electronics

MDPI

Article

SPD-Safe: Secure Administration of Railway Intelligent Transportation Systems

George Hatzivasilis [1,2,*], Konstantinos Fysarakis [3], Sotiris Ioannidis [4], Ilias Hatzakis [2], George Vardakis [2], Nikos Papadakis [2] and George Spanoudakis [3]

1 Institute of Computer Science, Foundation for Research and Technology—Hellas, Vassilika Vouton, GR-70013 Heraklion, Greece
2 Electrical and Computer Engineering, Hellenic Mediterranean University (HMU), Estavromenos, GR-71410 Heraklion, Greece; hatzakis@cs.teicrete.gr (I.H.); gvardakis@cs.hmu.gr (G.V.); npapadak@cs.hmu.gr (N.P.)
3 Sphynx Technology Solutions AG, Innovation Department, 6300 Zug, Switzerland; fysarakis@sphynx.ch (K.F.); spanoudakis@sphynx.ch (G.S.)
4 Electrical and Computer Engineering, Akrotiri Campus, Technical University of Crete, GR-73100 Chania, Greece; sotiris@ece.tuc.gr
* Correspondence: hatzivas@ics.forth.gr; Tel.: +30-2810-391600

Abstract: The railway transport system is critical infrastructure that is exposed to numerous man-made and natural threats, thus protecting this physical asset is imperative. Cyber security, privacy, and dependability (SPD) are also important, as the railway operation relies on cyber-physical systems (CPS) systems. This work presents SPD-Safe—an administration framework for railway CPS, leveraging artificial intelligence for monitoring and managing the system in real-time. The network layer protections integrated provide the core security properties of confidentiality, integrity, and authentication, along with energy-aware secure routing and authorization. The effectiveness in mitigating attacks and the efficiency under normal operation are assessed through simulations with the average delay in real equipment being 0.2–0.6 s. SPD metrics are incorporated together with safety semantics for the application environment. Considering an intelligent transportation scenario, SPD-Safe is deployed on railway critical infrastructure, safeguarding one outdoor setting on the railway's tracks and one in-carriage setting on a freight train that contains dangerous cargo. As demonstrated, SPD-Safe provides higher security and scalability, while enhancing safety response procedures. Nonetheless, emergence response operations require a seamless interoperation of the railway system with emergency authorities' equipment (e.g., drones). Therefore, a secure integration with external systems is considered as future work.

Keywords: intelligent transportation; railway; CPS; security; safety; critical infrastructure

Citation: Hatzivasilis, G.; Fysarakis, K.; Ioannidis, S.; Hatzakis, I.; Vardakis, G.; Papadakis, N.; Spanoudakis, G. SPD-Safe: Secure Administration of Railway Intelligent Transportation Systems. *Electronics* **2021**, *10*, 92. https://doi.org/10.3390/electronics10010092

Received: 18 November 2020
Accepted: 29 December 2020
Published: 5 January 2021

1. Introduction

Railways continue to be one of the main transport systems nowadays [1,2], covering public, private, and military needs over a wide operational area. Thus, railway assets are an attractive target for malicious actors and are exposed to various threats, from natural events to man-made ones, such as terrorism or vandalism (e.g., [3–6]).

The associated risks are exacerbated by the fact that railway infrastructure assets are typically placed along the route, including remote areas where physically protecting them is challenging. Moreover, railway premises have a large attack surface (due to their numerous electronic and electrical parts, such as power supply, switches, scheduling, and other subsystems), but often reside far from the main stations. While auditing for physical threats is quite important [7], the premises are usually inspected remotely through cameras. Sensory equipment is also deployed to monitor environmental parameters. The goal is to prevent potential intruders [8–10], avoid machinery overheating, and detect fires. Since the

interconnection of this monitoring equipment is, at least partly, wireless, it can become a target of several types of attacks.

In this context and considering that a successful attack could damage the railway's operation or even cause severe injuries and deaths, cybersecurity is an important consideration for such interconnected critical systems [11,12]. Attackers can disrupt communications (e.g., through jammers) or even infiltrate the networks and take control of critical equipment [13]. Cyber-attacks on the command-and-control centers (C&C) and the information systems are also feasible [13,14]. Thus, the secure interconnection of all the deployed elements and platforms is important, and the cyber and physical security of the critical infrastructure becomes imperative [9,10].

As sketched above, safety is another design factor and one that is closely related to security. Cargo and passengers are transported in high volumes each day, covering long distances. In the past, railway accidents have caused a number of deaths, along with significant financial losses [15]. While the introduction of electronic controllers reduced the occurrence of such situations [16], safety risks cannot be ignored, considering the wide railway coverage that still includes aspects such as uninspected car-crossings, system malfunctions (like signal loss), and, of course, the human factor [17,18].

Within the ever-changing technological landscape, there is currently a move from automated to intelligent cyber-physical systems (CPS), motivated by the speedy infiltration of the Internet of Things (IoT) and cloud computing and enabled by wireless networking [19–22]. Wireless sensor networks (WSNs) [23] can cover the wide railway operational territory, gathering and processing pieces of ambient knowledge, while gateways can be used to transmit the data to the controlling center or a cloud service. The railway controlling software at the backend can, then, collect and integrate the spatial information and manage the underlying subsystems [24,25]. Therefore, WSNs are an ideal solution for covering the railway operating area, including the railway routes and various scattered shelters.

However, the railway cyber infrastructure and networks currently only adopt rudimentary defenses (e.g., cryptography), which provide protection against the most basic threats, forfeiting effective ways of detecting advanced cyber-attacks [26]. While initially designed as closed systems, current infrastructure networks are vulnerable to various network layer attacks, like blackhole, badmouthing, and jamming attacks [27].

Motivated by the above, this work presents "SPD-Safe", (security, privacy, and dependability (SPD)), an administration framework for railway CPS, aiming to enhance the security, privacy, dependability, and safety of the intelligent railway infrastructure, while enabling services for monitoring and managing the overall setting. The framework integrates mechanisms for mitigating cyber-attacks attempting to disrupt communications or compromise infrastructure assets, and periodic malfunctioning of assets is also taken into consideration. SPD-Safe can act as an intelligent communications-based train control (CBTC) system for railway CPS, leveraging artificial intelligence (AI) to manage the system at runtime. The system uses standardized solutions, and its building blocks can be easily retrofitted in current deployments.

In addition to the detailed description of the proposed framework, a preliminary implementation is described and evaluated, concentrating on the management of: (a) In-carriage, and (b) on-route sub-systems. WSNs are deployed inside the carriage and by the railway tracks to safeguard carriages that transfer dangerous freight and to help avoid crashes with objects blocking the train's route (like stuck vehicles on rail track crossings), respectively. Furthermore, smart cameras are installed to improve the physical security of the critical infrastructure. In the context of the two use cases (a) and (b) above, through SPD-Safe the railway CPS is configured in real-time to tackle ongoing cyber-attacks and control safety-related incidents. This hands-on validation was developed and demonstrated under the EU-funded project new embedded Systems arcHItecturE for multi-Layer Dependable solutions (nSHIELD) [28], with the cooperation of major industrial partners in the railway and defense domains, including Ansaldo STS (http://www.railway-technology.com/contractors/signal/ansaldo-sts/), Selex ES (now Leonardo

S.p.A.: https://www.leonardocompany.com/en/home), and HAI (http://www.haicorp.com/en/). Simulation analysis was also conducted during the design phase, utilizing the security-aware Cyber-Physical Systems (CPS) Simulator Framework (COSSIM) [29], paving the way for the final installation of the proposed system, as presented in the following sections.

The rest of the paper is structured as follows: In Section 2, related work on railway signaling systems is reviewed. In Section 3, the middleware platform and intelligent agent technologies that manage the underlying equipment are presented. In Section 4, the network layer protection mechanisms are detailed. In Section 5, the implementation details of SPD-Safe are provided and the application in the railway setting is demonstrated. The proposed system is also compared with relevant systems in Section 6, while Section 7 features the concluding remarks.

2. Materials and Methods—Related Work

Smart transportation ecosystems involve, among others, passenger services as well as critical infrastructure-related applications and the associated safeguards. The fundamental goals in this context include "green" (i.e., environment-friendly) operation, improved performance and efficacy, as well as enhanced security and safety.

Railways, in specific, rely on signaling systems that direct the trains' traffic. Infrastructure control and management is achieved via various telecommunication means that are installed on carriages and tracks. Communication between track equipment and trains is achieved via CBTC signaling systems [30–32] enabling the railway's management and infrastructure control. For the European Union (EU), the international wireless communications standard for railways includes the European Train Control System (ETCS) [33]. The communication baseline is implemented by the Global System for Mobile Communications—Railway (GSM-R) [34], which is further enhanced with the General Packet Radio Service (GPRS) [35] and forms the base of an intelligent transportation application. ETCS utilizes trackside equipment that transmits information regarding the route to unified controlling equipment within the train cab. Thus, all lineside data are passed wirelessly to the driver, without requiring the direct observation of lineside visual signals, as was the case in legacy railway settings. The adoption of ETCS results in more and longer running trains, with increased traffic and railway management capabilities.

In addition to the signaling developments, WSNs can now cover a wide railway operational area, gathering ambient data. Embedded systems implement intelligence solutions encompassing the underlying critical assets as well the interlinked smart city ecosystems. Related frameworks for intelligent monitoring of the critical infrastructure have already been proposed in the literature (e.g., [36,37]). The Integrated System for Transport Infrastructure surveillance and Monitoring by Electromagnetic Sensing (ISTIMES) project [36] implements a transport infrastructure surveillance and monitoring system with electromagnetic sensing. Distributed and local sensory equipment (e.g., optic fiber sensors, infrared thermography, low-frequency geographical techniques, etc.) are utilized to perform non-destructive electromagnetic sensing and monitoring of the critical infrastructure. The Cloud to Infrared Thermography (Cloud2IR) [37] deploys an infrared and environmental Structural Health Monitoring (SHM) information system. The software architecture enables multi-sensor connection and the interplay with cloud computing services (e.g., data aggregation, system management, etc.). However, the heterogeneity of the deployed equipment and diverse demands of the various applications make the administration of the underlying infrastructure a challenging task.

In parallel, as Service-oriented Architectures (SoAs) increase in popularity, a continuous effort to deploy SoAs within the Industrial IoT (IIoT) domain and the smart railway CPSs can be observed. Several technologies are proposed that support the required functionality, ranging from agent frameworks and middleware platforms, to communication protocols and data representation standards. Such state-of-the-art solutions are presented in the subsections that follow.

2.1. Management Platfroms and Reasoning Systems

Agent technologies constitute the typical option for modeling ambient intelligent systems that exchange information with the environment and user [38,39]. Intelligent agents inspect the surrounding setting and react to upcoming events at normal operation. Their AI modules process context-aware data, as collected from the surrounding environment by the attached devices.

Regarding the various agent technologies, 24 frameworks were analyzed in Kravari and Bassiliades [40], including the popular Java Agent DEvelopment framework (JADE), Agent Globe (A-GLOBE), and Jason (the hero's name from Greek mythology). JADE implements the relevant standards for Semantic Web and the Foundation for Intelligent Physical Agents (FIPA: http://www.fipa.org/) (e.g., the Agent Communication Language (ACL) [41]). The platform is easy to learn and user-friendly, while offering portability and compatibility with all Java Virtual Machines (JVMs). The open-source and stable developer versions operate with several programming languages, such as Java, Jess, and Prolog. The agent communication is fast, and the overall framework is efficient and scalable. Moreover, JADE supports strong user authentication and cryptographic solutions—i.e., JADE security (JADE-S)—along with Hypertext Transfer Protocol Secure (HTTPS). The framework is widely-used and is deployed in several fields, including reasoning in multiple domains, general purpose applications, mobile computing, and e-commerce. The study of Kravari and Bassiliades [40] also infers that JADE is the most popular framework due to the pure Java design and the co-operation with several web systems. In addition, five respectable organizations (France Telecom, Motorola, Profactor, TILAB, and Whitestein TEchnologies AG) supervise the framework [40].

