IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 11, 2021, accepted July 20, 2021, date of publication July 27, 2021, date of current version August 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3100708

ELEGANT: Security of Critical Infrastructures With Digital Twins

BRUNO SOUSA, (Member, IEEE), MIGUEL ARIEIRO, VASCO PEREIRA, JOÃO CORREIA, NUNO LOURENÇO, AND TIAGO CRUZ, (Senior Member, IEEE)

CISUC, DEI, University of Coimbra, Pólo II—Pinhal de Marroco, 3030-290 Coimbra, Portugal

Corresponding author: Bruno Sousa (bmsousa@dei.uc.pt)

This work was supported in part by the Experiment Enabling Security with Digital Twin Units (ELEGANT) in the 7th Open Call of the Fed4FIRE+ Project through the Horizon 2020 Research and Innovation Program under Agreement 732638, in part by the European Commission, and in part by the Swiss State Secretariat for Education, Research and Innovation.

ABSTRACT The past years have witnessed an increasing interest and concern regarding the development of security monitoring and management mechanisms for Critical Infrastructures, due to their vital role in ensuring the availability of many essential services. This task is not easy due to the specific characteristics of such systems, and the natural resistance of Critical Infrastructures operators against actions implying downtime. Digital Twins, as accurate virtual models of physical objects or processes, can provide a faithful environment for security analysis or evaluation of potential mitigation strategies to be deployed in face of specific situations. Nonetheless, their on-premises deployment can be expensive, implying a significant CAPEX whose return will depend on the ability to plan and deploy a suitable support infrastructure, as well as implementing efficient and scalable data collection and processing mechanisms capable of taking advantage of the acquired resources. This paper presents an off-premises approach to design and deploy Digital Twins to secure critical infrastructures, developed in the scope of the ELEGANT project. Such Digital Twins are built using real-time, high fidelity replicas of Programming Logic Controllers, coupled with scalable and efficient data collection processes, supporting the development and validation of Machine Learning models to mitigate security threats like Denial of Service attacks. The validation approach of ELEGANT, which leveraged from the capabilities of the Fed4Fire federated testbeds evaluated the feasibility of using cloudified Digital Twins, thus converting a significant part of the projected CAPEX for the in-premises model into on-demand, pay-as-you-go OPEX, eventually paving the way for the establishment of a DTaaS (Digital Twin as a Service) paradigm. The achieved results demonstrate that the data pipelines providing support for the ELEGANT Digital Twins have low impact in terms of resource usage in Denial of Service and Distributed Denial of Service attack scenarios, when higher volumes of data are generated.

INDEX TERMS Digital Twins, SCADA, pipelines, security, programmable logic controllers, DTaaS.

I. INTRODUCTION

Modern automation technologies, deployed in modern Industrial Control Systems (ICS) or Industrial Automation Control Systems (IACS), have become pervasive [1], playing a crucial part in ensuring the availability of essential and critical services (e.g., Smart Grid, Water Distribution, etc). Such systems include many Internet of Thing (IoT) and/or sensing or control components which are instrumental to manage physical processes - thus, any disruption in their operation may have catastrophic results. For this reason, operators and service utilities are often heavily regulated by standardisation and steering organisations, in order to ensure proper quality, security and privacy requirements. Compliance with such standards and procedures necessarily means that operators have to plan and deploy proper control and monitoring mechanisms.

In light of the above, security monitoring solutions, along with other mechanisms for preventive and reactive purposes, play a key role in the protection of physical infrastructures and processes. But securing ICS is not an easy task, for reasons such as the substantial differences between these systems and their IT counterparts [2], their increasing complexity, or the considerable amount of legacy technologies

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

still in use, further aggravated by the fact that deploying security mechanisms or migrating away from obsolete technologies often cannot be performed without disruption of service or prohibitive costs.

Between several domain-specific security techniques, monitoring of infrastructure components, such as Programmable Logic Controllers (PLCs), is crucial both for operational purposes and to detect anomalous behaviour [3] - moreover because many monitoring techniques entail a low-overhead and low-risk approach to protect IACS/ICS. In this perspective, Digital Twins constitute an interesting development that may be leveraged to further evolve IACS/ICS security analysis and monitoring techniques.

The term Digital Twin (DT) has been coined by NASA [4] to refer to an integrated system that uses available physical models, sensor updates and other assets to mirror the behaviour of its corresponding flying twin. The Digital Twin approach was proposed to accelerate the certification and to facilitate fleet management and sustainability of aerospace vehicles. Nowadays, several areas have benefited with the introduction of Digital Twins, including the creation and management of complex distributed control systems, cyber-physical systems security, Industry 4.0 systems or Industrial IoT platforms, among other domains [5]. Indeed, companies like General Electric (GE) employ Digital Twins in manufacturing processes to monitor and manage the operation of assets [6]. The power of Digital Twins relies on the advanced monitoring capabilities, but also on the possibility of proactively assuming the role of the component being monitored.

The application of Digital Twins introduces benefits in terms of security, as well as in the design and prototyping of systems with demanding requirements, such as real-time remote-control applications in mission critical scenarios requiring low latency and high levels of security and reliability [7]. The real advantage of Digital Twins relies in its fusion with other technologies, like Artificial Intelligence (AI), for enhanced security analysis or to support Decision Support Systems (DSS). Besides enhancing security, Digital Twins also contribute to the optimization of the production processes in real-time [8], through big data analysis that support advanced analytics, as well as efficient and proactive monitoring.

DTs enable the development of evolved security assessment and monitoring techniques that are complementary to existing Intrusion Detection and Protection (IDPS) mechanisms. In fact, not only there is a high degree of compatibility as, in some cases, some security mechanisms may be leveraged for DT coupling purposes, as it is the case for the Shadow Security Unit (SSU) [9]. Being originally conceived to provide runtime monitoring of PLC devices, by intercepting both network traffic and physical I/O channels, SSUs can be easily repurposed as synchronisation mechanisms, feeding a DT with a continuously updated snapshot of a device state. This calls for the deployment of coupling mechanisms by means of data pipelines, providing a way to synchronise information across the physical and DT domains, as well as with any components used for in-replica analytics, eventually based on Machine Learning (ML) techniques and monitoring processes.

Despite the potential gains with the deployment of DTs, their on-premises implementation can prove to be expensive for IACS/ICS operators, due to the required investment to set-up and maintain the support infrastructure. Additionally, security and safety monitoring mechanisms need also to be adapted to synchronise with the DT replica, feeding a simulated virtual model that can be employed as the basis to build more evolved capabilities, such as Decision Support Systems [8].

In this scope, the ELEGANT (EnabLing sEcurity with DiGitAl Twins) project aims to validate the use of DTs, built with emulated control components at large scale, to establish a safe ground for analysis, development and validation of suitable security detection techniques, outside the scope of the production infrastructure. Instead of pursuing an in-premises deployment model, ELEGANT focuses on leveraging the capabilities of the Fed4Fire federated testbeds [10], in order to evaluate the feasibility of using cloudified DTs, thus converting a significant part of the projected CAPEX for the in-premises model into on-demand, pay-as-you-go OPEX, eventually paving the way for the establishment of a DTaaS (Digital Twin as a Service) paradigm.

Overall, the contributions of the ELEGANT project can be summarised as such:

1) Design and validation of a data pipeline solution to secure critical components within the Fed4Fire+ project [10] federated testbeds.
2) Design and validation of an off-premises approach to deploy Digital Twins reducing the CAPEX, within adequate levels of performance and security.
3) Design and validation of ML models to identify different type of DoS attacks targeting critical infrastructures, considering the Digital Twin model information.
4) Large scale experimentation with different deployment models for Container Network Functions (CNFs) or Virtual Network Function (VNFs), which are aligned with the service containerization and microservice decoupling trends, also characteristic of the 5G service-driven reference architecture.
5) Publicly share the datasets that were collected to enhance the training and validation of the ML models to enable Digital Twins. The details on the datasets are fully disclosed at [11].

This paper will focus on the communication stream analysis and learning components required to enable efficient digital replication of critical systems through DTs, as well as demonstrating their use to develop and validate ML-based detection techniques for network attack detection.

The remainder of the paper is organised as follows: Section II introduces the background associated with Critical Infrastructures and overviews relevant works. Section III provide the motivation for the ELEGANT project, while Section IV introduces the ELEGANT architecture to

enable DTs and models to detect different types of DoS attacks. Section V details the reference scenario that was established to validate the ELEGANT off-premises approach, in a distributed testbed. Section VI documents the achieved results, and Section VII concludes the paper.

II. BACKGROUND

This section intends to introduce the reader to several relevant aspects regarding the Critical Infrastructure (CI) security and Industrial Control System (ICS) domains, with an emphasis on the technologies and topics deemed relevant in the scope of the ELEGANT project.

A. CRITICAL INFRASTRUCTURE SECURITY

When it comes to cybersecurity, the CI domain constitutes a multidimensional challenge, in the sense that the topic involves several aspects such as the protection of physical process and communications infrastructures or hardware/-software lifecycle and asset management, just to name a few. Due to the specific characteristics of such infrastructures, these aspects often cannot be adequately handled using strategies and tools inherited from the IT world [12], requiring a domain-specific approach.

As ICS become more complex and distributed, with an increased number of attached actuators/sensors and smart devices, existing security threats are also becoming more challenging. This situation is further aggravated by the fact that many vulnerable legacy ICS technologies are still widespread, many of them designed at a time when operating safety and reliability were primal design concerns, discarding cybersecurity aspects. To some extent, the primacy of availability as the main concern over all other aspects influenced the CI mindset to consider technological maturity as a guarantee of reliability.

