IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 October 2022, accepted 16 November 2022, date of publication 28 November 2022,
date of current version 2 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3225074

RESEARCH ARTICLE

Cybersecurity for Industrial Internet of Things: Architecture, Models and Lessons Learned

GEORGE BRAVOS[1], (Member, IEEE), ANTONIO J. CABRERA[2], CAMILO CORREA[3], DRAGAN DANILOVIĆ[4], NIKOLAOS EVANGELIOU[1], GILAD EZOV[5], ZORAN GAJICA[4], DUŠAN JAKOVETIĆ[6], (Member, IEEE), LEONIDAS KALLIPOLITIS[7], MILAN LUKIĆ[8], (Member, IEEE), JULIEN MASCOLO[9], DAVIDE MASERA[9], RAÚL MAZO[10], IVAN MEZEI[8], (Senior Member, IEEE), ANDREAS MIAOUDAKIS[11], NEMANJA MILOŠEVIĆ[6], WILLIAM OLIFF[12], JACQUES ROBIN[3,13], MICHAIL SMYRLIS[11], GEORGIA SAKELLARI[12], GIORGOS STAMATIS[1], DUŠAN STAMENKOVIĆ[6], SRĐAN ŠKRBIĆ[6], CARINE SOUVEYET[3], SPYRIDON VANTOLAS[7], GIORGOS VASILIADIS[14,15], AND DEJAN VUKOBRATOVIĆ[8], (Senior Member, IEEE)

[1]Information Technology for Market Leadership (ITML), Athens, 115 25 Athina, Greece
[2]Infineon Technologies AG (IFAG), 85579 Neubiberg, Germany
[3]University of Paris 1 Panthéon-Sorbonne (UP1PS), 75231 Paris, France
[4]A1, Serbia (VIP), 11070 Belgrade, Serbia
[5]IBM Research (IBM), Haifa 3498825, Israel
[6]Faculty of Sciences (UNSPMF), University of Novi Sad, 21000 Novi Sad, Serbia
[7]AEGIS IT Research (AEGIS), 38106 Braunschweig, Germany
[8]Faculty of Technical Sciences, University of Novi Sad, 21000 Novi Sad, Serbia
[9]Centro Ricerche Fiat (CRF), 10043 Orbassano, Italy
[10]ENSTA-Bretagne, 29200 Brest, France
[11]Sphynx Technology Solutions AG (STS), 6300 Zug, Switzerland
[12]School of Computing and Mathematical Sciences, University of Greenwich (UoG), SE10 9LS London, U.K.
[13]ESIEA, 94200 Ivry-sur-Seine, France
[14]Foundation for Research and Technology (FORTH), 70013 Heraklion, Greece
[15]Department of Management Science and Technology, Hellenic Mediterranean University, 731 33 Chania, Greece

Corresponding author: Nemanja Milošević (nmilosev@dmi.uns.ac.rs)

This work was supported by the H2020 Project C4IIoT Funded by the European Commission under Agreement 833828.

ABSTRACT Modern industrial systems now, more than ever, require secure and efficient ways of communication. The trend of making connected, smart architectures is beginning to show in various fields of the industry such as manufacturing and logistics. The number of IoT (Internet of Things) devices used in such systems is naturally increasing and industry leaders want to define business processes which are reliable, reproducible, and can be effortlessly monitored. With the rise in number of connected industrial systems, the number of used IoT devices also grows and with that some challenges arise. Cybersecurity in these types of systems is crucial for their wide adoption. Without safety in communication and threat detection and prevention techniques, it can be very difficult to use smart, connected systems in the industry setting. In this paper we describe two real-world examples of such systems while focusing on our architectural choices and lessons learned. We demonstrate our vision for implementing a connected industrial system with secure data flow and threat detection and mitigation strategies on real-world data and IoT devices. While our system is not an off-the-shelf product, our architecture design and results show advantages of using technologies such as Deep Learning for threat detection and Blockchain enhanced communication in industrial IoT systems and how these technologies can be implemented. We demonstrate empirical results of various components of our system and also the performance of our system as-a-whole.

INDEX TERMS Anomaly detection, blockchain, cybersecurity, deep learning, Internet of Things.

I. INTRODUCTION

Despite the fact that the IIoT (Industrial Internet of Things) has a profound impact on many industry domains, a major

The associate editor coordinating the review of this manuscript and approving it for publication was Stefano Scanzio.

barrier towards IIoT adoption lies in cybersecurity issues that make it extremely difficult to harness its full potential: IIoT systems dramatically increase the attack surface (introducing new security threats due to newly connected devices and protocols, making them more vulnerable to interference), the disruption of process controls, the theft of intellectual property, the loss of corporate data, and the industrial espionage. The C4IIoT project (Cyber security 4.0: Protecting the Industrial Internet of Things[1]) provides and demonstrates a novel and unified IIoT cybersecurity framework for malicious and anomalous behaviour anticipation, detection, mitigation, and end-user informing. The framework provides a holistic and disruptive security-enabling solution for minimizing attack potentials in IIoT systems so the industry can benefit from the advantages of having connected or smart system architectures without the inherent risk which these architectures may bring. The framework itself represents an example of how similar systems can be built, and it went through many development iterations with various changes and improvements. While many of the components of the framework are open source and available for general usage, we do not offer a singular ready to use framework, but rather share our findings and results in building the framework.

In this paper, we provide an overview of the end-to-end framework developed in the context of C4IIoT for ensuring cybersecurity in two targeted industrial IoT applications, Smart Logistics and Smart Factory applications. While the paper gives an overview of the overall system architecture, it focuses in more detail on the several aspects of the overall system. The paper describes in detail the system operation, real world deployment, evaluation, and the derived lessons learned and guidelines with respect to detecting anomalies in IIoT data, as well as response to a concrete relevant attack, the data manipulation attack, in the context of Smart Logistics and Smart Factory, for a real world industrial environment. Through the layers of the system architecture we will demonstrate how privacy-preserving and other techniques can be used to enhance the system so it can detect faults and malicious activity in a privacy aware manner. There is a limited number of works that describes end-to-end cybersecurity systems tailored for Smart Logistics (SL) and Smart Factory (SF) applications, and limited evaluation results have been reported on threat detection performance based on deep learning in real environments of SL and SF systems. We use those two use cases to demonstrate how IIoT devices can be used in a secure way and what benefits this brings to the overall system and the industrial end-user who may implement the overall system.

A. RELATED WORK

We next provide an overview of a representative set of works on Cybersecurity in IIoT systems. In this context, a relevant challenge is to provide protocols and mechanisms that ensure secure IIoT devices' communication. A related effort to this is the application of fully homomorphic encryption (FHE) to enable arithmetic operations over the encrypted data, The authors of [1] develop a FHE approach that supports blending of arithmetic operations over real numbers. IIoT devices' security can be also improved via effective authentication schemes. To this end, Li et al. [2] propose a privacy-preserving biometric-based authentication scheme. In the context of IIoT secure authentication and communication, blockchain is also becoming a trending research direction: there have been several approaches to integrate or leverage blockchains in ensuring IIoT security [3], [4], [5]. Lipps et. al. [6] present a Static Random Access Memory (SRAM) based hybrid cryptosystem as a solution to securing communication and authentication in the IIoT. Our proposed architecture is in line with the described trends; for example, it uses the blockchain technology in order to ensure a state-of-the-art secure IIoT device communication.

Machine Learning and Deep Learning algorithms have been applied in IoT systems for various tasks and across multiple branches of the industry [7], [8], [9], [10], [11], [12]. For the manufacturing sector, Zhang et al. [7] used LSTM (Long Short Term Memory Recurrent Neural Networks) to predict the working condition of industrial equipment, in order to enhance operation quality. Yan et al. [8] utilize autoencoders in order to determine remaining useful life in machines. The authors of [9] used CNNs (Convolutional Neural Networks) for classification of production items into the defected and non-defected classes. Deep, fully-connected neural networks (DNN) have been used to detect malicious traffic in IoT networks [10]. The work of Wang et al. [11] proposes the use off CNNs and LSTMs to learn features of network traffic which are then used to differentiate between good and malicious network traffic.

Since sensitive and confidential data are constantly being shared across the networks, a major prevailing concern in industrial IoT systems is on data protection issues [13]. In this setting, among the aforementioned Machine Learning (ML) and Deep Learning (DL) models and their applications, anomaly detection is of significant interest [12]. In the process of detection of anomalies in IoT-generated data, some authors focus on detecting faults in device operation or communication errors in a complex IoT environment like smart city [14], while others focus on detecting different types of security threats, such as device tampering, botnet, IoT pivot, malware analysis and distributed denial-of service (DDoS) attacks [15], [16].

Some of the DL-based models utilized for anomaly detection include, e.g., recurrent neural networks [17], autoencoders [12], [18], etc. A related challenge is on ensuring data privacy. In this context, differential privacy has been proposed that safeguards privacy of data by adding a controlled amount of random noise to the data [19]. We also consider differential privacy in our framework; as detailed ahead, we adopt differentially private variants of principal component analysis (PCA) and the KMeans

[1] Project Website – https://www.c4iiot.eu/

clustering algorithm for anomaly detection by making use of the diffprivlib library [20].

When deploying machine learning models in IIoT environments, algorithm scalability and IoT edge platforms resource limitations should be taken into account [21]. Indeed, processing and computational power of the underlying hardware can present itself as a major constraint, as in some cases, the deployed Machine learning models' size could be required to go as low as a few kilobytes, while the hardware usually has only a low-level CPU. More demanding machine learning models can be supported by following an edge-to-fog-to-cloud architecture design, e.g., [22]. Therein, the data generated by IoT devices gets communicated to fog servers (e.g., mobile operator gateways) and is subsequently further transferred to the cloud, where more powerful machine learning (anomaly detection) models are deployed. In order to shorten response times, anomaly detection in these systems can take place not only at the cloud, but also at the edge or fog. As edge devices usually have limited computational and storage capabilities, only low-to-moderate complexity models (with a potentially limited performance) can be deployed at the edge. More powerful models are then deployed at the fog or cloud, at the expense of longer response times.

Following an edge-fog architectural pattern, Savic et al. [12] propose to integrate Deep Learning based anomaly detection as a service into a mobile IoT communication architecture. The proposed architecture embeds autoencoder-based anomaly detection modules both at the IoT devices, and in the mobile core network. Thus, the presented method balances between the system responsiveness and models complexity versus accuracy. Our framework here also follows an edge-fog-cloud strategy, as we deploy various DL and ML models to detect anomalies at all three layers of the edge-to-fog-to-cloud architecture.

B. CONTRIBUTIONS AND INNOVATIONS

We now summarize our main contributions and contrast them with existing work. While existing studies usually focus on a specific and fragmented aspect of cybersecurity in IIoT (e.g., homomorphic encryption, deep learning application for a specific attack type, etc.), there is a limited body of literature that reports on design and deployment of end-to-end cybersecurity systems for a targeted industrial domain. We contribute to bridging this gap by providing design, deployment, and evaluation of a comprehensive end-to-end cybersecurity system for the targeted industrial application, namely the smart logistics and the smart factory use cases in the manufacturing context. We report here on the system architecture and its integrative parts' descriptions, and we illustrate and report on the system operation and deployment on real industrial data for 1) detecting anomalies in IIoT readings; and 2) generating response to data manipulation attacks. The development, evaluation, and deployment of the system has lead to a number of lessons learned that we describe in detail later.

The developed system features several innovations that we will now describe. To tackle cybersecurity flaws and data leakages, we employ modern, encrypted means of communication using blockchain for additional security (described in Section II-F), multiple levels of machine learning enabled anomaly detection models (described in Section II-N) trained to detect attacks and faults, mitigation engines (described in Section II-I and Section II-H) to provide a meaningful response (described in Section II-G and Section II-P) to these possible anomalies in minimal time. Another distinctive feature of the system is a flexible anomaly detection approach that adaptively triggers anomaly detection modules at different system layers (edge, field gateway, cloud), hence trading off accuracy and response times (Sections II-N and II-H).