Regarding middleware systems and messaging protocols, a comparative analysis of relevant IoT solutions (Constrained Application Protocol (CoAP), Message Queuing Telemetry Transport (MQTT), and Devices Profile for Web Services (DPWS)) was conducted in Fysarakis et al. [42]. DPWS [43], by the Organization for the Advancement of Structured Information Standards (OASIS: https://www.oasis-open.org/), constitutes the benchmark in terms of ease-of-design. The framework is flexible and robust in terms of service eventing, discovery, and subscription; following initialization, the underlying devices can discover the provided services and communicate in a seamless manner.

Finally, concerning the deductive rule engines that enable the AI reasoning features, National Aeronautics and Space Administration (NASA) examines the capabilities of several related approaches (Jess, Drools, Microsoft Business Rule Engine, Official Production System Java (OPSJ), and Intelligence Logiciel (ILOG)), as described in [44,45]. Jess is efficient and excels in many categories. It works with dynamic facts and dynamism in variables and rules and is appropriate for NASA's mission-critical applications as well as many other research areas [46,47].

2.2. Intelligent Railway Systems

To the best of the authors' knowledge, only a few multi-agent systems (MAS) have been developed and tested on actual railway environments [48,49]. In addition, despite strong industrial involvement, not all developed agent capabilities are fully used in practice and thus, the full potential of agent technologies is not exploited fully. Three indicative installations on railway settings are: Train Integrity (TrainIntegrity) [50], Condition Monitoring of a Light Rail Vehicle and Track (CMLRVT) [51], and Sensor Networks for Railways (SENSORAIL) [52].

TrainIntegrity utilizes WSNs to check the integrity of cargo trains [50]. The nodes consist of the RCM 3400 RabbitCore module and sense environmental parameters. The WSN raises an alarm if it infers that an unexpected change has occurred in the train's composition. CMLRVT is built and tested on a tramway operation in Poland. The system consists of a dispersed sensor network that is installed on the vehicles and railway infrastructure, along with the data acquisition component and a data server that maintains the artifacts of the management and analysis procedures. At first, the system collects data during

normal operation, which is stored in the server. Then, the new pieces of knowledge that are sensed by the devices are compared with the nominal values. The detected variations are further analyzed by the system, revealing safety-related incidents (e.g., rail cracks) that are presented to the user through a dedicated application. However, neither of the two systems considers security issues at the network link, nor do they integrate and manage the heterogeneous underlying embedded systems.

SENSORAIL [52] is an early warning system for the railway monitoring infrastructure. WSNs collect and integrate data to enable the detection of structural failures and security threats. Sensor clusters communicate information towards distant controlling centers through GSM-R/GPRS mobile equipment. The integration of heterogeneous sensors is managed by a component referred to as "scalable software architecture for the integration of heterogeneous sensor systems" (SeNsIM) [53], while the detection of events is made by a model-based data correlation component, called "novel framework for the detection of attacks to critical infrastructure" (DETECT) [54]. SENSORAIL specifies the examined threats in the Event Description Language (EDL) [55] and maintains them within a scenario repository. Upcoming events are stored in a history database and model-checking is performed at runtime. Regarding the middleware and agent platform, SeNsIM does not utilize any semantic technologies and does not support related standards, thus lacking in terms of interoperability and ease of integration with existing setups. Moreover, SENSORAIL does not include any protection mechanisms, solely focusing on the detection of threats.

2.3. Network Layer Protection

Several schemes are suggested in the literature for protecting communication in WSNs, attempting to address pertinent security concerns (e.g., [56–59]).

The Reputation-based Framework for Sensor Networks (RFSN) [56] authenticates underlying nodes with the Timed Efficient Stream Less Tolerant Authentication protocol (μTESLA) [60], implementing a beta Bayesian formulation for fading and evaluating the reputation of the routing operation and the legitimacy of the reported sensed variables. Ariadne [57] also utilizes a TESLA variant for authentication, collecting feedback regarding the successful delivery of packets to choose optimum communication paths and avoid malicious behavior. The Cooperative Secure routing protocol based on ARAN (CSRAN) [58] integrates digital certificates and asymmetric cryptography for authentication. As in the case of RFSN, it uses a Bayesian distribution for fading and, when a node detects malicious activity, it automatically re-routes communication from that point on. The Secure Resilient Reputation-based Routing (SR3) [59] adopts lightweight cryptography (LWC) and symmetric modules for security and authentication. Fading is accomplished by a First In, First Out (FIFO) finite list. The system combines reputation with a reinforced random walk algorithm, producing enhanced load-balancing at the cost of a high intermediate forwarding node count.

Despite the plethora of proposed solutions, and while most can tackle basic security attacks and malfunctioning cases, there are still various open avenues for attackers, including flooding attacks in congested periods, topology-related attacks, and jamming [61,62].

3. Administration of IoT Deployments

Considering the landscape sketched above, this section presents the proposed SPD-Safe solution, and more specifically the deployed platform and the reasoning process of each SPD-Safe agent. The core reasoning engine has been previously presented by the authors in [63]. This version enhances the network layer security and is applied in a mobile setting, forming an intelligent transportation system that complies with the real-time requirements of CBTC for railways.

SPD-Safe comprises a framework that integrates variants of the aforementioned primitives (i.e., agents, middleware, rule engine, and network layer protection) across all system layers to implement an efficient, scalable, practical, and easy-to-deploy and maintain solution, with adequate reasoning and management capabilities. From top-to-bottom, the

system consists of four layers: (i) An overlay with intelligent agents that control distinct subsystems; (ii) a middleware platform that enables communication between the agents and underlying networks; (iii) the network layer that consists of interconnected IoT devices; and (iv) the node layer that represents the devices themselves. The core technological building blocks will be detailed in the subsections that follow.

3.1. Agent Technologies & Middleware Solutions

SPD-Safe utilizes JADE [64] as the top-layer multi-agent system. It adopts and implements standardized approaches to agent deployment, such as the ACL [41] by FIPA. The JADE-S add-on [65,66] safeguards communication at the overlay and offers built-in security functionality for confidentiality, integrity, authentication, and authorization.

Then, each agent is ported as a bundle in the middleware platform Open Service Gateway initiative (OSGi) [67]. Through it, an agent can monitor the underlying subsystem, enhancing real-time management. Network gateways also deploy a controlling bundle in the same platform, defining the offered functionality as a service in the DPWS standard [43]. The agent and the related network bundles interchange well-structured semantic data, as defined in the OASIS standardized Common Alerting Protocol (CAP) [68]. The OSGi platform also provides its own built-in security features for the inner-platform communication, limiting bundle functionality to pre-defined capabilities and protecting both the agent and controller bundles.

Here, other than these built-in features that are provided by the deployed platforms at the overlay and middleware layers, SPD-Safe integrates an additional defense mechanism for the network and node layers, namely Secure Route (SecRoute) [61], a security protocol that protects the wireless ad hoc communication of the underlying embedded devices. This protocol counters several types of threats and attacks at the network link, protects the nodes' assets and their resource consumption, and acts as an intrusion detection module for the upper layers. When a security incident is recorded, the network gateway bundle will send related CAP messages to the responsible agent, which may take further action. SecRoute is detailed in the next section.

Metrics that evaluate the various system aspects are now an integral feature of the development cycle. They offer a quantitative indication regarding the compliance with the targeted requirements of the application domain. An evaluation method for the estimation of the security, privacy, and dependability (SPD) properties for configurable embedded systems is presented by the authors in Hatzivasilis et al. [63]. For every configuration option, the metrics derive a triple vector of <Security, Privacy, Dependability>, whereby the vector's factors are assigned a value from 0–100, representing no to full protection respectively. SPD-Safe adopts this methodology to enable a metric-driven SPD- and safety-aware administration, where the reasoning procedure triggers runtime system adaptations to reach specific SPD goals [69].

3.2. Reasoning Capabilities & Conflict Resolution

The artificial intelligence (AI) behavior of each agent is developed in the rule engine Jess [70,71]. For knowledge representation and reasoning, the Jess-EC [72] is used. The latter is an Event Calculus (EC) [73] implementation in Jess, offering the required semantics modeling. SPD-Safe's software layers are illustrated in Figure 1.

Each agents' AI procedure implements automated temporal, casual, and epistemic reasoning with real-time events, action preconditions, rule priorities, indirect effects, context-sensitive side-effects, as well as the common law of inertia. Moreover, the reasoning capabilities can cope with the requirements of dynamic and partially known or uncertain domains.

However, as agents exchange information, contradictory reasoning results may occur due to the local viewpoint of each entity and the lack of global knowledge. Thus, for resolving conflicts, SPD-Safe introduces the epistemic mechanism of share theories [63]. The participating entities send the involved theory rules to a mediator agent, along with

the recently sensed local events. The mediator combines these elements and performs a reasoning operation that determines the final outcome and the state of the conflicting assets.

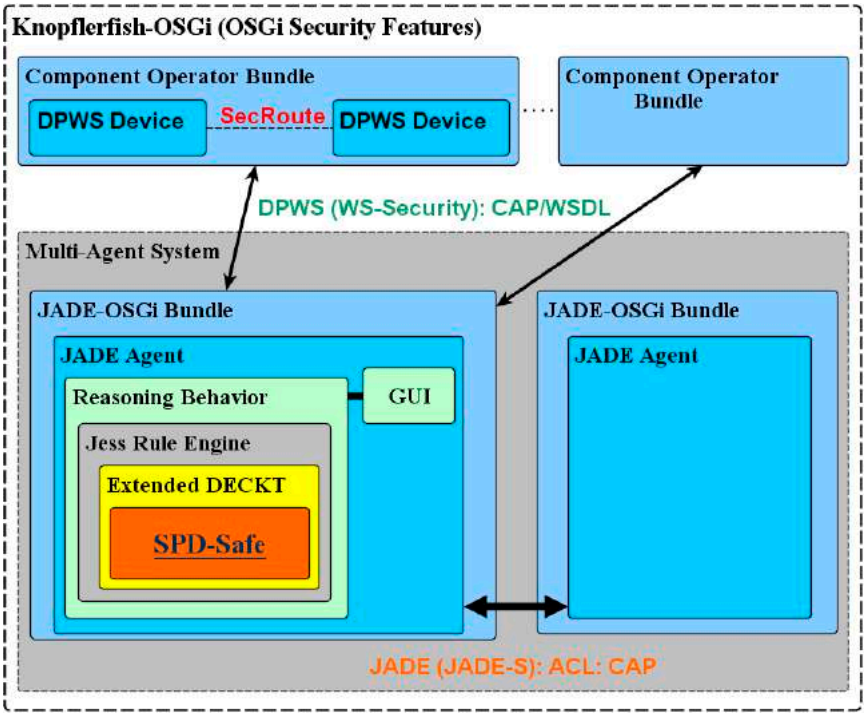

Figure 1. The software layers of the proposed security, privacy, and dependability (SPD)-Safe framework.

Nevertheless, if an agent utilizes protected data that must be maintained locally and not distributed (e.g., confidential information regarding user policies or system settings), it will not be able to contribute in the share theory with its full knowledge. For this occasion, an alternative relational grading mechanism, called certainty degree [63], resolves the affair quickly and efficiently. The mechanism utilizes subjective criteria as well as the agents' roles and hierarchy, marshaling the problem without constructing the related share theory and retaining the system's coherency. Thus, the certainty degree is applied in affairs where reasoning with locally protected data is involved, otherwise a share theory is constructed.

