proposing the use of Machine Learning to enhance safety in cyber-physical systems [2], to the best of our knowledge, within the scope of using intents for securing the safety function, this is the first time such solution has been described. We improve the decision baseline by integrating operational intents. We compare intent-based policies to the aggregated contextual input derived from multiple sensors. We check to see whether the current situation falls under an operational intent defined during initial configuration.
- We demonstrate both a rule-based and ML-based approach to implement our solution using well-known industrial and ML development tools. We verify our implemented solution against default configurations using under-attack and no-attack scenarios.
- Existing solutions to this problem propose to resolve security-safety conflicts during risk analysis, i.e., in design time. Yet, such an approach is limited by the ability of risk assessors to identify all possible conflicts of security-safety. This is practically impossible due to unknown unknowns phenomenon as well as the infeasible cost of taking precautions for very low probability but very high impact conflicts. Our intent-based method provides a run-time solution to prevent situations that are unimaginable in design time.

The rest of the paper is organized as follows. In Section II, we provide a background for the concepts of functional safety, security for safety and intent-based technologies. In Section III, related work is presented and then in Section IV, we provide the details of our proposed solution. We give implementation details and results in Section V and finally in Section VI, we present the conclusions.

2 Background

2.1 Functional Safety

Safety in industrial automation systems is regulated by different standards than those that regulate security. One such standard is IEC 61508 where safety is defined as freedom from unacceptable risk of physical injury or harm to human health, either directly, or indirectly. Functional safety is part of the overall safety and depends on a system or equipment operating correctly in response to its inputs. In other words, functional safety is correct working (functioning) of control systems and communication systems to maintain safety. Any system that carries out safety functions is called a

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

safety-related system (SRS). An SRS may be separate from the control system or the control system may itself carry out safety functions. Functionally, SRS is intended to detect a potentially hazardous state of an operation and place the system into a "safe state" before it reaches that hazardous state. Using simple logic operations with inputs received from safety sensors, a safety function is implemented in software to prevent harmful effects which can be caused by Equipment Under Control (EUC). EUC can be any industrial system such as a robot, machine, wind turbine or automated guided vehicle controlled by automation with a controller, often called Programmable Logic Controller (PLC).

An SRS is generally composed of the following components: (1) Safety Sensor(s), attached to the EUC or standalone, (2) Safety Actuator, attached to the EUC or standalone, (3) Safety Application/Program, which is a simple computer application running on a safety controller either on a dedicated hardware (e.g., safety PLC), or on the same hardware controlling the EUC (main PLC approved for failsafe operation). It is also possible to control both the EUC and the safety function using software controllers which run on industry-grade computers (e.g. industrial PC). Software controllers allow integration of high-level programming languages into the automation function.

Communication networks play a very important role to transfer information from safety sensors to safety controllers, and from safety controllers to safety actuators with a high level of integrity and availability required in these transmissions. Communication networks in most automation systems rely on industrial Ethernet protocols such as Profinet, and dedicated protocols are used to transmit safety data over these communication channels [3]. One of the most commonly used safety protocols is Profisafe [4]. As shown in Figure 1, Profisafe is an application layer protocol operating at OSI layer 7 and can detect systematic and stochastic failures resulting from node and network faults leading to potential hazard. Profisafe takes many measures (Cyclic Redundancy Check (CRC), timestamp, sequence number, sender/receiver name) to deterministically discover all possible faults and hazards infiltrated by the black channel and to keep error probability under a certain limit. Black channel assumption allows underlying communication channel to use any wired or wireless connectivity technology without requiring that devices involved are certified for certain availability or error rates. Profisafe can be used in almost all wired and wireless industrial communication protocols and access technologies including Profinet, Profibus, Industrial Ethernet, Wi-Fi, Bluetooth and 5G. Also, Profisafe is completely independent of the base transmission channel, and works equally well over copper wires, fiber optics, wireless communication, links, or backplanes since it does not make any assumptions regarding transmission rates or error detection mechanisms. Safety function response time (SFRT) is an important metric to be calculated for the Profisafe protocol, to determine how fast the time-critical safety applications respond to a hazardous situation and safe state can be achieved again according to IEC 61784-3-3 standard [5]. SFRT is calculated from the total worst case delay times of each SRS component including the period time of the transmission bus (black channel) or buses if more than one transmission channels is used. The worst case delay of an SRS component is related with its failsafe watchdog timer period, which is defined by equation 1 in IEC 61784-3-3, where T_{BUS} is period time of the transmission bus, D_{Ack} and H_{Ack} are device and host acknowledgement time respectively.

$$Fs_WD_Time = 2.T_{BUS} + D_{Ack} + H_{Ack} \quad (1)$$

As a realistic example which has been calculated from an automation test bed in the study of Akerberg et al. [6], the failsafe watchdog timer period is estimated as in equation 1, where the two involved transmission buses are namely Profinet Industrial Ethernet protocol period time (Tcy_{PNIO}, set as 128 ms) and WirelessHART (Tcy_{WH}, set as 3400 ms), DAT and HAT which are device (sensor) and host (ABB 800xA) acknowledge time (both are set as 6ms), and $WCDT_{GW}$ which is worst case delay time gateway (set as 50ms).

$$Fs_WD_Time = 2.Tcy_{PNIO} + 2.Tcy_{WH} + DAT + HAT + WCDT_{GW} \quad (2)$$

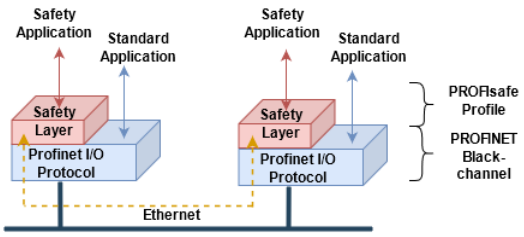

Fig. 1: Profisafe running at application layer

As depicted in Figure 2, there is a continuous cyclic communication between safety controller and safety sensor/actuator. In normal operation of a safety-related system, the safety program on the Safety Controller periodically polls Safety Sensors, which in turn send their measurements about safety-related ambient conditions such as temperature or pressure. Measurement values are compared to pre-set threshold values in the Safety Program. If the received sensor measurement falls outside of the range dictated by the threshold values, a potentially dangerous situation is detected, and the Safety Program sends the control command to the Safety Actuator to perform an action to mitigate damage to people, equipment or environment. Thus, this whole critical process is composed of: (i) sharing ambient condition information measured by sensors, (ii) assessing potential dangerous situations by the safety controller and (iii) commanding safety action to safety actuators. If the process is not performed correctly and timely by the overall safety function, unwanted consequences can occur.

2.2 Intent-Based Technologies

Intent-based technologies is a paradigm shift allowing users and operators to manage the operations of the system by stating at a high-level "what to do," rather than specifying the low-level details of "how to do" [7]. Intents can be defined as high-level policies in the form of a statement of targets or requirements that a system should meet (what to

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

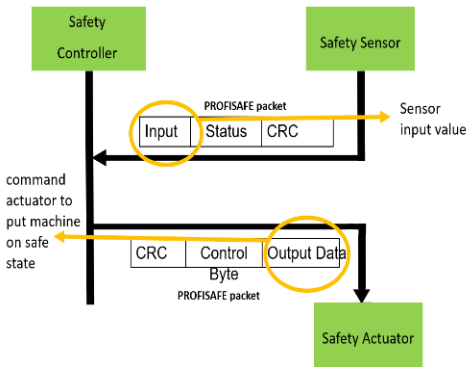

Fig. 2: Safety communication

do), without prescribing how to reach them (how to do). Intent-based approaches can be embraced by different parts of a system or in an end-to-end fashion and at different phases of the system e.g., only at operational phase or for the full life-cycle. This is a result of their abstracted, declarative and vendor agnostic features.

In some cases, "intents" may be confused with "policies". In fact, intents can be interpreted as particular type of policies to govern choices in the behavior of a system. On the other hand, the term "policy" has a broader meaning, and is used to define actions, conditions and declarations of events.

If one considers the case for a device operation, then an example policy would be: *When an operator comes in the vicinity of the safety sensor closer than one meter, stop the operation of the device*. And, an intent expressing the same thing would be: *Harmful operation is not permitted*. Intent-based technologies are being advertised in the context of intent-based networking, management, and in the context of AI/ML.

Policies are mostly used in conjunction with rules and rule-based decisions. On the other hand, enforcement for intents are not fixed, but rather they are flexible. This flexibility gives the freedom of choice between rule-based or AI/ML based decisions for the intents as we employ in this study.