Further, we use differential privacy methods (described in Section II-N1), so even in the case of malicious intrusions to the system, the integrity and privacy of the data provider is preserved. We also introduce both hardware-level and software-level security (described in Section II-M) for protecting the data sources and the system in general. Next, we design and fabricate custom IIoT edge devices tailored to the targeted use cases, and deploy them in a real industrial environment. (Sections II-L and II-K) Finally, we develop a privacy-preserving cloud-based malware analysis framework (described in II-O) that utilises trusted execution environments and is able to offload the analysis of suspicious files to the cloud confidentially.

C. PAPER ORGANIZATION

We provide here a paper road map and organization. Section II is devoted to the description of the developed end-to-end cybersecurity system (Figure 1). The end-to-end system, as detailed below (Figure 1), consists of three layers (edge, field gateway, cloud), and three levels of protection (hardware-enabled, device-to-device, and cognitive). Section II then proceeds as follows. Sections II-A to II-E provide the system overview, including the requirements and targeted use cases, and describe the three mentioned architecture layers. Sections II-F to II-L and II-O then detail the system in a component-by-component fashion. Subsections II-M and II-N describe the system components that correspond to three different levels. Section III is devoted to empirical evaluation and demonstration of the system. Specifically, we evaluate and describe communication performance III-A, anomaly detection performance III-B, and a system response example with respect to a data manipulation attack (Section III-C). Finally, section IV provides lessons learned that arose in the system development and evaluation.

In more detail, Section II represents the central part of the paper, and it is split into 14 subsections (refer to the architecture figure (Figure 1) for component roles and positioning, as required):

- II-A describes the system requirements and plausible threats

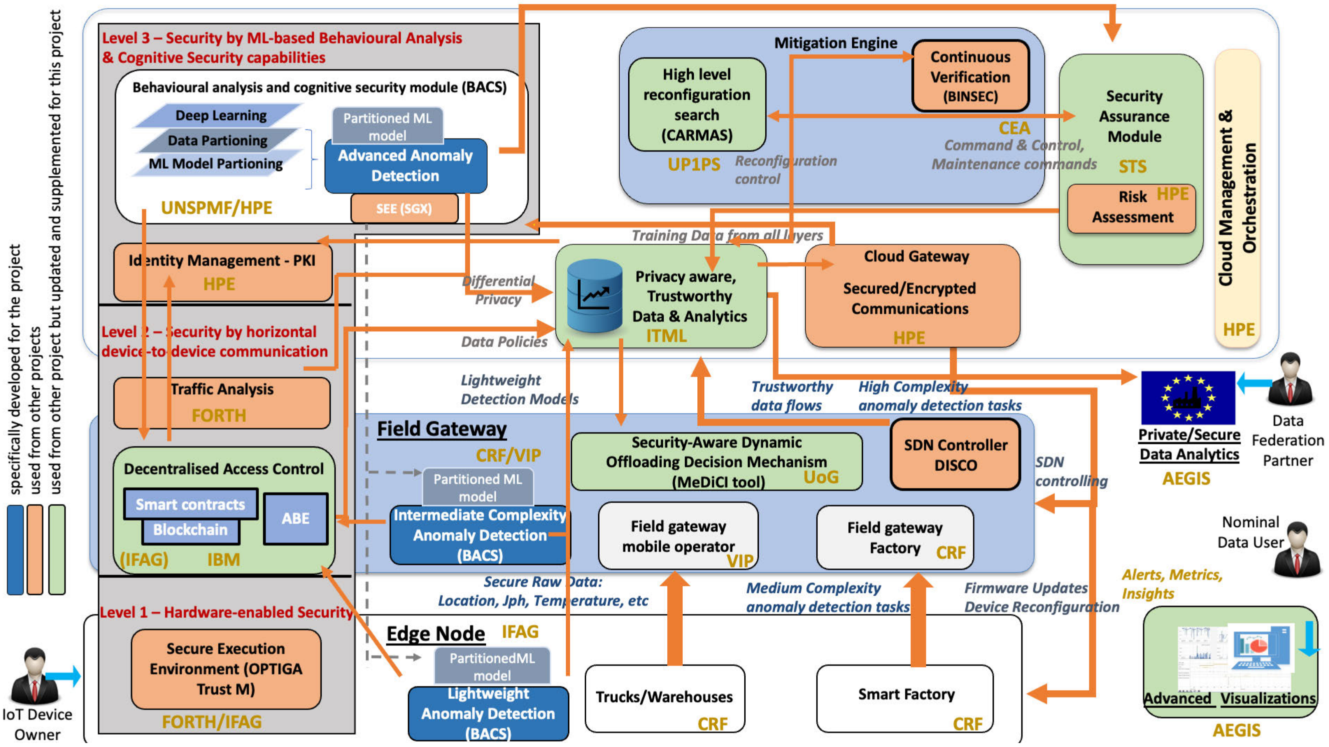

specifically developed for the project
used from other projects
used from other project but updated and supplemented for this project
Level 3 – Security by ML-based Behavioural Analysis & Cognitive Security capabilities
Behavioural analysis and cognitive security module (BACS)
Deep Learning
Data Partioning
ML Model Partioning
Partitioned ML model
Advanced Anomaly Detection
UNSPMF/HPE
SEE (SGX)
Identity Management - PKI
HPE
Level 2 – Security by horizontal device-to device communication
Traffic Analysis
FORTH
Decentralised Access Control
Smart contracts
Blockchain
ABE
(IFAG)
IBM
Level 1 – Hardware-enabled Security
Secure Execution Environment (OPTIGA Trust M)
FORTH/IFAG
IoT Device Owner
Mitigation Engine
High level reconfiguration search (CARMAS)
UP1PS
Reconfiguration control
Continuous Verification (BINSEC)
CEA
Command & Control, Maintenance commands
Security Assurance Module
STS
Risk Assessment
HPE
Cloud Management & Orchestration
HPE
Training Data from all layers
Differential Privacy
Privacy aware, Trustworthy Data & Analytics
ITML
Data Policies
Cloud Gateway
Secured/Encrypted Communications
HPE
Lightweight Detection Models
Trustworthy data flows
High Complexity anomaly detection tasks
Private/Secure Data Analytics
AEGIS
Data Federation Partner
Field Gateway
CRF/VIP
Partitioned ML model
Intermediate Complexity Anomaly Detection (BACS)
Security-Aware Dynamic Offloading Decision Mechanism (MeDiCI tool)
UoG
SDN Controller DISCO
SDN controlling
Field gateway mobile operator
VIP
Field gateway Factory
CRF
Nominal Data User
Alerts, Metrics, Insights
Secure Raw Data: Location, Jph, Temperature, etc
Medium Complexity anomaly detection tasks
Firmware Updates Device Reconfiguration
Edge Node
IFAG
PartitionedML model
Lightweight Anomaly Detection (BACS)
Trucks/Warehouses
CRF
Smart Factory
CRF
Advanced Visualizations
AEGIS

FIGURE 1. C4IIoT cybersecurity architecture high-level overview.

- Sections II-B and II-C explain two specific use cases, namely *Smart Factory Use Case* and *Smart Logistics Use Case*.
- Section II-D describes different architecture layers
- Decentralized Access Control has been described in II-F, it controls the access of the data by different entities
- Security Assurance Module, described in II-G, is responsible for monitoring the C4IIoT architecture
- Security-aware dynamic offloading decision mechanism section II-H describes the decision component of the architecture
- Section II-I describes the SDN component DISCO
- Mitigation engine – High-level reconfiguration search is presented in section II-J
- Edge devices for Smart Logistics and Smart Factory use cases are described in sections II-K and II-L
- Important part of the C4IIoT is Hardware Enabled Security, which is presented in section II-M
- Anomaly detection models are discussed in section Behavioural Analysis & Cognitive Security (BACS) II-N
- Finally, Confidential Malware Analysis in the Cloud is discussed in section II-O

II. ARCHITECTURE OVERVIEW

A. HIGH-LEVEL ARCHITECTURE VIEW

The architecture of the proposed framework is divided into three logical layers: Edge, Field gateway (fog) and Cloud (Fig. 1). The architecture enables threat detection at three different security levels (left side of Fig. 1): hardware-enabled security (level-1), security enabled by horizontal device-to-device communication through distributed ledger technologies (level-2), and security enabled by context-aware intelligence for detecting anomalous or malicious behaviour (level-3). The framework also provides a comprehensive solution for advanced visualization, mitigation, informing end-users and activating data federation partners.

In relation to cyber-physical impact, we use the standard confidentiality-integrity-availability (CIA) [23] triad for first-order impact in cyberspace (relating to the computation and digital communication), and the impact classification for physical space (relating to sensing and actuation).

The C4IIoT architecture is designed for two specific use cases described as follows.

B. SMART FACTORY USE CASE

The Smart Factory Use Case is related to the Manufacturing process in a factory environment. In this scenario, we have Autonomous Ground Vehicles (AGVs) which are used in the manufacturing process. AGVs are connected devices and provide real-time data which can be analyzed and monitored. The devices are equipped with sensors (for acceleration and velocity) which are used to monitor their movement and detect possible faults (e.g. collisions) and possible malicious attacks (cybersecurity attacks) which can lead to equipment malfunction or other issues. In these devices we employ software hardening through inclusion of anomaly detection models (both on device and more advanced offloaded models) and low-level runtime isolation through the use of Intel SGX (Intel Software Guard Extensions [24]).

C. SMART LOGISTICS USE CASE

The Smart Logistics Use Case implies the usage of edge devices outside of the factory environment. The edge devices (described in Sec. II-L) are attached to the containers carrying car parts between the manufacturing and the assembly plant. The key requirement is the ability to track the parameters of interest such as geolocation, vibrations, magnetic field, velocity, acceleration and temperature autonomously over extended periods of time. By nature, the edge devices used in this use case differ significantly from those used in the Smart Factory use case. Thereby, we showcase the overall architecture effectiveness while respecting the heterogeneous nature of the edge layer, usual for IoT applications. The edge node, designed and developed within the project to suit the requirements of the Smart Logistics Use Case, achieves significantly better power efficiency (with a consumption in the deep sleep mode with an order of magnitude of microamps), at the expense of having a constrained speed, memory capacity and processing power.

For the Smart Logistics Use Case, we utilize for communication the NB-IoT (Narrowband IoT) technology with EGPRS fallback, which further increases efficiency regarding the power consumption as well as the bandwidth usage. The key enhancement in the cyber security domain comes from the fact that the Field Gateway is deployed within the premises of the Mobile Network Operator (MNO). When edge devices containing SIM cards from a pre-assigned list connect to the network using a specialized APN, they get static IP addresses within the same sub-network as the Field Gateway. This network consisting of the FG and the edge devices is private and inaccessible by external entities, and therefore inherently secure. The already high level of security can be further enhanced by deploying lightweight end-to-end security protocols to secure the message exchange between edge nodes and the Field Gateway. It also features lightweight anomaly detection [12]. More bandwidth-intensive security protocols are deployed to secure the communication between the Field Gateway and the cloud layer. To the best of our knowledge, such approach based on utilization of an in-network Field Gateway has not been used so far in industrial IoT applications, and therefore represents one of the contributions introduced here.

D. EDGE, FIELD GATEWAY, AND CLOUD LAYERS

We now describe the overall system from the 3 layers perspective (Figure 1).

The edge node layer includes the devices and sensors that provide the data that feed the whole framework in both Smart Factory (based on Raspberry Pi like devices) and Smart Logistics (based on in-house designed IoT devices) use cases. It also features lightweight anomaly detection modules and hardware-enabled security. The devices are described in more detail in Sections II-K and II-L.