3.3. SPD Measurement

The SPD multi-metric methodology [69] measures the provided protection level of a system and its various configurations. The system's perimeter is identified and the data sources, entry, and exit points are recorded. Then, the mechanisms that protect each of these elements are assessed based on the standardized Criteria Evaluation Methodology (CEM) [69]. This involves the attack potential risk analysis that evaluates the attacker's motive to misuse specific system elements, expertise, and the resources that they are willing to devote for an attack. Henceforth, five parameters are examined for the analysis of a potential threat:

- Required time: The time that it is required to perform a specific attack (e.g., in days or weeks);
- Expertise: The technical skills and knowledge that the attacking group can exhibit (such as copy-cat, advanced, or expert);
- Knowledge of the target: Familiarity with the targeted system and its operation (e.g., public, sensitive, or critical information concerning some subsystems, etc.);
- Window of opportunity: The attacker may require appreciable access to the system in order to exploit a vulnerability and avoid detection;
- Resources: The software, hardware, or other equipment that is necessary to perform an attack (such as specialized or common resources).

The method does not investigate every possible attack but educes a good indication of the defense status in accordance with standard ratings. The protection level for each of the three SPD properties is calculated by integrating the risk analysis with the efficacy of the installed defenses against known attacks and/or other limitations (e.g., based on the latest reports from Computer Emergency Response Teams (CERTs) or Common Vulnerabilities and Exposures (CVE) repositories). The result is a value in the range of 0–100, where 0 represents the absence of defense mechanisms and 100 represents full protection. The final outcome is a vector of <Security, Privacy, Dependability>, which represents the total SPD value of the currently composed setting of the system. The SPDs of different system configurations can be estimated either in advance or at runtime. The first option is leveraged by the AI units of SPD-Safe in order to perform proactive and/or automated changes in the state architecture when a safety or security event occurs. The second option provides indications to the human operator in order to take decisions and make manual interventions.

Therefore, the protection status of all mechanisms and their integration in the demonstration examples are pre-calculated based on this method, as described in Sections 4 and 5. Then, automated administration policies are triggered in response to real-time events, as presented in Section 5.

These features enable the implementation of a relative novel protection strategy, called Moving Target Defenses (MTDs) [74]. When a system is stable, it is seen as a "sitting duck" by the attacker, who has plenty of time to analyze it, detect potential vulnerabilities, and exploit them. With MTD, a system that is aware of the defense level of its various components, their configurations, and the integration of all of them, can alter the setting automatically or semi-automatically in a periodic fashion. The AI modules are always keeping the system in a secure state, while the different configuration and architectural sets increase the system states that have to be analyzed by the attacker. In addition, the time that a specific setting remains active is determined by the time required for an average hacker to analyze it (i.e., based on the "Required Time" factor of the attack potential risk analysis). Performing attacks is becoming quite hard, while the window of opportunity for the malicious entities has significantly decreased.

3.4. AI Processing & Performance

The reasoning component of Jess implements the RETE algorithm (Latin word for net, meaning network in this domain) [75]. This is the most widely-used pattern matching technique for rule-based systems and is optimized for speed. Scalability and performance are affected by the three factors of: (i) The rules' volume (R), (ii) the average number of patterns in the left-hand-side of each rule (P), and (iii) the facts in the working memory (F). Computational complexity is linear to the working memory size and in the order of O (RPF). For each SPD agent in the railway mission-critical applications that are examined in the following sections, the theory rules volume (R) is very low (around 30 rules per scenario). In order to reduce the pattern-matching space, unique identifiers are assigned to every modeled entity, and therefore, occurring events affect specifically defined parameters, keeping the pattern-matching ration low (P) and in the order of 1–3. Performance is mostly influenced by the number of facts (F). In the demonstrated cases, it requires 10–20 facts per scenario. The computational overhead for an SPD agent is in the range of a nanosecond with additionally 50 bytes in memory.

For a central agent that collects information from the whole railway system, it requires around 500 facts and 40 rules to model the underlying setting. At boot time, the reasoning engine takes to run around 1.6 s, 87 MB for code, and 45 MB in RAM. Then, a reasoning process for a theory and a few hundreds of facts would require 0.002 s on average, representing the actual delay that affects the applications.

3.5. Relevant Methodologies for Secure IoT Modeling

Over time, several solutions have been proposed that try to resolve the open issues of capturing the security posture of an IoT or other system and facilitate its administration [76–80]. Eby et al. [76] integrated the Simple Modeling Language for Embedded Systems (SMoLES) with the Security Model Analysis Language (SMAL) [76]. SMAL provides security extensions to the composition meta-model of the Domain Specific Modeling Language (DSML) [77] and can express access control policies for IoT applications. The resulting framework is called SMoLES Security (SMoLES-SEC). However, its reasoning capabilities are bounded due to the constrained expressiveness of the underlying SMAL. Furthermore, SMoLES-SEC cannot deduce which security characteristics hold after the compositions of two components or the final security status of the composed system.

Service Dependency Trees (SDTs) [78] support the verification of service secure composition in IoT ecosystems. The IoT devices/nodes construct their own SDT. For each provided service, the relevant SDT defines the potential external service nodes that the service is depending on. The nodes are also aware of all recursive SDTs for their composed services. Thus, secure service composition is performed by enabling integration only with SDTs where all paths and involved entities are trusted. On the other hand, creating a SDT for a real IoT application is not trivial, while trustworthiness and consistency in an actual complex and dynamic environment may be challenging.

Albanese et al. [79] utilize attack surface metrics in order to evaluate the security aspects of system and materialize MTDs strategies. This solution calculates the distance of the security surface of the various system states. The goal is to administrate responses against ongoing attacks as well as to deduce a system setting that exhibits specific desirable parameters. Techniques for assessing and reducing the cost for the defender are also included.

Savola and Sihvonen [80] propose a MTD approach based on a multi-metric-driven management framework. The overall solution has been applied in an e-health digital environment for chronic diseases [80], where three metric types are considered. Risk-driven security assurance and engineering metrics are defined at deployment-time to offer an early assessment on the deployed defense mechanisms and their effectiveness. Continuous security monitoring metrics are determined at operational-time, enabling the security correctness assessment, enhanced systematization, and traceability of the various product requirements and involved metrics. Thereupon, automated adaptive decision-making metrics are assigned at operational-time and accomplish a higher quality security effectiveness understanding in operational security auditing and future versioning of the system. The method supports continuous security monitoring and automated metric-driven security-related actions.

Table 1 presents the outcomes of the qualitative analysis. The modeling expressiveness of SPD-Safe is quite general and can also be utilized in complex and dynamic systems. Moreover, it assesses all three security, privacy, and dependability properties and can evaluate their status both before a composition is performed and after the integration of the system. As with the other relevant approaches, the MTD features are driven by metrics and SPD-Safe provides a concrete implementation of this modern defense type. The overall solution fits with the distributed nature of IoT ecosystems and can resolve conflicts that may arise due to knowledge sharing between the various entities.

Table 1. AI (artificial intelligence) modeling features.

Feature	SPD-Safe [This Paper]	SMoLES-SEC [76]	SDT [78]	Attack Surface MTD [79]	Multi-Metric-Driven MTD [80]
System composition					
Expressiveness generality	Y	N	Y	N	N
Dynamicity	Y	Y	Y	N	N

Table 1. *Cont.*

Feature	SPD-Safe [This Paper]	SMoLES-SEC [76]	SDT [78]	Attack Surface MTD [79]	Multi-Metric-Driven MTD [80]
			Validation		
Pre-composition	Y	Y	Y	N	N
Post-composition	Y	N	N	N	N
			Evaluated Properties		
Security	Y	Y	Y	Y	Y
Privacy	Y	N	N	N	N
Dependability	Y	P	P	N	P
			Artificial Intelligence		
Distributed reasoning/processing	Y	N	Y	N	Y
Conflict resolution	Y	N	N	N	N
MTD	Y	N	N	Y	Y

Y(es), N(o), P(atrial). Service Dependency Trees (SDTs); Moving Target Defenses (MTDs); Simple Modeling Language for Embedded Systems Security (SMoLES-SEC).

4. Network Layer Security

The protection of the network link is essential in order to safeguard the underlying systems of critical railway infrastructure (e.g., WSN, signaling equipment, surveillance, etc.). For this purpose, as mentioned, SecRoute [61] is developed; a novel defence primitive that provides the core security properties for authentication, integrity, and confidentiality, along with energy-aware secure routing and authorization.

The secure routing protocol protects the involved entities from malicious operations while improving performance and offering load-balancing. It consists of three main primitives:

- The cryptographic service with the Timed Efficient Stream Less Tolerant Authentication protocol (μTESLA) [60], which implements message authentication, confidentiality, and integrity;
- The efficient secure routing service with the Self-Channel Observation Trust and Reputation System (SCOTRES) [62] that safeguards the communication link against ad-hoc routing attacks and network layer vulnerabilities;
- The authorization service with the Policy-Based Access Control framework (PBAC) [42], which offers authorization and access control based on policies.

Table 2 summarizes the overall security properties that are provided by the integrated network layer defense mechanism and the relevant threats and attacks that are countered, while a brief analysis is presented in the subsections that follow. More details regarding the three services are presented in the relevant papers for μTELSA [60], SCOTRES [62], and PBAC [42], respectively.

Figure 2 presents the block diagram of the main SPD-Safe modules and their connection.

Table 2. Protection aspects of the SPD-Safe's network layer security.

Primitive	System Property	Countered Threats
μTESLA	Authentication	Impersonation, Sybil attacks
	Integrity	Data tampering, modification, interruption
	Forward security	Replay attacks
	Confidentiality (optional)	Disclosure

Table 2. *Cont.*

Primitive	System Property	Countered Threats
SCOTRES	Topology-awareness	Attacks on topology-significant entities
	Energy-awareness & Load-balancing	Energy dissipation, overloading attacks on congested periods
	Channel health	Jamming attacks
	Reputation	Malicious or selfish activity on the network operations of: - Routing (link spoofing, routing table poisoning, HELLO flooding, false link break, loops, nonexistent paths), - Forwarding (blackhole, grayhole, sleep deprivation), - Or making recommendations (badmouth, ballot-stuffing)
	Trust	Overall misbehavior on the previously mentioned networking perspectives
	Secure routing	General attacks on the pure routing protocol (e.g., Denial of Service (DoS), inject arbitrary packets)
PBAC	Authorization based on policies	Unauthorized access

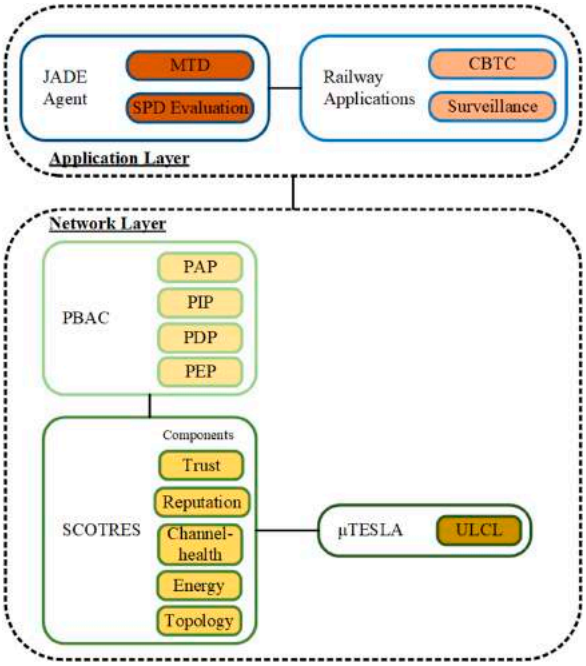

Figure 2. The building blocks of the SPD-Safe framework.

