Digital Communications and Networks(DCN)

HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/dcan

Smart and collaborative industrial IoT: A federated learning and data space approach

Bahar Farahani[*a], **Amin Karimi Monsefi**[b]

[a]**Cyberspace Research Institute, Shahid Beheshti University, Tehran 1983969411, Iran**
[b]**Department of Computer Science and Engineering, Ohio State University, Columbus, OH 43210, USA**

Abstract

Industry 4.0 has become a reality by fusing the Industrial Internet of Things (IIoT) and Artificial Intelligence (AI), providing huge opportunities in the way manufacturing companies operate. However, the adoption of this paradigm shift, particularly in the field of smart factories and production, is still in its infancy, suffering from various issues, such as the lack of high-quality data, data with high-class imbalance, or poor diversity leading to inaccurate AI models. However, data is severely fragmented across different silos owned by several parties for a range of reasons, such as compliance and legal concerns, preventing discovery and insight-driven IIoT innovation. Notably, valuable and even vital information often remains unutilized as the rise and adoption of AI and IoT in parallel with the concerns and challenges associated with privacy and security. This adversely influences inter- and intra-organization collaborative use of IIoT data. To tackle these challenges, this article leverages emerging multi-party technologies, privacy-enhancing techniques (e.g., Federated Learning), and AI approaches to present a holistic, decentralized architecture to form a foundation and cradle for a cross-company collaboration platform and a federated data space to tackle the creeping fragmented data landscape. Moreover, to evaluate the efficiency of the proposed reference model, a collaborative predictive diagnostics and maintenance case study is mapped to an edge-enabled IIoT architecture. Experimental results show the potential advantages of using the proposed approach for multi-party applications accelerating sovereign data sharing through Findable, Accessible, Interoperable, and Reusable (FAIR) principles.

KEYWORDS:
Industry 4.0, Industrial Internet of Things (IIoT), Artificial Intelligence (AI), Edge computing, Fog computing, Cloud computing, Predictive maintenance

1. Introduction

Industry 4.0 is characterized by the convergence of the Industrial Internet of Things (IIoT), Artificial Intelligence (AI) — including Augmented Intelligence, big data analytics, Machine Learning (ML), and Deep Learning (DL) — and Edge-Fog-Cloud Computing driving the next wave of the digital transformation [1–4]. Cross-company collaboration (e.g., multi-party computation, pooled analyses, data sharing, and data exchanging within a network of collaborators/organizations) is a prerequisite for alleviating severe fragmentation of data to unlock the full value of Intelligent IIoT. Another hurdle in Industry 4.0 is the lack of appropriate standards and interoperability for multi-party computation. Indeed, there is no consensus on any reference model or best practices that specify how IIoT and AI can be fused in a multi-party environment.

The adoption of IIoT in Industry 4.0 enables automation as never seen before [5, 6]. IIoT turns multiple devices, sensors, actuators, and machines into coherent ecosystems by creating a smart network facilitating data exchange [3, 7]. IIoT can also bridge the gap between the traditional Cyber-physical systems in manufacturing and Information Technology (IT) sys-

*Corresponding author: Bahar Farahani (email: b_farahani@sbu.ac.ir).

tems, enabling us to collect data automatically from a variety of systems. IIoT generates enormous quantities of data, which is infeasible for humans to ingest and analyze that data efficiently and effectively. AI and augmented intelligence acquire the power to unlock the value of IoT data by adding a layer of intelligence and analyzing copious amounts of data in numerous applications and bringing insight to human decision-makers allowing them to evaluate the best course of action [8–14].

Data is the fuel powering AI, IIoT, and digital transformation. However, data is healthy when it is accessible and utilized across organizations. However, IIoT data is currently severely fragmented across silos owned/controlled by different parties, each having access to only part of the picture for a range of inhibitor reasons. (e.g., different technologies and data standards, data sovereignty, security, privacy, lack of trust, data governance policies across companies, intellectual property rights, bureaucratic hurdles, compliance and legal concerns, slowing down discovery and technology-driven innovations).

Indeed, conventional ML/DL-based solutions (e.g., failure prediction and anomaly detection) typically cannot be directly utilized in decentralized IIoT applications with distributed edge devices. The reasons are twofold: i) privacy concerns: integration of IIoT data across different parties typically results in a powerful MLs; however, companies/factories and the corresponding edge devices are reluctant to share their privacy-sensitive collected data with each other, resulting in data silos and data islands [15–17]; ii) lack of high-quality data: high quality and labeled data are usually difficult and expensive to collect, resulting in significant degradation of the generalization and robustness ability of the model. Although a few techniques have been recently applied to address these issues (e.g., transfer learning techniques, knowledge transfer methods, and synthetic data generation), they cannot outperform solutions that try to enlarge the training dataset in terms of accuracy and reliability [15, 16].

To address the data sharing challenges, a few reference models, such as Gaia-X [18] and International Data Spaces Association (IDSA) [19], have recently been developed as a key step towards the industrial data economy based on interoperability, portability, sovereignty, and trustworthiness values [20]. Particularly, GAIA-X creates a federated data space adhering to the concept of a shared economy that enables service and data sharing while incorporating sovereign policies. Unfortunately, even with the recent advances in sovereign data sharing technologies, multi-party computation and collaborative analysis across different data sources are still challenging. Thereby, many organizations cannot share data and wring insights due to concerns around privacy, data sovereignty, and competitive advantages.

In parallel to this endeavor to establish a standard data sharing/exchanging framework, the Privacy-Preserving Machine Learning (PPML), Multi Party Computation (MPC), and Federated Learning (FL) techniques have recently gained considerable attention across vertical industries from smart city to smart manufacturing to address insufficient high-quality training data, security/privacy preservation, communication cost, network overhead, data collection cost, and scalability challenges. Despite the recent advances, multi-party and PPML-based fault diagnosis, prediction and predicate maintenance solutions have been seldomly explored. Notably, the authors of [21] proposed an FL-based solution to forecast device failures. Ge et al. [22] proposed a federated Support Vector Machine (SVM) and federated random forest algorithms for failure prediction of production lines. W. Zhang et al. [16] presented a decentralized data-driven machinery fault diagnosis by utilizing federated learning methods. In [15], an FL-based deep anomaly detection framework centered on an attention mechanism-based convolutional neural network long short-term model is proposed to detect anomalies accurately. However, to the best of our knowledge, the above methods have not been (adequately) mapped and integrated into IIoT, which significantly limited their applications and usage in Industry 4.0 and smart factories use cases.

In line with these efforts, the central theme and contributions of this paper include:

- Reporting novel methodologies, theories, technologies, techniques, and solutions for federated and multi-party data analytics.
- Presenting a holistic, privacy-preserving reference architecture intending to further facilitate a competitive and secure industrial data economy as well as the innovation process by providing secure, trustable environments for innovators.
- Establishing a scalable multi-party computation framework, scaling up privacy enhancement techniques, and discussing a holistic Collaborative Condition Monitoring (CCM) and Predictive Maintenance (PdM) use case that enables the creation of accurate AI models while striking a balance between the security/privacy of all involved participants and the utility and reliability of ML models. Compared to the existing predictive maintenance solutions, this is the first work that strives to map FL-based CCM/PdM to decentralized IIoT and data sharing frameworks to the best of our knowledge.

The remainder of this paper is organized as follows. Section 2 overviews the existing reference architectures, from the three-tier architecture to the gateway-mediated architecture and system of systems architecture pattern, and the corresponding design con-

siderations. Section 3 elaborates on data economy, cross-company collaboration technologies as well as privacy-preserving machine learning techniques. Section 4 presents a holistic reference architecture by combining existing models' unique advantages and features, particularly tailored for collaborative cross-company applications. In addition, the model details are discussed using a holistic use case on collaborative predictive maintenance. Section 5 discusses the experimental results. Finally, Section 6 concludes the paper.

2. Existing IIoT architectures

Architectural patterns characterize the most elemental and common IIoT implementation features understood and recognized as reference points for developing real-life IIoT solutions. The most well-known patterns include: i) Three-tiered architectural pattern, ii) Gateway-mediated edge connectivity and management architecture pattern, iii) Layered databus pattern, and iv) System of systems architecture pattern. Architecture patterns are simpler abstract perspectives of an IIoT system implementation that is repeated in many IIoT systems while allowing variants [23]. For example, a three-tier IIoT system pattern does not exclude multiple tier implementations (e.g., multiple edge tier instances) or many-to-many connections among tier instances and instances of the subsequent tier. In other words, every tier and its connections are only represented once in the pattern [23].