ICS, and particularly Supervisory Control and Data Acquisition (SCADA) systems - which constitute the focus of this paper, are often vulnerable to cyberattacks [13] against field devices (e.g., sensors or actuators), control elements (e.g., PLCs, Remote Terminal Units), or even control center/process control devices like Human Machine Interfaces (HMI), or other components. As a result of this situation, incidents involving SCADA systems have been in the headlines over the past years [13], [14]. An example of this is the recent attack against a water treatment facility, involving the modification of the dosage of chemical components [14]. To better understand the specific reasons for this, the next subsections will delve into some of the most relevant aspects regarding such architectures.

B. SCADA SYSTEMS

SCADA systems have been one of the cornerstones of modern ICS technology, being used on a diverse range of industries and infrastructures, from power generation and distribution to water and gas distribution or factory automation. Such systems made their debut in the 1960s, having inherited several characteristics along the way, many of which survive up to this day. In its simplest form, SCADA systems, as illustrated in Figure 1, may include the following components:

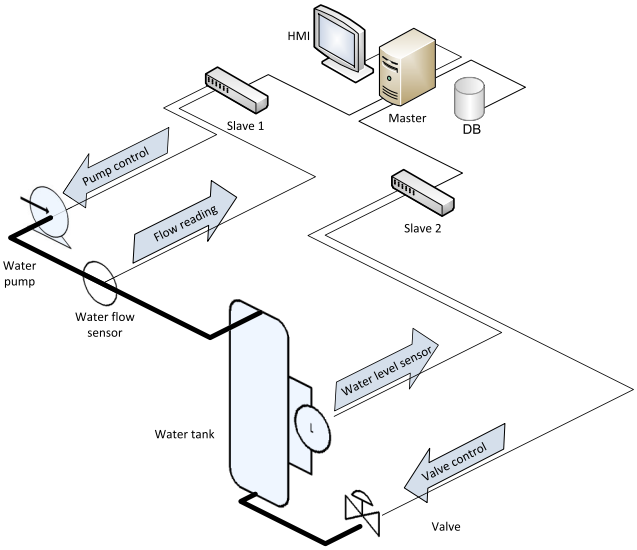

FIGURE 1. Example of a basic SCADA system architecture (adapted from [15]).

- **Master Stations** handle process supervision tasks, controlling and monitoring slave devices - such hosts are usually placed on the process network. Frequently, such stations also encompass HMI (Human-Machine Interface) capabilities, for process visualization, as well as being connected to services and applications, such as historian databases, for process-related data logging.
- **Slave devices**, among which RTUs (Remote Terminal Units) and PLCs are included, are located on the control network level. These devices, usually based on embedded systems, are connected to the field-level sensors and actuators, constituting the cyber-physical mediation layer of the SCADA system, being in charge of most process monitoring and control activities, under supervision of one or more Master Stations. RTUs have limited processing and control capabilities, being attached to process sensors/actuators under control of the master stations - by comparison, PLCs are more capable and autonomous, supporting different programming languages and more sophisticated control capabilities. PLCs are often used instead of RTUs in many application scenarios, due to a more appealing cost/capability ratio.
- **Field devices** constitute the physical interface with the supervised process, both for data acquisition (via sensors) and/or execution of actions controlling its behaviour (via actuators). Such components are deployed on the field network level.

Such components are linked together by means of protocols and technologies, such as CAN Bus [16], RS-485 [17], Industrial Ethernet [18], COTS Ethernet or TCP/IP. Communications between the Master station and the PLC/RTUs is

supported by using SCADA-specific protocols, like Modbus over TCP/IP [19], IEC 60870-5-104 (IEC 104) [20] or the Distributed Network Protocol 3 (DNP3) [21], among others, making it possible to acquire process data, as well as to execute actions.

Particularly, SCADA communication protocols such as Modbus [22], developed by Modicon in 1979, are still widespread in production environments. Modbus is a client-server application layer protocol, being often used to support communications between automation devices such as PLCs or RTUs and the supervisory stations, being considered *de facto* industry standard [23]. The original version of the protocol (Modbus Serial RTU) was later adapted for TCP/IP frame encapsulation (Modbus TCP variant), which the version normally used on Ethernet LANs. Its simplicity, which is part of the reasons for its popularity (as per the framing structure depicted in Figure 2), also makes it insecure. This is due to the fact that Modbus does not support native protection mechanisms, such as encryption. Indeed, only recently the ModBus/TCP security protocol was specified, relying on TLS to enable encrypted data transport, payload integrity protection, and resistance against replay attacks [24]. Nonetheless, the Modbus security protocol has not been widely deployed due to its potential overhead in terms of computing and hardware resources, which renders it incompatible with the limited capabilities of many SCADA devices, such as certain RTU/PLC models [25], [26], while also being incompatible with existing equipment.

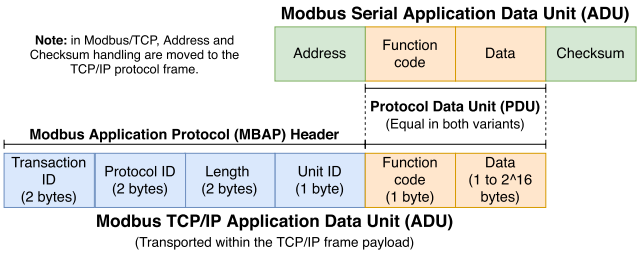

FIGURE 2. Modbus framing format.

C. NETWORK ATTACKS AGAINST SCADA SYSTEMS

Successful attacks against SCADA targets are often the ultimate result of several exposed weaknesses and vulnerabilities which provide suitable intrusion or disruption vectors for further exploitation. These may be leveraged to deploy network, service or process-level attacks, whose ultimate outcomes may range from loss of process visibility to asset destruction and even loss of human life, in extreme cases.

Among those, network-level Denial of Service (DoS) attacks will constitute the main concern for this paper. Such attacks constitute a serious threat to critical infrastructures, which can explore different techniques [27] such as resource exhaustion, for service degradation (i.e., disturb data collection or monitoring processes), or vulnerability exploitation, to explore flaws or holes in applications or in the design of protocols. Network-level DoS attacks are commonly performed from spoofed IP addresses (i.e., false IP addresses) or hijacked devices to avoid/hamper identification of their origin – when performed using multiple nodes, such attacks lead to Distributed Denial of Service (DDoS) incidents.

Flooding and amplification are two types of DoS attacks that may impact several devices in a ICS, leading to network and computing resource exhaustion [27]. For instance, SYN Flooding involves sending a high volume of SYN messages to a critical target (e.g., PLCs) in ICS. This type of attack exploits the TCP three-way handshake mechanism, to target either the destination IP of the stream or a third-party device, that will receive the ACK messages because the source IP was spoofed (in this latter case it becomes a reflection attack). This may cause the exhaustion of resources in the target nodes (half-open sockets consume resources) or at the communications network level (by consuming the available bandwidth).

Amplification attacks rely mainly on UDP traffic, where small size packets are sent to the targeted components. Such traffic, with the origin in spoofed IP addresses, leads to saturation of the available bandwidth, impacting protocols like Modbus/TCP. Such DoS attacks impact the response times of Modbus requests, the performance of the protocol and the monitoring of Modbus devices [28]. Indeed, many critical infrastructures are very sensitive to delay (and delay variations), thus any impact on the performance of the protocol in terms of response time, and the number of requests that can be handled in specific time periods, may have serious consequences.

D. THE CASE FOR DIGITAL TWINS

It is recognised that critical infrastructure operators must deploy stronger, adaptable and resilient cyber-defence solutions. However, designing these solutions is not a straightforward process, as it is well known that there is no single solution capable of addressing the specific requirements and needs of all CI. This is further true if we consider that the most sophisticated attacks against cyber-physical systems usually take advantage of process-specific knowledge to maximise their impact.

Standards such as the ISA/IEC 62443 [29] series have contributed to fill the gap regarding the establishment of security guidelines specifically dedicated to ICS. Such guidelines are perfectly coherent with the ISO 270xx standards series, being complemented by other recommendations such as the NERC CIP requirements [30], the NIST Guide to ICS Security [31], the ENISA Recommendations for ICS Protection [32] or the IAEA Computer Security at Nuclear Facilities Guide [33], just to name a few.

Despite these efforts, CIP protection remains a delicate issue due to its specific characteristics, which often advise against deploying inline monitoring mechanisms or executing aggressive pentesting campaigns (as recommended by NIST SP800-82 guidelines [31]), due to the possible cost of downtime associated to a potential failure or

unsuccessful procedure. Thus, operators are often advised to resort to simulation or out-of-band mechanisms to deal with vulnerability or risk assessment, as well as predictive analysis of potential cascading faults.

From this perspective, Digital Twins constitute an interesting approach - these consist of virtual systems providing a real-time digital counterpart (a ''live replica'') of a physical object or process. DTs necessarily involve emulating parts of the system, using real inputs collected from the field to feed models, which allow to make comparisons with the real operating parameters, for ongoing monitoring. Their development necessarily requires at least two fundamental resources: modelling, at least for parts of the system, and proper data collection mechanisms to feed such models.

The benefits of DTs are manifold: they can provide a real-time ICS replica whose behaviour can be accurately compared to the production system, in order to search for faults or potential incidents and, if properly designed, a DT can also provide a faithful environment for security analysis or evaluation of potential mitigation strategies to be deployed in face of specific situations. When coped with minimally invasive mechanisms such as the Shadow Security Unit (SSU) [3], [34] DTs can be fed with real-time data collected from the production environment.

Nevertheless, implementing an on-premises DT can prove to be expensive for an ICS operator, due to the required investment required to set-up and maintain the entire support infrastructure, both for the DT and also for the required security and safety monitoring mechanisms.