2.3 Security attacks on safety systems

Today's automation networks are susceptible to interception by malicious parties. Most safety protocols do not impose any embedded cryptographic security measures to protect either the authenticity or the integrity of packets while collecting status information from safety sensors. Therefore, attackers can modify the packets sent by safety sensors to fool the safety controller into assessing the safety situation in a false way as follows:

- In case of a real emergency situation (emergency button pressed or sensor measurements indicate a potentially hazardous situation such as rotor speed is unexpectedly high or worker is closer than the safe distance), the message sent by the safety sensor might be modified maliciously to indicate that there is no emergency. This will endanger the worker and/or machinery and/or environment.
- In case of no emergency situation, a safety sensor's message might be maliciously modified on-purpose to make it flagged as if there were a true emergency (emergency button pressed, or sensor measurement is above threshold). Then, the safety controller will force the system into an emergency shut-down or slowing-down in normal situation where it is not needed causing decrease in availability.
- An attacker could insert new unauthorized safety packets in the network and/or alter messages sent by authorized parties without any concern to satisfy naive integrity check mechanisms like CRC after these modifications. The reason is that safety protocols activate fail-safe mode whenever any packet error is detected, which force the system into emergency stop, causing decreased availability in case of no emergency situation.

Other than the attacks above, attackers can exploit the correlation between safety and security which may create conflicting requirements like opening of emergency doors, which is good for safety but results in decreased security. Such cases can be created by fooling the system into thinking that a safety-critical situation exists while a sensor can be fooled to send a false sensor value according to raw sensor measurement values despite there is no safety-critical situation. Existing message integrity checking methods, either CRC or cyrpto-based, cannot prevent such attacks since the packet from safety sensor to safety controller is not changed but, rather, the safety sensor is fooled.

3 Related Work

An overview of different approaches for how safety and security concerns are jointly considered for industrial infrastructures is provided by Kriaa et al. [8]. They classified the approaches as either generic or model-based methods by identifying possible dependencies of safety and security which ensures efficient risk management.

Johan et al. explained in [9] how an attack on Profisafe can be performed. They implemented this attack on a test network and showed that safety-related process data carried by Profisafe could be changed without any of the safety measures in the protocol detecting this attack. These types of attacks on safety protocols are not difficult to perform from security perspective since these protocols only rely on CRC without using cryptographic means. Thus, modification of the data and re-computing of the CRC by any attacker is straightforward and cannot be detected.

A simple solution to the problem mentioned above would be embedding cryptographic primitives into the integrity check using MAC/MIC (Message Authentication Code/Message Integrity Code). Akerberg et al. [6] proposed the "security module" concept as an additional security layer on top of the communication layer and below the safety layer where Profisafe operates. In this way, it would be possible to ensure integrity and authenticity of safety-related packets.

However, implementing crypto-based protocols in industrial automation systems is difficult due to specific characteristics of the underlying infrastructure with lots of non-IT devices. It puts burden on formation and transmission of

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

packets with additional computations and overhead bytes for security which creates negative effect on timing performance. Additionally, implementing the solution in Profisafe with off-the-shelf hardware and software components is not realistic.

The conflicting relationship between safety and security is a drawback which can be used by malicious parties to fool the safety controller. As an example, a European luxury car manufacturer noticed that one of its models was a target for theft comparably more often than other car models [10]. The reason for this was understood when an apprehended thief confessed that this model of car easily opened even when its doors were locked. If someone jumped on the roof of the car, doors would unlock automatically. The designers of the car included this as a safety measure where the doors of the car would unlock if the car was involved in an accident and rolled over. To check for roll-over situations, they only verified whether enough pressure/weight was being applied on the roof via pressure sensors. Hence, the process of increasing the safety of the system actually decreased the security of the same system. In fact, there are other examples of such cases [11] where conflicting requirements of safety and security result in weaknesses associated with one of two. The existing solutions propose to resolve security-safety conflicts during risk analysis, i.e., in design time. In [12], the authors introduced a common approach where safety and security are investigated together throughout the life cycle of the whole system. The approach uses IEC 61508 [13] and IEC 15408 [14] standards, where the first one is used for creating safety related systems and the second one is used to evaluate a system's security properties. Taken together, these two standards present a complementary framework to resolve conflicts between safety and security at the requirement and the functional level.

There are studies addressing new forms of attacks like product tampering and service interruption targeting manufacturing systems. In [15], the authors provided a survey on existing ML-based solutions for fault detection, prediction, and prevention in Industry 4.0. In [16], they proposed a novel threat intelligence technique consisting of two main components for detecting cyber threats in industrial systems. One module handles the huge amount of data sources consisting of sensors, actuators and network nodes. The other module monitors and detects abnormal activities based on Beta Mixture Model and Hidden Markov Mechanism. They evaluated their scheme on a combination of two well-known data sets (CPS data set [17] and UNSW-NB15 data set [18]) where the results show that their scheme outperformed other anomaly detection methods. However, the requirement of a large number of data samples for parameter estimation is one drawback of the approach of the study. In [19], an intrusion detection system based on context-awareness and DNN (Deep Neural Network) was presented. The anomaly signs can be effectively detected by this method and the risk of cyber-attacks can be decreased.

Intent-based technologies are mostly used for Quality-of-Service (QoS) and performance management of networks. Although there are some works proposing their use in conventional computer and mobile networks, there is hardly any studies proposing how intent-based approaches can be used in industrial automation systems which are composed of cyber-physical components. The authors in [20] proposed an end-to-end model for intent recognition from voice in a smart home. Typically, automatic speech recognition (ASR) is used in voice based intent recognition in which the intents are classified from the transcriptions. Since the intent classifier might be impacted by the errors at the ASR stage, the proposed end-to-end model classifies intents directly from raw speech data as an input, thus preventing error propagation.

4 Proposed Solution

In this paper, we propose a method which utilizes operational intents to ensure security of the safety function in cyber-physical systems including smart manufacturing. As an alternative to current operation of functional safety protocols like Profisafe, which relies only on raw sensor data transmitted in safety protocol messages from safety sensors to safety controllers, we propose the use of a collective set of sensor data received from all relevant sensors. Data is gathered not only from safety sensors but also other sensors in the system as well as in the environment to make intelligent decisions on the current safety situation. Our method checks the combined context information derived from data collected by all relevant sensors on and around the cyber-physical system. This approach determines whether the current situation falls under the operational intent defined during initial configuration, which emphasizes safe and secure operation of the individual components as well as the overall system. In this way, aggregated context information of the individual or overall cyber-physical systems is used to check if there is indeed an actual emergency or if the situation is normal, aligned with the system's operational and business intents. This intelligent control, based on comparison of current context with the programmed intents, is enhanced by use of ML/AI to increase the effectiveness and efficiency of the decision process as later explained in the subsection titled AI/ML Integration.

The main steps followed in the proposed method during a potential safety event detection are depicted in Figure 3 and explained below.

1) The value measured by the safety sensor is updated with a new measurement caused by a potential event.

2) The Profisafe Safety Program sends a Profisafe protocol message to poll the sensor input (Polling is periodically performed according to previously configured intervals).

3) The safety sensor replies to the Profisafe Safety Program with a message containing the new measurement.

4) Instead of directly making a decision based on the received raw sensor measurement as done in conventional methods, the Safety Program shares this measurement value with the Safety Decision Program.

5.1) Triggered by receiving the raw sensor measurement, the Safety Decision Program fetches the most recent value of the combined context information from the Context server.

5.2) Similar to the step above, the Safety Decision Program fetches the current value of the intent-driven safety policies from the Intent server.

5.3) The Safety Decision Program makes an intelligent decision on the current safety-related situation by comparing the combined context information against intent-driven

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

policies either in a rule-based fashion or automatically by ML component. Details of both approaches are explained in detail in the ML/AI Integration part.

6) If any real safety-critical and/or emergency is detected, the Safety Decision Program notifies the Profisafe Safety Program. If there is no emergency detected, nothing is sent to the Profisafe Safety Program.

7) If the Profisafe Safety Program receives from Safety Decision Program a message indicating an emergency, it sends a Profisafe message to the Safety Actuator with content describing the programmed action in the Safety Program.

8) The Safety Actuator performs the safety action commanded in the Profisafe message.