2.1. Three-tier architecture

A three-tiered architecture is made up of the platform, enterprise, and edge tiers that handle particular data flow and control roles that are part of usage tasks. The tiers are connected through three networks as outlined in Fig.1 [23, 24]:

- Edge tier: This tier gathers data from edge nodes via a proximity network. Its characteristics include responsibility for proximity network characteristics (e.g., location, governance, and distribution) depending on the needs of each use case.
- Platform tier: This tier is responsible for receiving, processing, and forwarding control commands sent by the enterprise tier and received by the edge tier. It is responsible for combining processes and analyzing data flow from the edge as well as other tiers and handles device/asset management functions. This tier also provides general services like analytics or data query.
- Enterprise tier: This tier is responsible for implementing applications specific to particular domains, decision support systems, and providing end-user interfaces. The enterprise tier also receives data from and sends control commands to the platform and edge tiers.

The three-tier architecture pattern integrates essential components such as applications, platforms, and management services that can map to functional domains [23, 25]. From the viewpoint of the Industrial Internet Reference Architecture (IIRA) model, the edge tier handles the majority of the control domain. In contrast, the platform tier handles the majority of operation and information domains. The enterprise tier is responsible for handling business and application domains. This mapping illustrates a straightforward partitioning of functions across tiers. In real-world systems, functions are mapped depending on the specifics of system requirements and use cases. For example, information domain functions may take place in or near the edge tier in conjunction with applicable rules or logic that facilitate edge computing [23, 25].

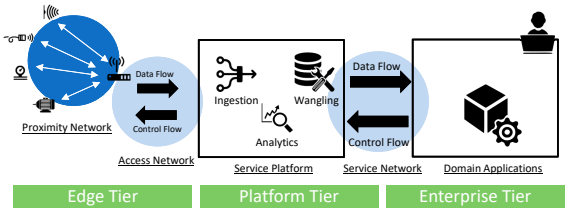

Fig. 1: Three-tier architecture.

2.2. Gateway-mediated edge connectivity and management architecture pattern

The Gateway-mediated edge connectivity and management architecture pattern use a gateway to span a Wide Area Network (WAN) and provide local connectivity for the IIoT edge, as illustrated in Fig.2. In this pattern, the gateway serves a WAN endpoint and isolates the local edge network. This pattern enables localized controls and operations (e.g., computing and edge analytics). The primary benefit of this pattern is the reduction of IIoT system complexity, allowing scalability of the network and managed assets. However, this pattern may not be suitable for systems with mobile assets that do not facilitate stable clusters within the local network. The edge gateway can also serve as a device and asset management point as well as a point for data management where data aggregation, analytics, processing, and control logic are utilized locally [23].

2.3. Layered databus architecture pattern

The layered databus architecture pattern is commonly utilized in IIoT systems across several industries because it provides secure, low-latency, and peer-to-peer communication across a system's logic layers [23, 25]. It is most beneficial for systems that handle the direct interactions (i.e., edge analytics, control, and local monitoring) among field applications. As shown in Fig.3, smart machines utilize databuses

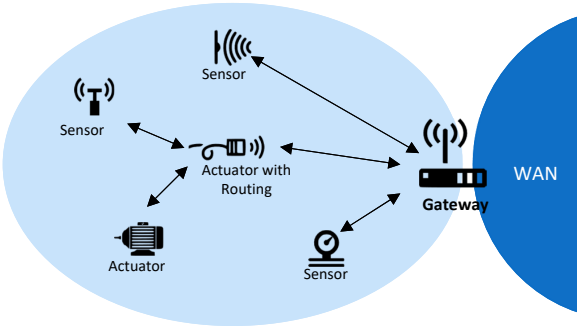

Fig. 2: Gateway-mediated edge connectivity and management architecture pattern.

to manage analytics, control, and automation in the lowest architecture level [23, 25]. Systems in the layers above utilize a separate databus to provide supervisory monitoring and control. Combining systems into a "system of systems" allows complicated, cloud-based analytics and control supervision. Databuses serve as a logically connected space that utilizes a common schema to facilitate communication between endpoints. Databuses use a data-centered, publish-and-subscribe model of communications. Databus applications subscribe to the needed data and publish the generated information. Messages are then communicated between nodes. This foundational communications model includes discovery (i.e., where to send data) and delivery (i.e., where and when to send data) [23]. These publish and subscribe systems are good at rapidly routing and propagating large amounts of time-sensitive information, specifically when delivery mechanisms are not dependable. Every layer in a databus uses a common data model, which allows interoperable communication within the layer. Adapters can be utilized to match data models between layers, bridge or separate security domains, or serve as a point of interface to integrate diverse protocols or legacy systems. Transitions commonly take place between layers where they reduce and filter data, which is vital because the breadth of analysis and control expands with each layer. The amount of data is usually lessened to suit the higher latency, greater abstraction, and broader scope [23].

2.4. System of systems architecture pattern

At high levels, IIoT systems can be configured to work as a component of a more extensive system, which creates a system of systems. This pattern increases system complexity, which can comprehensively impact subsequent and nearby system's trustworthiness or characteristics [23].

3. Secure cross-company collaboration via data spaces

Digitization drives and empowers innovative business models, and data is a primary element of busi-

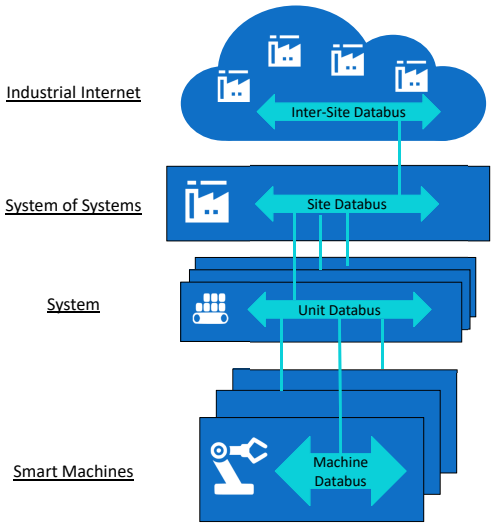

Fig. 3: Layered databus architecture pattern.

ness success. In this context, systematically collecting, processing, and accessing data has become a primary differentiator in the industrial sector. Due to technological advances in edge computing and connectivity, companies can utilize data that has historically been siloed to improve machine performance [26]. They can also integrate data to increase the value of production lines. However, not many companies have the internal resources or skills to discover these value sources methodically; therefore, system integrators and those who manufacture solutions equipment have begun creating solutions as additional services. This is an appealing income source that works to balance dropping profit margins across many industries. In unusual cases, completely new business plans capable of entirely replacing traditional models have been revealed [26].

Industrial data sharing ecosystems have the capability to support massive growth and enable organizations to optimize processes and create new products or businesses. In our connected environment, data sharing can be a strong facilitator for all parties – those who provide data, create new services, or disrupt markets with new appealing offers. This kind of innovation typically requires data from a variety of sources or organizations, such as machine integrators, machine users, or component manufacturers. However, sharing data outside business boundaries has become one of the primary obstacles to more full-scale adoption. Ecosystem participants are faced with multiple obstacles, including [10, 26]:

- Culture and mindset challenges: The creation of new value propositions relies on the ability to share data across bigger industrial ecosystems that include participants with limited trust in each other. Parties may be capable of analyzing shared data to discover confidential information. In ad-

dition, many companies do not participate in data sharing models due to the perceived inability to control data that leaves the organization.

- Challenges around ontological needs: Data that is shared has to be easy to interpret and integrate with additional data. It has to be clearly understood by all ecosystem participants; therefore, a common ontology or language has to be agreed upon by all parties [26].

- Privacy and security: Data/information sharing and collaboration across departments/companies help break down data silos; however, protecting business secrets is vital. Therefore, organizations typically face many challenges in ensuring they adhere to privacy and security regulations [10].

- Competition: Although data sharing among factories can lead to more robust predictive models, enterprises are typically reluctant to exchange data because they fear losing their competitive edge to competitors [10].