III. MOTIVATION

The ELEGANT project establishes its motivation considering the security issues in Critical Infrastructures (CI) discussed in the previous section, namely Denial of Service (DoS) attacks. ELEGANT also considers Digital Twins that relying on current communication protocols trends like the Modbus/TCP provide off-premises solutions to enhance the security and monitoring in CIs.

Motivation #1 Usage of insecure protocols such as Modbus/TCP [19] is still widespread [3] for reasons such as hardware and computational overhead or compatibility with existing systems, as the migration to secure protocols relying on TLS such as the Modbus Security protocol [24] would imply extensive infrastructure updates with associated downtime. In this regard, approaches that reduce the impact on production systems (e.g., no replacement of devices/sensors or firmware upgrade), but add extra security levels, are needed.

Motivation #2 Standards devoted to ICS security, such as NIST SP 800-82 [31], advise against the execution of many security assessment procedures in production environments, due to the sensitive nature of automation equipment and processes. Additionally, several ML techniques require training based on specific data sets for each use-case scenario - a requirement that transfer learning approaches may not fully address and which requires experimentation on the production environment, something that is out of question for most operators. In this regard, solutions that rely on DTs, supported by efficient data collection mechanisms, can provide safe, off-path and even off-premise environments for analysis and development of security solutions.

Motivation #3 Assess how the design of Digital Twins can be promoted to enhance security analysis without downtime in critical components. The design of Digital Twins for enhanced security levels is not a trivial task, either in the modelling or data collection processes. Indeed, the complexity can scale according to the dimension, number of devices in the ICS infrastructure or may require costly and customised hardware solutions. Enabling DT as accurate virtual models can rely on the advances of virtualization techniques like containers/micro-services.

Motivation #4 The deployment of on-premises DTs is also a complex task and can introduce high costs, as well as adding more complexity to management processes that are already complex. The ELEGANT project decided to go for an off-premises approach, deploying the DT on the Fed4Fire project [10] with federated testbeds. The rationale for this strategy was focused on evaluating an alternative approach for DT deployment, allowing ICS operators to shift the bulk of the CAPEX into OPEX while maintaining an adequate level of performance.

IV. ELEGANT: SCALABLE AND SECURE CRITICAL COMPONENTS

This section details the ELEGANT architecture to enable off-premise Digital Twins, the approach for scalable and secure data pipelines to enhance the DTs' accuracy and ML models to detect different types of Denial of Service attacks.

A. ELEGANT ARCHITECTURE

The ELEGANT architecture is designed to enable Digital Twins following an off-premise approach with scalable data pipelines to secure and monitor critical infrastructures. Data pipelines correspond to the path on which data is transmitted, stored, processed and analysed. The data pipeline constitutes the fundamental mechanism allowing for the communication of data and the respective processing and analysis supported by ML models [35]. Also, the same data pipelines allow for bridging the physical and DT domains, providing the means to synchronize them, by providing the state information which is necessary to feed the simulated/emulated components.

The ELEGANT project also considers critical elements that can be distributed over different locations - remote sites, and a central site to process the collected data from distributed sites. In this regard, the following components are required at all sites:

- **PLC slave(s)** - act as SCADA slave devices that are connected to the field-level sensors/actuators on each site. To monitor and manage devices, PLC slave(s) components implement the control logic in the form of PLC programs.

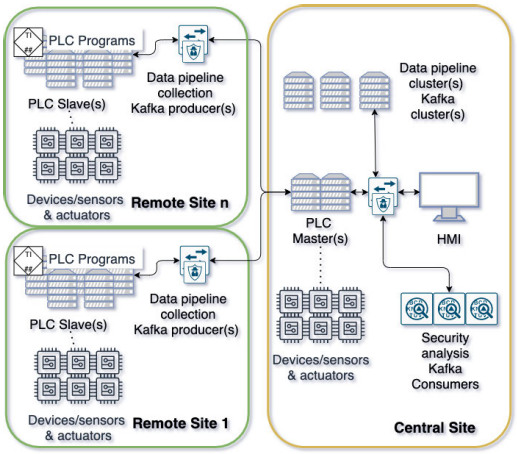

FIGURE 3. Proposed architecture for secure data pipelines.

- **Data collection elements** - compliant with scalable approaches for data collection. For instance implementing Kafka producer(s) that stream data to Kafka client(s).

The central site, besides including all components required in all sites, also includes a set of components to allow security analysis and other processes in the data pipeline, including:

- **Data Pipeline Cluster(s)** - nodes organised in cluster(s) with efficient mechanisms to allow the processing of high volumes of data in real-time, and with data retention policies for off-line analysis. Such clusters can be implemented with multiple Kafka broker nodes that enable stream processing in a scalable fashion and for different types of application/services [36].
- **Human Machine Interface (HMI)** - nodes with security dashboards for human monitoring. Dashboards are organised per site and aggregate the monitoring information regarding the status of components.
- **PLC Master(s)** - nodes performing the monitoring and actuation in the critical components through Modbus TCP/IP traffic. These components, along with HMI, assure the functionalities of the SCADA master devices.
- **Security Analysis elements** - components with ML models performing the analysis of the streamed data to detect Denial of Service attacks. Such elements can be instantiated as Kafka consumers consume data per required flow analysis. This allows to have distinct types of analysis per site, for instance considering different threat models.

The architecture also includes monitoring services in all the components to allow an integrated approach assessing resource usage and the behaviour of components.

B. CRITICAL COMPONENTS DISTRIBUTION

Figure 4 illustrates the deployment of the ELEGANT architecture in multiple sites, which are interconnected through a central site. Such kind of ICS deployment is also vulnerable to different type of threat models: i) external to the sites, where malicious users attack ICS components from the outside network (i.e., from Internet); ii) inside each site, where attackers can be personnel managing the ICS infrastructure, considered as insider threats.

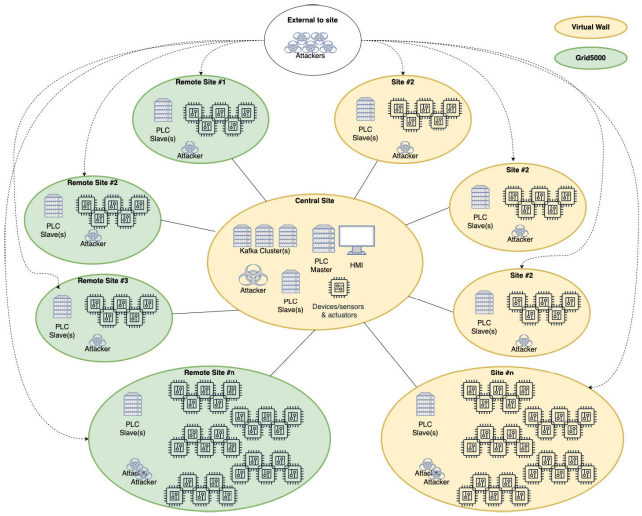

FIGURE 4. Multi site scenario.

The distribution of components can also consider different factors such as geographical locations, as exemplified in the Figure 4, where the Virtual Wall sites are on Belgium and Grid5000 are on France.

Key enablers for the distributed critical components include security mechanisms between the interconnection of sites like Virtual Private Networks (VPN), the Modbus/TCP protocol [37], and monitoring agents. As stated earlier, the Modbus/TCP protocol is one of the most employed protocols in critical infrastructures. The PLC nodes supporting Modbus/TCP, sense data from in-field devices and stream it to the PLC master nodes in the central sites via polling mechanisms.

Figure 5 illustrates how the information of the distributed devices is collected and modelled to enable Digital Twins. Each PLC slave implements the control logic, as specified in the PLC programs, which are developed in the Ladder Logic language [38], using graphical diagrams to express the circuits (i.e., connections to devices) and the relay logic. Within PLC programs and the Modbus/TCP protocol the information of devices is mapped into specific type of registers considering the data and controls supported by the respective device. The discrete input coils (i.e., identified as %IX) contain values of devices with two possible states: connected - 1/true, and disconnected - 0/false. The analog registers (i.e., identified as %IW or %QW depending if they are input or outputs) allow to store information from devices which measurements vary in a scale/range and that can be mapped to 16 bits (e.g., water level sensors).

The registers of each PLC client are mapped into the PLC master to enable the functional modelling of each PLC distributed in the diverse sites. This mapping relies on the polling mechanisms of Modbus/TCP, which have also associated specific function codes, according to the operation

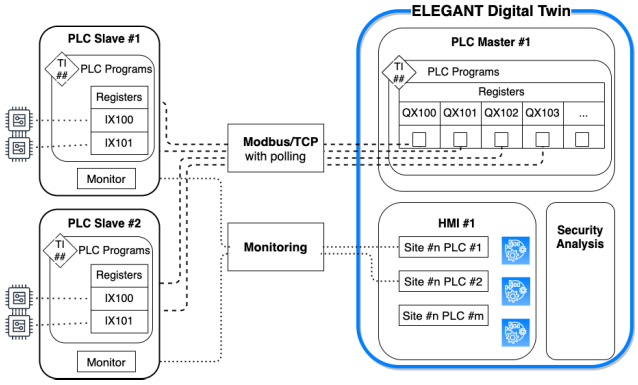

FIGURE 5. Distribution of critical components.

to be performed. For instance, the function code 0×02 implies reading discrete input coil registers, while the function code 0×05 writes values to output coil registers.

The monitoring component also enables the aggregation of data in the HMI, using an approach that relies on time-series databases (e.g., Prometheus, InfluxDB, etc), on which informational dashboards can be designed to convey real-time status information of the monitored devices, as well as their resource usage.