4.1 Solution Architecture

In principle, our method can work with any safety protocol. Below, we describe an implementation with Profisafe, which is one of the most common one compatible with almost all industrial communication technologies. Figure 4 depicts how this proposed method can be implemented in a case where a cyber-physical system, e.g. a robot arm, is controlled with a standard PLC and it is safety-protected with a Profisafe Safety Program running on a Safety PLC. Below, we provide a layer-by-layer explanation of the proposed solution architecture.

4.1.1 Enforcement Layer

As in any automation deployment with a safety function, there are hardware such as Safety Sensors and Safety Actuators at the bottom level of our proposed architecture. We name this level as the enforcement layer since this is where the system directly interacts with the physical phenomenon through sensors to monitor the safety situation and actuators to enforce the safety action decided. These sensors and actuators might be either embedded on the cyber-physical system or separately installed as depicted in the figure. There is also a manual Emergency Button connected to the safety controller via a Remote Safety IO device.

4.1.2 Decision Layer

Besides the components mentioned above, which exist in conventional safety function implementations, our method includes some additional elements used for making intelligent safety decisions based on intent and context. One of them is "Safety Decision Program", which is a computer program working in coordination with the Profisafe Safety Program to make the decision for whether any emergency exists based on the two main input parameters. Both of these programs are located in the decision layer where safety-related actions to be applied by the enforcement layer are made.

4.1.3 Configuration Layer

The first input to be sent to the Safety Decision Program from the configuration layer is the safety and security related intents coming from the "Intent Server". This input includes the intent-based policy statements translated from the normal operational intent of the cyber-physical system as defined by the person(s) in charge of this system, for instance the factory manager in a manufacturing use-case. The second main input to Safety Decision Program comes from the "Context Server", which is the data collection and aggregation point of all device-related and ambient sensors where a combined context information is produced by aggregation of all relevant incoming raw sensor data. Using this aggregated context information has an outstanding advantage. Current emergency status information for the cyber-physical system produced by multiple sensor data would have higher entropy than a single safety sensor data. This means that the system will be more reliable because it is harder to spoof an aggregated context information to spoof a single sensor data. In the system there is also a Database attached to the Context Server, the Intent Server, and the Safety Decision Program. The Database keeps historical records of both intent statements configured by system owners as well as individual sensor measurements and combined context information, with the purpose of serving as training data set for the optional ML component. Note that the Safety Decision Program receives instantaneous intent and context information directly from Intent and Context Servers respectively. The Safety Decision Program interface towards to Database is for receiving historical data to train an ML model and also for updating the data set with safety decisions, which later can be used for training purposes.

4.2 Example Use-Case from Manufacturing

Our proposed method can be applied in a smart manufacturing scenario where both automation and safety control messages are transmitted over a wireless channel. Such a scenario showcases a common production line where a robot arm mounts car parts in a semi-isolated area named robot cell. Functional safety runs over Profisafe with a light curtain that detects any dangerous movements, like someone reaching into the cell where the robot operates, which results in emergency stopping of the robot to prevent harm to robot and/or intruding worker. A light curtain, is a very common type of safety sensors used in various manufacturing use cases. It is a quite simple device composed of arrays of light, infrared or other types of light transmitters and receivers positioned opposite to each other. The transmitter constantly sends a light/laser beam to the receiver's photo/laser sensor and whenever these sensors do not receive any light/laser, they understand that an object is in between and blocking the light. Thus, the primary function of a light curtain as a safety sensor is to constantly scan for potential interruption of the light/laser beam by a human or object, and if the light/laser intensity measurement by the sensor is below a threshold or zero, the Profisafe safety program stops the robot operation as an emergency case since this measurement indicates that an object is about to enter the robot cell.

As stated previously, the decision made by the Profisafe safety program relies only on a single sensor measurement, that is the light intensity value, to detect the safety-critical situation of an object which will soon enter the dangerous production zone. This manufacturing use case is prone to malicious falsification of the safety situation. For instance, a commonly known issue with these light curtain devices is that their detection performance is badly affected by