- Technological design and platform services challenges: Even when organizations understand the benefits of data sharing within industrial ecosystems, participants often do not have the capability to provide data efficiently and effectively. Companies may not have the internal technology to provide data, create an environment to process or integrate data from various sources or utilize advanced analytical methodologies [26].

- Data management challenges: It can be difficult for companies to manage data in a way that makes quick automated data access due to a lack of internal capabilities needed to draw data from internal applications [26].

Enterprises have shared data laterally with trustworthy partners for many years. This has included exchanging data with vendors to help manage inventories or automating invoicing or payment processes. However, creating an ecosystem is much more complex because monitoring and enforcing appropriate data use is difficult. To address the increased need for sovereign data exchange, various initiatives have recently been taken to establish and facilitate a uniform standard and protocol. In this context, the concept of "data spaces" has evolved. Typically, the term "data space" describes the kind of relationship between trustworthy partners that equally utilize elevated standards and strict rules when sharing or storing data. However, when it comes to the idea of data space, it is vitally important that data is held at the source rather than centrally so that data is only shared by semantic interoperability as needed. Among data sharing standard protocols, GAIA-X and the International Data Space (IDS) are the most widely used today to create secure and federated industrial data spaces [18, 19].

GAIA-X and IDSA also introduce the concept of usage control as an extension of the classic access control. The usage control defines a set of policies describing the conditions and requirements for data sharing and handling among participants. It also proposes several methods for traceability, data monitoring, and technical enforcement of those policies. Although protocols like GAIA-X/IDSA can strike a balance between data monetization options with compliance, data security, and privacy regulations to some extent; however, they are not really designed to bridge the gap between privacy/security and (AI/ML) utility. Indeed, state-of-the-art data sharing frameworks, such as GAIA/X/IDSA, strive to establish the foundations and cradle of an open, transparent data exchanging ecosystem by addressing remote access to data, distributed data governance, and access control challenges to achieve a secure data economy. Although these technologies can be helpful, they have limitations. These technologies cannot fundamentally remove barriers around privacy, legal obstacles, policies, and conflict of interest. For instance, the General Data Protection Regulation (GDPR) prohibits transfers of even pseudonymized data regardless of the underlying data sharing standards [27]. Over the past few years, it has become apparent that companies/factories are unwilling to adopt these data sharing frameworks for real world challenges as they are afraid to lose their competitive edge by sharing their business secrets and sensitive data.

Recently, PPML techniques have evolved independently of data sharing protocols to tackle the privacy challenge by preventing data owners' privacy compromise and protecting against data leakage [28, 29]. PPML is used to design and create models while protecting data privacy. Privacy-Preserving methods empower several participants to partner to train an ML model without disclosing original information that includes private data. Additionally, as mentioned, MPC enables enterprises to analyze private information owned by other enterprises without disclosing input data. This means MPC can analyze information and handle computing using data from several parties because no single participant receives additional information around any other participant's input data. Each party receives identical outputs that are available to every participant.

3.1. Secret sharing

Secret sharing, originally established by Shamir and Blakely in 1979, refers to a technique allowing one participant to disperse pieces of a secret to all other participants in such a way that no single party can reveal the entire secret independently [30]. In short, a participant breaks a secret into multiple pieces and shares pieces with each party. Each participant then

completes individual computations without having access to input values. Ultimately, the combined results of each party's computation reveals the accurate output [31].

However, secret sharing also has disadvantages, including extended computing time because of the interplay between parties, lack of categorical data support, and reduced accuracy for complicated computations (i.e., non-linear functions) because of approximate implementation.

3.2. Homomorphic Encryption (HE)

Homomorphic Encryption (HE) is another centralized ML technique that uses cipher-texts with computing. HE produces decrypted results to generate a result that matches what would have been provided if the computation was done utilizing plain text. In this technique, participants encrypt raw data prior to sending it to the server to be analyzed. Each operation is done using encrypted data, and the outcomes are encrypted with the same key. In addition, the information does not require decryption before analysis [31–35].

3.3. Differential Privacy (DP)

Differential Privacy (DP), a technique that protects data privacy by adding random noise, was initially created by Cinthia Dwork. Adding noise makes it impossible to discover user identities. This is an innovative method because it safeguards privacy and allows insightful data analysis. Adding the noise protects user data, but when data points are aggregated, the noise is averaged, which divulges an outcome closer to the original [31, 36–38]. Typically, DP systems utilize global or local privacy. Globally private systems need a trustworthy party, known as a curator, capable of accessing the raw data of multiple parties. The curator can analyze the data and include noise once the computation is complete. On the other hand, locally private systems do not include a curator. Each participant is required to add noise to data prior to sharing information. Local systems typically include an aggregator that simultaneously collects data from an extensive group. Globally private systems usually maintain higher accuracy because noise-less data is analyzed. Every participant must trust the system's curator for a global system to work. Localized DP systems are considered safer because each data point includes noise, making obtaining individual data pieces useless. Noise can also be removed so aggregators can collect the local data to analyze dataset trends [36–38].

3.4. Federated Learning (FL)

Initially developed by Google, Federated Learning (FL) or collaborative learning is a decentralized ML technique that brings code to data instead of bringing data to the code. This method relies on a centralized server and several client servers, which enables private data to remain in its starting location. A central server is utilized for training a model with proxy data. Then, that model is given to clients for local training. Each client uses local data for multiple training iterations. Next, each participant is given a copy of the model's parameters, trains the model using private data, and returns the model's results to the centralized server. Next, the server generated a general model by aggregating the local models. Then, the generated global model is transferred back to local clients. The steps are repeated through several iterations until accuracy reaches an acceptable level. There are two different types of FL: Horizontal FL and Vertical FL [31, 39–43].

- Horizontal FL: This method of FL is used when datasets include identical feature spaces but have different spaces between samples [44].
- Vertical FL: This method, referred to as feature-based FL, is used when datasets include identical sample ID spaces but divergent feature spaces [44].

4. Proposed reference architecture: privacy-preserving collaborative edge-fog-cloud IIoT

To address the shortcomings of the data sharing solutions (e.g., GAIA-X/IDSA), supplementary privacy-enhancing measures are needed to achieve an adequate level of privacy-preserving and data protection, accelerating ML-enabled collaborative use cases. Unforgettably, the scope of the previous attempts to solve challenges and obstacles toward cross-company collaborative application is minimal. Either they focus on data-transferring protocols or privacy-enhancing techniques. Moreover, to the best of our knowledge, the existing solutions have not been thoroughly discussed in the fusion of decentralized IIoT architectures with data sharing and privacy-preserving MPC. Thereby, holistic solutions require combining and integrating data sharing technologies with privacy-enhancing/preserving techniques and finally mapping it to decentralized IIoT architectures.

To address the shortcoming mentioned above, collapse isolated data islands, and bridge the gap between reaping the benefits of AI/ML and privacy/regulations concerns, in this section, we present our proposed architecture for secure and privacy-preserving cross-company collaboration. The 5-layer pyramid was initially developed with the foundational assumption that every layer has a unique task and would be hosted by a private LAN [23, 45]. However, integrating edge-fog-cloud computing facilitates further innovation in IIoT solutions by lowering vital application latency and, more adeptly, handling the massive amount of data created by IoT devices (See Fig.4). As a result, the separation between each of the five layers

is blurring. Using cloud capabilities, including scalability, performance, virtualization, life cycle management, and multi-tenancy, an enhanced IIoT can be generated. Cloud computing's ability to host adaptable storage and computing services offers numerous novel solutions for industrial applications and systems. The cloud is capable of handling a diverse array of auxiliary elements and services that interact with and improve the abilities of IIoT. Quickly evolving advancements in storage, communication, and computing power, combined with cloud benefits, facilitate the creation of industrial systems based on Service-Oriented Architecture (SOA) with functions housed in the cloud and on devices. Modern IT and Operational Technology (OT) systems are moving to cloud-based environments as Manufacturing Execution System (MES), Enterprise Resource Planning (ERP), and Supervisory Control and Data Acquisition (SCADA) providers grow their businesses by expanding throughout the stack [45]. A 2015 IDC survey revealed that 68% of manufacturers utilize private clouds to host applications and 66% utilize public clouds. Cloud architectures still mainly benefit IT operations, but innovative technology may address many of the technical issues that have prevented OT applications (i.e., SCADA and MES) from adopting the cloud. MES and SCADA suppliers are utilizing evolving technology to offer cloud-based applications along with conventional, on-site solutions.