Field gateway (FG) is an intermediate (fog) layer of our architecture (Figure 1), acting as an element that communicates with the edge nodes on one side and the cloud on the other. FG possesses intermediate computational power compared to the low power edge node layer and the high power cloud layer. It can perform computationally more intensive security-related tasks compared to the edge node layer. FG features significantly lower latency compared to the cloud since it resides closer to the edge nodes. It supports both edge nodes that utilize wired network connectivity (local or internet) as well as those based on cellular protocols. The role of the field gateway is dual. Besides acting as an SDN (Software-defined networking)-enabled network node, it will also include a local offloading/outsourcing decision mechanism. Field gateways are located in mobile operator premises for the Smart Logistics case and within factory premises for the Smart Factory case, protected in both cases by a strong security network infrastructure. In both cases, field gateways are server computers. Communication between FG and the edge nodes relies on either industrial wireless protocols in Smart Factory or on Low Power Wide Area Networks (LPWAN) technologies in Smart Logistics scenario, provided by the mobile operator. A preferred technology in the latter case is Narrowband IoT (NB-IoT) which is suitable in terms of large area coverage, small power consumption and the support of massive number of devices. GPRS is used as a backup communication technology. A distinctive feature of the system is the offloading mechanism, realized through the MEDICI tool (Section II-H), also hosted at FG. Namely, if the confidence in an anomaly detected at the FG is low, the data is offloaded from field gateways to the cloud layer through the cloud gateway, using an SDN-enabled switch/router and the offloading mechanism.

The cloud layer is where advanced anomaly detection and privacy-aware data analytics take place using data aggregated and offloaded from the field gateways. In case an anomaly is detected, relevant features about the detected anomalies (e.g., location/sensor ID, priority level, etc.) are sent to the mitigation engine. The mitigation egine then decides, through the CARMAS component described in Section II-J (Figure 1) if an action needs to be taken and searches for a new configuration of the system, that is either resistant to the detected anomaly or at least minimizes the damage that it can cause. The system is then reconfigured through the SDN controller. At the same time, a continuous verification tool performs verification of IoT device's firmware and takes an action if a change is detected in the corresponding segment.

Figure 1 includes all the components that constitute the system, as well as directed arrows that indicate their communication to implement the above described operation mechanisms. It also includes the color code that indicates whether the components are developed in the context of C4IIoT or have been reused. The Figure also clearly indicates the mappings between the components and the layers, i.e., it described where each component is deployed. The Figure also showcases which components contribute to which level (hardware-enabled, device-to-device, cognitive) of the introduced security mechanisms. These levels are detailed in Subsections II-M, II-F, and II-N. We now proceed with a component-by-component system description.

E. COMMUNICATION THROUGH LAYERS – DATA FUSION BUS (DFB)

For all layers of the architecture it is very important to have safe and reliable means of internal communication. For this reason our proposed architecture uses the Data Fusion Bus component which is a secure message queue implementation based on Apache Kafka. Distributed message queues such as Apache Kafka allow for easy to implement message-based communication in large systems. They also allow asynchronous communication where needed and they are often designed with high-throughput, scalable and high availability systems in mind. This characterstic combined with extensive usage of container and container management software (Kubernetes[2] was used for our system) allows for very high and dynamic, adjustable scalability of our system. Apache Kafka is also very compatible with various platforms and programming languages, and it offers various built-in stream operations for advanced message processing, filtering and so on. All system components which we describe in following sections use this single message bus to communicate with each other. Components publish information to various message queue topics, and other components subscribe to topics which are relevant to them. The messages are also permanently stored for future analysis. It is important to clarify that the IIoT data (or payload, which is to be found in various messages) is encrypted so even in case of a system breach the data remains private. This process of payload encryption will be described in Section II-F.

F. DECENTRALIZED ACCESS CONTROL

The C4IIoT architecture includes a decentralized access control (DAC) solution, allowing to control the access to data by various entities, to enable auditability of various events and policies, and to verify the integrity of data items. One core element of the DAC is applying encryption in order to restrict access to data such as sensor readings. We apply ciphertext-policy attribute-based encryption (CP-ABE), a type of public-key encryption where data consumers, such as the C4IIoT analytics service (BACS, Section II-N), are each granted with a personal secret key that is associated with a set of attributes characterizing its holder (for example: organization, role, purpose of consuming the data, etc.). Entities generating the sensor readings encrypt them with a public key and specify an access policy to the encrypted data as part of the process, describing who shall be allowed to decrypt it in the ‘‘language of attributes’’. This mechanism has built-in elements of decentralization. Once a data item is encrypted, no central authority is required to evaluate the access policy and grant access to the data. The decentralized access control also relies on Hyperledger Fabric (HLF), which is a permission-enabled blockchain with support for executing smart contracts. HLF enables auditability of events and access policies as well as assure the integrity of data in C4IIoT. When sensor readings are created in the edge nodes, or when being stored on the cloud storage service, a corresponding record is logged in the HLF channel. These tamper-proof records include a pointer to the place where the data item is stored, a hash of the data taken in the time when it was created, and the CP-ABE access policy used to encrypt it. This solution allows the entities involved in C4IIoT to monitor and verify the integrity of the data and detect data manipulation attacks.

G. SECURITY ASSURANCE MODULE

The Security Assurance Module (SAM) is a model driver tool responsible for monitoring, testing, and assessing the runtime operations of the C4IIoT architecture. This component is auditing critical components and processes of the infrastructure while leveraging monitoring mechanisms developed in the context of project. SAM provides an evidence-based view of the security posture of the C4IIoT architecture, with accountability provisions for changes that occur in said posture and the analysis of their cascading effects, supporting the runtime checking based on sets of associated claims and assessments. The real time, continuous assessment of the security posture of the C4IIoT architecture is enabled by a purpose-built Event Captor Module based on Elasticsearch [25](ELK stack), which is responsible for creating and aggregating events as the required evidence from multiple sources related to the operation of individual components, as well as the overarching processes where these components are involved in. Those events are digested by the EVEREST tool (details are described below in this section), where specific rules are set end checked in real-time for violations. This way, SAM supports several built-in security assessments addressing the CIA principles along with custom metrics.

In more detail, the SAM is comprised of five primary modules:

1) **Cyber System Asset Loader:** The component responsible for receiving the system's asset model for the target organization. This model includes the assets of the organization and their relations, security properties for these assets, the threats that may violate these properties, and the security controls that protect the assets and is based on STS's Assurance Model.
2) **Vulnerability Analyzer (VA):** The Vulnerability Analyzer is responsible to identify known vulnerabilities of assets defined within an organisations' asset model based on the well established National Vulnerability Database (NVD) of NIST [26]. The module includes two components, (a) the vulnerability loader and (b) the vulnerability database.

[2] https://kubernetes.io/

3) **Dynamic Tester:** The component responsible for executing dynamic testing assessment (e.g. penetration testing). The module incorporates varies open-source tools such as OpenVAS [27], in order to assess the occurrence and exploitability of identified vulnerabilities in the target system.
4) **Event Captor (EC) Module:** The Event Captor Module is a tool that creates a variety of event types related with the assured system (e.g. user logins and, file accesses) and pushes them towards EVEREST for evaluation. Data and events are mostly collected through Elasticsearch based on lightweight shippers (namely Beats), such as Filebeat, Metricbeat, Packetbeat, etc., that forwards and centralizes log data. Data can also be collected through Logstash8, an open server-side data processing pipeline that ingests data from a multitude of sources transforms it, and then sends it to Elasticsearch. The Event Captor is initiated through the respective REST calls from EVEREST.
5) **EVEREST:** A run-time monitoring (reasoning) engine, for a defined set of tools based on the event calculus reasoning [28]. It offers an API for establishing such monitoring rules. EVEREST consumes the run time events from the applications's monitored properties (through the EC), and evaluates the defined rules. The outcome of EVEREST is the real-time assessment for the rules validation or violations.

H. SECURITY-AWARE DYNAMIC OFFLOADING DECISION MECHANISM

The security-aware dynamic offloading decision component of the C4IIoT architecture consists of a Multi-critEria DecIsion support meChanism for IoT offloading (MEDICI) [29]. MEDICI resides in the FG and dynamically decides which anomaly detection model (BACS II-N, either in the FG or the Cloud layer) needs to be triggered if further investigation is deemed necessary by the edge anomaly detection model (BACS, II-N). MEDICI takes into account metrics such as the accuracy and confidence of an anomaly detection model, the inference or execution time of an anomaly detection task, the network transmission time to offload the task data to another device and any delays incurred by the network. Using lightweight estimation techniques and previously historical data it predicts the future values of these metrics in order to decide which anomaly detection task should be executed and where, so that the overall time to detect an anomaly will be reduced without compromising the detection accuracy.

As seen in Figure 2, MEDICI consists of three services called Request, Response, and Network. The role of the *Request Server* is to handle incoming offloading requests from the edge devices and forward them to the *Decision Maker* where the actual decision on which anomaly detection model should be triggered is made. Whenever an anomaly detection task is executed, relevant execution history information (e.g. inference times, task size and confidence levels) and network data (e.g. round-trip latency and transmission times) are gathered by the *Response Server* and *Network Server* respectively. History information and network data are then stored in dynamic storage (*Execution Map* and *Network Map*) to inform future decisions.

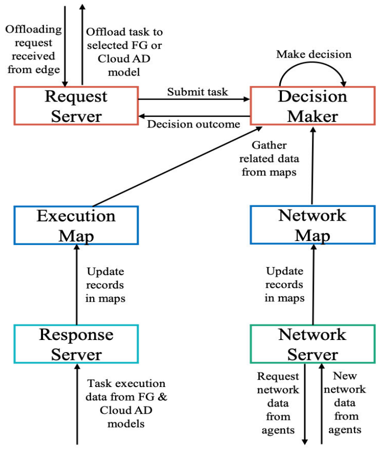

Offloading request received from edge
Offload task to selected FG or Cloud AD model
Make decision
Request Server
Submit task
Decision outcome
Decision Maker
Gather related data from maps
Execution Map
Network Map
Update records in maps
Update records in maps
Response Server
Network Server
Task execution data from FG & Cloud AD models
Request network data from agents
New network data from agents

FIGURE 2. Overview of the MEDICI service.

1) CONTINUOUS VERIFICATION (BINSEC)

BINSEC is a platform for static analysis of software binaries using formal methods. BINSEC operates at binary level, i.e. on executable programs (e.g. .exe files) after they have been compiled from source, which is typical for programs written in languages like C, C++, Rust, Go, Fortran, Ada or Pascal.

Binary executables contain machine code, a combination of low-level instructions, designed for execution by a specific processor. Therefore, BINSEC must analyse for different hardware architectures which have different instruction sets.

Typical examples of its usage include vulnerability discovery (finding new security bugs in existing programs), vulnerability analysis (finding the specific conditions under which a vulnerability can be exploited by an attacker, e.g. Stuxnet-alike [30]), reverse engineering (understanding the behaviour of a program without access to its source code), malware analysis (reverse engineering on obfuscated binary code produced to make reverse engineering more difficult), program verification at the binary level (proving that a binary program meets some property, for instance that a binary program does not suffer from buffer overflows), program verification in programs containing assembly/binary fragments (verifying a program at the source level, using BINSEC to help it understand the assembly parts or external binary-only libraries used by the program).

For integration of BINSEC in the system architecture, the key constraints for integrating a binary code analysis for continuous verification in the mitigation engine are automation (minimal intervention from the user), precision (low rate of false positives), and security guarantees (necessity to not trade security for less automation). In the light of these constraints, we concluded that, in general, methods

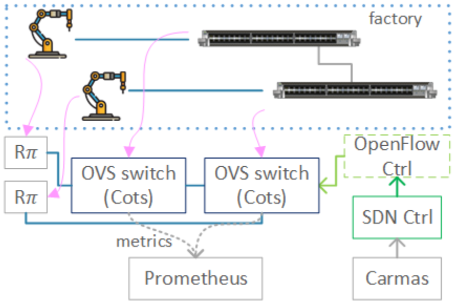


FIGURE 3. Overview of the SDN service.

like static analysis are not practical for the project, because they become imprecise without user annotations, and report many false alarms. They can still be used to guarantee the security of critical components that are unlikely to change, but it is unwise to focus on them. Thus, our goal is to focus on complete methods, that can find bugs instead of proving their absence. Even if these methods provide lower security guarantees than sound methods, their ability to be more automated and the fact that they mostly report true errors are in line with the focus of the project and the system in general.