C. SCALABLE PIPELINES FOR SECURITY

Human Machine Interface (HMI) components are relevant in critical infrastructures to gather information regarding critical components in informative dashboards for human operators. HMI components, besides providing real-time status of the critical components, also enable simple mechanisms to detect abnormal behaviour, based on simple rules/thresholds. For instance, if metrics are not polled within the configured interval, an alarm can be generated, as well when the values in the PLC addresses change quickly. Nonetheless, they can provide erroneous information to operators on successful attacks like Man-in-the-Middle (MiTM).

Secure data pipelines can include components and functionalities that are able to collect critical traffic, exchanged between Modbus slave and Modbus master nodes, and propagate the gathered data streams for enhanced security analysis.

Data pipeline cluster(s) can rely on the distributed streaming capabilities of Kafka for different types of application/services [36]. It is employed in several domains, ranging from IoT security, data collection for automation, event processing in networks, among others. Kafka also supports a failure tolerant architecture with multiple brokers configured in cluster mode. Data is replicated between cluster nodes according to configured policies, the most frequent topic data is stored in specific partitions for performance or reliability purposes.

In the ELEGANT architecture, depicted in Figure 3, the data collection points or Kafka producers are distributed in the diverse sites, to allow the collection of data in a scalable fashion. Such collection points capture network level data, which includes the Modbus/TCP traffic and do some processing (e.g., data format conversion) and stream it to the Kafka cluster(s).

Within the complete set of data from PLCs, enhanced security analysis can be performed. The network level along with the critical information data (i.e., Modbus/TCP) allow Digital Twins to identify with higher accuracy attacks targeting PLC nodes. Each critical component like the PLCs in each site, also includes a monitoring component inspecting the information in the Modbus registers and resource usage feeding the real-time dashboards in the HMI. Despite the duplication of data, this approach is useful for accuracy in the Digital Twins components, as information does not solely relies on a single origin/process, which could be compromised or not considered trustful.

D. SECURITY ANALYSIS WITH MACHINE LEARNING

Security analysis in critical infrastructures with SCADA components has been relying on correlation engines [39], that do not scale to complex infrastructures. In particular, the engines rely on static rules, which are complex to manage and have limited detection scope. Early versions of Shadow Security Unit for SCADA systems [34] used an embedded correlation engine for event aggregation and processing purposes, whose outputs was sent to a Security Information and Event Management (SIEM) service (often based on classic rule-based correlation engines). More recently, SSUs have evolved to include advanced security mechanisms based on AI models without the restrictions associated with pure, rule-based, correlation technologies [9].

Modern IACS/ICS security solutions have evolved towards incorporating anomaly-based techniques based on data-driven approaches or ML models, to deal with threats such as Denial of Service (DoS) and Man-in-the-Middle (MiTM) attacks. In such context, solutions have considered aspects related to the computing capabilities of critical components [40], as well as the capabilities for distributed processing [41].

The detection of DoS threats can rely on models that detect SYN flooding attacks by analysing the arrival rate, or difference between SYN and SYN-ACK packets, or SYN and FIN packets, which should be coherent as per the handshake schemes of TCP/IP [42]. The analysis relies on statistical features of the TCP/IP headers and employs different types of ML models like Decision Trees and Artificial Neural Networks for flooding detection. With accuracy concerns other ML techniques are explored to detect DDoS in industrial IoT networks [43]. For instance, Random Forests are combined with Artificial Neural Networks to achieve accuracy levels of 99%, and considering features like network flow characteristics such as packet-length, interval between packets and the protocol used.

Other approaches [44] employ Deep Learning algorithms but with the restriction of requiring more computational resources and increased training times of the deep neural networks. Additionally, the data used for training is not always available, limiting the reproducibility or employment of such models in critical infrastructures.

The ELEGANT architecture enables off-premise data collection, and off-path security analysis, thus reducing the potential impacts on running processes of critical infrastructures. The data required to train ML models for anomaly detection can also be streamed to the Kafka cluster(s) without disrupting PLC nodes. As an example, attacks to a 'virtual' PLC slave (i.e., not participating in the monitoring of critical processes) can be performed considering the patterns of DoS attacks [45]. Such data within the relevant features is then fed into the data pipeline for analysis with ML models.

V. REFERENCE SCENARIO

This section details the reference scenario that was employed to evaluate and assess the performance of the ELEGANT architecture as enablers of accurate Digital Twins.

A. EXPERIMENTATION TESTBED

The distributed ELEGANT architecture was evaluated in Grid5000 (g5k), virtualWall1 (vWall1) and virtualWall2 (vWall2) Fed4FIRE+ testbeds [10]. The federated testbeds are made available by the Fed4FIRE+ consortium for research. The vWall1 testbed was employed to validate solutions for the components pictured in Figure 6.

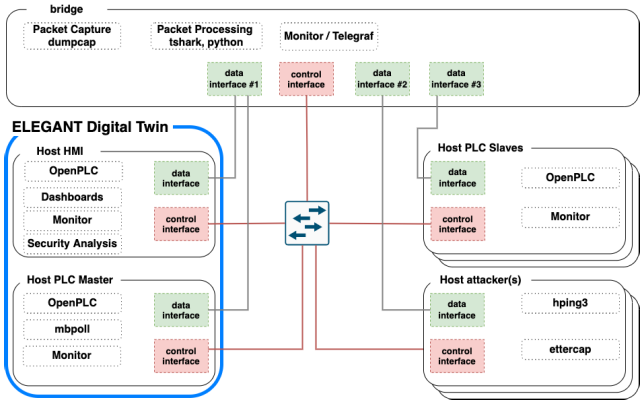

FIGURE 6. Experimentation scenario and nodes interfaces.

The central site was deployed in the vWall2 testbed and the remaining sites were implemented in the g5k and vWall1 testbeds. Since the g5k testbed is located in France, while the others are in Belgium, there was the need to interconnect such sites using VLAN technologies available in the interconnection links between testbeds managed by different entities.

The components have data and control interfaces, as highlighted in Figure 6. The data interfaces include the exchange of critical information between PLC slaves and PLC masters via the ModBus/TCP protocol. This information is mainly employed to monitor and control critical devices, for instance to stop a water pump. OpenPLC [46] is employed as the open-source solution to enable PLC slave and PLC master nodes, as it brings support for Modbus/TCP, includes APIs to upload PLC programs, and supports other communication protocols like DNP3 [21].

TABLE 1. Reference nodes in the experimentation.

Node(s)	Intel Platform	Description
bta1, btsiteb	Xeon E5645	Bridge nodes at central site and site B for data collection.
plca1, plca2	Xeon E5645	PLC master and PLC slave at site B.
kafkasrv	Xeon E3-1220	Kafka cluster node.
hmi	Xeon E3-1220	Human Machine Interface with security analysis.
Attack	Xeon E5645	Attacker nodes.

The control interface of nodes uses a specific physical network card and is relevant to manage the data collection process. For instance, through such interface the PLC nodes can be instrumented to modify the polling intervals, which are set at 100ms by default. The bridge nodes are placed in the path of data interfaces of critical components to enable data collection points without impacting critical information flows (Modbus/TCP) exchanged on the data interfaces. The data collection points placed at the bridge nodes stream the data to the Kafka cluster(s), as per the configured Kafka topics, thus enabling the handling of high data volume in a scalable fashion.

The components were deployed in physical servers with different computation power, as summarised in Table 1, that depicts the nodes' information employed in the experiment participating actively in the data pipeline.

B. MODBUS/TCP SETTINGS

Regarding the reference architecture, there are both horizontal (PLC-PLC) and vertical (PLC-HMI) communication patterns. The PLC master, placed in the central site has multiple configured slaves, as per the PLC slaves in the diverse sites. The polling mechanisms of Modbus/TCP to collect critical information is configured in intervals of 100ms (default in OpenPLC). In addition, the PLC master is configured to map all the registries of PLC slaves in different addresses, considering the type of PLC programs running in the PLC slaves of the distributed sites.

On real-world deployments the polling interval can be reduced to values around 40ms, leading to rates around 25pkts/s [47]. Despite introducing higher volumes of traffic in the network, it has the advantage of allowing a faster detection of anomalous behaviour. The value of 25pkts/s is considered as the reference rate for the Modbus/TCP polling process. In other words, values bellow this threshold trigger an alarm in the HMI dashboard.

PLC master node(s) query the diverse PLCs slaves, which implement the sensing logic to gather information from sensors and to actuate as per the control logic of PLC programs. In the experimentation scenario the PLC slaves have the role of sensing and actuating on the collected data (see next section). OpenPLC was also employed as it allow to emulate devices with different characteristics (e.g., ESP8266, Arduino).

C. WATER LEVEL PLC PROGRAM

Each PLC slave executes the logic specified in the Water level PLC program. As stated, OpenPLC provides APIs and a web interface to upload such programs on PLCs. Figure 7 depicts the Water level program in execution, after the successful compilation of the instructions specific in the Ladder Logic language.

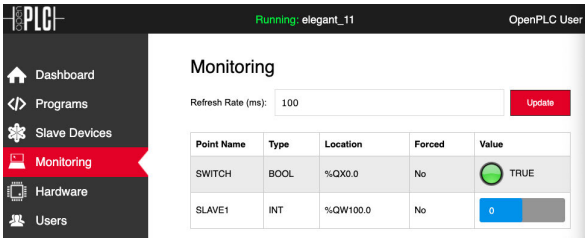

FIGURE 7. ELEGANT water level PL program.