Edge/Fog computing is capable of providing device-level intelligence services that optimize computing, storage, and communication resources [46, 47]. When it comes to industrial applications, edge computing could be utilized to empower agile connectivity and smart decision-making while optimizing data transfer and providing real-time control for legacy industrial and automation systems. Edge computing and the industrial cloud each have different inherent advantages. Edge computing is best utilized to manage localized, short-term data analysis needed in real-time to control execution and facilitate real-time decision making. On the other hand, industrial clouds can be utilized to handle the global, long-term analysis of big data outside of real-time to support business decisions and guide preventative maintenance. Edge computing is not meant to replace the industrial cloud. In fact, these two options should be used in coordination to satisfy the needs of diverse industrial situations. Industrial edge computing can support cloud applications needed for data preprocessing and mining, while the industrial cloud is capable of deploying models and rules centered on big data analysis.

The interconnection of cloud computing and edge/Fog computing is essential for supporting even more IIoT application opportunities reliant upon machine learning or artificial intelligence. While cloud computing can resolve issues around data-based solutions, storage or processing capacity, and service creation/management, it has challenges when it comes to the service requirements of time-sensitive IoT applications. Latency difficulties are not expected to improve immediately because networks are designed to improve bandwidth and link efficiency. Extensive cloud computing adoption has reduced the need for edge device storage and allowed edge devices to dispense with complex computing. However, edge computing allows edge devices to offload highly resilient, low latency transmission tasks in the event of the increased network backhaul traffic. While neither model is perfect, harmonious collaboration allows end-point edge devices/machines to benefit from both, including [1, 7, 48]:

- Reduced network load: Moving computing nearer to the data sources more evenly distributes the network load and lessens the network backhaul load; therefore, backhaul data rates do not require increasing, and data links to the network edge are available to services [1, 7, 48].
- Latency-aware computing: As the network load is reduced, edge devices can process more quickly. Edge/fog computing is useful in improving Quality of Service (QoS) and allows latency-sensitive applications to offload tasks in alignment with latency needs [1, 7, 48].
- Native mobility support: Locating resources nearer to edge devices enables the network to react quickly and accurately to user mobility. As edge/fog nodes communicate horizontally, requests and task offloading are handled even within high-mobility situations [1, 7, 48].
- Provision of context: Resources nearer to end users allow resources to generate content relevant to a specific location [1, 7, 48].
- No single point of failure: Resources distributed across a network mean that other edge/fog nodes can take over if links are disabled or edge/fog nodes are knocked offline by a cyberattack, supporting continued functionality without interruption [1, 7, 48].
- Increased battery life: Edge devices can depend on task offloading because of low processing delays in edge/fog computing. This expands battery life and lessens power use which enables long-term autonomy [1, 7, 48].
- Reduced energy use: Because most data processing occurs near the data source, long-distance data transfers are unnecessary, reducing network energy use. In addition, because power loads are distributed across the network, it is easier to meet energy needs with renewable energy or current power grid infrastructure [1, 7, 48].

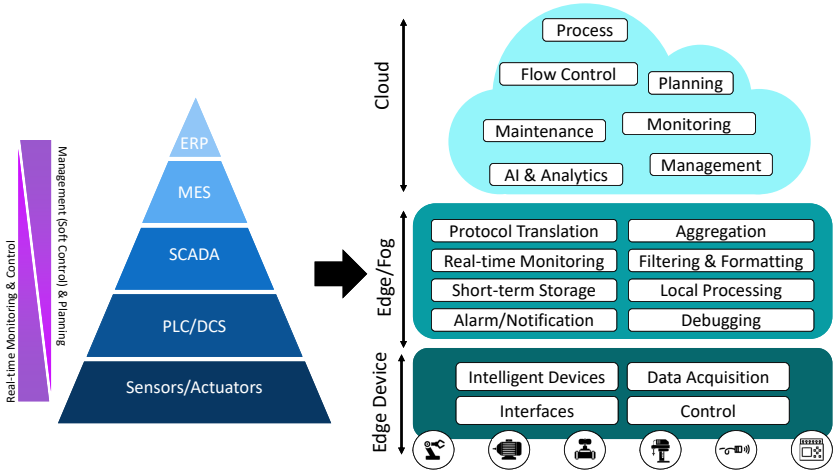

Fig. 4: Legacy 5-level reference architecture and its transformation to edge-fog-cloud computing.

- Heavy load support: The cloud holds more resources than the most powerful edge/fog node; therefore, an overwhelmed edge/fog node can send complex computing to the cloud for execution in exchange for a longer network processing time [1, 7, 48].
- Unlimited storage: While edge/fog computing works to lessen traffic on the network backhaul, edge devices may occasionally require extensive storage. Edge devices can depend on the cloud to expand resources and meet unlimited storage requirements [1, 7, 48].

Generally, ERP systems and systems responsible for manufacturing execution (i.e., Levels 4-5) are the best candidates to be transferred to an industrial cloud based on high data volume and fewer real-time needs (See Fig.4). However, Levels 0, 1, and 2 typically should stay on the manufacturing floor because of their higher real-time requirements. Determining if an application should be cloud-hosted is dependent upon the industry and application needs. Usually, the higher an application is within a stack, the less vital it is to have high communication availability for edge devices, low latency, and bi-directionality. Cloud-based architecture has met challenges in addressing these three areas, which restricts the transition of MES and SCADA systems to the cloud for some industries, such as manufacturing. As new technology overcomes these challenges, transitioning to the cloud becomes more likely for an ever-growing number of industrial use cases.

When edge computing and the industrial cloud are combined, the architecture is generally separated into three layers, as outlined below and depicted in Fig.4 [49, 50]:

- Cloud layer: The uppermost layer holds industrial cloud platforms that provide a variety of applications for maintenance, design, management, and manufacturing. It is worth noting that legacy ERP, product life cycle management, CRM systems, and manufacturing execution could be transitioned to an industrial cloud to lower operating costs. Additionally, cutting-edge applications housed on industrial clouds (i.e., supply chain analysis, energy use optimization, and device operation analysis) may be improved through real-time edge computing device data collection. It is also possible that such services could be handled by third parties and managed using a local cloud rather than a public cloud.
- Edge/Fog layer: This layer is known as the industrial edge/fog gateway. It deploys algorithms and manages data acquisition from Edge Computing Nodes (ECNs) while balancing storage, computing, and network resource use. This middle layer is responsible for timely development and deployment using a model-based organization for modular services. Edge computing nodes also tap networks to monitor packet transactions between terminals, cloud servers, clients, and IoT systems. This enables ECNs to alter packets to supply add-on options. For example, suppose a new device with an unknown network protocol is detected by a network. In that case, the edge gateway can automatically configure or translate the protocol or update security to safeguard ECNs. Edge gateways can also detect IoT system attacks and restrict access. ECNs are also capable of halting IT system attacks similarly. Within the middle layer, ECNs furnish two kinds of manufacturing functions:
  - Data analysis: As large amounts of data are gathered from endpoints, data is filtered and buffered to summarize technical models. Such models are further refined by using distributed reasoning through machine learning, knowledge rules, and statistical

analysis. Transmitting results to the industrial cloud optimizes the supply chain, logistics, ordering, and other services that assist manufacturers, vendors, and clients.

  - Management: ECNs can deploy and execute industrial applications using APIs responsible for configuring and maintaining industrial applications. In addition, all service and application instances could be managed using APIs. Connected applications related to utilization rate, equipment effectiveness, production capacity, and energy usage offer features for application development as well as life cycle management.

- End-point device layer: This layer is composed of distributed nodes responsible for handling at least one, if not more, functionalities such as that of a closed-loop real-time programmable controller, sensor, and actuator. These functions may be handled by constantly changing combinations of industrial edge nodes based on real-time feedback from closed loops.