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

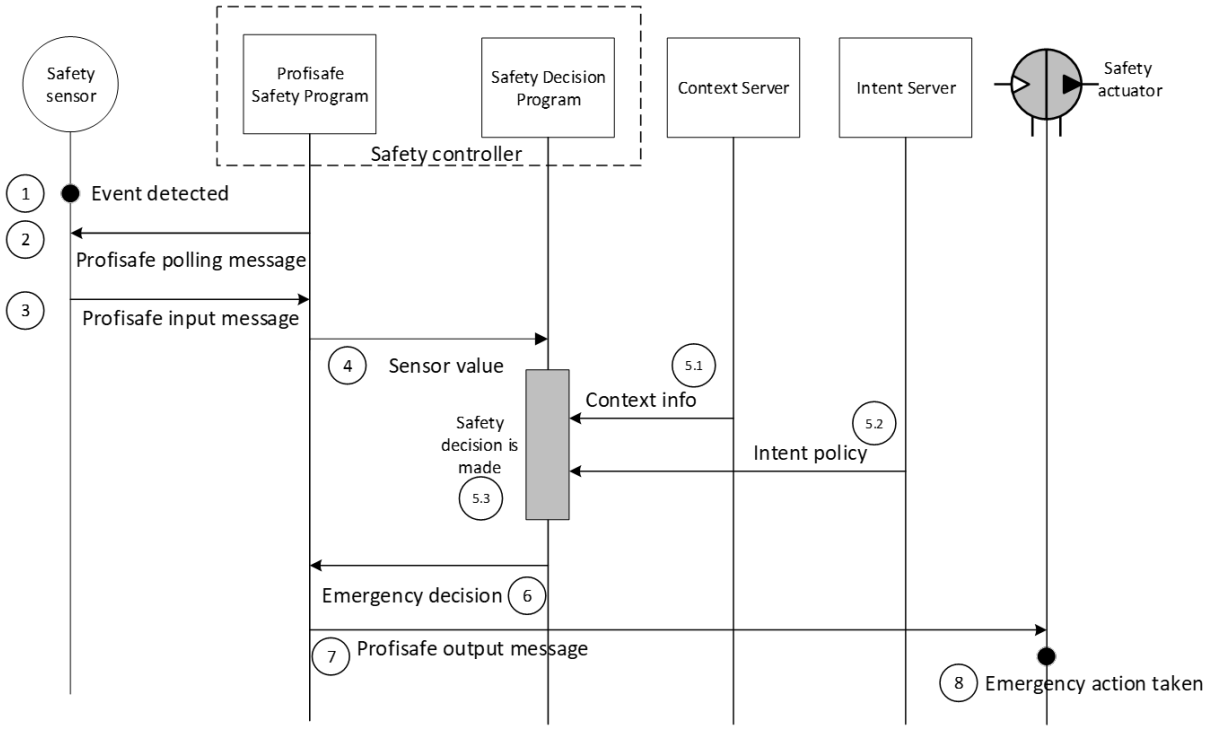

Fig. 3: Main steps in the proposed protocol

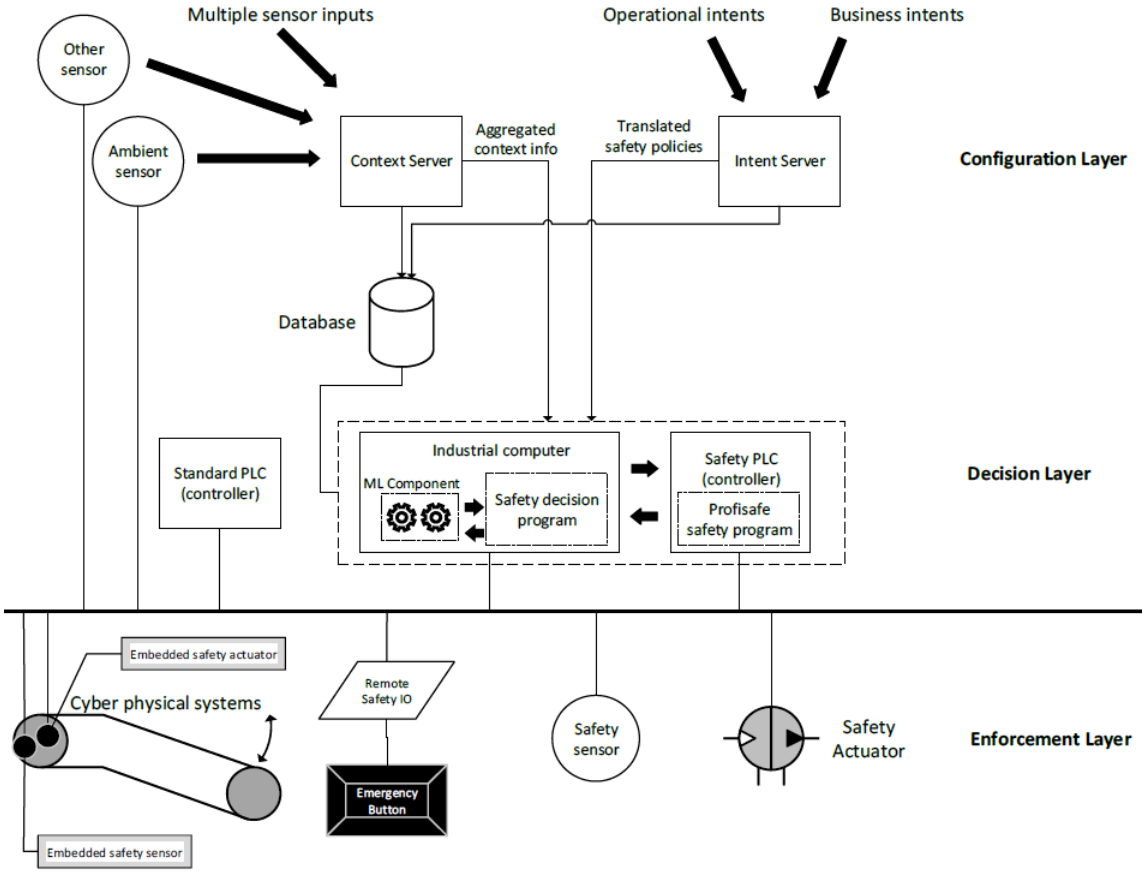

Fig. 4: Intent-based and AI/ML integrated functional safety model

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

interference coming from other light sources or reflective surfaces. In the operating manual of one of the largest safety sensor producers, it is stated that safety light curtain systems operating in close proximity to each other can interfere with each other. For instance, infrared light beams of the sender of system 1 can interfere with the receiver of system 2. This can disrupt the protective function of system 2 and also would mean that the operator is at risk. This interference problem can be exploited by malicious parties who purposefully direct light onto the receiving sensor of the safety curtain to manipulate the intensity measurement,which is then forwarded to the Profisafe safety program. If an attacker sends laser or some other type of light beam targeting the sensor of the safety curtain while a worker attempts to reach into the robot cell, this can cause the safety curtain to falsely think there is no interruption in the light signal. Therefore, emergency mode may not be triggered, and that worker could be injured or worse. Such an attack can be launched either locally by some intruding unauthorized person or a malicious insider physically holding an interfering light source, or it could be launched remotely by hacking into a nearby source of light like a security camera with flashlight or a connected IoT lamp.

Such an attack targeting the connected smart manufacturing scenario described above can be prevented by our proposed method of intelligent comparison of the more comprehensive set of sensor measurements combining context information with intent-based policy statements. Instead of using the single sensor measurement of light intensity at light curtain sensor, the vector comprising of [light intensity, robot speed, safety mat pressure, motion detection] can be used as the input to the safety decision program. Motion detection can be performed using already installed or easy to install equipment like a security camera or motion sensor. Similarly, safety mat is a commonly used equipment for measuring pressure on the floor of the entrance to a hazard zone, like a robot cell, to detect when someone is dangerously close.

Rotational speed of the robot is already measured by integrated sensors since this is needed in robot control applications. Thus, as opposed to Profisafe, which does not make use of these additional sensors in safety decision, our method combines them together with the single sensor of Profisafe, light curtain, to have an extended input set in the form of a vector. This context vector serves as a better input to detect a potentially dangerous intrusion of a worker into the robot cell. For example, if light intensity at receiving light curtain sensor is below the threshold, robot speed is above the threshold, safety mat pressure is above the threshold and the motion detection sensor measurement is positive, then this context vector implies that there is a potentially dangerous situation.

If an attacker maliciously interferes with a light sensor using another strong light source to fool the light curtain into making the wrong decision, this will be detected based on the information from the combined context vector. For instance, if light intensity at a receiving light curtain sensor is above the threshold indicating no intruding object but the safety mat pressure is above the threshold, a motion detection sensor measurement is positive, and the robot speed is also above threshold, then our method flags a dangerous safety event.

On the other hand, when the safety mat pressure is below the threshold, the motion detection sensor measurement is negative, and the robot speed is also below the threshold, our method does not flag a dangerous safety event even if the light intensity at a receiving light curtain sensor is below the threshold indicating an intruding object.

The first case represents an attack to cause harm to people/equipment by trying to fool the safety system into deciding that there is no emergency when there actually is, while the latter case represents an attack to cause disruption on availability by trying to fool the safety system into deciding that there is emergency when there actually is not. These examples show the improved ability of our solution to better assess the existing situation from a safety perspective thanks to the richer information provided by the combined context vector compared to reliance on Profisafe utilizing a single source of information from a safety sensor.

4.3 Realization for Manufacturing Use-Case

Our method follows a different path compared to Profisafe since all four main parameters are involved in a functional safety system, namely, input, decision parameter, decision logic and output. How we use this aggregated context information, raw sensor data and intents is explained below for this smart manufacturing example in comparison with the current Profisafe-based method.

Default decision method (Profisafe Safety Program)

- *Input* is the raw sensor measurement denoted as Lcurtain(t), that is the light intensity measurement at time t, of the receiving light sensor of light curtain.
- *The decision parameter* is a preset fixed threshold value (Tlight-intensity)
- *The decision logic* in the Profisafe Safety Program is based on only one comparison operation (is the light intensity sensor measurement below the threshold).
- *Output* can have only two possible values: "Stop the robot" / "Don't stop the robot".

Thus, the Profisafe Safety Program makes its safety decision by checking whether Lcurtain(t) $<$ Tlight-intensity. Lcurtain(t) is received periodically from the light curtain sensor and when a value smaller than the threshold is observed the robot is stopped due to a safety-critical situation.

The proposed Intent based safety decision method

- *Input* is the combined context information vector [Lcurtain(t), Vrobot(t), Psafety-mat(t), M(t)] where Lcurtain(t), Vrobot(t), Psafety-mat(t) and M(t) are respectively light intensity, robot speed, safety mat pressure and motion detection. M(t) is Boolean where M(t) $= 1$ if motion is detected, and 0 otherwise.
- *The decision parameters* are policy statements received from the intent server and have been configured there by the system owner. For this smart-manufacturing example, a high level intent statement could be "Robot cell operation continues as long as

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

there is no dangerous situation that requires emergency action". Then, the corresponding translated Safety Policies would be:

  - "SP1=Dangerous situation requires human presence and fast robot movement",
  - "SP2=Dangerous situation is not possible unless a moving human/object is present nearby the robot cell",
  - "SP3=Dangerous situation requires the robot cell immediately take emergency action",
  - "SP4=Emergency action is full stop in highly dangerous situations and slowing down in potentially dangerous situations",
  - "SP5=Highly dangerous situations require high certainty in Normal mode and less certainty in Crowded mode" where Normal mode and Crowded mode can be defined using the number of people around the place.

- *The decision Logic* in our proposed Safety Decision Program is either rule-based as explained below or utilizes ML/AI to make the safety decision.
- *Output* can have more than two possible values: "Stop the robot" / "Slow down the robot speed to level speed1" / "Slow down the robot speed to level speed2" / ... / "Don't stop the robot".

Thus, the Safety Decision Program (SDP) makes its safety decision by evaluating the current input set [Lcurtain(t), Vrobot(t), Psafety-mat(t), M(t)] against the configured intents mapped to safety policies, namely SP1, SP2, SP3, SP4, SP5. Note that how this evaluation is performed depends on the actual implementation in several aspects. For instance, statements in the intent safety policy could be converted into expressions like:

- SP1: $\forall$ Dangerous situation, Psafety-mat(t) $>$ Tpressure $\wedge$ Vrobot(t) $>$ Tspeed;
- SP2: $\forall$ Dangerous situation, M(t) $= 1$;
- SP3: If Psafety-mat(t) $>$ Tpressure $\wedge$ Vrobot(t) $>$ Tspeed $\wedge$ M(t) $= 1$ then SDP(output) $=$ Emergency Action;
- SP4: Emergency Action $= \{$Full Stop If Highly Dangerous; Else, decrease speed one level down$\}$;
- SP5: Highly Dangerous $= \{$Dangerous situation with Lcurtain(t) $<$ Tlight-intensity if Normal mode or Dangerous situation if Crowded mode$\}$.

As stated previously, intent-driven safety policies can be represented in different forms considering the type of processing computer program such as conventional and machine learning programs.