I. MITIGATION ENGINE – SDN CONTROLLER

The SDN component (DISCO) built for C4IIoT manages the network infrastructure. It provides network connectivity to the factory (the smart factory use case) for day to day operation. It also protects the network from malicious behavior by enforcing mitigations. As shown in the Fig. 3, many containerized services implement the SDN service:

1) The SDN Controller takes as input mitigation requirements from the CARMAS (II-J) component using a REST API. The input mitigation requirement request describes the wanted end-result across the whole network infrastructure. In other words the desired network configuration is provided. The request details are high-level and generic and lack any specific implementation details. From the point of view of CARMAS, the details of an intent instantiation are irrelevant. For example, an exclusion of a network device (from a software defined network) can be enforced on every infrastructure network switch. But it can also be only enforced on the switches on the data path used by the target device. This latter example can save some space in the unrelated flow tables.
2) The OpenFlow Controller manages the network switches. It pushes OpenFlow [31] rules inside them. The SDN Controller configures the OpenFlow Controller through a set a network policies. A policy is a specific configuration constraint to a specific infrastructure context. A set of low level network policies can provides a higher level network intent. Faucet [32] is the base of this OpenFlow Controller.
3) The SDN Controller also uses extra side services. Such as a Metric Controller that monitors the network switches using OpenFlow. A Prometheus database that stores the gathered metrics. A Grafana UI that displays the infrastructure's state.

For the smart factory use case, the SDN fabric is built as shown in the Fig. 3. Some Raspberry Pi mimics the communicating factory hardware. Some COTS hosts act as the rack mounted network switches. They run an Open vSwitch [33] daemon. It listens to OpenFlow rules and steer the data path accordingly to the SDN rules.

J. MITIGATION ENGINE – HIGH-LEVEL RECONFIGURATION SEARCH (CARMAS)

The *Cyber-Attack Runtime Mitigation Action Search (CARMAS)* component is presented in detail in [34]. It takes as input from the Advanced Visualization Toolkit (AVT), a set of detected attack actions, each accompanied with the business loss that they would bring about if left unmitigated. It produces and sends as response to the AVT a proposed list of network reconfiguration action sets to mitigate the input attack, in decreasing order of their estimated business loss reduction. CARMAS assembles five sub-components in a Docker containerized REST web service:

1) The *Inference Engine (IE)*, a general-purpose, application-independent, rule-based artificial intelligence automated reasoner [35] to interpret application-specific, *Knowledge Bases (KB)* in a formal yet executable language that parsimoniously integrates the constraint [36], [37], logic [38], [39], object-oriented [40], [41] and service-oriented programming paradigms [42], resulting in what we coin *Constraint Object-Oriented Logic Programming as a Service (COOLPS)*;
2) The ontology, a COOLPS KB representing an application-independent, conceptual model of IIoT networks, vulnerabilities, attack actions and network reconfiguration mitigation actions and their relations;
3) The *Constraint Optimization Problem (COP)* builder, a COOLPS KB which interpretation by the IE builds the COP of finding the best mitigation for the input suspected ongoing cyberattack;
4) The COP solver, a COOLPS KB which interpretation by the IE solves the built COP;
5) The service-oriented wrapper, a COOLPS KB which interpretation by the IE starts a web server to communicate with the AVT, and then, for each REST request, (a) convert its JSON payload into a query submitted to the COP builder, (b) forwards the builder's answer as a query to the COP solver, (c) converts the solver's answer into a JSON payload for the REST response that (d) the web server then sends back to the AVT.

The COP builder takes as parameter a heuristic function to build a COP that tailors relevant general knowledge that its reuses from the ontology to the specific attack input description received from AVT. In [34], we show that such heuristically built COP can then be solved by the COP solver in a few seconds even for large coordinated attacks involving up to 10 attack actions targeting up to 15 network assets.

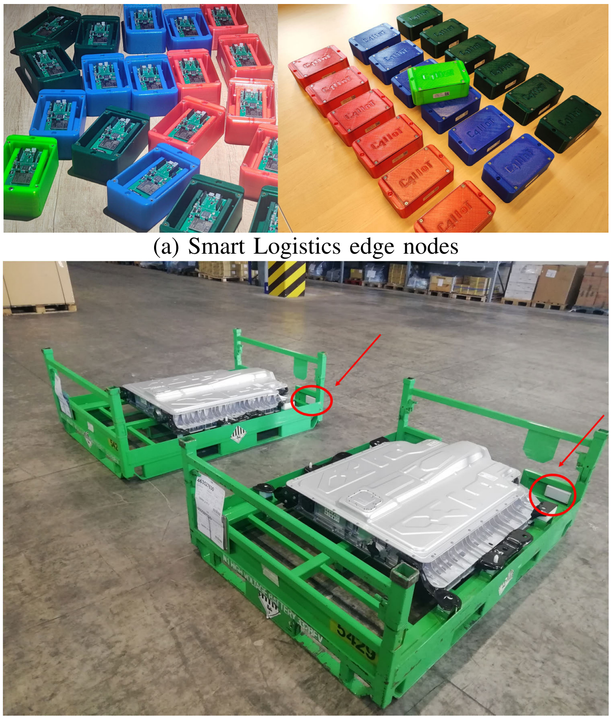
K. SMART FACTORY EDGE DEVICES

The edge node for the Smart Factory use case is designed following the established requirements, according to which, these nodes are placed on autonomous ground vehicles (AGVs). The AGVs are located in the factory, where they have virtually unlimited energy. These vehicles have a big energy consumption, as for example a direct tap a power input or a set a big batteries, hence, the IoT devices embedded in these vehicles do not have particular power restrictions. Due to this feature, IoT nodes contain a relatively powerful microprocessor, such as the ones found on Raspberry Pi devices.

Attached to it, there are different connected devices, which form the node and add a hardware security layer. This layer is achieved through the use of several Hardware Security Modules (HSMs). On one hand, an Infineon OPTIGA$^{\text{TM}}$ TPM2.0 [43] is incorporated via SPI (Serial Peripheral Interface) communication. TPM2.0 is a type of HSM which stores keys and performs cryptographic operations on the node. This component is essential as the keys used in the blockchain network are stored in this device. Two more HSMs are also involved in the Smart Factory edge node. On the one hand, there is one HSM in the form factor card that uses an NFC reader to authenticate the user at the node. On the other hand, the same mechanism is also established but through a USB Dongle, which contains an HSM where the cryptographic keys of the user are stored, in this way, it makes possible to carry out the authentication process.

L. SMART LOGISTICS EDGE DEVICES

The edge node devices for Smart Logistics use case have been custom built, designed and manufactured in-house within the C4IIoT project, aiming to be low power due to battery-powered operation, but still capable of running lightweight anomaly detection tasks. The CPU is a low-power ARM Cortex M0+ operating at 16MHz with 32kB/256kB of SRAM/FLASH memory for the data/application code. Such limited resources are fit for the required tasks of sensor data acquisition, secured communication with the field gateway, and lightweight anomaly detection. For wireless connectivity, it features NB-IoT and LTE-M, new 3GPP communication standards. Where LTE carrier might be unavailable, EGPRS fallback is supported to ensure connectivity in such areas. Geolocalization is supported by the integrated GNSS module. The following onboard sensors providing data for anomaly detection modules are available: accelerometer, magnetic field sensor, air temperature, humidity and pressure sensor, and the illumination sensor. Secured-by-hardware functionality is provided by OPTIGA$^{\text{TM}}$ Trust M crypto chip. The edge node devices in their housings, and mounted on the containers are depicted in Fig. 4.


(a) Smart Logistics edge nodes

(b) Edge nodes mounted on containers

FIGURE 4. Edge nodes mounted on containers in the smart logistics use case.

M. HARDWARE ENABLED SECURITY

Hardware-enabled security plays a significant role in the C4IIoT architecture, since it forms the base for many modules that are built on top of it, including the decentralized access control (DAC) technologies. The Hardware Security Module (HSM) that has been added in the edge nodes offers robustness and trustworthiness both in the operations and the data collected from the IoT devices. There are different HSMs included in the two scenarios: Smart Factory and Smart Logistics. The main difference between these two scenarios is the constraints of the IoT devices themselves. In the Smart Factory use case, the IoT devices are placed in automatic guided vehicles (AGVs). The work is carried out inside the factory, hence there are no energy constraints, enabling IoT nodes to contain a high power microprocessor, such as a Raspberry Pi platform. In the inbound Smart Logistics scenario, the IoT devices are attached to containers, which results to significant power restrictions due to the fact the devices are powered by batteries only. In this case, IoT devices developed with low-power microcontrollers are essential.

Both the hardware and software running on IoT devices is conditioned by this feature. In the case of software the anomaly detection mechanisms are lighter than in other layers of the architecture. Furthermore, the hardware elements embedded in the IoT device also vary from the different scenarios. In the Smart Factory use case the hardware security module used in the Raspberry Pi is the Infineon OPTIGA$^{\text{TM}}$

TPM2.0. OPTIGA^{TM} TPM2.0, as we will see later, is a suitable HSM for the Raspberry Pi, as it offers more robust operations and provides more advanced mechanisms for operating system devices. In the inbound Smart Logistics use case, the HSM used is the Infineon OPTIGA^{TM} Trust M. OPTIGA^{TM} Trust M offers a low power crypto-controller which performs security mechanisms in the microcontroller.

1) HORIZONTAL DEVICE-TO-DEVICE SECURITY

Within the C4IIoT architecture, data aggregated by the IIoT devices (edge nodes) may not be shared among different devices or with data federation partners in raw form. Furthermore, a data federation partner may want to protect its data and analytics results from its competitors, or to monetize its data selectively. This, Level-2 security, of the C4IIoT framework has been built upon multiple technologies brought by the partners in order to support this. One of them is IBM's decentralized access control (DAC) solution that utilizes distributed ledger technologies (Blockchain) and attribute-based encryption (ABE) in order to restrict access to data using privacy-aware policies, enable auditability of events and access policies and assure the integrity of data in C4IIoT. Infineon's secure element technology that allows to protect sensitive information integrates with the DAC and further strengthen the security and trustworthiness in C4IIoT by securely storing the secret keys the edge nodes use to interact with the Blockchain. Our identity management solution (developed by HPE - Hewlett Packard Enterprises) completes the Level-2 security mechanism by providing a public key infrastructure (PKI) that enables to manage and authenticate identities in C4IIoT using certificates and cryptographic materials.

N. COGNITIVE SECURITY

The Behavioural Analysis & Cognitive Security (BACS) Framework includes behavioural models based on advanced deep learning techniques to provide more contextual information and form an advanced anomaly detection model for the entire IoT ecosystem. The framework takes into account that each device is not isolated and includes its interactions with other devices. BACS is depicted in Fig. 5 and it consists of the following three main packages:

- BACSCL (BACS Cloud Layer) performing anomaly detection based on deep autoencoder forests (unsupervised Anomaly Detection (AD)) and deep neural network forests (supervised AD) implemented in Python using the Tensorflow 2 library.
- BACSPY providing anomaly detection based on outlier detection, classification and representation learning algorithms implemented in Python using Tensorflow 2, scikit-learn and PyOD libraries
- BACSC contains lightweight anomaly detection routines implemented in C for constrained microcontroller and based on IIoT devices planned for the Smart Logistics use case.

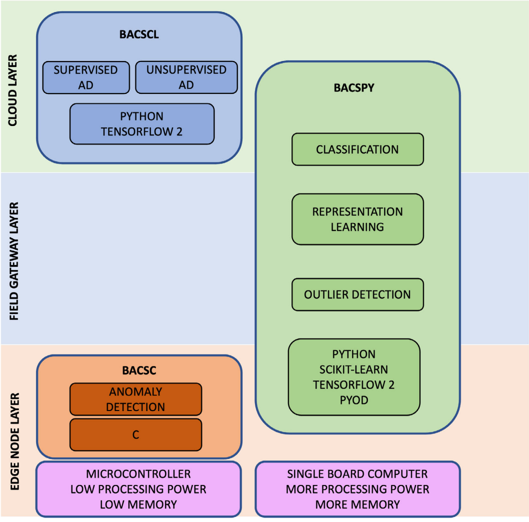


FIGURE 5. BACS framework.