4.1. Cryptographic Service—μTESLAs

μTELSA is a building-block for the Sensor Protocols for Information via Negotiation (SPIN) [81]. Loose time synchronization is required between the receiver and sender, with μTESLA utilizing broadcast messages and symmetric cryptography to implement the aforementioned core cryptographic properties. The security functionality of asymmetric cryptography is achieved by utilizing keyed Message Authentication Code (MAC) operations. In brief, the sender includes a keyed MAC on every transmitted packet, where this key is initially known only to this entity. Receivers maintain the received packets without authenticating the sender at this point. Shortly after, the key is revealed by the sender and then the receiver authenticates the packet and proceeds to further processing. Otherwise, the receiver discards the unauthenticated packets after a time-slot.

The protocol μTESLA is efficient and exhibits low computational and communicational overheads. It also tolerates packet loss and scales well for large networks. We use the Ultra-Lightweight Cryptographic Library (ULCL) [61] in order to develop the cryptographic functionality of μTESLA, adopting the Secure Hash Algorithm (SHA) with 256-bits message digest (SHA-256) for the MAC computations and the Advanced Encryption Standard (AES) with 256-bits cryptographic keys (AES-256) for the encryption/decryption.

4.2. Secure Routing Service—SCOTRES

After authenticating a package with μTESLA, SCOTRES evaluates the sender's trustworthiness and its contribution to the network [62]. SCOTRES is a secure routing system for wireless ad-hoc systems that is based on trust computing and is designed around the intricacies of CPS solutions. It maximizes the information that is inferred regarding the network state, based on the knowledge that a node already processes. It safeguards communication against Internet-originating attacks or compromised equipment and jammers. The overall setting is utilized for real-time monitoring of IoT and CPS applications and their management through the cloud.

SCOTRES consists of five components that rate different aspects of the networking operation: (i) The topology-aware component improves the traffic load-balancing and defends distant entities from being isolated; (ii) the energy-aware component estimates the remaining energy of each node, defending the network against energy dissipation and other relevant threats; (iii) the channel-health component identifies jamming in the wireless medium, constraining its effects by routing communication through unaffected paths; (iv) the reputation component ranks a node's fair use of the network resources for routing, forwarding, and recommending activities; and (v) finally, the trust component aggregates all these pieces of knowledge and evaluates the trustworthiness and overall cooperativeness of network entities. Performance and security analyses for the five components have been conducted in [62].

4.3. Authorization Service—PBAC

After verifying the message's legitimacy, the receiver node must decide if it will perform the requested action or not. The PBAC framework is used to implement this authorization functionality. The framework manages direct access to a smart device's resources as determined by a pre-defined collection of policies and rules that are modeled on the OASIS standards DPWS [43] and the eXtensible Access control Markup Language (XACML) [82]. PBAC consists of four components that are placed between the backend infrastructure and devices: The Policy Administrator Point (PAP) and Policy Information Point (PIP) that maintain the attribute values for creating and managing policies in a central repository, the Policy Decision Point (PDP) that runs on a trusted gateway node with sufficient computational capabilities, evaluates the request, and renders the authorization decision, and the Policy Enforcement Point (PEP) that enforces authorization at the end device and makes decision requests. These are combined to provide fine-grained, policy-based access control on assets from remote endpoints (like control stations, sensors, or cameras). Therefore, the specification of an active policy set can be used to define the

rights to access to acquired resources (e.g., sensed data and video/audio streams), the rights to update the settings, and even the rights to push notifications of emergency alerts (e.g., blocked routes and train crashes).

4.4. Performance Evaluation

To assess the performance and validate the feasibility of the proposed approach, SecRoute is deployed on an embedded system which features BeagleBone (http://beagleboard.org/bone) devices and is integrated with the Distance Source Routing (DSR) protocol (DSR Uppsala University: http://dsruu.sourceforge.net/). BeagleBone is a low-cost and credit-card-sized device with ARM architecture, executing compact Linux operating systems (ARM Cortex-A8 processor at 720 MHz, 256 MB RAM, Ubuntu Linux). The devices sense environmental conditions, like humidity and temperature, and exchange data wirelessly with a central processing unit via a USB-WiFi.

We measure the processing overhead for SecRoute under normal operation without attacks taking place. Table 3 details the resource consumption of the proposed network layer defense. As indicated in the results, the calculation of reputation is the most compute-intensive part, as it maintains a history with previous interactions, which also increases the overall resource demands for trust computations. The requirements of authentication, routing, forwarding, as well as policy check are low. For the end-to-end interaction, the network latency is also low, ranging between 0.2–0.6 s on average. In an in-carriage setting where the distance among the nodes is short (a few meters) the transmission overhead is minimal, while the maximum delay is recorded for outdoor deployment where the nodes are placed hundreds of meters away from each other. The two scenarios are detailed in the next section.

Table 3. Resource allocation for SecRoute on BeagleBone devices.

Component	ROM (KB)	RAM (KB)	CPU (ms)
Cryptographic service			
Authentication	3.7	2.22	0.0020
Encryption	25.0	10.41	0.0028
Authenticated encryption	28.7	12.63	0.0048
Secure routing service			
Direct trust	5.6	4899.00	677.52
Reputation evaluation	20.0	1621.00	108.97
Indirect trust (recommendations)	30.0	185.00	37.90
Total trust	2.9	45,756.00	9.48
Accept route request	2.0	0.00	104.23
Suitable route selection	15.0	40.00	33.17
Authorization service			
PBAC policy check	24.7	36.00	7.50
Total resource consumption			
Total SecRoute	210.4	46,000.00	1652.50
DSR	310.0	90,000.00	2300.00
SecRoute_DSR	520.4	136,000.00	3952.50

4.5. Comparison with Other Protocols

Efficiency and security analysis of the proposed network layer solution and five relevant systems (RFSN, Ariadne, CSRAN, and SR3) have been presented by the authors in [62]. Table 4 summarizes the main features of the examined secure routing protocols.

Table 4. Secure routing protocols.

Property	SecRoute	RFSN	Ariadne	CSRAN	SR3
Authentication	μTESLA	μTESLA	TESLA	Certificates	LWC
Routing method	DSR	DSR	DSR	ARAN	Random walk
Reputation Fading	Bayesian	Bayesian	NO	Bayesian	FIFO *
Load-balancing	YES	NO	NO	NO	Partially
Energy-aware	YES	NO	NO	NO	NO
Anti-jamming	YES	NO	NO	NO	NO
Authorization	YES	NO	NO	NO	NO

* FIFO: First In First Out finite list.

The secure network communication link of SPD-Safe is compared with the five most relevant proposals for protecting WSNs. Simulation analysis has been performed in the Network Simulator 2 (NS2: http://www.isi.edu/nsnam/ns/), analyzing the performance of each scheme and the provided protection level on a medium-size WSN with 50 nodes [62]. Four attack cases are considered for blackhole, ballot-based attacks for link-spoofing, topology- and energy-aware attacks, and jamming. For each setting, several experiments have been conducted, with the attackers' participation in the network ranging from 10–50%. Figure 3 presents the evaluation of the simulation results. SecRoute counters the attacks and outperforms the relevant schemes, providing the highest level of security and demonstrating the best energy- and load-balancing characteristics.

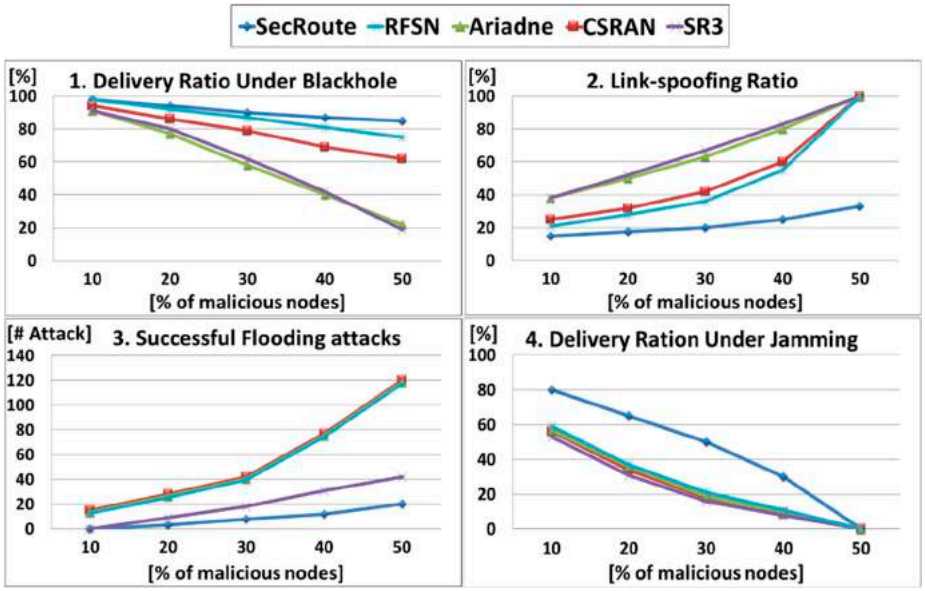

Figure 3. Simulation results for the evaluation of the network layer security solutions against four attack scenarios.

5. SPD-Safe Demonstration

5.1. Railway CPS Architecture

This section details the demonstration and evaluation of the whole SPD-Safe framework in the context of protecting and managing a railway CPS. In the proof-of-concept setting, our proposal assesses and manages the system and ambient ecosystem with the goal of safeguarding the trains' carriages and railway's routes. The hardware platforms incorporate embedded devices that control smart equipment (e.g., cameras and electronic doors), inspect environmental conditions, and exchange information wirelessly. Furthermore, the PBAC framework is applied for the control of the physical access for personnel, determined by access rights that are specified in XACML policies. Every agent manages a smart subsystem, like a train or a station. Backend agents can also run at the cloud in order to gather

high level information, perform big data analysis, and enable interaction with external systems and actuators. These agents run on virtual machines deployed on the research cloud platform GRNET Virtual MAchines (ViMA: http://vime.grnet.gr/about/info/en/). Figure 4 illustrates the railway system architecture. The whole setting is administered by a master agent (MA) at the C&C. At the edge, simple and more lightweight agents (SAs) protect the local subsystems (applying access control, lightweight data analysis, incident detection, etc.) and exchange information with the MA (i.e., security/safety events and response strategies). The MA can, optionally, forward data to a cloud SA for storing or in-depth analysis. The cloud SA also presents high-level knowledge to end-users as well as the current SPD status of the railway infrastructure.