The underlying logic considers a process for water level control in a tank, using a level sensor and a water pump (labelled as SWITCH in Figure 7) which is activated if a certain threshold (corresponding to setpoints in the PLC Ladder Logic language) is achieved. The Water level program, defines a setpoint of 10, as per line 2 in the program listing 1. The Water level program, performs actuation by considering the monitored values and their attachment to the setpoint (line 6). If exceeding is verified (line 8) then actuation is performed (line 9), otherwise the actuation considers the default value (zero, as per line 11).

```
1   VAR
2      setpoint : INT := 10;
3      GT2_OUT : BOOL;
4   END_VAR
5
6   GT2_OUT := GT(SLAVE1, setpoint);
7   SWITCH := GT2_OUT;
8   IF SWITCH=TRUE THEN
9      ACTUATOR := 1;
10  ELSE
11     ACTUATOR := 0;
12  END_IF;
```

Listing. 1. Code fragments of the PLC program.

D. DENIAL OF SERVICE ATTACKS

Two types of Denial of Service attacks have been considered, as per their impact and probability of occurrence in critical infrastructures [27]:

- **Amplification/Volumetric** based on UDP traffic with spoofed IP addresses with small UDP packets (60 bytes) aiming to overload the network of critical components, more specifically the data interfaces networks.
- **Flooding** based on TCP SYN traffic with spoofed IP addresses, which is sent with different rates and with packet size of around 120 bytes.

DoS attacks have been performed considering multiple configurations. Each test was performed using the *hping3* tool [48] with the ability to randomise source node IP (*–rand-source* option). In addition, two traffic rate classes were considered for the inter-arrival packet times. Rate 1 includes packet interval rates in a ratio of 10 times higher than the normal rate of the Modbus/TCP polling process. Rate 2 assumes the maximum flood that can be performed with the attack node (*--flood* option), as summarised in Table 2.

TABLE 2. DoS attack rate settings.

Type	Rate	Interval (ms)	N. packets
Flood	1	40	2500
Flood	2	< 1	Max. supported by node
Amp	n/a	< 1	Max. supported by node

The attack tests using the hping3 tool were also performed in a distributed fashion (identified as DDoS), using multiple nodes in the testbeds. The synchronisation of attacks was performed through the jFed multiple commands functionality. The DoS targeted the PLC master in the central site, acting as the SCADA master, polling information from PLCs in other sites. Both types of attacks were instrumented to use port 502 as the destination port, which is the default port of Modbus/TCP.

E. DATA COLLECTION

The data collection was performed at the bridge nodes, acting as data collecting points, through the use of dumpcap, tshark tools [49] and custom Python scripts. The data processing in the Data pipeline includes diverse steps: capture, processing, and retention as illustrated in Fig. 8.

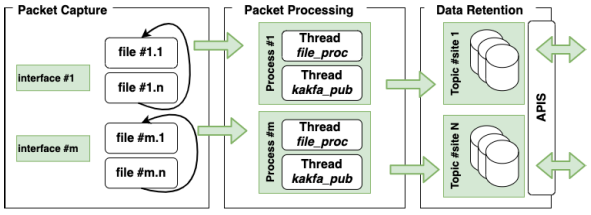

FIGURE 8. Data processing steps.

The capture was implemented using the *dumpcap* tool from the *wireshark* project, which was configured to dump network traffic in a ring buffer with file sizes between 2 and 20MB. Such option introduces flexibility, since the capture does not require specific hardware, only disk space to keep collected data locally in the Packet CAPture (PCAP) format. The split between the capture and the processing introduces reliability in the process, since packet loss in capture process is avoided (initially, these functionalities were performed in a single step, which led to high and non-acceptable packet losses, higher than 15%, in attack scenarios). The ring buffer configuration of *dumpcap* also allows to configure the data collection in terms of duration or intervals that should be kept in the collection process.

The processing step was performed through Python scripts with support for multi-threading. Such scripts include specific processes to convert data from PCAP format (which is the result of the previous step) to JSON/CSV format in order to be streamed and published in the configured Kafka topics (one per site). One thread was employed to perform the reading, considering the timestamp of the last processed packet, and another thread performs the data conversion and respective publishing in the configured Kafka topics. In order to support high volumes of data, the data for each site is streamed to a specific topic (that can be stored in a specific Kafka partition, or as per other configuration policies). The processing of the captured traffic in the data collection points includes the creation of Kafka messages in JSON/CSV format with fields of the captured packets like *IP.src*, *IP.dst*, *ETH.mac.src*, *ETH.mac.dst*, among others and respective forwarding to the site topic(s) configured in the Kafka cluster.

The data retention step, besides including the storage of data, also makes available standard APIs for data consumption, per different policies (e.g., time based, partitions with specific event streams [50]).

F. SECURITY ANALYSIS WITH MACHINE LEARNING MODELS

The security analysis has mainly the purpose of detecting anomalous activity due to DoS and DDoS attacks. In concrete, the model developed analyses the incoming traffic and classifies the flow as Normal (Label 0) or as Attack traffic (Label 1). The classification considers features in the TCP/IP header, the packet timestamp information, the packet length, the TCP sequence numbers, flags and options. In addition, the features of the framing format, as illustrated in Figure 2, are considered from the Modbus/TCP protocol: transaction ID, protocol ID, length, data size, function code, byte count, response time.

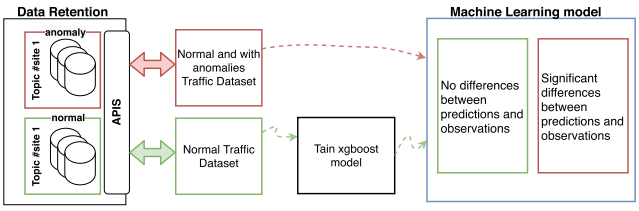

FIGURE 9. Machine learning security analysis pipeline.

Figure 9 presents an overview of the model architecture and pipeline, including the training and validation steps. The model was integrated with the ELEGANT data pipeline, one of the key goals of this work, and the model output is integrated in the HMI informative dashboards. Overall, the ML approach is a classifier that receives traffic flow time-windows as input with the set of features and outputs if the flow is normal or if it is an attack. Moreover, the classification is done in two steps:

1) regressing the behaviour of the sampled data;
2) a classification rule based on a threshold applied to the error of the observed behaviour.

In the training phase the system trains a model that learns what is considered as normal behaviour, that corresponds to situations with captured traffic during regular polling mechanisms between PLC slaves and PLC master without any disruption or attack. The results obtained in the training phase provide the baseline for the prediction error in regular monitoring and control between PLC nodes. The model, after being deployed for security analysis, compares the observed traffic in a configured time-window (e.g., 60s in the experiments), with the predictions of what the traffic should be for that period. If the error between what is observed and what is predicted surpasses a certain threshold with statistically significant differences, it is classified as an *Attack*. On the contrary, if there are no statistically significant differences between what is observed and what is predicted, the traffic is classified as *Normal*.

The system deployed for testing and production is based on Extreme Gradient Boost (xgboost) [51]. The reason for employing this technique is related with a binary classification problem, where two types of traffic, the *Normal* and the *Attack* must be distinguished. Within the binary classification problem, there are four possible outcomes: True Positive (TP); False Positive (FP); True Negative (TN) and; False Negative (TN); where a positive is when an Attack occurs.

The accuracy of the ML model for the security analysis considers 3 metrics, that are based on the confusion matrix covering the types of outcomes from the classification task:

- **Precision** is concerned with measuring the percentage of traffic classified as attack that was correctly classified, as formulated in Equation 1.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

- **Recall** measures the percentage of actual attacks that were correctly classified, as per Equation 2

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

- **F1-measure** is an weighted average between the precision and the recall, as per Equation 3

$$F1 - Measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

The model achieves best results when the value of the mentioned metrics are maximised. Considering the positive class as an attack, Precision shows how well the model is robust to falsely identify normal traffic as an attack whereas Recall provides feedback on how fair the system is able to identify the attacks. In the particular case for accurate Digital Twins, we argue that high recall values are important but avoiding low Precision values is also key, so the F1-measure is important to capture a balance between these two metrics.

VI. RESULTS

This section presents and discusses the achieved results validating the ELEGANT towards accurate Digital Twins for security and monitoring. The validation first presents the overhead in the data pipelines steps, in terms of memory and CPU overhead, as discussed in section VI-A, and in terms of input & output impact, as discussed in section VI-B. The overhead of the background processes, considered as user services to avoid overlapping with the processes terminology in ICS, is also discussed in section VI-C, such services enable the diverse steps in the data pipeline. The discussion regarding the security analysis services, the machine learning models, is performed in section VI-D, while section VI-E discusses and presents the performance of the devised xgboost ML models.

This section discusses the experimentation results in the Denial of Service - DoS, in the Distributed Denial of Service - DDoS attacks, performed in diverse variants as summarised in Table 2 and considering the characteristics of the nodes presented in Table 1.

A. MEMORY AND CPU OVERHEAD IN THE DATA PIPELINE

This subsection discusses the overhead in terms of CPU, system and memory usage in the data collection and processing steps of the data pipeline.

Figure 10 illustrates the overall CPU usage of the elements participating in the data pipeline, during the DoS and DDoS attacks. One can observe that the DDoS attacks have an higher impact in the CPU usage of the diverse elements, since the DDoS attacks have higher usage ratios, in particular for the user processes - *usage_user*, such as OpenPLC in the bridge nodes, InfluxDB in HMI, among others. Such impact is more evident in the elements performing the capture of traffic (i.e., bta1, btsiteb), where in some cases it can go up to 60%. The amplification attacks lead to higher CPU usage ratios.