The models available in BACS are:

1) `TFAutoAD` - Auto-Encoder model that has three variants: one that works on Edge layer, other that works in FG, and third one that works in Cloud layer.
2) `TFAutoDeepAD` - Deep Auto-Encoder model
3) `TFAutoVAEAD` - Variational Auto-Encoder model.
4) `TFAutoDeepVAEAD` - Deep Variational Auto-Encoder.
5) `TFAutoWideVAEAD` - Wide Variational Auto-Encoder
6) `TFAutoFCNAD` - AD model based on Fully Connected architecture.
7) `TFAutoDeepFCNAD` - Deep AD model based on Fully Connected architecture.
8) `TFAutoLSTMAD` - AD model based on LSTM architecture.
9) `TFAutoGRUAD` - AD model based on GRU architecture.
10) `TFAutoRNNAD` - AD model based on RNN architecture.
11) `TFAutoProphetAD` - AD model that uses Facebook's state of the art model Prophet [44]
12) `Kmeans_DPLAD` - differentially private kMeans AD model.
13) `PCA_DPLAD` - differentially private PCA AD model.
14) `EE_SKLAD` - Scikit Learn implementation of Elliptic Envelope AD model.
15) `SVM_SKLAD` - Scikit Learn implementation of One-Class SVM AD model.
16) `LOF_SKLAD` - Scikit Learn implementation of Local Outlier Factor AD model.
17) `IF_SKLAD` - Scikit Learn implementation of Isolation Forest AD model.
18) `ABOD_PyODAD` - PyOD implementation Angle-base Outlier Detection model.

19) `KNN_PyODAD` - PyOD implementation of kNN model for Outlier Detection.
20) `PCA_PyODAD` - PyOD implementation of PCA for Outlier Detection model.
21) `HBO_PyODAD` - PyOD implementation of Histogram-based Outlier Detection model.

1) PRIVACY AWARE, TRUSTWORTHY DATA & ANALYTICS

In order to preserve private information in privacy-sensitive data, we deploy differentially private methods. A method that analysis certain data at the input and produces a certain output is said to be differentially private, if by looking at the output, one cannot determine whether any individual's data was included in the original dataset or not for analysis. In other words, the guarantee of a differentially private algorithm is that its behavior hardly changes when a single individual joins or leaves the dataset – anything the algorithm might output on a database containing some individual's information is almost as likely to have come from a database without that individual's information. Some of the common practices to achieve differential privacy are i) adding noise to the samples during the training process (indistinguishability of samples); and ii) adding noise to the gradient computed on a sample (indistinguishability of gradients of samples). Specifically, in the context of C4IIoT, two differential privacy outlier detection models – PCA and KMeansm have been utilized. Both of these methods are implemented using diffprivlib – a general-purpose library for experimenting with, investigating and developing applications in, differential privacy [20].

O. CONFIDENTIAL MALWARE ANALYSIS IN THE CLOUD

C4IIoT has also developed an (optional) cloud-based solution that can be used to detect malicious files and identify malware. The entire system is designed with privacy-preserving guarantees regarding the processing of sensitive and/or critical files in third-party clouds that may not considered trusted. The entire processing is performed in the cloud-based server, encapsulated inside hardware assisted enclaves, base on Intel SGX[3] enclaves, which communicates with the clients through a network TLS-terminated connection. This encapsulation enables the protection of the malware analysis, and most importantly the privacy of the user's data. In addition, by having the entire malware analysis modules on a cloud-based server, the C4IIoT is alleviated from the need to maintain multiple analysis tools in different layers or even entities.

The cloud-based server is able to accept multiple connections and perform the analysis on the incoming data. For the analysis of the received data, the server maintains an updated signature set of know-threats (acquired by open-source tools, such as ClamAV [45]), or even behavioral analysis models for combating evolved attacks. The suspicious data are received in encrypted format by the cloud-based server and are forwarded inside the Intel SGX enclave that host the malware analysis engine. Once inside the secure enclave, the data are decrypted and prepared for processing. The cryptographic keys required for the authentication and the successful decryption of the data reside exclusively inside the SGX enclave. In this way, the secret keys and sensitive or critical data are never present in plain-text format in the server's file system or DRAM and they remain inaccessible even by the server's host/provider. Moreover, even if the non-SGX part of the cloud-based server or the hosting infrastructure gets compromised, the keys and the private user data cannot be obtained.

When the analysis of the suspicious data finishes, the results are send back as a status report. The report generation is also performed inside the secure enclave, so it cannot be accessible and ensure that attackers or honest-but-curious entities, such as the cloud provider, will not obtain any information about the data. In combination with protecting the analysis inside the enclaves, we also eliminate the possibility of malicious entities injecting custom code and observe the generated report in order to infer information that could threaten the privacy of the user's data.

This way of encapsulation and software protection can be used in combination with our BACS (II-N) component to achieve even higher level of security. BACS includes built and ready-to-use special Docker images which can be used with Intel SGX platform compatible runtimes.

P. USER INTERFACE

The C4IIoT user interface offers active real-time monitoring, historical analysis and possible mitigation actions for specific attacks. The real-time monitoring consists on data produced by the edge enriched with information from the applied anomaly detection mechanisms, and information from the network traffic analysis. An alerting mechanism informs the end-user about detected attacks and provides the option to select among suggested mitigation actions. Finally, the historical analysis help users to understand the evolution of the edge devices behaviour and look for potential relationships or behavioural patterns among the monitored assets.

1) ADVANCED VISUALISATIONS

The Advanced Visualization Toolkit (AVT) provides the means to visualise several indicators deriving from the analysis of data coming from the Edge layer. It enables the end-user to explore data in a high level through several interconnected, interactive visualisations that also allow drilling into more detailed information to reveal hidden relationships and insights. It also supports a timeline analysis component and multiple visualisations. These visualisations include a set of interactive graphs and charts that form the heart of the AVT. Bar charts, line charts and pie charts are some of the standard forms of data representations. Different

[3] Project website – https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html

TABLE 1. Various types of attacks and responsible system components for detection and mitigation. The list is not comprehensive and includes some common attacks.

Type of Attack	Responsible component
Denial of Service	SAM, BINSEC, DISCO
Malware	SAM, TEE, BINSEC, DISCO
Manipulation of hardware & software	SAM, TEE, BINSEC
Manipulation of Information	SAM, DAC, DFB, TEE, BACS
Targeted attacks	SAM, BINSEC, DISCO, BACS
Abuse of personal data	SAM, DAC, TEE
Brute force	SAM, DISCO
Man-in-the-Middle attack / Session hijacking	SAM, DAC, DISCO, BACS
IoT communication protocol hijacking	SAM, DAC, DISCO
Network reconnaissance	SAM, DISCO
Vandalism and theft	DISCO
Unintentional change of data or configuration in the OT system	DISCO
Erroneous use or administration of devices and systems	DISCO
Damage caused by a third party	DISCO
Failure or malfunction of a sensor / actuator	DISCO, BACS
Failure or malfunction of a control system	DISCO, BACS
Software vulnerabilities exploitation	SAM, BINSEC
Failure or disruption of service providers	DISCO
Communication network outage	DISCO
Violation of rules and regulations / Breach of legislation / Abuse of personal data	SAM, DAC
Failure to meet contractual requirements	SAM

visualizations are used to display different types of data, in order to make the understanding of data easier.

III. SYSTEM PERFORMANCE AND VALIDATION

We now present several evaluation results for the C4IIoT framework. Specifically, we demonstrate validity and performance of the framework by considering anomaly detection validity and performance, as well as the overall system and communication performance.

The C4IIoT framework supports detection of various types of cybersecurity attacks, through the utilization of its different components (Figure 1). Table 1 shows the mapping between various types of attacks and the responsible components. In this paper, we demonstrate the framework capabilities on a concrete example of the Data Manipulation Attack.

We next proceed by describing the setup and data preparation for the subsequent evaluations (for anomaly detection). Then, the overall system and communication performance will be described in more detail in Section III-A, the anomaly detection performance is discussed in III-B, and the data Manipulation Attack testing will be detailed in Section III-C.

For anomaly detection components, a thorough evaluation of both unsupervised and supervised BACS models was planned to be done on labeled real datasets. While real data for nominal operation in the industrial environment has been acquired, acquisition of labeled data with positive labels that correspond to real anomalies in the industrial environment is difficult to acquire. Henceforth, we work with real nominal (normal operation) data and generate on top of it synthetic anomalies. We evaluated the BACS component with synthetic anomalies, where one can differentiate between i) *isolated*, and ii) *continuous* anomalies.

- *Single (isolated) anomalies* are formed in a single timestamp via one of 3 strategies, which are discussed bellow. This type of anomalies reflect the anomalous behaviour of sensors.
- *Continuous anomalies* stretch over a time interval and they affect 10 distinct consecutive data samples. This type of anomalies reflect the behaviour of physical values that are measured by the sensors.

The strategy that we use to form anomalies is to replace a value x from the column c in the original dataset with:

- `make_zero` – zero;
- `make_small` – a sample from distribution
$$X_{\text{small}} = x\,(1 - U[0.1, 0.4));$$
- `make_large` – a sample from distribution
$$X_{\text{large}} = x\,(1 + U[0.1, 0.4));$$
- `randomize` – a sample from distribution
$$X_{\text{randomize}} = U[\min_x\{x|x \in c\}, \max_x\{x|x \in c\}).$$

Here, $U[a, b)$ denotes the uniform distribution on the interval $[a, b)$, $b > a$.

A. COMMUNICATION PERFORMANCE

Speed and reliability, along with data integrity, are important aspects of the architecture. The Data Fusion Bus is a component based on Apache Kafka [46], which offers management capabilities of the system through an API, which offers a fast, reliable and secure way to transfer data between the components and the layers of C4IIoT architecture. Kafka topics (queues) are secured through certificates issued by a private Certificate Authority, and the certificate users can either be allowed or banned from using these topics. Access to topics is provided following the principle of least privilege, in order to ensure that in a case where the certificate is compromised, access will be restricted to the minimum possible. This architecture protects the transferred data, along with preventing any unauthorised user to inject data into the system.

Although most topics cause very little traffic ($\sim$ 0.1 messages/second), the main topic which aggregates the data from Edge of both the Smart Logistics and the Smart Factory, may be used to transfer messages at a very higher rate. The traffic was measured using a Prometheus [47] to receive the metrics from Kafka and a Grafana [48] interface to depict them. In the testing environment, using approximately ten simulated devices sending a message every 7-9 seconds a traffic of 5kB/s was measured. This kind of throughput is very low for Kafka infrastructure, strongly suggesting that even hundreds of devices, either AGVs or Smart Logistics could be easily supported by this architecture.

Data from Kafka is consumed by various components of the C4IIoT architecture, at almost real time, allowing the detection of possible cyber attacks to be reported fast. All components that need to process the data, consume them at the rate they can, therefore avoiding an overflow of data which could potentially disrupt their service. Additionally, data which is needed by more than one components, is consumed by the relevant Kafka topic without delay since it becomes available for all at the same time. In the case one of the components crashes, since the data remain available in the Kafka topic, when it is up again, it can resume operation from the point it was stopped.

B. COGNITIVE SECURITY PERFORMANCE

In this section we describe the C4IIoT framework performance with regards to the BACS evaluation (anomaly detection). We evaluate both use casesm Smart Logistics and Smart Factory, in order to validate our approach. For the Smart Logistics use case we operate in an unsupervised fashion, where we only measure the system response time and not anomaly detection accuracy. Here, we report that all our models successfully converge and behave normally during training. For the Smart Factory use case, we employ our synthetic anomaly generation process described in the previous section. In addition to the system response time, we also measure validation metrics (such as accuracy and F1-score) obtained in testing with the synthetic dataset.

It is important to emphasize that BACS was designed to be compatible with any tabular dataset, and to work with both unsupervised and supervised data. This was achieved through the definition of abstractions in the code which help with unknown data sources and making BACS configurable to adapt to these situations.

All hyperparameters (such as epochs of training, learning rate, neural network architectures and so on) were decided though a trial-and-error process. A hyperparameter optimization process such as grid search would surely benefit the system, but it has not been performed due to time constraints.