Integrating end-point edge devices, edge/fog nodes, and the cloud produces a hierarchy-based IIoT model that improves the function of IIoT systems. As computing evolves, hierarchical paradigms transform the IIoT by combining millions of distributed devices, centralized cloud servers, and many edge servers. Rather than viewing each of these as a separate element, it is common for many IIoT applications to enable elements to work together to create reliable services with various location or time needs. Hierarchical edge-fog-cloud models facilitate AI activities such as decision making, machine learning, and big data analytics near data creation points, at the edge, or by uploading to the cloud or edge/fog to facilitate complicated computing from smart IIoT objects. The adaptability of edge-fog-cloud hierarchies regarding latency and computing power suggests that such a model can support AI/ML applications in an efficient and scalable manner.

Collaborative condition monitoring and predictive maintenance: Component suppliers create multiple individual components. The individual pieces are then transferred to a machine supplier that creates a machine using parts from various suppliers. Operators then utilize several different machines within production systems. The manufacturer of the component may want to analyze the data of a specific component installed inside a machine used by another company to ensure the efficiency and safety of the machine or component [51]. While this common scenario seems simple, it generates questions about who owns the component's data, who should be allowed to access the data, and for what purposes the data should be accessed [52, 53]. Additional issues arise about how the data can be monetized and how the information should be standardly formatted. For the operators and component and machine manufacturers to collaboratively gather, assess, and present condition monitoring using sensors, all parties need a trustworthy model for data sharing, communal rules for multi-party authentication, and shared agreement regarding access control [51–53].

Condition Monitoring (CM) can ensure that all pieces of the production chain fit together so that components and machines function dependably for long periods. Traditional CM gathers and analyzes all operational information, including temperature, vibration, or other physical values related to a machine's status. Currently, such data is exclusively shared bilaterally between machine operators and suppliers, and participants can only see their piece of the chain rather than the entire picture. Optimizing the whole system is challenging, but CCM, originally proposed by the GAIA-X Industrial Use Case Group [18], allows data to be shared multilaterally along the entire chain [52, 53]. With CCM, the system can be optimized if each participant makes data accessible on an independent digital platform. For example, a machine or component's operational life can be prolonged, or data-driven predictive maintenance services can be created, benefiting every member of the chain. CCM collects and shares data multilaterally across the whole network. Typical CCM scenarios include [52–54]:

- Component provider: Component providers produce a variety of drives that incorporate sensors. Components hold an Asset Administration Shell (AAS) that includes data fields for information about service life and dependability.

- Machine suppliers: Suppliers create machines that include manufacturing components. The machines also include AASs with data fields for information about service life and dependability. In short, a machine's AAS is made up of component AASs. Vital data produced during the running of the machine is saved to the component and machine AASs; however, the machine AAS is tasked with transferring data collected from components and the machine to an independent platform.

- Factory operator: The operator uses the machine within the production process, adding application data such as temperature and maintenance intervals to collect data using the AAS data fields.

CCM problems can be mapped to the AAS, edge-fog-cloud IIoT, federated data sharing, and PPML technologies as proposed in Fig.5. The proposed reference model consists of the following main components:

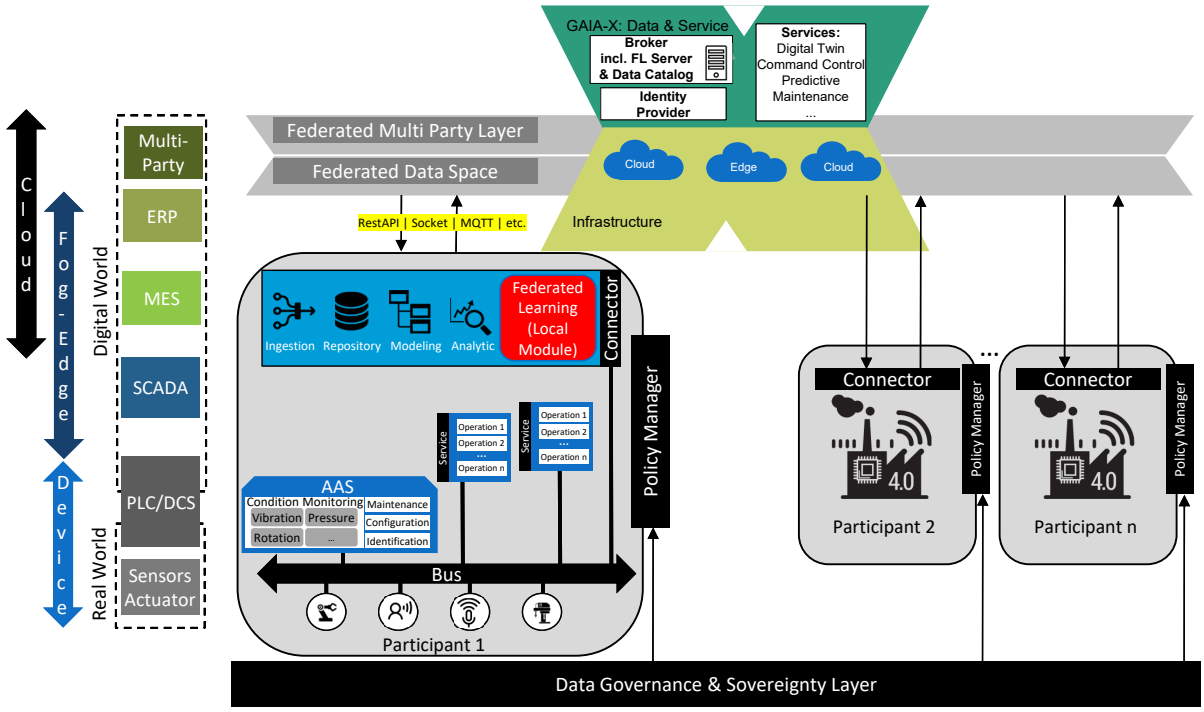

Fig. 5: Proposed architecture based on GAIA-X and FL for privacy-preserving distributed Collaborative Condition Monitoring (CCM) and predictive maintenance.

- Participant: participants represent organizations engaged in the multi-party federated data ecosystem. They can be either an asset owner, a data consumer, a smart service provider, a smart service consumer, or an enabler.

- Asset: An Asset is either a device (e.g., PLC controllers, sensors, and actuators) or a virtual asset that provides specific data/information about an entity (e.g., human workers). Each asset also provides callable services with the corresponding interfaces.

- Bus: It is a virtual media including several software components to establish end-to-end communication between system components. With the help of appropriate gateways, the bus can also bridge different network protocols.

- Asset Administration Shell (AAS): ASS is the digital representation of each asset by capturing and exposing different aspects (e.g., functional and non-functional attributes) of the asset, allowing consumers to select them based on their requirements.

- Edge node: The node offers several features, such as 1) Registry: it stores the self-Description file of each AAS, enabling registration and lookup of all existing AASs; 2) Aggregator: it ingests and aggregates all the data/information coming from different AASs; 3) Data management and processing: as the volume, variety, velocity, and veracity (four V's of big data) of industrial data grow, data management must be reimagined. Data management includes data ingestion and integration, creation of physical data storage (e.g., database, data warehouse, file system, and data lake), data modeling as well as cleansing, wrangling, and transformation. This layer can also be utilized for data reporting and visualization based on structured data storage sources like data warehouses, distributed file systems, and databases; 4) Local data analytics: the edge can also implement and provide various AI services incorporated in different scenarios and use cases. It handles both the reactive side of analytics (i.e. diagnostic, descriptive) as well as the proactive side (i.e., prescriptive, exploratory, and predictive), producing information about outliers, aggregates, significance level, prediction, trends, and useful insights that drive decisions surrounding control, monitoring, and optimization. The AI techniques used in this layer can include evolutionary algorithms used in intelligent search and optimization, machine learning needed to address data-driven problems, affective computing for human-based and social intelligence, as well as probabilistic reasoning needed to handle representational problems based on knowledge. The edge can also be equipped to run PPML algorithms, including federated learning. These AI techniques work together to develop high-level intelligence and generate inputs for other components; 5) Connector: a connector is a gateway

for two-way data exchange. This includes protocol translation, security, switching, routing, and networking analytics. It provides a connection between the shop floor environment to the cloud. It also enables sovereign peer-to-peer data sharing across company borders. Note that the functionality of the connector and the running applications/services can be extended or upgraded online.