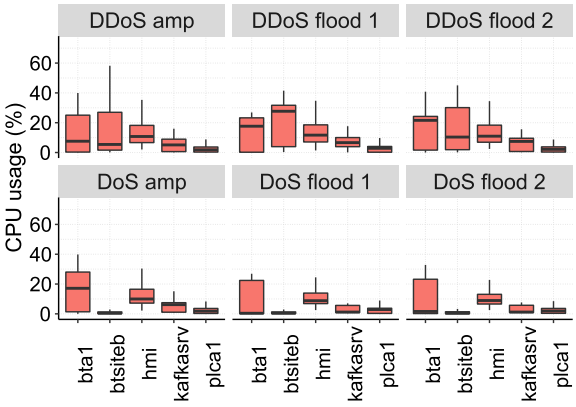

FIGURE 10. CPU usage in the data pipeline.

The memory usage overhead is illustrated in Fig. 11 for the DDoS and DoS attacks. The node with higher memory usage ratios are the HMI and *kafkasrv* nodes, which are associated with the stream processing and data retention steps of the data pipeline. The HMI node also hosts the security analysis

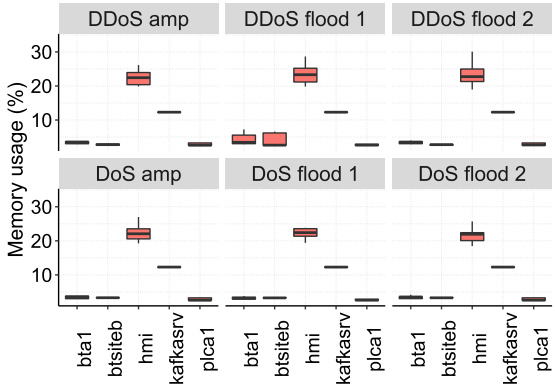

FIGURE 11. Memory usage in the data pipeline.

component of the Digital Twin (recall Figure 5). The security analysis only includes the devised ML model to detect DoS attacks according to the data being streamed in the pipeline.

The usage ratios do not increase with the higher volumes of traffic introduced by the distributed DoS attacks (as per the CPU usage), since the processing of the security analysis component is performed in a time-window period (60s). For instance, the consumer of Kafka feeding ML model acquires data using the timestamp approach for the offsets to fetch data considering the current time and the previous 60 seconds. The normal fetching processes consider the current position of messages in the topics that were read and do not rely on time information [50]. Higher time-window flow periods, above 60s, will lead to higher memory usage.

The system load of the diverse nodes is depicted in Figure 12, considering the load in the previous minute - *load1*, and on the previous 5 minutes - *load5*.

The system load is higher in the distributed DoS attacks, in particular in the bridge nodes that performing the capture of packets. The attacks last two minutes in average, which impacts the load in the 1 minute period - load1 metric. The HMI, hosting the security analysis component also has an increased system load due to higher volumes of information that need to be analysed in the distributed DoS attacks.

B. DATA PIPELINE INPUT & OUTPUT IMPACT

As stated in section V-E the data pipeline includes several steps: data collection, data processing and data retention in Kafka components. The data collection step includes the capture of network traffic in PCAP files and their storage in the local file system of the bridge nodes for reliability purposes. Such step impacts the disk input output - diskIO performance in terms of write and read operations, for write times, and on the respective waiting times for an operation.

Figure 13 depicts the number of read operations, which illustrates HMI as the node that is more impacted with the data pipeline steps. Such performance impact relies in the fact of using time-series databases to feed the visualisation dashboards. The number of read operations is almost constant in the Kafka nodes in the different types of attacks.

When considering the write operations, as pictured in Figure 14, it is clear that the diskIO performance in the

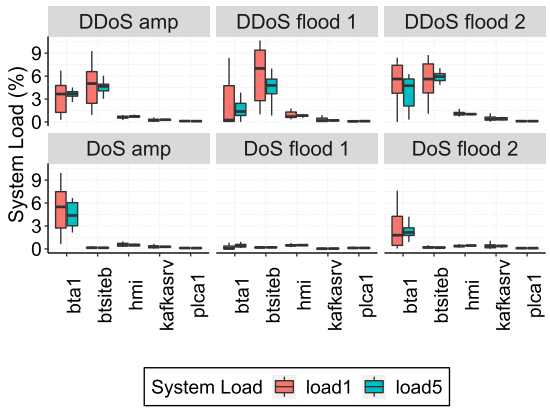

FIGURE 12. System load in the data pipeline.

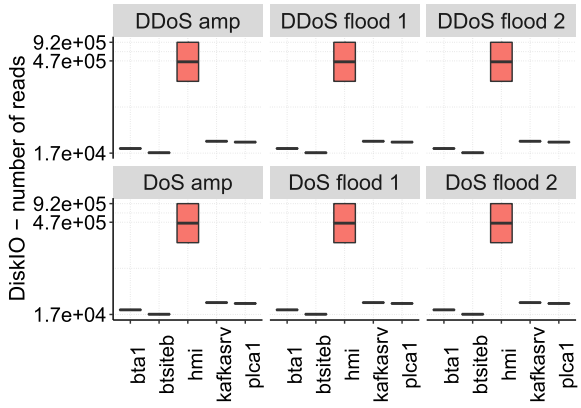

FIGURE 13. DiskIO number of read operations.

data capture step is not impacted by the volume of traffic that is generated in the distributed DoS attacks. This puts in evidence that the packet ring buffer option available in the *dumpcap* tool scales well for high volumes of data that need to be collected.

The data retention and processing at the *kafkasrv* and HMI nodes leads to a high number of disk operations, in particular for the write operations. HMI also aggregates real-time database series, necessary for the monitoring component of the Digital Twin, justifying the higher values, in comparison to the remaining nodes.

The bottleneck in terms of input and output must also consider the waiting times of the respective operations. That is considering the required number of operations, the average time (in milliseconds) the I/O requests have waited for the availability of the disk device. Figure 15 depicts the wait time for the write operations, where the wait times increase with the distributed DoS (DDoS) attacks in the nodes participating in the different steps of the data pipeline. For instance, *btsiteb* in the DDoS attacks has higher wait times in the write operations, in particular in the amplification attacks where the volume of data being captured is higher. Despite, not pictured, the waiting times for the read operations is also higher in the HMI node, due to the employment time-series database to feed the data visualised in the dashboards.

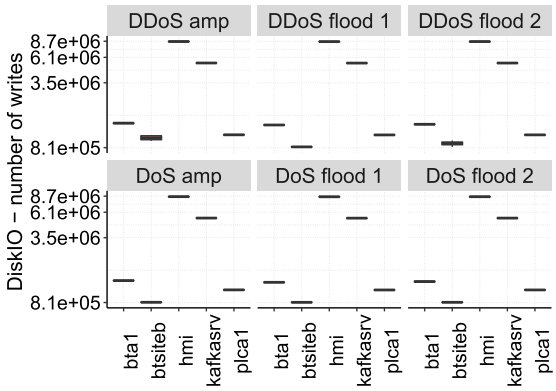

FIGURE 14. DiskIO number of writes operations.

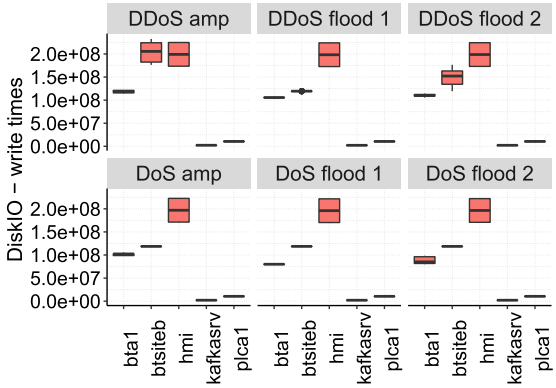

FIGURE 15. DiskIO write waiting times in milliseconds.

Figure 16 illustrates the bytes received and sent over the network interfaces of the ELEGANT components.

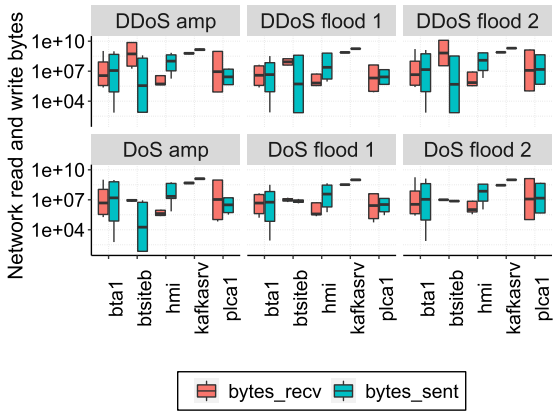

FIGURE 16. Network usage in terms of read and sent bytes.

The distributed DoS attacks lead to high volumes of traffic traversing the ELEGANT components, as highlighted in the number of bytes received and written in the *btsiteb* node.

The type of attack also impacts the critical components, where the SYN flooding attack - flood 2 attack, occurring in higher frequency intervals, leads to higher values in the sent bytes. Which means that PLC components process received packets, and send the respective replies to conclude the

negotiation of the TCP handshake process. Even if this handshake is to be performed with malicious, and forged IP addresses employed by attacking nodes.

C. USER SERVICES OVERHEAD IN THE DATA PIPELINE

This subsection presents the overhead of the user services, implemented as scripts and running as background processes in the Linux hosts to validate the diverse steps of the data pipeline. The data collection step includes the *dumpcap*, *tshark* processes. The data processing step includes the Python scripts to process the PCAP files and stream it to the data pipeline cluster. On the other hand, the data storage/retention step includes the Java processes running in the Kafka nodes. The Monitoring includes the *telegraf*, *influxdb*, *chronograf* processes. It should be noticed that the user services performing the security analysis are not included here, they are reported in section VI-D.

Figure 17 illustrates CPU utilisation rate of the diverse user services employed in the data pipeline steps.