In order to make a decision about the current safety status, triggered by receiving Lcurtain(t), the Safety Decision Program compares the current context vector's value ([Lcurtain(t), Vrobot(t), Psafety-mat(t), M(t)]) with conditions dictated by the intent-based safety policies (SP1, SP2, SP3, SP4, SP5) and checks whether this comparison results in a match with any safety policies indicating an emergency situation.

For the robot cell example, having the context information [Lcurtain(t) $<$ Tlight-intensity $\wedge$ Psafety-mat(t) $>$ Tpressure $\wedge$ Vrobot(t) $>$ Tspeed $\wedge$ M(t)=1] will lead to an emergency action of Full Stop since the context information matches one of the intent policy statements which indicates a highly dangerous situation.

For this example, the full emergency stop will also occur in Crowded mode even if Lcurtain(t) $\geq$ Tlight-intensity, to prevent harm to human even in case of a possible falsification from a malicious interference attack since it is riskier situation compared to normal mode.

On the other hand, if only the light sensor triggers (Lcurtain(t) $<$ Tlight-intensity), but all other sensors affecting the vector do not indicate anything out of the ordinary (no movement, no pressure on the floor, nor too high speed of the robot), the conclusion would be that there is no danger. This prevents an attack on availability which aims to cause the system to take an emergency action. In addition to SP1, ..., SP5, there could be additional policies defined, for example, to reduce the speed of the robot. According to such a policy, the robot speed will be decreased by one level if [(Lcurtain(t) $<$ Tlight-intensity $\wedge$ Psafety-mat(t) $>$ Tpressure) $\vee$ ((Lcurtain(t) $<$ Tlight-intensity $\wedge$ M(t) $= 1$)] meaning if at least two sensor measurements, e.g., light curtain and safety mat or light curtain and motion sensor, indicate a potentially dangerous situation. This same action can also be taken when utilizing an ML model to make estimates of possible dangerous situations that are about to occur. The feature of having multiple levels of robot speed which can be adapted based on sensor readings makes our method to capture the best compromise for availability and safety. As stated earlier, additional policies would need to be defined to capture all other possible combination of context information indicated by different sensors. This is not needed if an ML component is utilized since it makes automated decisions once the ML model is appropriately trained.

4.4 ADVANTAGES OF THE PROPOSED SOLUTION

The main advantage of our method over Profisafe is that the use of intent-driven policies utilizing a larger set of input data provides a more accurate decision compared to just looking at one value as done in Profisafe. Our method has the advantage of utilizing a stronger baseline for making decisions, namely, the intent-driven safety policies, when comparing this extended input set to decision parameter, which is only the fixed threshold value in Profisafe. One more advantage of using intent-driven policy statements as decision parameter is the flexibility and ease of adapting them to changes in the operational environment. As explained in the example scenario above for the manufacturing-use case with the robot cell, in intent-based approach it is possible to make different safety decisions depending on different operational modes of the robot cell such as Normal and Crowded mode. With the intent-driven approach, policy statements for these different modes are converted into appropriate policy rule expressions. In this way, the owner/manager of the underlying cyber-physical system only needs to select the operating mode and then our method makes the relevant safety decision best fitting to the current situation.

Once a safety-critical incident is detected, our method is able to apply an action selected from multiple options most

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

appropriate for the current situation and intended operation of the system. In other words, rather than the Boolean option in Profisafe for actuating the safety measure such as either switch to fail-safe mode (emergency stop) or stay in the normal mode (continue rotation), our method can command actuators among multiple options such as slow down to speed 1, slow down to speed 2, . . . , slow down to speed n, and emergency stop. This allows the system to make a good compromise between availability and safety due to its ability of making more precise assessments of a current situation. Predictions for potential future dangers can be made as explained in the AI/ML Integration subsection.

Another advantage of the proposed method is that the program implementing the safety function will not rely directly on raw sensor data, which is prone to malicious modification and spoofing or which can be altered accidentally. Thus, the cyber-physical system will be able to make reasonably correct decisions on when to trigger the safety function. One other major advantage is that our method will neither require any change in the safety protocol itself nor any major change in the deployment of the safety system since the Safety Decision Program can be integrated with existing conventional safety program code as a function or it can run as a separate application communicating with the safety program as depicted in Figure 4.

Furthermore, an extension to the proposed method can be done to meet requirements of highly safety sensitive systems as follows. When the intent-based system suspects that safety systems/sensors are fooled with a probability, it will still trigger the safety mode but then it will also verify the real situation using other sensors or event logs (online or offline). If it is detected that there is an on-purpose malicious activity to make the safety system produce wrong decisions, it increments an associated reputation counter to feed into the Safety Decision Program as an additional parameter for future evaluations. In this way, it will be assured that the safety function will promptly take the necessary safety actions even in suspicious cases. Based on the further analysis of the event, as discussed earlier, the system can also classify events as genuine or attacks. Based on the frequency and type of attacks, i.e. sensor value manipulations, the system learns to classify similar situations as likely attacks or genuine safety situations, thereby acquiring an ability to prevent constant stopping of the cyber-physical system in case of multiple spoofing attempts performed one after another.

4.5 AI/ML Integration

Integration of AI/ML into the decision-making mechanism of functional safety systems can be useful especially when the number of information sources (e.g. sensors, network parameters) feeding the system becomes large. As the number and variety of data affecting the decision-making mechanism in the system increases, it becomes more difficult to program and manage. Another advantage of employing ML in safety protocols is that it may become possible not only to detect complex safety-critical situations but also to predict an impending safety-critical situation before it realizes into existence. In this sense, an optional ML component can evaluate the safety status by taking the context information along with the intent-based policies as input and decide whether the current situation is safety-critical or not.

The ML component can be developed based on (i) supervised learning or (ii) reinforcement learning (RL), each of which is described below.

(i) Supervised Learning

To develop a useful ML model by supervised learning, a training dataset needs to be built by collecting historical safety decisions along with associated context-related data and intent-based policy. Figure 5 illustrates content of such a dataset. Through training over such a dataset, the ML model can learn the relationship between safety decisions and associated context information. Once sufficiently trained, it can classify whether a given context information should raise a safety-related alert or not.

Safety Decision		Intent*	Context Information						
			Sensor-1	…	Sensor-n	Other data-1	…	Other data-n	
No		1	Value	…	Value	Value	…	Value	Input-1
No		1	Value	…	Value	Value	…	Value	
Yes		1	Value	…	Value	Value	…	Value	
No	+	2	Value	…	Value	Value	…	Value	
No		3	Value	…	Value	Value	…	Value	
…..		….	…..	…	……	…..	…	…..	
Yes		2	Value	…	Value	Value	…	Value	
No		4	Value	…	Value	Value	…	Value	
No		m	Value	…	Value	Value	…	Value	

(a) Illustration of a dataset to be used for training ML model

Intent Category	Intent-driven policy statements
1	Stop the system when there is emergency
2	Stop machinery before someone gets hurt
3	Do not allow equipment to be overheated
4	Raise alert when an abnormal situation appears
….	….
m	Do not harm people

(b) Sample intent policy statements in smart-manufacturing domain

Fig. 5: Sample structure of training dataset for supervised learning

To develop an ML model that can predict impending safety-critical situations, the dataset can be constructed as time-series data such that records are ordered based on time of occurrence. These inputs may come from aggregated context data that can be input to the ML model for detecting safety critical situations. Then, an ML model (e.g Model-I in Figure 6) can be built to extract hidden or visible trends in the time-series dataset and can estimate future variations of context-data. Thus the ML Model-I can predict future values of the context-data within an acceptable error-range. Then,

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

another ML model (e.g. Model-II in Figure 6) can analyze whether the predicted future input would lead to a safety critical situation or not, which makes it possible to estimate a safety-critical situation before it happens. Alternatively, this goal can be achieved by defining buffer areas, representing an additional new state, that signal impending danger. This process eliminates sharp transitions between safety critical and non-safety critical states.