- Federated platform: It consists of different infrastructure (e.g., cloud) providers creating a decentralized and federated ecosystem, such as GAIA-X and IDS [18, 19]. A single cloud typically benefits from a multi-layer architecture consisting of a data injection layer, integration layer, hierarchical data storage (database, data warehouse, and data lake) layer, data processing/analytics layer, data visualization and presentation layer, and orchestration/management layer. The cloud can complement the edge by providing hyperscale resources (e.g., storage and processing). It can also be utilized to host sophisticated ML/DL algorithms and services, e.g., an FL aggregator server that needs considerable bandwidth to communicate with all clients.

5. Experimental results

In this section, we evaluate the efficiency of the proposed collaborative framework using a novel FL-based predictive maintenance. Predictive maintenance, as a use case of CCM, aims to estimate when maintenance should be performed by forecasting possible future defects/failures in equipment. In this experiment, we considered two scenarios:

- Baseline approach (Scenario I): In scenario I, we assume all participants trust each other; thus, sharing their data without any privacy concerns. Therefore, all local data are combined in a single repository to be used for model training.

- FL-based privacy-preserving approach (Scenario II): In scenario II, we assume there are several participants (e.g., owners of privacy-sensitive data) that do not trust each other. However, they understand the importance of collaboration to unlock the value of data and train a holistic predictive maintenance model. Therefore, they utilize the architecture presented in Fig.5 and FL to train a model jointly.

5.1. Dataset

We used an open predictive maintenance dataset from one of Schwan's factories [55]. This dataset is composed of three different parts consisting of 100 machines. One of them includes the characteristics of the machines, such as age and model of the machine ("model 1, model 2, model 3, and model 4"). Another part of this dataset is about the telemetry of machines' situation in each hour. This part of the dataset contains "data-time, machine-id, volt, rotate, pressure and vibration." The last part of the dataset is about machine failures in terms of timestamp (date-time). We joined the telemetry part and machine characteristics in a table and then normalized all columns of this table between zero to one.

Telemetry data is sequential data; thereby, we decided to use Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) simultaneously for prediction. To do this experiment, we chose several conditions of data. We collect H hours of the machine's situation history and predict whether the next h hours the machine faces a failure or not. We run our model for several (h, H). Table 1 shows several (h, H) that we used in our experiments. The columns and rows of this table show the size of the history window (H) and prediction window (h), respectively. For example, data-03 shows that we picked 48 hours of our data as historical data, and we predict that in the next 24 hours whether the machine i faces a failure or not. We split data into three parts, months 1 to 9 are used in the training phase, month 10 is used for the validation phase, and months 11 and 12 are reserved for the test phase.

5.2. Computation platform

For our experiment, we used an Ubuntu system running on a server with a 2.30GHz Intel(R) Xeon(R) Gold 5118 CPU and 16 GB RAM.

5.3. Scenario I: Baseline approach; Ignoring privacy concerns

In scenario I, companies decide to collaborate by directly sharing their data and jointly training a holistic model for the predictive maintenance process. All local datasets are collected and aggregated in a common database for further analysis. Fig.6 shows our model. The model contains three parts. Input data go to CNN and RNN parts simultaneously, and the outputs of these two go to a fully connected neural network. In this figure, x is a tensor, showing the input of each convolution block and H represents the size of the history window. We trained our model with the combined data using Adam method as an optimizer with Binary Cross-entropy as the loss function. Training of the models will be stopped if we do not get any improvement for more than ten epochs at the loss value of the validation data.

Table 2 shows the results of scenario I in terms of runtime, accuracy, recall, and precision. According to Table 2, better results are obtained by increasing the history window size. However, the runtime is longer. The best result is related to "data-05", with the cost of long runtime. As shown, "data-05" has the most extensive history window and the smallest prediction

Table 1: Different data keys in terms of diverse prediction windows and history windows (h,H).

h \ H	24 hours	36 hours	48 hours	60 hours	72 hours
24 hours	data-01	data-02	data-03	data-04	data-05
48 hours	data-11	data-12	data-13	data-14	data-15
72 hours	data-21	data-22	data-23	data-24	data-25
96 hours	data-31	data-32	data-33	data-34	data-35

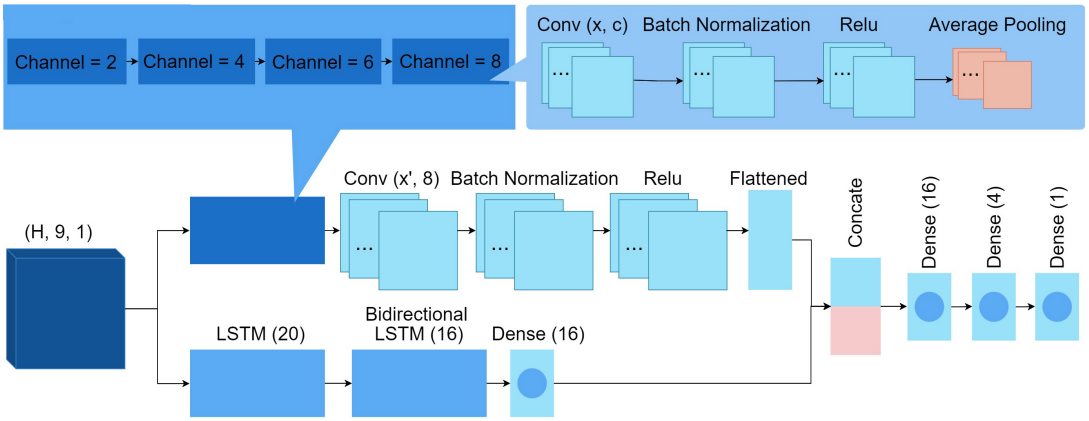

Fig. 6: Proposed deep learning model for failure prediction.

window. Results of "data-3" show that failures are not perfectly predictable by telemetry data of machines, and more close data to the failure time are required.

It should be noted that for accurate calculation of FL runtime, one needs to incorporate the following three main components: (1) the model training time of models by local clients; (2) communication time between local clients and the aggregator server (e.g., uploading model parameters to the server and downloading the aggregated model to local clients); 3) aggregation time in the server. Our proposed model contains 10149 trainable parameters. Considering the fact that these parameters are floating-point numbers, the corresponding model size is almost 121 KB. Suppose the FL clients and the aggregator server are connected via LTE with an average uploading speed of 7-8 Mbps and an average download rate of 12-30 Mbps. This results in 0.157 seconds as the communication overhead between clients and the server for each iteration of FL, which is negligible compared to the training time of local models.

5.4. Scenario II: FL-based privacy-preserving approach

Unlike traditional central training, in scenario II, several participants with their own training dataset are reluctant to disclose any raw data. However, to break down the data silos while addressing the data-privacy concerns, they exploit the proposed solution and FL techniques to train a model in a collaborative manner. To simulate this scenario, we split our dataset into X parts. We have data from 100 machines; in each part, we have 100 / X machines of data. For example, if

Table 2: Results of the deep learning model for scenario I when the privacy issue is ignored.

Data Key	Time (s)	Accuracy	Recall	Precision
data-01	29143	93.4	94.57	98.14
data-02	28948	95.91	97.35	98.33
data-03	28911	97.67	97.92	98.51
data-04	28924	98.55	99.84	98.56
data-05	29479	98.59	99.81	99.07
data-11	28736	93.63	93.49	99.28
data-12	29045	94.86	94.81	99.41
data-13	28014	97.92	98.34	99.37
data-14	28121	98.01	97.55	99.24
data-15	28175	98.19	98.03	99.63
data-21	29846	92.98	94.15	99.28
data-22	29943	92.48	92.73	99.11
data-23	30371	94.09	94.26	99.45
data-24	30089	96.94	97.01	99.86
data-25	30127	97.37	96.99	99.45
data-31	28337	70.86	69.81	98.17
data-32	28642	73.04	72.43	98.95
data-33	28834	71.67	71.49	97.81
data-34	28964	65.98	63.99	98.97
data-35	28469	73.49	73.53	98.57

X=4, part one contains data of machines number between 1 to 25, part two is 26 to 50, part three is 51 to 75, and part four is 76 to 100. For running this experiment, we used "data-04" data from Table 1. The training phase has stopped after model convergence. Similar to the previous experiment, we used Fig.6 as our model, with Adam method as an optimizer and Binary Cross-entropy as a loss function. Note that we use FedAVG as the aggregation approach on the FL server.