In the results ahead, there are a few notation specifics that we now explain for clarity. If there are several models with the same name, we also annotate them with the architecture layer they belong to (e.g., TFAutoAD [Edge]). The TFAutoAD (TensorFlow autoencoder) has the same configuration in all the three architecture layers (edge, field gateway, cloud), but the layers themselves are different by design in operating with varying window lengths. This means that the input to the models is not the same across the architecture layers, and that is why we add the layer designation in the naming conventions. The window lenghts used are 1, 5, and 10 for the Edge, Field Gateway and Cloud layers, respectively. The models having "TF" in their name are implemented in TensorFlow 2 (neural networks), "SKLAD" are implemented with scikit-learn library, "PyOD" with the PyOD library, and "DPL" with the diffprivlib library.

In Table 2 we present performance for all the supported BACS models. Column *Inference Time* represents total time to perform inference for the entire dataset, while the column *Average Inference Time* represents the average response time for a specific model. In the Smart Logistics use case, and in similar systems in general, we expect the models to have low response times (e.g., less than 100ms), which is achieved here.

TABLE 2. Smart logistics use case BACS results. The column anomalies represents the number of detected anomalies on the training dataset (model sensitivity), Inf. Time represents the total model inference time for all dataset derived time series windows, and the Avg. Inf. Time represents the average model inference time which is the expected model response time in real-world usage. Times are in seconds. Models operate in the Cloud layer, unless otherwise specified in the model name.

Model	Anomalies	Inf. Time	Avg. Inf. Time
TFAutoAD (Edge)	60	3.10	0.0157
TFAutoAD (FG)	6	3.07	0.0159
TFAutoAD	56	3.02	0.0160
TFAutoDeepAD	50	3.02	0.0160
TFAutoDeepVAEAD	175	3.10	0.0165
TFAutoVAEAD	174	2.94	0.0156
TFAutoWideVAEAD	175	2.96	0.0157
EE_SKLAD	127	0.06	0.0003
SVM_SKLAD	180	0.03	0.0002
LOF_SKLAD	154	0.12	0.0006
IF_SKLAD	176	8.63	0.0459
ABOD_PyODAD	156	0.14	0.0007
KNN_PyODAD	150	0.04	0.0002
PCA_PyODAD	175	0.04	0.0002
HBO_PyODAD	164	0.04	0.0002
AE_PyODAD	175	5.02	0.0267
TFAutoFCNAD	0	2.93	0.0155
TFAutoDeepFCNAD	0	2.97	0.0158
TFAutoLSTMAD	8	3.14	0.0167
TFAutoGRUAD	9	2.93	0.0156
TFAutoRNNAD	86	2.96	0.0157
Kmeans_DPLAD	2	0.07	0.0004
AutoEnsembleAD	36	31.17	0.1658

TABLE 3. Smart factory use case BACS results. Same notes as table 2.

Model	Anomalies	Inf. Time	Avg. Inf. Time
TFAutoAD (Edge)	323	50.57	0.0156
SVM_SKLAD (Edge)	324	0.44	0.0001
TFAutoAD (FG)	323	50.48	0.0156
SVM_SKLAD (FG)	321	0.49	0.0002
TFAutoAD	323	50.46	0.0156
TFAutoDeepAD	323	50.53	0.0157
TFAutoDeepVAEAD	323	49.46	0.0153
TFAutoVAEAD	322	49.26	0.0153
TFAutoWideVAEAD	327	49.43	0.0153
EE_SKLAD	323	0.89	0.0003
LOF_SKLAD	411	2.42	0.0008
IF_SKLAD	484	150.24	0.0466
SVM_SKLAD	321	0.51	0.0002
ABOD_PyODAD	470	2.24	0.0007
KNN_PyODAD	281	0.73	0.0002
PCA_PyODAD	323	0.67	0.0002
HBO_PyODAD	466	0.38	0.0001
AE_PyODAD	323	84.56	0.0262
TFAutoFCNAD	291	49.37	0.0153
TFAutoDeepFCNAD	355	50.01	0.0155
TFAutoLSTMAD	484	50.54	0.0157
TFAutoGRUAD	484	50.56	0.0157
TFAutoRNNAD	645	50.43	0.0156
TFAutoProphetAD	3040	8578.22	2.6525
Kmeans_DPLAD	1991	1.04	0.0003
PCA_DPLAD	3173	0.70	0.0002

In Table 3 and Table 4, we present similar results for the Smart Factory use case. The only exception is the Facebook Prophet (ProphetAD) model. As Prophet is a tool for univariate time series modelling, we had to create multiple models to process data per sensor, hence the slower response times.

Finally, in Table 5, we present the synthetic supervised learning results for our models when testing with continuously generated anomalies. From our testing, recurrent neural networks seem to perform best with this type of

TABLE 4. Smart factory use case BACS results when testing with synthetic singular anomalies (self-supervised results). Models operate in the Cloud layer, unless otherwise specified in the model name.

Model	Accuracy	Precision	Recall	F1
TFAutoAD (Edge)	0.90	0.50	0.94	0.65
SVM_SKLAD (Edge)	0.89	0.45	0.78	0.57
TFAutoAD (FG)	0.93	0.86	0.99	0.92
SVM_SKLAD (FG)	0.89	0.84	0.89	0.86
TFAutoAD	0.92	0.94	0.93	0.94
TFAutoDeepAD	0.49	0.79	0.28	0.42
TFAutoDeepVAEAD	0.52	0.84	0.31	0.45
TFAutoVAEAD	0.51	0.84	0.31	0.45
TFAutoWideVAEAD	0.52	0.83	0.31	0.45
EE_SKLAD	0.65	0.89	0.51	0.65
LOF_SKLAD	0.91	0.93	0.94	0.93
IF_SKLAD	0.52	0.80	0.35	0.49
SVM_SKLAD	0.74	0.91	0.66	0.76
ABOD_PyODAD	0.91	0.92	0.94	0.93
KNN_PyODAD	0.85	0.93	0.82	0.87
PCA_PyODAD	0.48	0.78	0.28	0.41
HBO_PyODAD	0.63	0.85	0.52	0.65
AE_PyODAD	0.48	0.78	0.26	0.39
TFAutoFCNAD	0.87	0.94	0.85	0.90
TFAutoDeepFCNAD	0.87	0.93	0.86	0.89
TFAutoLSTMAD	0.72	0.89	0.64	0.74
TFAutoGRUAD	0.62	0.85	0.50	0.63
TFAutoRNNAD	0.86	0.89	0.90	0.89
TFAutoProphetAD	0.15	0.10	0.95	0.18
Kmeans_DPLAD	0.50	0.63	0.56	0.59
PCA_DPLAD	0.62	0.64	0.94	0.76

TABLE 5. Smart factory use case BACS results when testing with synthetic continuous anomalies (self-supervised results). Models operate in the Cloud layer, unless otherwise specified in the model name.

Model	Accuracy	Precision	Recall	F1
TFAutoAD (Edge)	0.47	0.14	0.88	0.24
SVM_SKLAD (Edge)	0.53	0.13	0.67	0.21
TFAutoAD (FG)	0.61	0.50	0.99	0.67
SVM_SKLAD (FG)	0.61	0.51	0.86	0.64
TFAutoAD	0.70	0.72	0.87	0.79
TFAutoDeepAD	0.64	0.76	0.64	0.70
TFAutoDeepVAEAD	0.72	0.78	0.80	0.79
TFAutoVAEAD	0.72	0.78	0.79	0.79
TFAutoWideVAEAD	0.72	0.78	0.80	0.79
EE_SKLAD	0.53	0.72	0.44	0.55
LOF_SKLAD	0.77	0.75	0.95	0.84
IF_SKLAD	0.74	0.74	0.93	0.82
SVM_SKLAD	0.72	0.73	0.90	0.81
ABOD_PyODAD	0.74	0.73	0.96	0.83
KNN_PyODAD	0.63	0.70	0.73	0.71
PCA_PyODAD	0.64	0.77	0.64	0.70
HBO_PyODAD	0.55	0.72	0.49	0.58
AE_PyODAD	0.65	0.78	0.63	0.70
TFAutoFCNAD	0.65	0.72	0.73	0.73
TFAutoDeepFCNAD	0.64	0.73	0.70	0.71
TFAutoLSTMAD	0.60	0.74	0.60	0.66
TFAutoGRUAD	0.53	0.68	0.51	0.59
TFAutoRNNAD	0.69	0.72	0.85	0.78
TFAutoProphetAD	0.12	0.10	0.98	0.18
Kmeans_DPLAD	0.54	0.66	0.60	0.63
PCA_DPLAD	0.51	0.60	0.72	0.65

synthetic anomalies, with respectable accuracy and F1-scores. It is important to mention that the F1-score is the main metric we use to compare models here, as accuracy can often be misleading in anomaly detection scenarios when working with imbalanced data (less anomalies compared to normal data). In the results table we can also see that the Principal Component Analysis (PCA_DPLAD) with differential privacy included performs well while the K-Means algorithm from the same family of algorithms fails to converge, mostly due to it being more suitable for unsupervised analysis (e.g., clustering).

The above experiment with the synthetic anomalies generation has helped us greatly to detect implementation errors and to choose which models to use in which layers of the architecture in order to have an optimized performance. This leads, for example, to the choice that at the Cloud layer we use recurrent neural network models (GRU – Gated Recurrent Unit, specifically).

C. SYSTEM RESPONSE EXAMPLE: DATA MANIPULATION ATTACK

A data manipulation event corresponds to the scenario when the data is changed either in transit or at rest. This event is detected by comparing the hash of the data against the hash stored on the blockchain element of DAC. The hash of the data is stored on DAC at the Field Gateway for the Smart Logistics data and at the Edge for the Smart Factory use case. This hash is checked before the data are stored, by the storage connector, and if there is a mismatch a data manipulation event is reported, which is the case where the data have been somehow changed in between the different layers and components of C4IIoT (in transit). An event like this would suggest that a certificate which allows access to the DFB (Data Fusion Bus) has been compromised and therefore has to be invalidated. A different case of data manipulation is at rest, where the data are changed after they are stored in the database. In this case the users of the AVT can request a check of the data, initiating again a comparison of the hash of the data against the hash that is stored on DAC, and they are notified if there is a mismatch.

IV. LESSONS LEARNED & CONCLUSION

In this paper we described our effort to design and implement a secure and modern Industrial IoT System while providing useful details which can help other researchers in developing similar systems. While working together, the entire consortium of the C4IIOT Project advanced and learned about different aspects of cyber security in IIoT systems. Through various levels of security layers implemented in the components described here, we collected several lessons on how IIoT systems can be hardened and safely used in the industry. Specifically, we draw the following lessons learned from the aforementioned development process.

In the IoT ecosystem, sensors are resource-constrained devices that are mainly used for fine-grained monitoring of the infrastructure and the environment. The commoditization of trusted execution environments (TEE, such as OPTIGA^{TM} TPM or Intel SGX) can ensure that these operations are performed in a trustworthy manner.

Regarding anomaly detection models, validation is important and can be difficult to define. For example, if we were to use only accuracy score as a performance metric for our models, we would have very biased models as the data is very biased (less anomalies compared to normal data). By using other performance metrics (e.g., F1-score) we were able to create better models with respect to the natural class imbalance in the data. Moreover, it is very important to have good understanding of model hyperparameters and how to effectively tune them in order to obtain high performing models.

With regards to on-device anomaly detection performance, due to a low computational power and small memory capacity, it was practically infeasible to train the edge node anomaly detection autoencoder directly on the edge node device. Many data points would have to be stored at the device to train a model exhibiting an acceptable level of accuracy. Moreover, the training of autoencoders is a computationally intensive optimization process usually performed in many iterative steps. Finally, low computational power prevents any serious model validation and tuning of model hyperparameters. Consequently, we adopt a scheme in which edge node autoencoders are trained offline and an inference engine for feed-forward neural networks is directly integrated into the firmware of the edge node device enabling autoencoder-based anomaly detection on pretrained models.

In our experiments with anomaly detection models, we found that it is important to have different anomaly detection model types in place in order to be able to detect different types of anomalies: sequence-type models such as GRU or LSTM neural networks could detect continuous anomalies (anomalies spanning multiple measurements/timestamps) better than lighter "fully connected" models. On the other hand, for the singular anomalies, both model types performed similarly well.