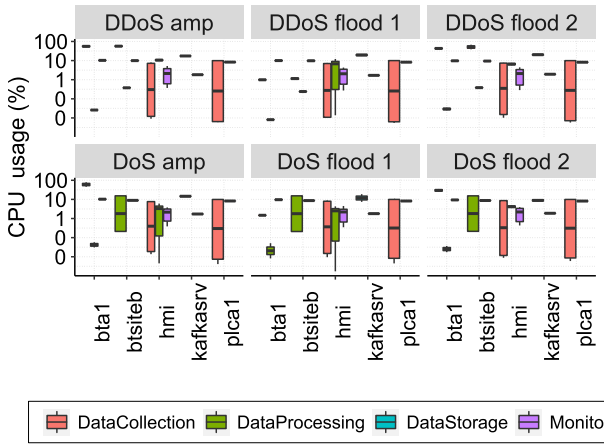

FIGURE 17. CPU usage by the user services.

The approach to implement the monitoring, to enable the Digital Twin approach introduces low overhead, since the CPU used by the programs/scripts is negligible. Opposed to the processes performing the data collection process, which can lead to significant CPU usage ratios around 60% in the distributed DoS attacks. As expected, with higher rates in the flooding and amplification DoS attacks lead to higher CPU utilisation.

The distributed DoS attacks also impact other steps in the data pipeline, although with lower impact severity. For instance, the user services of Kafka to retain/store the data consume more CPU in such type of attacks.

Figure 18 illustrates memory utilisation rate of the diverse data pipeline steps, assured by the respective user services.

Opposed to the trends verified in the CPU utilisation rate, the monitoring has an high impact in the HMI node in terms of the memory that is used. The HMI, as stated previously, aggregates diverse functionalities, which include the informed dashboards (powered with *chronograf* and *influxdb*)

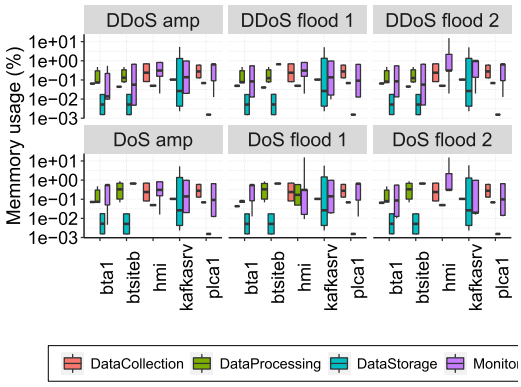

FIGURE 18. Memory usage by the user services.

as well as the security analysis. The impact of the security analysis is not pictured here (refer to subsection VI-D).

The memory required to picture the status of components in the dashboards increases with the type of DoS attack, where the flooding with maximum rates - flood2 leads to higher variation in the memory.

D. SECURITY OVERHEAD ANALYSIS

This section provides the results of the Security analysis overhead that is deployed on the HMI node. This overhead includes CPU and memory utilisation rates. The Security Analysis includes the processes of the Machine Learning models, with Python3 scripts running in docker containers.

As illustrated in Figure 19 the CPU utilisation increases according to the number of events that must be analysed. The distributed DoS attacks lead to higher CPU usage, rates above 10% in the amplification tests. In the DoS attacks from a single attacker the ratios are bellow 5% in all the test cases.

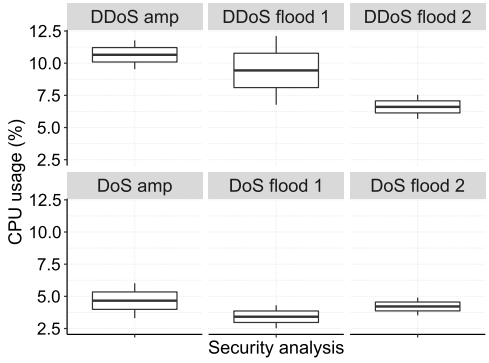

FIGURE 19. CPU usage by security analysis component.

As per the memory utilisation, depicted in Figure 20, the difference between the diverse attack tests is minimal (bellow 0.2%). In addition, the increased number of events to be analysed does not introduce significant impact, since the analysis is performed in 60s time windows, thus leading to the same memory footprint in a single analysis step.

E. MACHINE LEARNING MODEL PERFORMANCE

In this section we present and discuss the results obtained with the *xgboost* ML model for the security analysis,

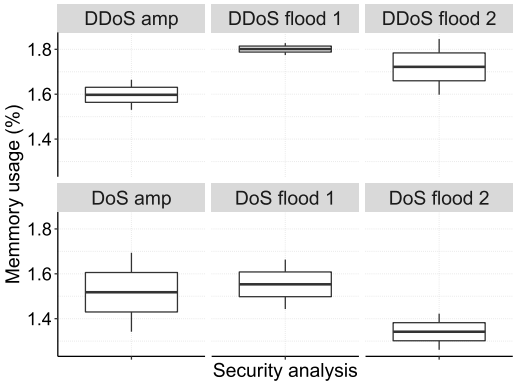

FIGURE 20. MEM usage by security analysis component.

TABLE 3. Digital Twins traffic classification results.

Traffic	Precision	Recall	F1-Score
Normal	1.00	0.96	0.98
Attack	0.98	1.00	0.99

in particular to detect DoS attacks. The system was implemented using the *scikit-learn* Python library [52] The reported results rely on the following configurations. In what concerns the hyper-parameters, we used a value of 0.1 of learning rate and a maximum tree depth for base learners of 20.

Our classifier was trained using 10 minutes (2 million data points) of traffic to establish what corresponds a normal traffic flow in the system. Such data was considered from the site running in *grid5000* testbed (recall Figure IV-B). After the training, the testing phase included a period of 20 minutes of combined normal and attack traffic, and the Precision, Recall and F1-Score were computed.

The results are summarised in Table 3 and show that the *xgboost* ML model is able to distinguish between normal and DoS traffic, attaining a high value (> 0.95) in all the metrics considered. Looking at the Recall for the Attack traffic ($= 1.00$), it shows that the system was able to detect and correctly identify all the traffic associated with an attack.

Looking at the Recall metric for the normal traffic, we can see that it has the lowest value, meaning that there is some normal traffic that is being classified as attack. These results are expected, since the *xgboost* model in the security analysis of the Digital Twin is adjusted to detect any abnormalities in the traffic, even if they result from a normal increase in the traffic flow in the system. As an example, such increase can correspond to the addition of PLC components in the diverse sites, upon new sensors that may be deployed and that require management. This result is particularly important to the conditions in which *xgboost* ML model will be operating, since classifying normal traffic as attack is far less dangerous than classifying Attack traffic as normal. In addition, such inaccuracy can also be detected with the monitoring data that is pictured in the informed dashboards of HMI (e.g., new devices being added in the sites).

VII. CONCLUSION

The ELEGANT project validated an off-premises approach to design and deploy Digital Twins to secure critical infrastructures. Such Digital Twins are built using real-time, high fidelity emulated replicas of Programming Logic Controllers (PLCs), coupled with scalable and efficient data collection processes, supporting the development and validation of ML models to mitigate security threats like Denial of Service (DoS) attacks, which can occur with different patterns (flooding and amplification).

The achieved results in ELEGANT to enable Digital Twins as accurate virtual models of physical objects or processes, demonstrate that DTs provide a faithful environment for security analysis or evaluation of potential mitigation strategies to be deployed in face of threats with high impact, such as distributed DoS attacks.

The approach to enable Digital Twins in ELEGANT has been motivated by the widespread of Modbus/TCP protocol with insecure deployment (no encryption) - Motivation#1, by the need of efficient data collection mechanisms able to provide safe and off-path environments for multiple analysis in DTs - Motivation#2, by the prohibited costs associated with on-premises monitoring and security analysis - Motivation#4, and by the need of solutions that can scale with the complexity of the IACS/ICS infrastructure - Motivation#3. The validation approach of ELEGANT, which leveraged from the capabilities of the Fed4Fire federated testbeds evaluated the feasibility of using cloudified DTs, thus converting a significant part of the projected CAPEX for the in-premises model into on-demand, pay-as-you-go OPEX, eventually paving the way for the establishment of a DTaaS (Digital Twin as a Service) paradigm.

Our next steps include further research on the enablement of DTaaS concept in SmartGrids powered with 5G networks, edge computing and associated virtualisation technologies.

REFERENCES

[1] G. Prettico, M. G. Flammini, N. Andreadou, S. Vitiello, G. Fulli, and M. Masera, "Distribution system operators observatory 2018—Overview of the electricity distribution system in Europe," Publications Office Eur. Union, Luxembourg, Tech. Rep. JRC113926 and EUR 29615 EN, 2019, doi: 10.2760/104777.

[2] C. Foglietta, D. Masucci, C. Palazzo, R. Santini, S. Panzieri, L. Rosa, T. Cruz, and L. Lev, "From detecting cyber-attacks to mitigating risk within a hybrid environment," *IEEE Syst. J.*, vol. 13, no. 1, pp. 424–435, Mar. 2019.

[3] V. Graveto, L. Rosa, T. Cruz, and P. Simões, "A stealth monitoring mechanism for cyber-physical systems," *Int. J. Crit. Infrastruct. Protection*, vol. 24, pp. 126–143, Mar. 2019.

[4] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and U.S. Air force vehicles," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf., 20th AIAA/ASME/AHS Adapt. Struct. Conf., 14th AIAA*, Apr. 2012, pp. 1–14.

[5] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," *Procedia CIRP*, vol. 61, pp. 335–340, Jan. 2017.