Table 3 shows the results of scenario II. As shown in the federated learning method, runtime decreases by increasing the number of clients. When we split data among different clients, accuracy decreases; however, this decrease is not meaningful, and when privacy is important, we can ignore this decrease. In addition, this table shows that with the increasing number of clients from 7 to 19, the rate of change in accuracy is shallow.

If none of the clients participate in the FL process and only use their local data to train their model, the results of Table 4 and Table 5 will be obtained. In this part, without limiting the generality of the proposed method, we assumed data is split into five and ten parts, respectively. Note that a different number of FL clients could be used in this experiment; however, it would not impact the general trend and our observations. As shown in the tables, the accuracy of this scenario is lower than when we use FL, in which models are aggregated on the server and the new model is sent to the clients.

Table 6 shows the result of learning when we have different volumes of datasets in clients. In this scenario, without limiting the generality of the proposed method, we have three clients; one of them has 20 machines (machine numbers 1 to 20), another has 35 machines (machine numbers 21 to 55), and the last client has 45 machines (machine numbers 56 to 100). The first three rows of this table show results without federated learning; row four is the result when we use FL to train the model with these three clients. As the result shows, FL leads to higher accuracy with the expense of elevated runtime.

5.5. *Evaluation of federated learning aggregation approaches*

In this experiment, we evaluated two main aggregation approaches in federated learning, FedAVG, and FedSGD, using 10 FL clients. In federated stochastic gradient descent (FedSGD), the server-side computation includes averaging the gradients proportional to the number of training data on each node. In Federated Averaging (FedAVG), the weights of the different local models are averaged to provide new weights and, therefore, a new model. We designed a federated learning system with five clients. We ran this experiment on ten cores of "Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz," and we initiated the weight of our model randomly. Fig.7 shows how model accuracy changes over time (10 FL iterations) when different aggregation approaches are used on the server. It is inferred that the FedAVG approach converges more rapidly than the FedSGD. In both approaches, the final model accuracy reaches around 95%. Regarding runtime, FedSGD requires 65 rounds to train the model, whereas FedAVG needs 60 rounds.

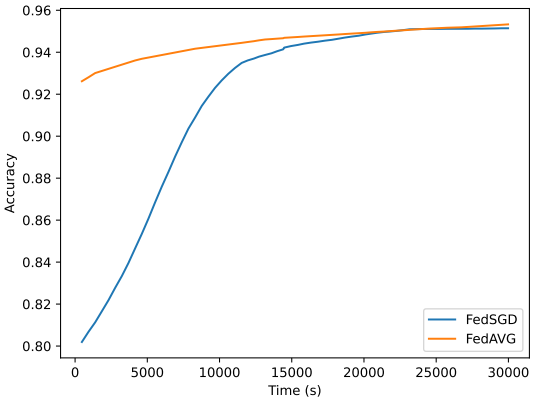

Fig. 7: The impact of FL aggregation approaches (FedAVG versus FedSGD) on the accuracy and runtime.

5.6. *Evaluation of clustered federated learning*

It has been observed that the performance of FL can decrease when local data distributions diverge. Over the past few years, several Clustered Federated Learning (CFL) and Federated Multi-Task Learning (FMTL) have been proposed to address this issue [56–59]. The main idea behind these techniques is to partition the clients by exploiting the similarities among either model parameters or local data to achieve a more specialized model for each group. In this study, we utilized k-means clustering to analyze the similarity among clients; To do so, we considered ten FL clients in this experiment. We tainted a general FL model for all clients by running five FL iterations. Next, we applied the K-means algorithm to cluster similar clients. Next, each cluster was trained separately in an FL model to create a customized model for each cluster by adjusting the previously generated general model. Table 7 illustrates the impact of the clustering technique and the number of clusters on the performance of the model. Based on the extracted results, it can be concluded that clustering and model personalization can improve the performance of FL.

5.7. *The role of proposed FL-based architecture on cross-company use cases*

As discussed in Section 4, the proposed architecture (See Fig.5) which is based on GAIA-X and FL, can bridge the gap between privacy and AI utility. Predictive maintenance and CCM, as potential use cases of this architecture, aim to assess when maintenance

Table 3: Results of scenario II for different numbers of clients.

Number of Clients	Time (s)	Loss Value	Accuracy	Recall	Precision
1	29021	0.056	98.58	99.86	98.66
2	20381	0.064	98.35	99.95	98.34
3	16669	0.089	97.61	99.96	97.58
4	12151	0.121	96.80	99.92	96.81
5	11915	0.129	96.54	99.97	96.51
6	11865	0.137	95.85	99.05	96.67
7	11601	0.125	96.50	99.97	96.47
8	11132	0.144	95.82	99.99	95.80
9	11043	0.187	95.15	99.92	95.22
10	10897	0.147	95.82	99.98	95.81
11	10527	0.191	95.46	98.60	95.35
12	10362	0.190	95.16	99.98	95.16
13	10176	0.178	95.23	99.99	95.23
14	9847	0.163	95.83	99.99	95.80
15	9432	0.165	95.35	99.99	95.34
16	9534	0.172	95.04	99.99	95.04
17	9347	0.166	95.28	99.99	95.27
18	9326	0.173	95.28	99.99	95.26
19	9247	0.182	95.09	99.99	95.09
20	9074	0.411	89.42	93.55	95.24

Table 4: Results of using only local data in order to train the model (Number of clients = 5).

Client # [Machines]	Time (s)	Loss Value	Accuracy	Recall	Precision
1 [1 - 20]	10716	0.139	96.23	99.92	96.41
2 [21 - 40]	10119	0.143	96.06	99.91	96.38
3 [41 - 60]	10347	0.145	96.01	99.91	96.36
4 [61 - 80]	10209	0.141	96.13	99.92	96.39
5 [81 - 100]	10578	0.138	96.25	99.94	96.47
FL	11915	0.129	96.54	99.97	96.51

Table 5: Results of using only local data in order to train the model (Number of clients = 10).

Client # [Machines]	Time (s)	Loss Value	Accuracy	Recall	Precision
1 [1 to 10]	9981	0.217	94.28	99.91	94.4
2 [11 to 20]	10147	0.196	94.53	99.93	94.65
3 [21 to 30]	9902	0.207	94.39	99.92	94.48
4 [31 to 40]	9945	0.209	94.42	99.92	94.55
5 [41 to 50]	9923	0.191	94.57	99.93	94.67
6 [51 to 60]	9976	0.221	94.03	99.9	94.23
7 [61 to 70]	9916	0.18	94.97	99.94	95.12
8 [71 to 80]	9957	0.197	94.62	99.93	94.68
9 [81 to 90]	9989	0.193	94.71	99.93	94.83
10 [91 to 100]	9927	0.184	94.83	99.94	94.89
FL	10942	0.152	95.53	99.97	95.61

Table 6: Results for different volumes of historical data in each client (Number of clients = 3).

Client # [Machines]	Time (s)	Loss Value	Accuracy	Recall	Precision
1 [1 - 20]	10716	0.139	96.23	99.92	96.41
2 [21 - 55]	12127	0.101	97.03	99.94	96.78
3 [56 - 100]	14410	0.096	97.21	99.94	97.06
FL	17261	0.091	97.37	99.96	97.24

Table 7: Clustered federated learning.

Number of Clusters (K in K-means)	Clusters	Accuracy of Each Cluster					
		#1	#2	#3	#4	#5	AVG
1	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]	95.82	-	-	-	-	95.82
2	[[1, 2, 5, 8], [3, 4, 6, 7, 9, 10]]	95.93	95.67	-	-	-	95.8
3	[[1, 2, 8], [5, 9], [3, 4, 6, 7, 10]]	96.13	96.47	96.01	-	-	96.23
4	[[1, 2, 8], [5, 9], [3, 10], [4, 6, 7]]	96.17	96.39	96.89	97.41	-	96.71
5	[[1, 8], [2], [5, 9], [3, 10], [4, 6, 7]]	96.98	97.95	96.47	96.93	97.51	97.16

should be performed by forecasting possible future defects/failures in equipment. By Bringing the code to distributed data instead of transferring and collecting privacy-sensitive data in one single server, we showed that the performance of AI/ML models could significantly be improved by FL-enabled cross-company collaboration and data sharing. This solution facilitates innovations in Industry 4.0 and IIoT by enabling multiple data owners to collaboratively train a model and unlock the value of data without sacrificing privacy/security.