The Apache Kafka message queue as one part of the data fusion bus (DFB) provided a solution for communication between modules which was easily adopted by the other modules using high level programming languages but it is more challenging for other technologies. Kafka Mirror makers are a good solution where Kafka cannot be deployed as a cluster, but only if the flow of the data is one way.

The SDN Controller uses the OpenFlow protocol that is not flexible enough and contains some ambiguities. However, the SDN switches have to implement the standard following their interpretations of it. This leads to incompatibilities between vendors. The SDN switches must be controlled by the SDN Controller. In order to do that, they listen for instructions on a port. This has been identified as a potential security hole.

There are several attack types which we left somewhat unexplored as they are not relevant to our concrete system and use cases. For example, real time monitoring and detection of ransomware attacks remains to be explored in our future work. Techniques such as auto-quarantine of malware or buggy software, full network communication halt of potentially compromised components can be used to somewhat reduce risks of data loss in such scenarios.

To conclude, through our system definition, implementation and validation, we display our vision of a secure connected platform in smart factory and smart logistics environments. We considered concepts important in IIoT scenarios through procedures such as mitigation strategies, encryption, deep learning based anomaly detection, data validity, etc. Our proposed architecture may also be applied to other new IIoT systems with some minimal or moderate technical modifications, while our results can be used as a baseline for new research in the field. We also emphasize the independence and portability of several components in our system. Most of them can be easily adapted and used as standalone services in other systems. To reduce verbosity some implementation details have been left out, but we encourage the readers to explore our project website,[4] where we include all the detailed descriptions for all of the mentioned components.

ACKNOWLEDGMENT

This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

REFERENCES

[1] K. Gai and M. Qiu, "Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3590–3598, Aug. 2018.

[2] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ECC-based provable secure authentication protocol with privacy preserving for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3599–3609, Aug. 2018.

[3] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.-K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Comput.*, vol. 5, no. 1, pp. 31–37, Jan./Feb. 2018.

[4] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Performance optimization for blockchain-enabled industrial Internet of Things (IIoT) systems: A deep reinforcement learning approach," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3559–3570, Jun. 2019.

[5] M. Banerjee, J. Lee, and K.-K. R. Choo, "A blockchain future for Internet of Things security: A position paper," *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 149–160, 2018.

[6] C. Lipps, P. Ahr, and H. D. Schotten, "How to secure the communication and authentication in the IIoT: A SRAM-based hybrid cryptosystem," in *Proc. 20th Eur. Conf. Cyber Warfare Secur.*, 2020, p. 232.

[7] W. Zhang, W. Guo, X. Liu, Y. Liu, J. Zhou, B. Li, Q. Lu, and S. Yang, "LSTM-based analysis of industrial IoT equipment," *IEEE Access*, vol. 6, pp. 23551–23560, 2018.

[8] H. Yan, J. Wan, C. Zhang, S. Tang, Q. Hua, and Z. Wang, "Industrial big data analytics for prediction of remaining useful life based on deep learning," *IEEE Access*, vol. 6, pp. 17190–17197, 2018.

[9] L. Li, K. Ota, and M. Dong, "Deep learning for smart industry: Efficient manufacture inspection system with fog computing," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4665–4673, Oct. 2018.

[10] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the Internet of Things," *Sensors*, vol. 19, no. 9, p. 1977, 2019.

[11] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, "Hast-IDS: Learning hierarchical spatial–temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2017.

[12] M. Savic, M. Lukic, D. Danilovic, Z. Bodroski, D. Bajovic, I. Mezei, D. Vukobratovic, S. Skrbic, and D. Jakovetic, "Deep learning anomaly detection for cellular IoT with applications in smart logistics," *IEEE Access*, vol. 9, pp. 59406–59419, 2021.

[4] Project Website – https://www.c4iiot.eu/

[13] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1606–1616, Apr. 2018.

[14] P. Bellini, D. Cenni, P. Nesi, and M. Soderi, "Anomaly detection on IoT data for smart city," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Sep. 2020, pp. 416–421.

[15] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1686–1721, 3rd Quart., 2020.

[16] F. Ullah, H. Naeem, and S. Jabbar, "Cyber security threats detection in Internet of Things using deep learning approach," *IEEE Access*, vol. 7, pp. 124379–124389, 2019.

[17] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1963–1971, Mar. 2020.

[18] Y. Koizumi, S. Murata, N. Harada, S. Saito, and H. Uematsu, "SNIPER: Few-shot learning for anomaly detection to minimize false-negative rate with ensured true-positive rate," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 915–919.

[19] M. A. Husnoo, A. Anwar, R. K. Chakrabortty, R. Doss, and M. J. Ryan, "Differential privacy for IoT-enabled critical infrastructure: A comprehensive survey," *IEEE Access*, vol. 9, pp. 153276–153304, 2021.

[20] N. Holohan, S. Braghin, P. M. Aonghusa, and K. Levacher, "Diffprivlib: The IBM differential privacy library," 2019, *arXiv:1907.02444*.

[21] M. G. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, "Machine learning at the network edge: A survey," *ACM Comput. Surveys*, vol. 54, no. 8, pp. 1–37, Nov. 2022.

[22] N. Mohan and J. Kangasharju, "Edge-fog cloud: A distributed cloud for Internet of Things computations," in *Proc. Cloudification Internet Things (CIoT)*, Nov. 2016, pp. 1–6.

[23] S. Samonas and D. Coss, "The CIA strikes back: Redefining confidentiality, integrity and availability in security," *J. Inf. Syst. Secur.*, vol. 10, no. 3, pp. 1–25, 2014.

[24] V. Costan and S. Devadas, "Intel SGX explained," Cryptol. ePrint Arch., Paper 2016/086, 2016. [Online]. Available: https://eprint.iacr.org/2016/086

[25] *Elasticsearch*. Accessed: Oct. 25, 2022. [Online]. Available: https://www.elastic.co/

[26] *The National Vulnerability Database*. Accessed: Oct. 25, 2022. [Online]. Available: https://nvd.nist.gov/

[27] *OpenVAS—Open Vulnerability Assessment Scanner*. Accessed: Oct. 25, 2022. [Online]. Available: https://www.openvas.org/

[28] M. Shanahan, "The event calculus explained," in *Artificial Intelligence Today*. Berlin, Germany: Springer, 1999, pp. 409–430.

[29] A. Jaddoa, G. Sakellari, E. Panaousis, G. Loukas, and P. G. Sarigiannidis, "Dynamic decision support for resource offloading in heterogeneous Internet of Things environments," *Simul. Model. Pract. Theory*, vol. 101, May 2020, Art. no. 102019.

[30] T. M. Chen and S. Abu-Nimeh, "Lessons from Stuxnet," *Computer*, vol. 44, no. 4, pp. 91–93, Apr. 2011.

[31] Open Networking Foundation. *OpenFlow Switch Specification Version 1.5.1*. Accessed: Oct. 25, 2022. [Online]. Available: https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

[32] Faucet Organisation. *Faucet*. Accessed: Oct. 25, 2022. [Online]. Available: https://github.com/faucetsdn/faucet

[33] Linux Foundation. *Open vSwitch*. Accessed: Oct. 25, 2022. [Online]. Available: https://www.openvswitch.org/

[34] C. Correa, J. Robin, R. Mazo, and S. Abreu, "Intelligent decision support for cybersecurity incident response teams: Autonomic architecture and mitigation search," in *Proc. 16th Int. Conf. Risks Secur. Internet Syst. (CRISIS)*. Ames, IA, USA: Springer, 2021, pp. 91–107.

[35] S. Russell and P. Norvig, *Artificial Intelligence, A Modern Approach*, 4th ed. London, U.K.: Pearson, 2020.

[36] T. Frühwirth and S. Abdennadher, *Essentials of Constraint Programming*. Berlin, Germany: Springer, 2003.

[37] M. Triska, "The finite domain constraint solver of SWI-Prolog," in *Proc. Int. Symp. Funct. Log. Program.* Berlin, Germany: Springer, 2012, pp. 307–316.

[38] M. Triska. (2021). *The Power of Prolog*. [Online]. Available: https://www.metalevel.at/prolog

[39] J. Wielemaker, T. Schrijvers, M. Triska, and T. Lager, "SWI-Prolog," *Theory Pract. Log. Program.*, vol. 12, nos. 1–2, pp. 67–96, 2012.

[40] M. Weisfeld, *The Object-Oriented Thought Process*. London, U.K.: Pearson, 2008.

[41] P. Moura, "Logtalk-design of an object-oriented logic programming language," Ph.D. thesis, Dept. Comput. Sci., Univ. Beira Interior, Covilha, Portugal, 2003.

[42] J. Wielemaker, Z. Huang, and L. Van Der Meij, "SWI-prolog and the web," *Theory Pract. Log. Program.*, vol. 8, no. 3, pp. 363–392, May 2008.

[43] J. Haid, "Hardware-based solutions secure machine identities in smart factories," *Boards Solutions*, pp. 10–13, 2016.

[44] S. J. Taylor and B. Letham, "Forecasting at scale," *Amer. Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[45] *Clamav | Cisco Talos Intelligence Group*. Accessed: Oct. 25, 2022. [Online]. Available: https://www.talosintelligence.com/clamav

[46] *Apache Kafka*. Accessed: Oct. 25, 2022. [Online]. Available: https://kafka.apache.org/

[47] *Prometheus*. Accessed: Oct. 25, 2022. [Online]. Available: https://prometheus.io/

[48] *Grafana*. Accessed: Oct. 25, 2022. [Online]. Available: https://grafana.com/

GEORGE BRAVOS (Member, IEEE) received the Diploma degree in electrical engineering and computer science from the National Technical University of Athens (NTUA), in 2002, and the Ph.D. degree from the University of Piraeus, in 2008. From 2002 to 2008, he worked as a Research Engineer at the University of Piraeus on several international research projects. From October 2009 to 2013, he worked as an Adjunct Professor at the Technical Educational Institution of Chalkida, Greece. From 2013 to 2019, he was a Postdoctoral Researcher at the Harokopion University of Athens. From 2014 to 2019, he worked as an Assistant Professor and the Director of IT programs with Hellenic American University. Since 2016, he has been the Director of Research and Development of ITML, coordinating the management of more than 20 EU H2020 projects. He has published more than 30 papers in peer-reviewed journals and conference proceedings. He is a member of the Technical Chamber of Greece.

ANTONIO J. CABRERA received the B.Eng. and M.Eng. degrees in computer engineering from the University of Granada, Granada, Spain, in 2018 and 2019, respectively. He is currently pursuing the Ph.D. degree in secure and reliable communication protocols in the Industrial IoT networks with the Infineon Technologies AG, University of Granada, Neubiberg, Germany. He is involved in different research projects covering topics related to the Industrial Internet of Things, security, cryptography, and virtualization environments. His current research interests include hardware security, blockchain technologies, and IoT embedded systems.

CAMILO CORREA received the degree in information systems engineering from Universidad EAFIT, Colombia, in 2018, and the master's degree in computer science applied to business administration from Université Paris 1 Panthéon-Sorbonne, France, in 2020, where he is currently pursuing the Ph.D. degree in computer science. Since 2019, he has been working as a Research Engineer (ingénieur d'études) with the CRI Laboratory, Université Paris 1 Panthéon-Sorbonne. His research interests include the intersection of declarative programming and software product lines and their applications.

DRAGAN DANILOVIĆ received the M.Sc. degree from the Department of Telecommunication, University of Belgrade, with his thesis on ''Performance analysis of dual cell and MIMO technologies in HSPA network.'' He is a dedicated telecommunications professional with more than 14 years of experience. He has thorough knowledge and practical experience in implementing and operating all segments of modern Radio Access Network (RAN). He is highly interested in IoT technology with experience in HW and SW developing IoT devices. He is currently a Radio Access Operations Team Leader with A1 Srbija d.o.o, Serbia. In recent years, he was devoted to NB-IoT introduction in mobile networks, parameter optimization, and testing NB IoT modules from different vendors in the laboratory and live network. His research interest includes cellular IoT technologies (NB-IoT and LTE-M).