[6] R. Saracco. (Jun. 2019). *Digital Twins: Where We Are Where We Go—I*. [Online]. Available: https://cmte.ieee.org/futuredirections/2019/06/29/digital-twins-where-we-are-where-we-go/

[7] H. Laaki, Y. Miche, and K. Tammi, "Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery," *IEEE Access*, vol. 7, pp. 20325–20336, 2019.

[8] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *IEEE Access*, vol. 6, pp. 3585–3593, 2018.
[9] L. Thames and D. Schaefer, Eds., *Cybersecurity for Industry 4.0: Analysis for Design and Manufacturing*, 1st ed. Springer, Apr. 2017, doi: 10.1007/978-3-319-50660-9.
[10] F. Consortium. (2021). *Fed4Fire Federation for Fire Plus*. [Online]. Available: https://www.fed4fire.eu/
[11] B. Sousa, T. Cruz, M. Arieiro, and V. Pereira, "An ELEGANT dataset with denial of service and man in the middle attacks," 2021, *arXiv:2103.09380*. [Online]. Available: https://arxiv.org/abs/2103.09380
[12] A. Miglani, N. Kumar, V. Chamola, and S. Zeadally, "Blockchain for internet of energy management: Review, solutions, and challenges," *Comput. Commun.*, vol. 151, pp. 395–418, Feb. 2020.
[13] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, and J. Lopez, "A survey of IoT-enabled cyberattacks: Assessing attack paths to critical infrastructures and services," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3453–3495, 4th Quart., 2018.
[14] M. G. (Mass.gov). (2021). *Cybersecurity Advisory for Public Water Suppliers*. [Online]. Available: https://www.mass.gov/service-details/cybersecurity-advisory-for-public-water-suppliers
[15] F. Caldeira, T. Cruz, P. Simões, and E. Monteiro, *Towards Protecting Critical Infrastructures*, 1st ed., vol. 1. Hershey, PA, USA: IGI Global, Jul. 2015, ch. 7, pp. 123–169.
[16] R. De Andrade, K. N. Hodel, J. F. Justo, A. M. Laganá, M. M. Santos, and Z. Gu, "Analytical and experimental performance evaluations of CAN-FD bus," *IEEE Access*, vol. 6, p. 21287–21295, 2018.
[17] Texas Instrument. (2014). *RS-485 Reference Guide*. [Online]. Available: http://www.ti.com/rs485
[18] R. Zurawski. *Industrial Communication Technology Handbook* (Industrial Information Technology), 2nd ed. New York, NY, USA: Taylor & Francis, 2014.
[19] M. Organization. (Oct. 2006). *Modbus Messaging on TCP/IP Implementation Guide V1.0b*. [Online]. Available: https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf
[20] *Telecontrol Equipment and Systems—Part 5: Transmission Protocols—Section 4: Definition and Coding of Application Information Elements*, document IEC 60870-5-4, 1993. [Online]. Available: https://webstore.iec.ch/publication/3749
[21] *IEEE Standard for Electric Power Systems Communications—Distributed Network Protocol (DNP3)*, IEEE Standard 1815-2010, 2010, pp. 1–775.
[22] Modicon. (Jun. 1996). *Modbus Protocol Reference Guide (PI-MBUS-300 Rev. J)*. [Online]. Available: https://www.modbus.org/docs/PI_MBUS_300.pdf
[23] W. J. Buchanan, *Modbus*. Boston, MA, USA: Springer, 2004, pp. 677–687.
[24] Modbus Organization. (Jul. 2018). *MODBUS/TCP Security Protocol Specification*. [Online]. Available: https://www.modbus.org/docs/MB-TCP-Security-v21_2018-07-24.pdf
[25] D. J. Kang, J. J. Lee, B. H. Kim, and D. Hur, "Proposal strategies of key management for data encryption in SCADA network of electric power systems," *Int. J. Elect. Power Energy Syst.*, vol. 33, no. 9, pp. 1521–1526, Nov. 2011.
[26] A. Rezai, P. Keshavarzi, and Z. Moravej, "Key management issue in SCADA networks: A review," *Eng. Sci. Technol., Int. J.*, vol. 20, no. 1, pp. 354–363, 2017.
[27] A. Huseinović, S. Mrdović, K. Bicakci, and S. Uludag, "A survey of denial-of-service attacks and solutions in the smart grid," *IEEE Access*, vol. 8, pp. 177447–177470, 2020.
[28] E. Gamess, B. Smith, and G. Francia, "Performance evaluation of modbus TCP in normal operation and under a distributed denial of service attack," *Int. J. Comput. Netw. Commun.*, vol. 12, no. 2, pp. 1–21, Mar. 2020.
[29] *Security for Industrial Automation and Control Systems—Models and Concepts*, document ISA/IEC-62443-1-1, 2017.
[30] North American Electric Corporation (NERC). *Critical Infrastructure Protection Standards*. Accessed: Jul. 28, 2021. [Online]. Available: https://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx
[31] A. K. Stouffer, A. J. Falco, and A. K. Scarfone, *Guide to Industrial Control Systems (ICS) Security: Supervisory Control and Data Acquisition (SCADA) Systems, Distributed Control Systems (DCS), and Other Control System Configurations Such as Programmable Logic Controllers (PLC)*, Standard NIST SP 800-82, 2011.
[32] ENISA. (2011). *Protecting Industrial Control Systems. Recommendations for Europe and Member States*. [Online]. Available: https://www.enisa.europa.eu/publications/protecting-industrial-control-systems.-recommendations-for-europe-and-member-states
[33] International Atomic Energy Agency (IAEA). (2011). *Computer Security at Nuclear Facilities Technical Guidance Reference Manual*. [Online]. Available: https://www.iaea.org/publications/8691/computer-security-at-nuclear-facilities
[34] T. Cruz, J. Barrigas, J. Proença, A. Graziano, S. Panzieri, L. Lev, and P. Simoes, "Improving network security monitoring for industrial control systems," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 878–881.
[35] O. Oleghe and K. Salonitis, "A framework for designing data pipelines for manufacturing systems," *Procedia CIRP*, vol. 93, pp. 724–729, Jan. 2020.
[36] K. Singh and D. S. Tomar, "Architecture, enabling technologies, security and privacy, and applications of Internet of Things: A survey," in *Proc. 2nd Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud), (I-SMAC), I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Aug. 2018, vol. 4, no. 5, pp. 642–646.
[37] L. Rosa, M. Freitas, S. Mazo, E. Monteiro, T. Cruz, and P. Simoes, "A comprehensive security analysis of a SCADA protocol: From OSINT to mitigation," *IEEE Access*, vol. 7, pp. 42156–42168, 2019.
[38] K.-H. John and M. Tiegelkamp, *Bibliography*. Berlin, Germany: Springer, 2001, pp. 365–367.
[39] F. Adamsky, M. Aubigny, F. Battisti, M. Carli, F. Cimorelli, T. Cruz, A. Di Giorgio, C. Foglietta, A. Galli, A. Giuseppi, F. Liberati, A. Neri, S. Panzieri, F. Pascucci, J. Proenca, P. Pucci, L. Rosa, and R. Soua, "Integrated protection of industrial control systems from cyber-attacks: The ATENA approach," *Int. J. Crit. Infrastruct. Protection*, vol. 21, pp. 72–82, Jun. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1874548217301798
[40] J. Granjal, J. Silva, and N. Lourenço, "Intrusion detection and prevention in CoAP wireless sensor networks using anomaly detection," *Sensors*, vol. 18, no. 8, p. 2445, Jul. 2018.
[41] L. Rosa, T. Cruz, M. B. D. Freitas, P. Quitério, J. Henriques, F. Caldeira, E. Monteiro, and P. Simões, "Intrusion and anomaly detection for the next-generation of industrial automation and control systems," *Future Gener. Comput. Syst.*, vol. 119, pp. 50–67, Jun. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X21000431
[42] M. S. Al-Hawawreh, "SYN flood attack detection in cloud environment based on TCP/IP header statistical features," in *Proc. 8th Int. Conf. Inf. Technol. (ICIT)*, May 2017, pp. 236–243.
[43] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 29–35.
[44] M. A. Amanullah, R. A. A. Habeeb, F. H. Nasaruddin, A. Gani, E. Ahmed, A. S. M. Nainar, N. M. Akim, and M. Imran, "Deep learning and big data technologies for IoT security," *Comput. Commun.*, vol. 151, pp. 495–517, Feb. 2020.
[45] I. Arbor Networks. *Digital Attack Maptop Daily DDoS Attacks Worldwide*. Accessed: Jul. 28, 2021. [Online]. Available: https://www.digitalattackmap.com/
[46] T. Alves. *OpenPLC—The First Fully Open Source Programmable Logic Controller*. Accessed: Jul. 28, 2021. [Online]. Available: https://www.openplcproject.com/
[47] M. Niedermaier, F. Fischer, D. Merli, and G. Sigl, "Network scanning and mapping for IIoT edge node device security," in *Proc. Int. Conf. Appl. Electron. (AE)*, Sep. 2019, pp. 1–6.
[48] Antirez. *Hping3 Network Tool*. Accessed: Jul. 28, 2021. [Online]. Available: https://github.com/antirez/hping
[49] Wireshark. *Wireshark User's Guide V3.5.0*. Accessed: Jul. 28, 2021. [Online]. Available: https://www.wireshark.org/docs/wsug_html_chunked/
[50] D. A. Dana Powers. *Kafka-Python Api—Kafkaconsumer*. Accessed: Jul. 28, 2021. [Online]. Available: https://kafka-python.readthedocs.io/en/master/apidoc/KafkaConsumer.html
[51] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, and H. Cho, "Xgboost: eXtreme gradient boosting," *R Package Version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.
[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.