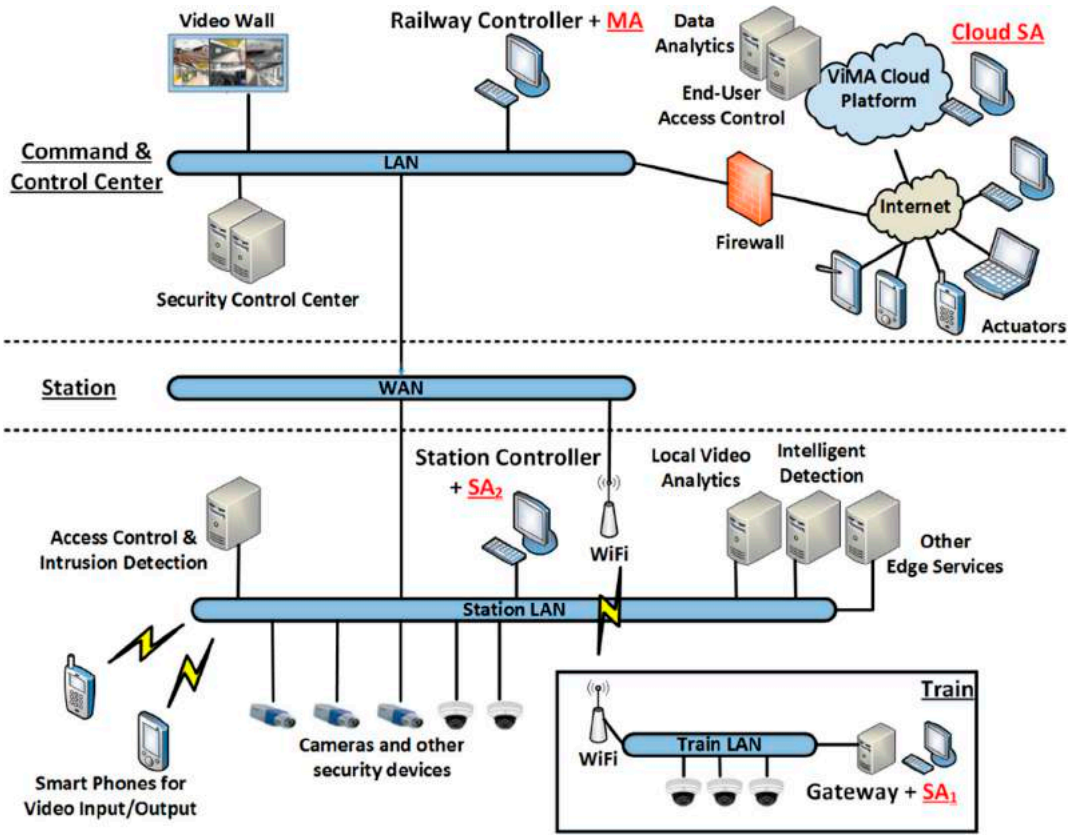

Figure 4. The smart railway use case architecture.

For this demonstration, the MA and the C&C services are deployed on a laptop. Both MA and cloud SA are installed on machines witha 2.1 GHz Intel Core i-7 processor, 8 GB of RAM, and the Ubuntu Linux Operating System (OS). The SAs are deployed on the BeagleBone devices at the edge systems.

As a case study, two deployments are evaluated. In the first indoor setting, which emulates in-carriage or shelter equipment, we test the system under normal operation and the aforementioned attacks on routing. In the second outdoor scenario, which emulates the on-route equipment, we examine the system's response to safety-related incidents. Both networks run the SecRoute protocol [61] to enable communication, protect the network layer against cyber-attacks, and act as an intrusion detection and incident response system for the upper layers.

5.2. Indoor Setting—Cyber-Security

The demonstration setting includes a carriage/shelter inspecting application, which is equipped with a surveillance system and WSNs. Those components are sensitive to

network layer threats, like jamming and blackhole attacks. The deployed network is depicted in Figure 5, where these devices are deployed in a shelter [28]:

- At the entrance, the smart camera inspects for physical intrusion;
- Two WSNs are deployed in the shelter. $WSN_{1\text{-}1}$ (green color) monitors light and temperature, and $WSN_{1\text{-}2}$ (red color) senses temperature. $WSN_{1\text{-}1}$ and $WSN_{1\text{-}2}$ utilize different hardware to enhance diversity and ensure redundancy for the monitored factors;
- A gateway interconnects the rest of the components with the C&C.

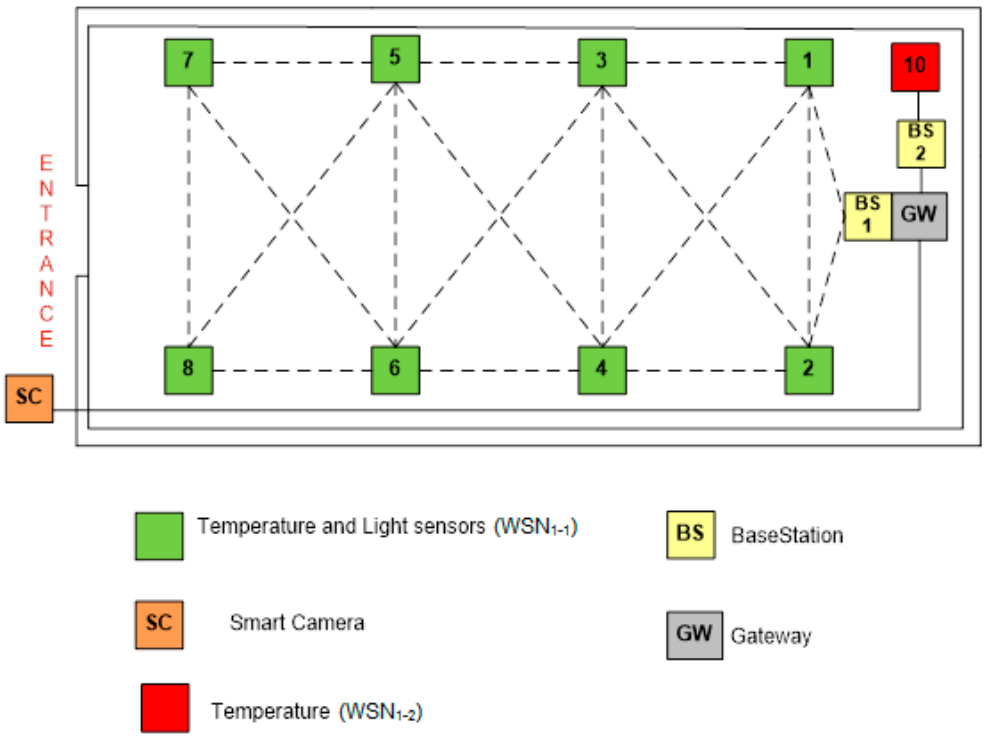

Figure 5. The internal wireless sensor network (WSN) for the carriage setting.

$WSN_{1\text{-}1}$ consists of eight Memsic Iris sensor nodes (16 MHz Atmel ATMega 1281 processor, 8 KB RAM, Contiki OS). The devices are battery powered and measure light and temperature. Furthermore, the smart camera is controlled by the node at the carriage's entrance. $WSN_{1\text{-}2}$ is installed for redundancy and is comprised of Zolertia Z1 sensor nodes (16 MHz MSP430 processor, 8 KB RAM, and Contiki OS) that collect temperature data. The two WSNs are monitored by two relevant simple agents ($SA_{1\text{-}1}$ and $SA_{1\text{-}2}$ respectively). Every device executes the PEP module of the PBAC framework. The devices also exchange data with the gateway, which runs the access policies for PBAC, and communicates with an MA which administrates the whole network.

The devices gather environmental information and send data to the relevant base station (laptop with WiFi connectivity). This component integrates and processes the received information. It also runs an application with which the user accesses and manages the overall testbed.

The different components are evaluated by the corresponding agents, who also estimate the aggregate SPD value of the whole system. The agents inspect their underlying domain, managing it based on an SPD-aware reasoning operation. Furthermore, the system is re-configurable at runtime according to the SPD protection and performance goals defined in the activated policy. Affected agents configure their subsystem's settings to raise the SPD value when attacks are performed and then return to normal when the attacks are over (to save resources). Regarding the adaptation capabilities integrated within the proof-of-concept, the cryptographic service provides three communication states: Plaintext, authenticated, as well as authenticated encryption. Additionally, the trust scheme

supports two trust evaluation states: Direct trust only, as well as a combination of direct and indirect trust.

The system begins with a moderate SPD configuration to conserve resources (i.e., authenticated communication and direct trust). If SPD-Safe observes malicious activity, it informs the system entities to raise their protection level. The relevant response actions are specified in a security policy (applicable to the specific device type), such as applying authenticated and encrypted communication with combined direct and indirect trust information. The SPD value and status of each system component is then altered as a response to the launched attacks, so as to achieve a sufficient level of protection. The WSNs comply with the current policies, becoming stricter to misbehavior and isolating the compromised nodes. The main protection mechanism against cyber-attacks (i.e., blackhole or link-spoofing) is provided by SecRoute, while the smart camera enhances physical protection. In the same way, the system returns to the previous (initial) state when the triggering conditions are over.

For $WSN_{1\text{-}1}$, we emulate scenarios where: (i) A node is malfunctioning due to low battery, and (ii) a compromised node launches a badmouth attack. In (i) the node is protected when a low energy level is observed by not including in traffic forwarding operations. The administrator gets notified accordingly. When the issue is fixed, the trust level is restored and the nodes' operational status returns back to normal. In case (ii), the compromised entity is detected when the attack rate reaches a threshold and it is blocked from routing operations. For $WSN_{1\text{-}2}$, we launch blackhole and jamming attacks against congested or topology significant components. The secure routing mechanism successfully identifies both attacks and mitigates them. Table 5 presents in detail the above-mentioned scenario phases. The SPD levels are depicted with: (i) Red for values of 0–50—i.e., a situation where the provided protection is low, the proper functionality may not be available, and the operator must take immediately the related countermeasures; (ii) yellow for values of 51–70—i.e., moderate protection but still safe operation; and (iii) green for values of 71–100—i.e., high levels of protection.

Table 5. Scenario steps of the smart transportation use case.

Event	Description	SPD State	Total <S, P, D> Value	SPD Visualization
1	Start of all components and services.Discovery/registration operations.	Initial State	<80, 70, 65>	
2	Bad-mouthing attack to $WSN_{1\text{-}1}$. MA (master agent) is alerted for the attack and commands the rest agents to increase security.	Security level decreases	<60, 70, 65>	
3	Security status is enhanced on all SAs (simple agents). MA is notified.	Security level increases	<85, 70, 65>	
4	$WSN_{1\text{-}1}$ counters bad-mouthing and $SA_{1\text{-}1}$ informs the MA.The MA requests from the SAs to restore the normal state (to conserve resources).	Security level returns to initial state	<80, 70, 65>	
5	Blackhole attack to $WSN_{1\text{-}2}$. MA is alerted for the attack and commands the rest agents to increase security.	Security level decreases	<50, 70, 65>	
6	Security status is enhanced on all SAs. MA is notified.	Security level increases	<85, 70, 65>	

Table 5. *Cont.*

Event	Description	SPD State	Total <S, P, D> Value	SPD Visualization
7	$WSN_{1\text{-}2}$ counters the blackhole attack and $SA_{1\text{-}2}$ informs the MA. The MA requests from the SAs to restore the normal state (to conserve resources).	Security level returns to initial state	<80, 70, 65>	
8	A node has died in $WSN_{1\text{-}2}$. MA is informed.	Dependability level decreases	<80, 70, 30>	
9	The dead node is replaced by the personnel. Dependability is restored.$SA_{1\text{-}1}$ reports the new status to MA.	Dependability level returns to initial state	<80, 70, 65>	
10	Simulated jamming attack against the network layer of $WSN_{1\text{-}2}$. MA is informed.	S & D levels decrease	<40, 70, 40>	
11	The trust-based routing component counters the attack. $SA_{1\text{-}2}$ reports new state to MA.	S & D levels return to initial state	<80, 70, 65>	

5.3. Outdoor Setting—Safety Scenario

For outdoor on-route defense, a similar WSN with four BeagleBone nodes is installed. The nodes are connected with a mains power supply and control a smart camera as well as weather sensors. In the emulated use-case, the nodes and related SAs are deployed on: (i) The passenger's station; (ii) the track; (iii) the carriage departure, and; (iv) all bridges and tunnels along the track. Figure 6 illustrates the on-route WSN_2 [28] along with the central MA and underlying $SA_{2\text{-}1}$–$SA_{2\text{-}4}$.