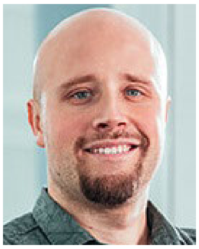
NIKOLAOS EVANGELIOU received the M.Eng. degree in electrical and computer engineering from the Technical University of Crete (TUC), Chania, Greece, in 2015, and the M.Sc. degree in information systems from the School of Information Sciences and Technology, Athens University of Economics and Business. He has been engaged in many projects related to big data and business intelligence, cybersecurity, and digital forensics. From 2018 to 2020, he worked as a Project Coordinator and Information Developer for Atos Greece, Athens Center of Excellence (CoE). He participated in multimillion-euro/dollar ICT projects covering a variety of roles both in technical and administrative level. In particular, he was responsible for establishing the project plan baseline, coordinating the technical teams and effectively cooperating with senior software engineers from Europe and America to create operational guidelines and solve any technical issues on documents. Furthermore, he has been a member of the Innovation Laboratories with a strong passion for new ideas and innovations. Since 2019, he has been a co-inventor of a patent in the domain of enterprise communications and collaboration. He is currently a Project Manager with ITML. In addition to all the above, he has experience in the field of financial technology (Fintech) having worked on solutions that facilitate the services of Banco Santander. Finally, he has been involved in EU-funded projects with a focus on information technology (cybersecurity, artificial intelligence, and machine learning). He is a member of the Technical Chamber of Greece.

GILAD EZOV received the B.Sc. degree in computer science from Technion—Israel Institute of Technology, Haifa, Israel, in 2018. Since 2017, he has been working as a Research Staff Member with the IBM Research Laboratory, Haifa. He has been working in the domains of security and privacy of data and AI, homomorphic encryption, cloud security, anomaly detection, and blockchain.

ZORAN GAJICA received the master's degree from the Faculty of Technical Sciences, University of Novi Sad. He is currently a Senior Radio Access Operations Expert with A1 d.o.o, Serbia, and a highly skilled professional with 17 years of experience in telecommunications. Besides his main focus on operations and maintenance of all generations of cellular mobile networks, his special interest is in IoT hardware and software development together with integration and testing of new devices in live mobile networks. He is currently involved in rollout and optimization of NBIoT and LTE-M technologies in live mobile networks.

DUŠAN JAKOVETIĆ (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisbon, Portugal, in 2013. He is currently an Associate Professor with the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Novi Sad, Serbia. His research interests include distributed inference and distributed optimization.

LEONIDAS KALLIPOLITIS received the B.Sc. degree in informatics and telecommunications and the M.Sc. degree in advanced information systems from the National and Kapodistrian University of Athens, Greece. He has been working as a Software Engineer, since 2008. He has great experience working as a Technical Coordinator of research projects and contributing to the analysis, design, and implementation of complex IT systems. He is particularly involved in web applications development and system integration in the domains of cybersecurity, digital forensics, and big data analytics. His research interests include novel data visualization techniques and latest cybersecurity concepts.



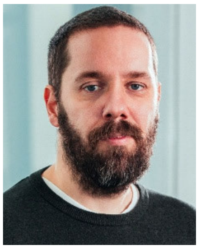
MILAN LUKIĆ (Member, IEEE) received the Ph.D. degree from the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 2015. He is currently an Assistant Professor with the Faculty of Technical Sciences, University of Novi Sad. He has been involved in numerous research projects covering topics of low-power embedded systems, data acquisition and monitoring, system automation and control, and energy efficiency. His research interests include embedded systems design, electronics, sensor and control systems in robotics, wireless sensor networks in general, and the Internet of Things.

JULIEN MASCOLO is currently a Senior Researcher and the Project Manager with the Factory Innovation Department, CRF. His areas of work in FCA regard the optimization of industrial processes (supply chain management, manufacturing, logistics, and product development process). In Stellantis, he coordinates internal projects related to the reengineering and optimization of the manufacturing processes.

DAVIDE MASERA received the master's degree in automotive engineering and the 2nd level master's degree (in "New technologies and processes for hybrid materials finalized to automotive mechanical and electronical applications") from the Politecnico di Torino, Italy, in 2015 and 2019, respectively. Since 2017, he has been working with Centro Ricerche Fiat as a Manufacturing Methods Specialist, where he was involved in innovation activities with Stellantis production departments and publicly funded projects (both Europeans and Italians). His fields of activity include flow optimization, physical element improvement, and data analysis applications development related to inbound logistics of the automotive assembly process.

RAÚL MAZO received the Engineering degree in informatics from University of Antioquia (Colombia), in 2005, and the M.S. degree in information systems, the Ph.D. degree in computer science, and the Habilitation to Lead Research (HDR) degree from University Panthéon Sorbonne, France, in 2008, 2011, and 2018, respectively.

He is currently working as a Full Professor at the École nationale supérieure de techniques avancées (ENSTA) Bretagne. He is also a Visiting Professor with EAFIT University. He is the Leader of the VariaMos Project and tool with which he has participated in several national, European and intercontinental research projects. He has published more than 100 scientific works on his research topics and received five awards for his contributions on these topics. His research interests include requirements engineering, variability management, and (dynamic) product line engineering.



IVAN MEZEI (Senior Member, IEEE) received the Ph.D. degree from the University of Novi Sad, Serbia, in 2012. He is currently working as an Associate Professor with the University of Novi Sad. He has been involved as a Principal or a Senior Researcher, a Work Package Leader, a Supervisor, or a Team Leader in a number of international research projects financed by the European Commission or by the companies. He has published two book chapters, less than 15 peer-reviewed journal articles, and less than 40 conference papers. His current research interests include LPWAN Internet of Things, localization and coordination in wireless sensors, actuator, and robot networks. He acted as a reviewer for less than 20 peer-reviewed journals and as a TPC Member for less than 40 conferences.

ANDREAS MIAOUDAKIS received the Diploma degree in electrical engineering in 1996 and the Ph.D. degree in electrical engineering from the University of Patras. He worked as a Research Engineer and a Teaching Assistant with the Applied Electronics Laboratory, University of Patras, and the Applied Informatics and Multimedia Department of TEI of Crete. He also worked as a Postdoctoral Fellow with the Telecommunications and Networks Laboratory, Foundation for Research and Technology—Hellas. He has a high participation in several research projects. He is currently the Chief Network and Communications Engineer with Sphynx Technology Solutions AG.

NEMANJA MILOŠEVIĆ received the B.S. degree in information technologies, the M.S. degree in software engineering, and the Ph.D. degree in computer science from the University of Novi Sad, Serbia, in 2014, 2016, and 2021, respectively.

Since 2016, he has been working as a Teaching Assistant with the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Serbia. His research interests include the development of novel deep-learning methods and their applications. He has published multiple related publications at different international venues.

WILLIAM OLIFF received the B.Sc. degree in software engineering from the University of Greenwich, in 2016. He is currently pursuing the Ph.D. degree in context-aware occupancy detection systems with the School of Computing and Mathematical Sciences, University of Greenwich. His research interests include indoor positioning, context awareness, the Internet of Things, and machine learning and applications.

JACQUES ROBIN received the Ph.D. degree in computer science from Columbia University. He previously worked at Sorbonne University, Paris, Thales Research and Technology, Palaiseau, France, and the Federal University of Pernambuco, Recife, Brazil. He is currently an Associate Professor with the Higher Education School of Informatics, Electronics and Automation (ESIEA), Paris; and a Researcher with both ESIEA's Laboratory of Data Science and Robotics (LDR) and the Center for Research in Informatics (CRI), University of Paris 1 Panthéon-Sorbonne. His research touches upon a wide range of topics in both artificial intelligence and software engineering.

MICHAIL SMYRLIS received the B.Sc. degree in computer science from the University of Crete. He is currently pursuing the Ph.D. degree as an external part-time student with the City University of London. Before joining SPHYNX, he worked as a Part-Time Research and Teaching Assistant at the City University of London, and a Software Engineer for EMPELOR GmbH (Switzerland). He is currently the Chief Software Engineer with SPHYNX Technology Solutions AG. He has expertise in the development of software solutions for platforms supporting big data analytics and security assurance. He has worked on several H2020 EU projects, including THREAT-ARREST, C4IIoT, CYRENE, HEIR, SPIDER, TOREADOR, and EVOTION.

GEORGIA SAKELLARI received the M.Sc. (M.B.A.) degree in Techno-Economic Systems and the M.Eng. degree in electrical and computer engineering from NTUA, Greece, and the Ph.D. degree in computer networks from the Imperial College London, in 2009. She is currently an Associate Professor of networked systems with the University of Greenwich, U.K. She is on the Editorial Board of *Simulation Modeling Practice and Theory* (Elsevier) and is serving as Expert Evaluator for the European Commission. Her research interests include computational offloading, edge computing, cloud computing, and network quality of service and security.

GIORGOS STAMATIS received the B.Sc. degree in information technology from Deree, The American College of Greece, Athens, Greece, in 2017. He joined Information Technology for Market Leadership (ITML), in 2018, first as a Software Developer and then as a Technical Manager. He worked in various EU and B2B projects. His research interests include software design and architecture, cybersecurity, and testing.

DUŠAN STAMENKOVIĆ received the B.S. degree in theoretical mathematics and the M.S. degree in data science from the University of Nisa, Serbia, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree in computer science. His research interests include deep learning, reinforcement learning, and graph neural networks. Since 2019, he has been working as a Scientific Researcher with the University of Novi Sad. He worked as a Research Intern at Telefonica, from 2020 and 2021, where he collaborated with Google. He has multiple publications in international conferences and journals.

SRĐAN ŠKRBIĆ is currently a Full Professor of computer science with the Division of Informatics, Faculty of Sciences, University of Novi Sad. In prior research, he was focused on applications of fuzzy logic in data storage and retrieval, while in recent years, he has focused on research topics in scientific and high-performance computing. He has published more than 50 research papers and participated in 17 international projects including H2020, Erasmus, Interreg transnational and cross-border, and SCOPES.

CARINE SOUVEYET has been a Full Professor in computer science with the University of Paris 1 Panthéon Sorbonne, since 2008, and the Dean of the Department of Mathematics and Computer Sciences, from 2015 to 2020. She has been a member of the CRI Research Team, since 1991. Her research activities, in recent years, aim to integrate into information systems engineering approaches, the specificities and innovations resulting from pervasive computing, infrastructures cloud computing, and the Internet of Systems with particular attention to context-based systems and systems with high variability (software product lines) which is summed up under the term pervasive information systems.

SPYRIDON VANTOLAS received the Diploma degree in electrical engineering and computer science from the University of Patras, Greece, and the M.Sc. degree in technoeconomic systems from the National Technical University of Athens, Greece. He has involved as a Entrepreneur and the Project Manager in several large-scale IT projects in the domains of telecommunications, marketing, news media, and travel. He has valuable experience in EU-funded projects as a Project Manager and a Technical Coordinator. He is involved in the field of business applications, big data analysis, marketing management and coordination, product commercialization, travel-related applications, dissemination and exploitation activities, and project management. He is a member of the Technical Chamber of Greece. Since 2019, he has been working as the Project Manager for AEGIS IT Research in several EU-funded projects under the H2020 Program.

GIORGOS VASILIADIS received the Ph.D. degree in computer science from the University of Crete, Greece, in 2015. He is currently an Assistant Professor with the Hellenic Mediterranean University and also a collaborating researcher with FORTH-ICS. Before that, he was a Scientist at the Qatar Computing Research Institute (2016–2017), and a Research Intern at Symantec Research Laboratories, USA, in 2013. He was a recipient of the Symantec Laboratories Graduate Fellowship and the Maria M. Manassaki Bequest Scholarship.

DEJAN VUKOBRATOVIĆ (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Novi Sad, Serbia, in 2008. From 2009 to 2010, he was a Marie Curie Intra-European Fellow at the University of Strathclyde, Glasgow, U.K. Since 2019, he has been a Full Professor with the Department of Power, Electronics and Communication Engineering, University of Novi Sad. His research interests include wireless communication systems and the Internet of Things.