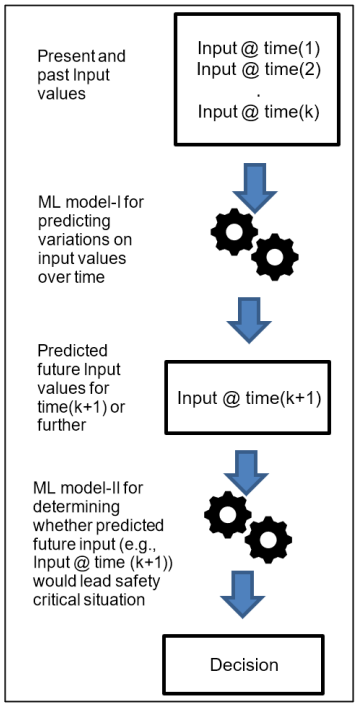

Fig. 6: Predicting impending safety-critical situations before they happen

(ii) Reinforcement Learning

In reinforcement learning, intelligent agents learn what to do in certain situations in complex environments to maximize the cumulative reward. They discover which actions yield the most reward by trying them during an exploration phase of the learning. In some cases, actions may affect not only the immediate reward but also subsequent rewards in forthcoming situations. Trial-and-error search and delayed reward are the two most important distinguishing properties of reinforcement learning [21].

Reinforcement learning is widely used in robotics [22], cyber-physical systems [23], [24], smart manufacturing [25], gaming [26] and many other domains [27], [28]. Since its learning mechanism is based on trial-and-error, reinforcement learning can be very useful especially in some real-world applications that are difficult to model, one of which could be detecting safety-critical situations in complex environments.

5 Implementation and Results

We used two different test setups to be able to implement both the rule-based and ML-based approach. The first setup is based on a PLC-based industrial development tool and is used to implement the proposed rule-based method. Since this setup does not allow the use of ML, we needed a second setup to implement our ML-based approach on a Reinforcement Learning development toolkit. In both setups, we implemented the same smart manufacturing scenario where there is a robot with a rotating arm in a production cell. A safety system is in place with a light curtain, a safety mat with a pressure sensor, and a motion detection sensor for ensuring safety as illustrated in Figure 7. Accordingly, during robot operation, if an object moving into the production cell, where the robot arm is rapidly rotating, causes a potentially dangerous situation, the robot switches to the fail-safe mode, e.g, stops. However, as we mention in the previous sections, a malicious act onto the light curtain can mislead the sensor and if the safety system relies only on the light intensity measurement, a hazardous situation can happen since the system will not switch to the fail-safe mode. We tested our proposed method in this scenario to verify whether our approach utilizing more safety-related information from different sensors could mitigate such an attack.

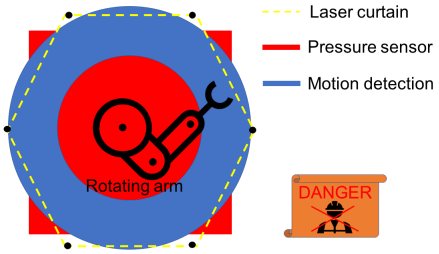

Fig. 7: A rotating arm where several sensors are deployed in the safety region

Details about the implementation and results are given below for the rule-based approach and the ML-based approach.

5.1 Rule-Based Approach

The development platform that we utilized in our work is the "Siemens Totally Integrated Automation (TIA) portal". This platform provides a complete package for both industrial software and automation hardware components. Using the TIA portal and its tools and features, the Siemens SIMATIC programmable logic controllers (PLC) and automation systems can be configured, programmed and simulated. In our study, we implemented our proposed method on the SIMATIC S7-1500 PLC which is a newer version of the automation system, released in 2012, and used for medium and upper performance ranges of applications [29]. We chose CPU 1516F-3 PN/DP as the PLC in our implementation because it is mostly used for fail-safe applications.

The TIA portal supports five types of programming languages to program S7-1500 PLCs: ladder logic (LAD), function block diagram (FBD), structured control language (SCL), statement list (STL) and sequential control (GRAPH). We realized our implementations in this work using FBD and LAD by linking binary logic functions of AND and OR, arithmetic instructions such as greater than and equal, and complex functions such as pre-defined safety-stop, which are represented as boxes.

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

In the test setup, we programmed the Profisafe based safety PLC with the safety program's standard logic using a single sensor and threshold, as well as the advanced logic of our proposed method using various sensors and intents. We simulated both normal and attack behaviours for both the default (conventional) safety program and the proposed method as explained below.

5.1.1 Verification of our implementation

We implemented two safety programs with and without applying our approach. We programmed the SIMATIC S7-1500 PLC on the TIA portal using functional block diagrams and simulated the implementations for verification. In the default method implementation, which is shown in Figure 8, the safety program can produce output for only the full stop state. In the implementation of our proposed method, there are two parts of the PLC program, one for slow state and one for full stop. The PLC ladder diagram for full stop is shown in Figure 9. The program parts of Full stop and slow state are referred to PLC networks in the TIA portal project, and they allow the safety system to produce two main outputs for putting the system into slow state or full stop respectively. After implementing the default method and the proposed method, we used the TIA portal simulation feature to test how outputs change first when we assume the inputs are applied without any attack, and then when there is an attack launched on the light curtain sensor. We explain the simulation results in more detail in the next subsection referring to these two simulations as "no attack case" (without attack) and "attack case" (with attack).

5.1.2 No attack case

In this case, we aim to verify that our proposed method is able to provide the same safety response as the default method. The implementation of the current method as seen in Figure 8, a built-in safety function block named $ESTOP$ takes input from the light curtain named $Lcurtain$. Whenever the light curtain detects an object, the $Lcurtain$ input changes, going from high to low, and the output of the block which is named $FullStop$ is triggered and becomes high, a logic 1. This represents that the safety program status changed to "full stop" to cease the robot arm's rotation. In the simulations, we observed that the PLC implementation of our proposed method given in Figure 9, showed the same response of switching to full stop in "Normal" (not crowded) mode by reading not only the light curtain input but also other safety sensors including safety pressure mat sensor, motion sensor and speed sensor of the robotic system. Besides, we observed stopping of the rotating arm in "Crowded" mode by taking inputs of three additional sensors independent of the light curtain input as indicated by intent policies.

5.1.3 Attack case

We fabricated an attack scenario to test the behaviors of both our proposed method and the default method. In the attack scenario, we assume the light curtain sensor of the safety area is accessible to an attacker from outside or a malicious insider. The light curtain detects an object when the object blocks the light between the light transmitter and receiver. If an attacker directs a constant external light source to the receiver, the receiver is not able to detect the objects which pass through the light curtain due to non-decreasing light intensity by the malicious light source. Our tests verified that the safety system with the default method depicted in Figure 8 was rendered ineffective for detecting safety situations. On the other hand, in our proposed method shown in Figure 9, even when the light curtain input was ineffective due to attack, the inputs from motion sensor, pressure mat and robot speed made the safety system trigger a full stop in Crowded mode, thereby detecting the objects passing through the safety area even under attack. We tested both methods in another attack scenario targeting availability where a malicious party puts an artificial blockage between transmitter and receiver of the light curtain when there is no person or device in and around the rotating robot arm. In this case, the default method made the robot stop unnecessarily when there is no danger, but in our proposed method the system continued to operate when none of the other three sensors but the light curtain indicated danger. Thus, our proposed method was able to maintain availability even under attack scenario.

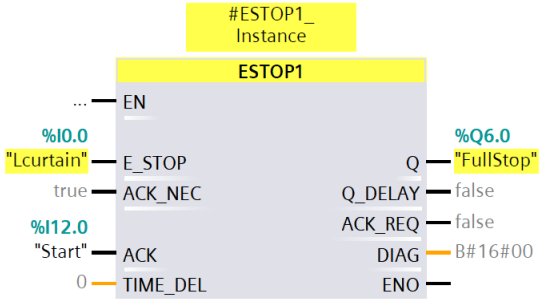

Fig. 8: Default method

5.1.4 Overhead Analysis

Regarding the overhead introduced by our proposed method and the default method, they can be compared by their code size and cycle times of the command execution on TIA Portal. In the simulation environment of the TIA portal, we tested the operations of both the proposed and the default methods' safety codes while under the normal mode and the attack mode. Diagnostic feature of the TIA portal showing the code and data memory in use and cycle time of the output of the PLC code are given in Table 1. These figures and results indicate an acceptable level of overhead of the proposed method considering the security advantages brought. Safety function is generally embedded separately into each piece of control equipment such as robot arms or autonomous guided vehicles. Therefore, the proposed method is expected not to be heavily impacted by the overhead induced even in large scale and more complex cyber-physical systems composed of tens or even hundreds of such machines. Safety software programmed for each device will have slightly increased cycle time, and non-centralized implementation of safety functions helps maintain a lower rate of memory usage since these individual overheads will not add up.