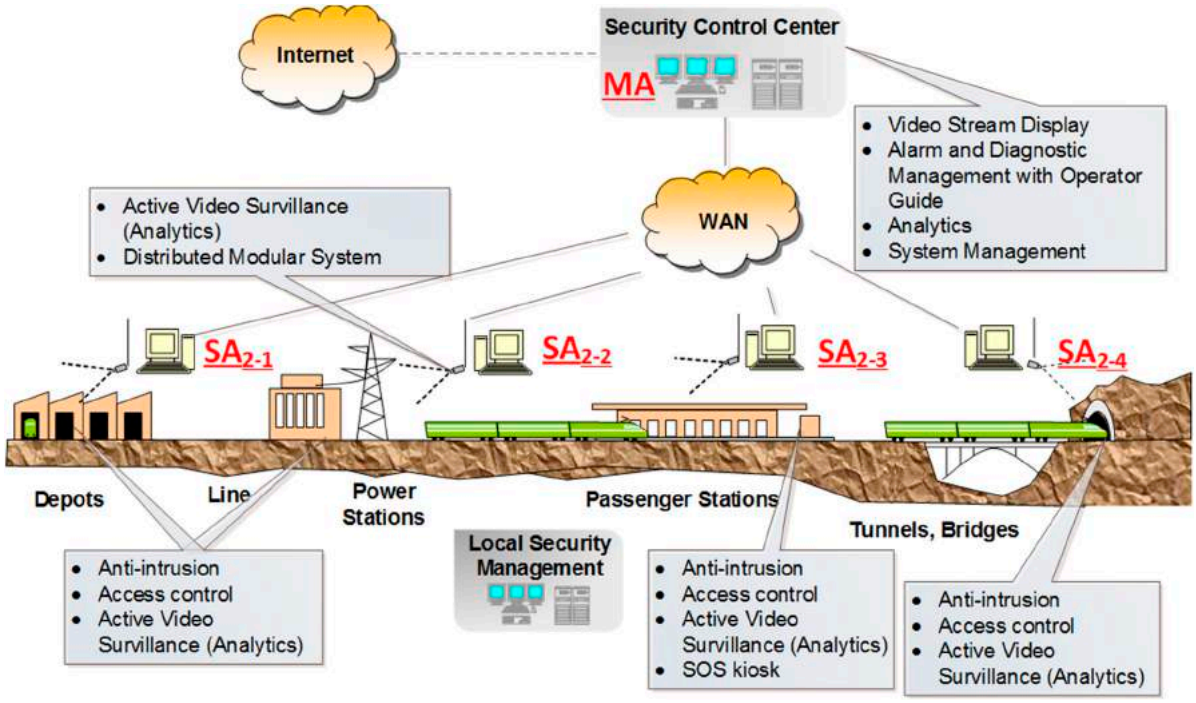

Figure 6. The outdoor WSN for the on-route scenario. The MA is deployed in the security control center and the four SAs are installed in the edge system.

Through the responsible SA, the networking components (e.g., sensors and cameras) send real-time information to a security control center and the related master agent. Figure 7 depicts the graphical user interface and the visualization of the information that is collected by the on-route equipment, as developed by Ansaldo STS.

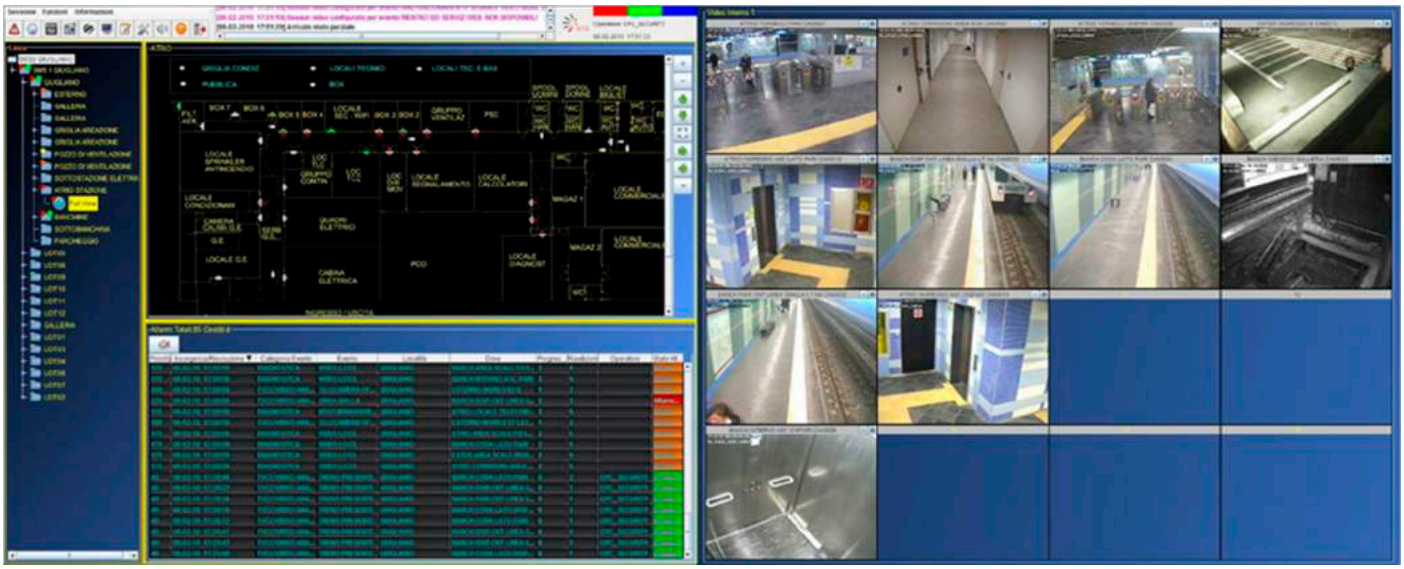

Figure 7. The railway on-route WSN graphical user interface.

In case of an emergency, the agents manage the system components to advise the personnel and assist the passengers. The demonstrated incident emulates the response strategy for a fire-alarm, where decisions concerning both safety and security must be taken. In Appendix A, the code sample Figure A1 describes the CAP message that indicates the fire alarm.

Normally, for the indoor setting, the personnel and passengers are allowed to open doors based on their access rights (as determined by safety and security rules). When fire is detected by the sensors, an alarm is triggered, and the associated agent is notified. The agent takes the decision to degrade the security status by unlocking all doors, therefore enabling the unhindered evacuation of the train. Furthermore, via GSM, the agent automatically transmits an SMS to the responsible authorities concerning this incident (including situation's severity, GPS coordinates) and alerts the neighboring entities to be aware (e.g., agents on nearby trains). The train agents that cross the area are also notified to perform related actions (such as stop to the nearest station or change route). Moreover, it is assumed that during normal operation the smart cameras capture frames at a low rate to preserve bandwidth. When the alarm is raised, the setting is reconfigured at runtime, offering a high framerate and continuous monitoring of the affected area. As the fire is extinguished and the damaged components are restored, the normal status is restored. The code shown in Figure A2 summarizes the main processing flow and the emergency response rules that perform the described actions (for more information regarding EC, please refer to Mueller [73]).

6. Discussion

6.1. Comparison

This subsection compares SPD-Safe with the related works presented in Section 2.2 (i.e., TrainIntegrity, CMLRVT, and SENSORAIL) in terms of features. Table 6 summarizes the comparison results.

Table 6. Smart railway systems.

Property	SDP-Safe	TrainIntegrity	CMLRVT	SENSORAIL
AI technologies	JADE/Jess	NO	NO	SeNsIM
Reasoning & processing	Distributed	Centralized	Centralized	Centralized
Conflict resolution	YES	NO	NO	NO
Security management	YES	NO	NO	NO
Safety management	YES	YES	YES	YES
Middleware	OSGi	NO	NO	NO
Network layer protection	SecRoute	NO	NO	NO
Cloud management	ViMA	NO	NO	NO

All the related smart railway systems identified adopt semantic representation and reasoning. The service-oriented approaches conform to the specific application aspects and, therefore, in all relevant systems the agents are uniquely responsible for specific operations. The conflicting patterns are also not examined in most of these designs, limiting their applicability to specific deployments.

Furthermore, the three related systems do not use any management middleware for embedded devices. This approach is quite limiting in the IoT era, where high volumes of heterogeneous equipment have to be deployed and co-function. The systems also neglect the popular agent frameworks which, among others, provide efficient agent-related functionality and implement relevant standards. The reasoning operation is developed with general purpose programming languages, ignoring the advantages offered by the deductive rule-based techniques. Mechanisms for resolving conflicts, when implemented, are based either on epistemic or relational reasoning. More importantly, these related systems do not safeguard security, privacy, and dependability, and do not utilize any built-in protection technologies.

Conversely, SPD-Safe is a solution focusing on the SPD management of IoT and CPS settings. The SPD modeling is based on well-structured metrics that analyze the various configuration options of a multi-layered system. The AI process adjusts the railway CPS and counters attacks at runtime. SPD-Safe integrates state-of-the-art technological building blocks and platforms for the implementation of reasoning, as well as the management of devices and agents. Epistemic and relational reasoning are incorporated for resolving conflicts. Furthermore, the proposed framework adopts standardized technologies, from semantic standards to communication protocols and authorization schemes.

6.2. Future Work

SPD-Safe integrates several technologies in a secure manner. It preserves the SPD properties, enables active defenses and countermeasures, and can facilitate emergency response operations.

Active and offensive types of defenses are proposing nowadays, as the next step to enhance protection and mitigate threats, that the mainstream passive mechanisms (e.g., cryptography, network slicing, anti-viruses, etc.) cannot tackle. MTD is such an approach. It is becoming harder to analyze the system and exploit its vulnerabilities. Furthermore, in conjunction with other intrusion detection techniques, it can mitigate or even block some type of ongoing attacks. Nevertheless, more research is needed in order to make guidelines for the implementation of effective MTD policies as well as strategies to mitigate more advance attacks.

Moreover, safety-related events require the participation of relevant authorities. In modern settings, emergency authorities possess their own equipment, which is utilized during safety incidents. The cooperation of the involved systems becomes vital when it comes to rescuing lives. For the effectiveness of the response services, the systems must authenticate and authorize the various participants and exchange information (e.g., sensors' data,

surveillance video, etc.) in real time. The seamless interoperation will be examined in future extensions of SPD-Safe.

7. Conclusions

This paper introduced SPD-Safe, an administration framework for IoT settings in ambient secure and safety-critical domains, applied to protect a railway CPS. For secure connectivity, an innovative secure routing protocol was integrated in the network layer. The protocol covers all core security properties (confidentiality, integrity, and authentication) and features policy-based authorization. It was found to be energy efficient and could effectively counter a variety of attacks, providing defense against several threats that are not mitigated by existing solutions. For smart monitoring and automatic adaptation, smart agents were deployed at the edge systems and backend infrastructure, and performed the required AI processes. A multi-agent system was developed in the JADE platform and integrated on the OSGi middleware for the management of DPWS-enabled equipment, also utilizing various built-in protection mechanisms. The core reasoning process was implemented in Event Calculus. The SPD validation and metric-driven administration were modeled as a heuristic framework in a security-related theory. The implementation of MTDs was enabled, providing extra protection against attacks that were not mitigated by passive defenses. Furthermore, the system models a safety-related theory and implemented associated AI ambient strategies and plans. The two features were incorporated to administrate the underlying components, considering both the SPD and safety aspects. To validate the proposed approach, SPD-Safe was deployed to administrate WSNs on a complex railway CPS testbed, where the underlying components were successfully configured at runtime and mitigate security-related attacks, while AI reactive plans preserved the safety of personnel and passengers in emergency situations. The average delay in real equipment was around 0.2–0.6 s.

In terms of future work, advances in MTD solutions and integration with emergency response services were considered. MTDs are coming to the foreground nowadays and are expected to play a significant role in future defense strategies as AI becomes an integral part of new generation systems. Safety critical systems, such as the railway ones, must provide an adequate means to collaborate with emergency authorities and support their operations. Facilitating emergency response and a rapid restoration of service must also be considered by modern smart railway installations.