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

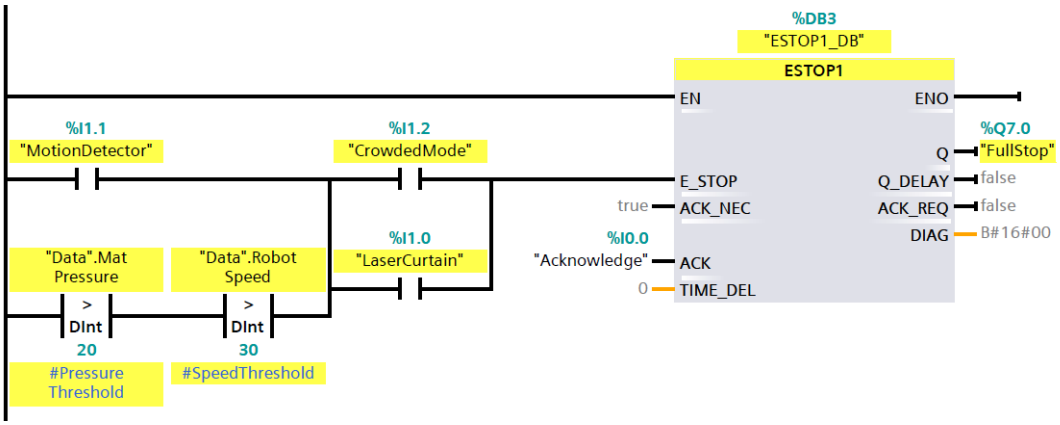

Fig. 9: Our proposed method for full stop

TABLE 1: Data and cycle time comparison

	Code Memory	Data Memory	Cycle Time
Default Method	6.1 kB.	4.5 kB.	0.24 cyc.
Proposed Method	6.6 kB.	4.6 kB.	0.29 cyc.

TABLE 2: States in the environment

State	Light	Pressure	Motion
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

5.2 Reinforcement Learning Implementation

We realized our implementation in the OpenAI Gym [30], which is a toolkit for developing and comparing reinforcement learning algorithms. In our experimental setup, we simulated three different sensor signals, which are light curtain, pressure mat sensor, and motion detector as depicted in Figure 7. We modeled these sensors as operating in binary mode for alarming (i.e. ON/OFF). Thus, there may be eight different states that can be observed in the environment as shown in Table 2. Moreover, in order to include malicious attack scenarios that cause sensors to malfunction, we added Gaussian noise to the operating range of the sensors, causing them to malfunction to a certain extent.

The agent in our RL setup is a rotating arm that can take two discrete actions: (i) ROTATE or (ii) NOT ROTATE. To define a reward mechanism, we identified four different situations as illustrated in Figure 10: (I) there is an object, i.e. a human, in the safety zone, so the robotic arm is in stop mode, which results in a positive reward because this is a desired situation, (II) there is a human in the safety zone and the robotic arm is rotating, which results in a negative reward because this is an undesired situation, (III) there is nobody in the safety zone and the robotic arm is rotating, which results in a positive reward because this is a desired situation, and (IV) there is nobody in the safety zone and the robotic arm is in stop mode, which results in a negative reward because this is not a desired situation.

We developed a Q-learning based reinforcement learning algorithm that seeks to find the best possible next action given its current state, in order to maximise the reward it receives. A traditional Q-learning algorithm [21], [31] is given in Equation 3 where R_{t+1} is the reward obtained when transitioning from state S_t to S_{t+1}, and α is the learning rate ($0 < \alpha \leq 1$). The term $\underset{a}{max}\, Q(S_{t+1}, a)$ corresponds to the maximum expected future reward and γ is the discount factor. Thus the Q-learning algorithm keeps updating the current Q-value (i.e. $Q(S_t, A_t)$) with its observations after taking an action.

$$Q^{new}(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \underset{a}{max}\, Q(S_{t+1}, a) - Q(S_t, A_t)\right] \quad (3)$$

Figure 11 shows the learning performance of our model for 1000 episodes, plotted based on 50-episode moving average values. It can be seen from Figure 11 that the model learns to get the maximum possible reward (i.e. 50 points in our experimental setting, depending on episode length and reward function) after training 1000 episodes. After completing such a training process, the model can be deployed in a test or work environment and expected to run with 100% accuracy because it has learned to work to achieve the maximum possible reward value in an episode. Indeed, when we evaluated the trained model over an additional 100 test episodes, we found that all the actions it took were correct.

Note that RL eliminated the need for manually defining a rule (i.e. the action to take) for each state, which would be getting harder as the number of states grows. It is also noteworthy that as mentioned earlier, although we have set the sensors to work somewhat incorrectly in order to simulate malicious attacks, the model manages to achieve the maximum reward points, which shows that it is able to resist such attacks.

6 Conclusion

Traditional functional safety systems are designed to operate using rule-based decisions relying on sensor measurements. Current safety systems are unable to detect intentful attacks that may come from malicious parties. In this paper, we propose an intelligent context-aware intent-based solution to detect and prevent on-purpose attacks

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

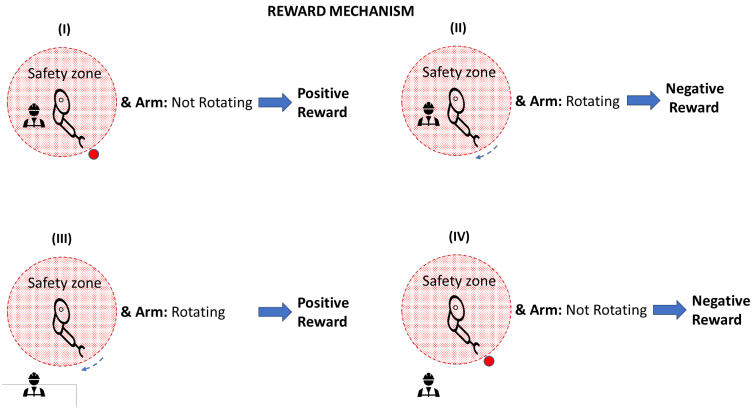

Fig. 10: T
he reward mechanism in our RL model: (I) there is a human in the safety zone and the robotic arm is in the stop mode, which results in a positive reward because this is a desired situation, (II) there is a human in the safety zone and the robotic arm is rotating, which results in a negative reward because this is an undesired situation, (III) there is nobody in the safety zone and the robotic arm is rotating, which results in a positive reward because this is a desired situation, and (IV) there is nobody in the safety zone and the robotic arm is in the stop mode, which results in a negative reward because this is not a desired situation.

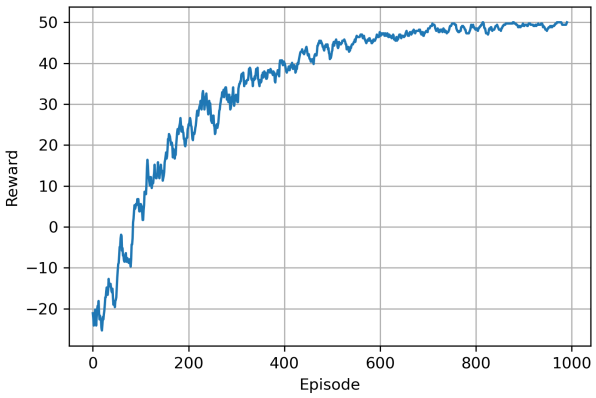

Fig. 11: Learning performance of our model

targeting critical safety functions in cyber-physical systems that cannot be prevented by current safety systems.

The operation of existing functional safety protocols such as Profisafe only rely on raw sensor data transmitted from safety sensors to safety controllers. We improved the current operation of such protocols with the use of a collective set of sensor data received from all relevant sensors to make intelligent decisions on the current safety situation. In this way, aggregated context information of the individual or overall cyber-physical systems are used to check whether there is indeed an emergency or if the current situation is normally aligned with the system's operational and business intents.

We also improved the decision baseline by integrating operational intents for comparison with the comprehensive contextual input collected from multiple sensors to detect the safety situation and potential malicious behavior. Consequently, the decision baseline is aligned with what the system is expected to do from both safety and security perspective, thereby making this safety system more resistant to attacks which try to manipulate the decision baseline.

We demonstrated both the rule-based and the ML-based methods through implementation on two test setups. For the rule-based approach, we coded two safety programs, one using our approach and the other using the default existing safety methodology. Then, we verified the response of these two programs when there was no attack and when there was a simulated attack. In the no attack case, both programs operated as expected, going to full stop when an object triggered the sensors. In the attack case simulation where we assumed an external attacker disabled the light curtain, our proposed method still operated as expected by going to full stop based on other sensors detecting the foreign object despite the light curtain malfunction. However, in the default program, safety program depends on only the light curtain. Malfunction caused by an attacker adversely impacted operation of the critical safety function by allowing the foreign object to enter the safety zone undetected. Moreover, by comparing the memory utilization and execution cycle times of these two implementations, we showed that our method incurs an acceptable level of overhead while successfully preventing attacks on the safety function.

For the ML-based approach, we developed an RL implementation to control a rotating arm against possible safety violations upon entering the danger zone in an environment where a variety of different sensors such as light curtain, motion detector and pressure sensor are deployed. We demonstrated that the agent (i.e. the rotating arm) can learn what action to take in the face of varying combinations of sensor data by automatically detecting safety violations without manual programming.

ACKNOWLEDGMENTS

This work was funded by The Scientific and Technological Research Council of Turkey, under 1515 Frontier RD Laboratories Support Program with project no: 5169902.

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

REFERENCES

[1] T. Novak and A. Treytl, "Functional safety and system security in automation systems a life cycle model," 10 2008, pp. 311 – 318.

[2] S. Xu and I. Koren and C. Mani Krishna, "Adaptive workload adjustment for cyber-physical systems using deep reinforcement learning", Sustainable Computing: Informatics and Systems, vol. 30, pp. 100525, 2021.

[3] P. Organisation, "Profinet," 2022. [Online]. Available: https://www.profibus.com/technology/profinet

[4] P. Organisation, "Profisafe," 2022. [Online]. Available: https://www.profibus.com/technology/profisafe

[5] IEC 61784-3-3:2010, Industrial communication networks - Profiles - Part 3-3: Functional safety fieldbuses - Additional specifications for CPF3, *European Committee for Electrotechnical Standardization, Brussels*, 2010.

[6] J. Åkerberg, M. Gidlund, T. Lennvall, J. Neander, and M. Björkman, "Efficient integration of secure and safety critical industrial wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, pp. 1–13, 2011.

[7] B. K. Saha, D. Tandur, L. Haab, and L. Podleski, "Intent-based networks: An industrial perspective," in *Proceedings of the 1st International Workshop on Future Industrial Communication Networks*, ser. FICN '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 35–40. [Online]. Available: https://doi.org/10.1145/3243318.3243324

[8] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand, "A survey of approaches combining safety and security for industrial control systems," *Reliability Engineering & System Safety*, vol. 139, pp. 156–178, 2015.

[9] J. Åkerberg and M. Björkman, "Exploring network security in profisafe," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2009, pp. 67–80.

[10] M. Sun, S. Mohan, L. Sha, and C. Gunter, "Addressing safety and security contradictions in cyber-physical systems," in *Proceedings of the 1st Workshop on Future Directions in Cyber-Physical Systems Security (CPSSW'09)*. Citeseer, 2009.

[11] N. Tuptuk and S. Hailes, "Security of smart manufacturing systems," *Journal of Manufacturing Systems*, vol. 47, pp. 93–106, 2018.

[12] T. Novak and A. Treytl, "Functional safety and system security in automation systems - a life cycle model," in *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, 2008, pp. 311–318.

[13] I. E. Commission *et al.*, "Iec 61508 functional safety of electrical/electronic/programmable electronic safety-related systems," *International Electrotechnical Commission, Geneva, Switzerland*, 1998.

[14] ISO/IEC 15408-1:2009, "Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model," 2009.

[15] A. Angelopoulos, E. Michailidis, N. Nomikos, T. Panagiotis, A. Hatziefremidis, S. Voliotis, and T. Zahariadis, "Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects," *Sensors*, vol. 20, p. 109, 12 2019.

[16] N. Moustafa, E. Adi, B. Turnbull, and J. Hu, "A new threat intelligence scheme for safeguarding industry 4.0 systems," *IEEE Access*, vol. 6, pp. 32 910–32 924, 2018.

[17] R. C. B. Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *2014 7th International Symposium on Resilient Control Systems (ISRCS)*. IEEE, 2014, pp. 1–8.

[18] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[19] S.-T. Park, G. Li, and J.-C. Hong, "A study on smart factory-based ambient intelligence context-aware intrusion detection system using machine learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1405–1412, 2020.

[20] T. Desot, F. Portet, and M. Vacher, "Towards end-to-end spoken intent recognition in smart home," in *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. IEEE, 2019, pp. 1–8.

[21] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[22] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013. [Online]. Available: https://doi.org/10.1177/0278364913495721

[23] C. Li and M. Qiu, *Reinforcement Learning for Cyber-Physical Systems: with Cybersecurity Case Studies*. CRC Press, 2019.

[24] X. Liu, H. Xu, W. Liao, and W. Yu, "Reinforcement learning for cyber-physical systems," in *2019 IEEE International Conference on Industrial Internet (ICII)*, 2019, pp. 318–327.

[25] B. I. Epureanu, X. Li, A. Nassehi, and Y. Koren, "Self-repair of smart manufacturing systems by deep reinforcement learning," *CIRP Annals*, vol. 69, no. 1, pp. 421–424, 2020.

[26] C. Beattie, T. Köppe, E. A. Duéñez-Guzmán, and J. Z. Leibo, "Deepmind lab2d," *ArXiv Preprint arXiv:2011.07027*, 2020.

[27] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[28] G. Li, R. Gomez, K. Nakamura, and B. He, "Human-centered reinforcement learning: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 4, pp. 337–349, 2019.

[29] H. Berger, *Automating with SIMATIC S7-1500: Configuring, Programming and Testing with STEP 7 Professional*. John Wiley & Sons, 2014.

[30] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *ArXiv:1606.01540*, 2016.

[31] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[32] M. Baezner and P. Robin, "Stuxnet," ETH Zurich, Tech. Rep., 2017.

[33] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

E. Tomur. received his Ph.D. in Information Systems from Middle East Technical University in 2008 and his M.Sc. and B.S. degrees both in Electronics Engineering from Bilkent University in 2001 and in 1999, respectively. He has been working as a Research Leader at Ericsson since January 2019 where he is leading the Ericsson Research Site in Turkey. Before joining Ericsson, he worked as R&D Manager in private sector companies, Technology Transfer Manager in universities where he gave courses and served as graduate thesis advisor. His technical expertise is on security of Internet of Things, M2M and Wireless Sensor Networks.

Z. Bilgin received his Ph.D. degree in computer science from the City University of New York, NY, and the B.S. and M.S. degrees in electrical&electronics engineering from the Gazi University, Ankara. He involved in many industrial and academic research projects related to telecommunication, IoT, 5G, cybersecurity, social computing, computer vision, smart grid and machine learning, and authored many scientific articles and inventions in these domains. He works as a Senior Specialist at Arcelik Research, and previously served as an Experienced Security Researcher at Ericsson Research in Turkey.

U. Gülen. received the B.Sc. degree in electronics and communication engineering in 2012, from Yildiz Technical University, Istanbul, Turkey, and the M.Sc. degree in 2014, from Bahcesehir University, Istanbul. He is currently a Ph.D. candidate in computer engineering at Bahcesehir University. He joined Ericsson Research Turkey in 2020 as an Experienced Security Researcher. His research interests are applied cryptography, network security and public-key cryptography on embedded systems and IoT devices.

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3251031

E. Ustundag Soykan. received her M.S. and Ph.D degrees in Computational Science and Engineering from Istanbul Technical University. She had worked at The Scientific and Technological Research Council, National Cryptology Institute for 13 years in security domain. She joined Ericsson Research in 2018, and is currently working as Master Security Researcher. She has published several papers in international conferences, mostly on information security and privacy. Her research interests include ML/AI security, privacy enhancing technologies, IoT security.

L. Karaçay received her Ph.D. and M.Sc. degrees both in Computer Science and Engineering in 2020 and 2012 from Sabanci University, Turkey. She had worked as a Teaching Assistant at Sabanci University for 7 years. She has some scientific research papers published in journals and conference proceeding in the area of security. She joined Ericsson Research in October 2019. She works as an Experienced Security Researcher at Ericsson Research in Turkey.

F. Karakoç received his Ph.D., M.Sc., and B.Sc. degrees in Computer Engineering from Istanbul Technical University. He has been working as a security researcher and 3GPP SA WG3 delegate at Ericsson since 2020. Before joining to Ericsson, he worked on information security and cryptography at private sector companies and public institutes. He also teaches information security and cryptography related courses at universities.

Authorized licensed use limited to: Ericsson. Downloaded on August 03,2023 at 11:07:11 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.