Author Contributions: Conceptualization, G.H. and K.F.; methodology, G.H. and S.I.; software, G.H.; validation, K.F., N.P., G.V., and I.H.; writing—original draft preparation, G.H.; writing—review and editing, K.F. and S.I.; Supervision and Project administration, S.I. and G.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding from the European Union Horizon's 2020 research and innovation program under the grant agreements No. 786890 (THREAT-ARREST), No. 830927 (CONCORDIA), and No. 269317 (nSHIELD).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A

The code sample Figure A1 describes the CAP message that indicates the fire alarm.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Body>
  <ns3:Notify xmlns="http://docs.oasis-open.org/wsrf/bf-2"
   xmlns:ns2="http://www.w3.org/2005/08/addressing"
   xmlns:ns3="http://docs.oasis-open.org/wsn/b-2"
   xmlns:ns4="http://protectrail.eu/model/events/resource"
   xmlns:ns5="urn:oasis:names:tc:emergency:cap:1.2"
   xmlns:ns6="http://docs.oasis-open.org/wsn/t-1">
   <ns3:NotificationMessage>
    <ns3:Topic Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Full">FireDetection</ns3:Topic>
    <ns3:Message>
     <ns5:alert>
     <ns5:identifier>urn:rixf:com.tuc.SPD-Safe:id/FireAlert_01</ns5:identifier>
      <ns5:sender>WSN2</ns5:sender>
      <ns5:sent>2018-10-19T10:43:09.000+02:00</ns5:sent>
      <ns5:status>Actual</ns5:status>
      <ns5:msgType>Alert</ns5:msgType>
      <ns5:source>urn:rixf:com.tuc.fireprotection/devices/WSN</ns5:source>
       <ns5:scope>Public</ns5:scope>
       <ns5:info>
         <ns5:category>Fire</ns5:category>
         <ns5:event>FireDetection</ns5:event>
         <ns5:responseType>Evacuate</ns5:responseType>
         <ns5:urgency>Immediate</ns5:urgency>
         <ns5:severity>Extreme</ns5:severity>
         <ns5:certainty>Observed</ns5:certainty>
         <ns5:parameter>
           <ns5:valueName>Area</ns5:valueName>
           <ns5:value>51.468928,16.858863 0.01</ns5:value>
         </ns5:parameter>
        </ns5:info>
      </ns5:alert>
     </ns3:Message>
    </ns3:NotificationMessage>
   </ns3:Notify>
 </soap:Body>
</soap:Envelope>
```

Figure A1. Simple Object Access Protocol (SOAP) message that contains the CAP alert for the fire alarm.

The code shown in Figure A2 summarizes the main processing flow and the emergency response rules that perform the described actions (for more information regarding EC, please refer to Mueller [73]).

Rule 1: Happens(EventDitection(WSN,Fire),t) ∧ HoldsAt(ControllingAgent(SA,WSN),t)
=>
Happens(InformAgent(SA,WSN,Fire),t+1)

Rule 2: Happens(InformAgent(SA,WSN,Fire),t)
=>
Happens(InformTrainOperator(SA,SA_TrainOperator,Fire),t+1) ∧ Happens(UnlockDoors(SA, SA_Train),t+1) ∧
Happens(InformMA(SA,MA,Fire),t+1) ∧
Happens(InformNearbyAgents(SA,Fire),t+1)

Rule 3: Happens(InformNearbyAgents(SA_i,Fire),t) ∧ HoldsAt(LocatedIn(SA_i,area),t) ∧ HoldsAt(NearbyAgents(SA_i, SA_j),t)
=>
Happens(AvoidArea(SA_j, SA_i_TrainOperator, area),t+1)

Rule 4: Happens(InformMA(SA,MA,Fire),t)
=>
Happens(InformSystemOperator(MA,SystemOperator,Fire),t+1) ∧ Happens(InformAuthorities(MA,FireBrigate,Fire),t+1)

Figure A2. EC rules that perform the main safety reasoning behavior of each intelligent agent.

References

1. Xu, S.; Zhu, G.; Ai, B.; Zhong, Z. *A Survey on High-Speed Railway Communications: A Radio Resource Management Perspective. Computer Communications*; Elsevier: Amsterdam, The Netherlands, 2016; Volume 86, pp. 12–28.
2. Chamoso, P.; González-Briones, A.; Rodríguez, S.; Corchado, J.M. Tendencies of Technologies and Platforms in Smart Cities: A State-of-the-Art Review. *Wirel. Commun. Mobile Comput.* **2018**, *2018*, 3086854.
3. Boudi, Z.; El Koursi, E.M.; Ghazel, M. The New Challenges of Rail Security. *J. Traffic Logist. Eng.* **2016**, *4*, 56–60. [CrossRef]
4. Kour, R.; Thaduri, A.; Karim, R. Railway Defender Kill Chain to Predict and Detect Cyber-Attacks. *J. Cyber Secur. Mobil.* **2019**, *9*, 47–90. [CrossRef]
5. Luxton, A.; Marinov, M. Terrorist Threat Mitigation Strategies for the Railways. *Sustainability* **2020**, *12*, 3408. [CrossRef]
6. Zhang, J.; Hu, F.; Wang, S.; Dai, Y.; Wang, Y. Structural vulnerability and intervention of high speed railway networks. *Phys. A Stat. Mech. Appl.* **2016**, *462*, 743–751. [CrossRef]
7. González-Briones, A.; Garcia-Martin, R.; de AlbaJuan, F.L.; Corchado, M. Agent-Based Platform for Monitoring the Pressure Status of Fire Extinguishers in a Building. In *International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 1233, pp. 373–384.
8. Catalano, A.; Bruno, F.A.; Galliano, C.; Pisco, M.; Persiano, G.V.; Cutolo, A.; Cusano, A. An optical fiber intrusion detection system for railway security. *Sens. Actuators A Phys.* **2017**, *253*, 91–100. [CrossRef]
9. Fraga-Lamas, P.T.; Fernández-Caramés, M.; Castedo, L. Towards the Internet of Smart Trains: A Review on Industrial IoT-Connected Railways. *Sensors* **2017**, *17*, 1457. [CrossRef]
10. Wang, Y.; Zhu, L.; Yu, Z.; Guo, B. An adaptive track segmentation algorithm for a railway intrusion detection system. *Sensor* **2019**, *19*, 2594. [CrossRef]
11. Gai, K.; Qiu, M.; Hassan, H. Secure Cyber Incident Analytics Framework using Monte Carlo Simulations for Financial Cybersecurity Insurance in Cloud Computing. In *Concurrency and Computation: Practice and Experience*; Wiley: Hoboken, NJ, USA, 2017; Volume 29, issue 7.
12. Chang, S.E.; Liu, A.Y.; Lin, S. Exploring privacy and trust for employee monitoring. *Ind. Manag. Data Syst.* **2015**, *115*, 88–106. [CrossRef]
13. Paganini, P. Modern Railroad Systems Vulnerable to Cyber Attacks. Security Affairs. 2016. Available online: http://securityaffairs.co/wordpress/43196/hacking/railroad-systems-vulnerabilities.html (accessed on 18 November 2020).
14. Bababeik, M.; Khademi, N.; Chen, A.; Nasiri, M.M. Vulnerability analysis of railway networks in case of multi-link blockage. *Transp. Res. Procedia* **2017**, *22*, 275–284. [CrossRef]
15. Khanmohamadi, M.; Bagheri, M.; Khademi, N.; Ghannadpour, S.F. A security vulnerability analysis model for dangerous goods transportation by rail–Case study: Chlorine transportation in Texas-Illinois. *Saf. Sci.* **2018**, *110*, 230–241. [CrossRef]
16. Salmane, H.; Khoudour, L.; Ruichek, Y. A Video-Analysis-Based Railway–Road Safety System for Detecting Hazard Situations at Level Crossings. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 596–609. [CrossRef]
17. Chernov, A.V.; Savvas, I.K.; Butakova, M.A. Detection of Point Anomalies in Railway Intelligent Control System Using Fast Clustering Techniques. In Proceedings of the 3rd International Scientific Conference Intelligent Information Technologies for Industry, Sochi, Russia, 17–21 September 2018; pp. 267–276.
18. Coppola, P.; Silvestri, F. Assessing travelers' safety and security perception in railway stations. *Case Stud. Transp. Policy* **2020**, *8*, 1127–1136. [CrossRef]
19. Mrazovic, P.; Eser, E.; Ferhatosmanoglu, H.; Larriba-Pey, J.L.; Matskin, M. Multi-vehicle Route Planning for Efficient Urban Freight Transport. In Proceedings of the 2018 International Conference on Intelligent Systems (IS), Funchal, Madeira, Portugal, 25–27 September 2018; pp. 744–753.
20. Zhu, C.; Shu, L.; Leung, V.C.M.; Guo, S.; Zhang, Y.; Yang, L.T. Secure multimedia Big Data in trust-assisted sensor-cloud for smart city. *IEEE Commun. Mag.* **2017**, *55*, 24–30. [CrossRef]
21. Chamoso, P.; De La Prieta, F.; De Paz, F.; Corchado, J.M. Swarm Agent-Based Architecture Suitable for Internet of Things and Smartcities. In *Distributed Computing and Artificial Intelligence, 12th International Conference*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 373, pp. 21–29.
22. Zhang, Q.; Chen, Z.; Leng, Y. Distributed fuzzy c-means algorithms for big sensor data based on cloud computing. *Int. J. Sens. Networks* **2015**, *18*, 32–39. [CrossRef]
23. Tsaramirsis, G.; Karamitsos, I.; Apostolopoulos, C. Smart Parking: An IoT application for Smart City. In Proceedings of the 10th INDIACom-2016 International Conference, New Delhi, India, 16–18 March 2016; pp. 2271–2275.
24. Yin, W.; He, S.; Zhang, Y.; Hou, J. A Product-Focused, Cloud-Based Approach to Door-to-Door Railway Freight Design. *IEEE Access* **2018**, *6*, 20822–20836. [CrossRef]
25. Dong, Q.; Hayashi, K.; Kaneko, M. An Optimized Link Layer Design for Communication-Based Train Control Systems Using WLAN. *IEEE Access* **2017**, *6*, 6865–6877. [CrossRef]
26. Fanian, F.; Rafsanjani, M.K. Cluster-based routing protocols in wireless sensor networks: A survey based on methodology. *J. Netw. Comput. Appl.* **2019**, *142*, 111–142. [CrossRef]
27. Khanna, N.; Sachdeva, M. Study of trust-based mechanism and its component model in MANET: Current research state, issues, and future recommendation. *Int. J. Commun. Syst.* **2019**, *32*, 1–23. [CrossRef]
28. Cesena, M. SHIELD Technology Demonstrators. In *Measurable and Composable Security, Privacy, and Dependability for Cyberphysical Systems*; CRC Press: Boca Raton, FL, USA, 2017; pp. 381–434.

29. Brokalakis, A.; Tampouratzis, N.; Nikitakis, A.; Andrianakis, S.; Papaefstathiou, I.; Dollas, A. An Open-Source Extendable, Highly-Accurate and Security Aware CPS Simulator. In Proceedings of the 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS), Ottawa, ON, Canada, 5–7 June 2017; pp. 81–88.
30. Farooq, J.; Soler, J. Radio Communication for Communications-Based Train Control (CBTC): A Tutorial and Survey. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1377–1402. [CrossRef]
31. Sun, W.; Yu, F.R.; Tang, T.; Bu, B. Energy-Efficient Communication-Based Train Control Systems with Packet Delay and Loss. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 452–468. [CrossRef]
32. Garcia-Loygorri, J.M.; Val, I.; Arriola, A.; Briso-Rodriguez, C. 2.6 GHz Intra-Consist Channel Model for Train Control and Management Systems. *IEEE Access* **2017**, *5*, 23052–23059. [CrossRef]
33. Fotso, S.J.T.; Frappier, M.; Laleau, R.; Mammar, A. Modeling the Hybrid ERTMS/ETCS Level 3 Standard Using a Formal Requirements Engineering Approach. In *International Conference on Abstract State Machines*; Alloy, B., Ed.; Springer: Berlin/Heidelberg, Germany, 2018; pp. 262–276.
34. Chetty, K.; Chen, Q.; Woodbridge, K. Train monitoring using GSM-R based passive radar. In Proceedings of the 2016 IEEE Radar Conference (RadarConf), Philadelphia, PA, USA, 1–6 May 2016; pp. 1–4.
35. Bates, R.J. GPRS: General Packet Radio Service. In *Book GPRS: General Packet Radio Service*; McGraw-Hill, Professional Telecom: New York, NY, USA, 2001.
36. Proto, M.; Bavusi, M.; Bernini, R.; Bigagli, L.; Bost, M.; Bourquin, F.; Cottineau, L.-M.; Cuomo, V.; Della Vecchia, P.; Dolce, M.; et al. Transport Infrastructure Surveillance and Monitoring by Electromagnetic Sensing: The ISTIMES Project. *Sensors* **2010**, *10*, 10620–10639. [CrossRef] [PubMed]
37. Crinière, A.; Dumoulin, J.; Mevel, L.; Andrade-Barroso, G. Cloud2IR an Infrared and Environmental SHM Information System. In Proceedings of the 13th Quantitative Infrared Thermography Conference (QIRT), Gdansk, Poland, 4–8 July 2016; pp. 226–235.
38. Xie, J.; Liu, C.-C. Multi-agent systems and their applications. *J. Int. Counc. Electr. Eng.* **2017**, *7*, 188–197. [CrossRef]
39. De la Prieta, F.; Rodríguez-González, S.; Chamoso, P.; Corchado, J.M.; Bajo, J. Survey of agent-based cloud computing applications. *Future Gener. Comput. Syst.* **2019**, *100*, 223–236. [CrossRef]
40. Kravari, K.; Bassiliades, N. A Survey of Agent Platforms. *J. Artif. Soc. Soc. Simul.* **2015**, *18*, 11–29. [CrossRef]
41. FIPA, "FIPA ACL Message Structure Specification," Foundation for Intelligent Physical Agents. 2002. Available online: http://www.fipa.org/specs/fipa00061/SC00061G.html (accessed on 18 November 2020).
42. Fysarakis, K.; Askoxylakis, I.; Soultatos, O.; Papaefstathiou, I.; Manifavas, C.; Katos, V. Which IoT Protocol? Comparing Standardized Approaches over a Common M2M Application. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–6.
43. OASIS. "Devices Profile for Web Services Version 1.1," Organization for the Advancement of Structured Information Standards. 2009. Available online: http://docs.oasis-open.org/ws-dd/dpws/1.1/os/wsdd-dpws-1.1-spec-os.pdf (accessed on 18 November 2020).
44. Thirumalainambi, R. Pitfalls of Jess for dynamic systems. In Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition (AIPR), Orlando, FL, USA, 9–12 July 2007; Volume 1, pp. 491–494.
45. Kumar, S.; Prasad, R. Importance of expert system shell in development of expert system. *Int. J. Innov. Res. Dev.* **2015**, *4*, 128–133.
46. Semmel, G.; Davis, S.; Leucht, K.; Rowe, D.; Kelly, A.; Boloni, L. Launch commit criteria monitoring agent. In Proceedings of the 4th International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Utrecht, The Netherlands, 25–29 July 2005; pp. 3–10.
47. Goseva-Popstojanova, K.; Tyo, J. Experience Report: Security Vulnerability Profiles of Mission Critical Software: Empirical Analysis of Security Related Bug Reports. In Proceedings of the 28th International Symposium on Software Reliability Engineering (ISSRE), Toulouse, France, 23–26 October 2017; pp. 152–163.
48. Leitao, P.; Karnouskos, S. *Industrial Agents: Emerging Applications of Software Agents in Industry*, 1st ed.; Elsevier Science: Amsterdam, The Netherlands, 2015; pp. 1–476.
49. Ghadimi, P.; Wang, C.; Lim, M.K.; Heavey, C. Intelligent sustainable supplier selection using multi-agent technology: Theory and application for Industry 4.0 supply chains. *Comput. Ind. Eng.* **2019**, *127*, 588–600. [CrossRef]
50. Scholten, H.; Westenberg, R.; Schoemaker, M. Sensing Train Integrity. In Proceedings of the IEEE Sensors Conference, Christchurch, New Zealand, 25–28 October 2009.
51. Firlik, B.; Chudzikiewicz, A. Condition monitoring of a light rail vehicle—From concept to implementation. *Key Eng. Mater.* **2012**, *518*, 66–75. [CrossRef]
52. Flammini, F.; Gaglione, A.; Ottello, F.; Pappalardo, A.; Pragliola, C.; Tedesco, A. Towards wireless sensor networks for railway infrastructure monitoring. In Proceedings of the Electrical Systems for Aircraft, Railway and Ship Propulsion (ESARS), Bologna, Italy, 19–21 October 2010.

53. Casola, V.; Gaglione, A.; Mazzeo, A. A reference architecture for sensor networks integration and management. In Proceedings of the 3rd International Conference on Geosensor Networks, Oxford, UK, 13–14 July 2009; pp. 158–168.
54. Flammini, F.; Gaglione, A.; Mazzocca, N.; Pragliola, C. DETECT: A novel framework for the detection of attacks to critical infrastructures. In *Safety, Reliability and Risk Analysis: Theory, Methods and Applications*; Taylor & Francis: Abingdon, UK, 2008; pp. 105–112.
55. Chakravarthy, S.; Mishra, D. Snoop: An expressive event specification language for active databases. *Data Knowl. Eng.* **1994**, *14*, 1–26. [CrossRef]
56. Ganeriwal, S.; Balzano, L.; Srivastava, M. Reputation-based framework for high integrity sensor networks. *ACM Trans. Sen. Netw.* **2008**, *4*. [CrossRef]
57. Hu, Y.-C.; Perrig, A.; Johnson, D.B. Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wirel. Netw.* **2005**, *11*, 21–38. [CrossRef]
58. Zhang, Y.; Xu, L.; Wang, X. A Cooperative Secure Routing Protocol based on Reputation System for Ad Hoc Networks. *J. Commun.* **2008**, *3*, 43–50. [CrossRef]
59. Altisen, K.; Devismes, S.; Jamet, R.; Lafourcade, P. SR3: Secure resilient reputation-based routing. In Proceedings of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), Cambridge, MA, USA, 20–23 May 2013; pp. 258–265.
60. Dhaheri, A.A.; Yeum, C.Y.; Damiani, E. New Two-Level μTESLA Protocol for IoT Environments. In Proceedings of the 2019 IEEE World Congress on Services (SERVICES), Milan, Italy, 8–13 July 2019; pp. 84–91.
61. Hatzivasilis, G.; Papaefstathiou, I.; Askoxylakis, I.; Fysarakis, K. SecRoute: End-to-end secure communications for wireless ad-hoc networks. In Proceedings of the 22nd IEEE Symposium on Computers and Communications (ISCC), Heraklion, Crete, Greece, 3–6 July 2017; pp. 558–563.
62. Hatzivasilis, G.; Papaefstathiou, I.; Manifavas, C. SCOTRES: Secure Routing for IoT and CPS. *IEEE Internet Things J.* **2017**, *4*, 2129–2141. [CrossRef]
63. Hatzivasilis, G.; Papaefstathiou, I.; Plexousakis, D.; Manifavas, C.; Papadakis, N. AmbISPDM: Managing embedded systems in ambient environment and disaster mitigation planning. In *Applied Intelligence*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 1–21.
64. Java Agent Development (JADE) Framework. Available online: http://jade.tilab.com/ (accessed on 18 November 2020).
65. Tilab, S.P.A. JADE Security Add-On Guide. 2005. Available online: http://jade.tilab.com/doc/tutorials/JADE_Security.pdf (accessed on 18 November 2020).
66. Ali, B.; Manzoor, U.; Zafar, B. eJADE-S: Encrypted JADE-S for Securing Multi-Agent Applications. In Proceedings of the International Conference on Artificial Intelligence (ICAI), Athens, Greece, 27–30 July 2015; pp. 548–554.
67. Open Services Gateway Initiative (OSGi). Available online: http://www.osgi.org/ (accessed on 18 November 2020).
68. OASIS. Common Alerting Protocol Version 1.2, Organization for the Advancement of Structured Information Standards. 2010. Available online: http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.pdf (accessed on 18 November 2020).
69. Hatzivasilis, G.; Papadakis, N.; Hatzakis, I.; Ioannidis, S.; Vardakis, G. AI-driven composition and security validation of an IoT ecosystem. *Appl. Sci.* **2020**, *10*, 4862. [CrossRef]
70. Friedman-Hill, E.J. Jess: The Rule Engine for Java Platform. Sandia National Laboratories. 2008. Available online: http://www.jessrules.com/docs/71/ (accessed on 18 November 2020).
71. Lu, Y.; Li, Q.; Zhou, Z.; Deng, Y. Ontology-based knowledge modeling for automated construction safety checking. *Saf. Sci.* **2015**, *79*, 11–18. [CrossRef]
72. Patkos, T.; Plexousakis, D.; Chibani, A.; Amirat, Y. An event calculus production rule system for reasoning in dynamic and uncertain domains. In *Theory and Practice of Logic Programming*; Cambridge University Press: Cambridge, UK, 2016; Volume 16, pp. 325–352.
73. Mueller, E.T. *Commonsense Reasoning*, 2nd ed.; Kaufmann, M., Ed.; Elsevier: Amsterdam, The Netherlands, 2015; pp. 1–516.
74. Lei, C.; Zhang, H.-Q.; Tan, J.-L.; Zhang, Y.-C.; Liu, X. Moving Target Defense Techniques: A Survey. *Secur. Commun. Netw.* **2018**, *2018*, 1–25. [CrossRef]
75. Berstel, B. Extending the RETE algorithm for event management. In Proceedings of the 9th International Symposium on Temporal Representation and Reasoning, Manchester, UK, 7–9 July 2002; pp. 49–51.
76. Eby, M.; Werner, J.; Karsai, G.; Ledeczi, A. Integrating security modeling into embedded system design. In Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS), Tucson, AZ, USA, 26–29 March 2007; pp. 221–228.
77. Kelly, S.; Tolvanen, J.-P. *Domain-Specific Modeling: Enabling Full Code Generation*; Wiley-IEEE Computer Society Pr.: Hoboken, NJ, USA, 2008; pp. 1–444.
78. Ko, H.; Jin, J.; Keoh, S.L. Secure Service Virtualization in IoT by Dynamic Service Dependency Verification. *IEEE Internet Things J.* **2016**, *3*, 1006–1014. [CrossRef]
79. Albanese, M.; Battista, E.; Jajodia, S.; Casola, V. Manipulating the Attacker's View of a System's Attack Surface. In Proceedings of the IEEE Conference on Communications and Network Security, San Francisco, CA, USA, 29–31 October 2014; pp. 472–480.

80. Savola, R.M.; Sihvonen, M. Metrics driven security management framework for e-health digital ecosystem focusing on chronic diseases. In Proceedings of the MEDES '12: International Conference on Management of Emergent Digital EcoSystems, Addis Ababa, Ethiopia, 28–31 October 2012; pp. 75–79. [CrossRef]
81. Ayyappan, B.; Kumar, P.M. Security protocols in WSN: A survey. In Proceedings of the 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM), Chennai, India, 23–24 March 2017; pp. 301–304.
82. Parducci, B.; Lockhart, H. *eXtensible Access Control Markup Language (XACML) Version 3.0*; OASIS Standard: Burlington, MA, USA, 2013; pp. 1–154.