6. Conclusions

The production and machine operation process create a massive amount of data that can serve as the foundation for innovative data-driven business models. However, depending on authorization, every participant in the value chain, from component providers to machine suppliers, and factory operators, can access only a part of the data. In other words, data are siloed and cannot be shared and (re)used beyond organizational boundaries across the entire ecosystem to unlock their actual value. This paper presented a holistic reference architecture facilitating cross-company, collaborative, and privacy-preserving use cases to tackle this issue. A novel case study of FL-based collaborative predictive maintenance was discussed and mapped to the reference model to shed light on the presented architecture's design and implementation details and considerations.

References

[1] F. Firouzi, B. Farahani, A. Marinšek, The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT), Information Systems 107 (2022) 101840.

[2] B. Farahani, F. Firouzi, M. Luecking, The convergence of IoT and Distributed Ledger Technologies (DLT): Opportunities, challenges, and solutions, Journal of Network and Computer Applications 177 (2021) 102936.

[3] F. Firouzi, B. Farahani, M. Weinberger, G. DePace, F. S. Aliee, IoT fundamentals: Definitions, architectures, challenges, and promises, in: F. Firouzi, K. Chakrabarty, S. Nassif (Eds.), Intelligent Internet of Things, Springer, Cham, 2020, pp. 3–50.

[4] P. Sandner, J. Gross, R. Richter, Convergence of blockchain, IoT, and AI, Frontiers Blockchain 3 (2020) 522600.

[5] C. Paniagua, J. Delsing, Industrial frameworks for internet of things: A survey, IEEE Systems Journal 15 (1) (2020) 1149–1159.

[6] S. R. Bader, M. Maleshkova, S. Lohmann, Structuring reference architectures for the industrial internet of things, Future Internet 11 (7) (2019) 151.

[7] F. Firouzi, S. Jiang, K. Chakrabarty, B. Farahani, M. Daneshmand, J. S. Song, K. Mankodiya, Fusion of IoT, AI, edge-fog-cloud, and blockchain: Challenges, solutions, and a case study in healthcare and medicine, IEEE Internet of Things Journal (2022) 1–22.

[8] F. Shi, H. Ning, W. Huangfu, F. Zhang, D. Wei, T. Hong, M. Daneshmand, Recent progress on the convergence of the internet of things and artificial intelligence, IEEE Network 34 (5) (2020) 8–15.

[9] M. J. Kaur, V. P. Mishra, P. Maheshwari, The convergence of digital twin, IoT, and machine learning: transforming data into action, in: M. Farsi, A. Daneshkhah, A. Hosseinian-Far, H. Jahankhani (Eds.), Digital twin technologies and smart cities, Springer, 2020, pp. 3–17.

[10] F. Firouzi, B. Farahani, M. Barzegari, M. Daneshmand, AI-driven data monetization: The other face of data in IoT-based smart and connected health, IEEE Internet of Things Journal 9 (8) (2020) 5581–5599.

[11] F. Firouzi, B. Farahani, F. Ye, M. Barzegari, Machine learning for IoT, in: F. Firouzi, K. Chakrabarty, S. Nassif (Eds.), Intelligent Internet of Things, Springer, Cham, 2020, pp. 243–313.

[12] C. Resende, D. Folgado, J. Oliveira, B. Franco, W. Moreira, A. Oliveira-Jr, A. Cavaleiro, R. Carvalho, Tip4. 0: Industrial internet of things platform for predictive maintenance, Sensors 21 (14) (2021) 4676.

[13] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, G. P. Li, Predictive maintenance in the industry 4.0: A systematic literature review, Computers & Industrial Engineering 150 (2020) 106889.

[14] S. Hassankhani Dolatabadi, I. Budinska, Systematic literature review predictive maintenance solutions for smes from the last decade, Machines 9 (9) (2021) 191.

[15] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, M. S. Hossain, Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach, IEEE Internet of Things Journal 8 (8) (2020) 6348–6358.

[16] W. Zhang, X. Li, H. Ma, Z. Luo, X. Li, Federated learning for machinery fault diagnosis with dynamic validation and self-supervision, Knowledge-Based Systems 213 (2021) 106679.

[17] J. Zhou, S. Zhang, Q. Lu, W. Dai, M. Chen, X. Liu, S. Pirttikangas, Y. Shi, W. Zhang, E. Herrera-Viedma, A survey on federated learning and its applications for accelerating industrial internet of things, arXiv preprint arXiv:2104.10501.

[18] GAIA-X, https://www.gaia-x.eu/, 2022 (accessed 14 March 2022).

[19] International data spaces association, https://www.internationaldataspaces.org/, 2022 (accessed 14 March 2022).

[20] Data sovereignty – critical success factor for the manufacturing industry, https://internationaldataspaces.org, 2022 (accessed 14 March 2022).

[21] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, L. Zhu, Blockchain-based federated learning for device failure detection in industrial IoT, IEEE Internet of Things Journal 8 (7) (2020) 5926–5937.

[22] N. Ge, G. Li, L. Zhang, Y. Liu, Failure prediction in produc-

tion line based on federated learning: an empirical study, Journal of Intelligent Manufacturing 33 (8) (2022) 2277–2294.
[23] S.-W. Lin, B. Miller, J. Durand, G. Bleakley, A. Chigani, R. Martin, B. Murphy, M. Crawford, The industrial internet of things volume g1: reference architecture, Industrial Internet Consortium 10 (2017) 10–46.
[24] F. Fraile, R. Sanchis, R. Poler, A. Ortiz, Reference models for digital manufacturing platforms, Applied Sciences 9 (20) (2019) 4433.
[25] A. P. Kalogeras, H. Rivano, L. Ferrarini, C. Alexakos, O. Iova, S. Rastegarpour, A. A. Mbacké, Cyber physical systems and internet of things: Emerging paradigms on smart cities, in: Proceedings of the 2019 First International Conference on Societal Automation (SA), IEEE, 2019, pp. 1–13.
[26] Data sharing in industrial ecosystems: Driving value across entire production lines, https://www.mckinsey.de/, 2022 (accessed 14 March 2022).
[27] H. B. Bentzen, R. Castro, R. Fears, G. Griffin, V. Ter Meulen, G. Ursin, Remove obstacles to sharing health data with researchers outside of the European Union, Nature Medicine 27 (8) (2021) 1329–1333.
[28] Q.-V. Pham, K. Dev, P. K. R. Maddikunta, T. R. Gadekallu, T. Huynh-The, et al., Fusion of federated learning and industrial internet of things: a survey, arXiv preprint arXiv:2101.00798.
[29] B. Jiang, J. Li, G. Yue, H. Song, Differential privacy for industrial internet of things: Opportunities, applications, and challenges, IEEE Internet of Things Journal 8 (13) (2021) 10430–10451.
[30] A. Shamir, How to share a secret, Communications of the ACM 22 (11) (1979) 612–613.
[31] A. Nadian-Ghomsheh, B. Farahani, M. Kavian, A hierarchical privacy-preserving IoT architecture for vision-based hand rehabilitation assessment, Multimedia Tools and Applications 80 (20) (2021) 31357–31380.
[32] A. Acar, H. Aksu, A. S. Uluagac, M. Conti, A survey on homomorphic encryption schemes: Theory and implementation, ACM Computing Surveys (CSUR) 51 (4) (2018) 1–35.
[33] I. Damgard, M. Geisler, M. Kroigard, Homomorphic encryption and secure comparison, International Journal of Applied Cryptography 1 (1) (2008) 22–31.
[34] Y. Aono, T. Hayashi, L. Wang, S. Moriai, et al., Privacy-preserving deep learning via additively homomorphic encryption, IEEE Transactions on Information Forensics and Security 13 (5) (2017) 1333–1345.
[35] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, K. Chen, Multi-key privacy-preserving deep learning in cloud computing, Future Generation Computer Systems 74 (2017) 76–85.
[36] Z. Ji, Z. C. Lipton, C. Elkan, Differential privacy and machine learning: a survey and review, arXiv preprint arXiv:1412.7584.
[37] C. Dwork, Differential privacy: A survey of results, in: Proceedings of the International conference on theory and applications of models of computation, Springer, 2008, pp. 1–19.
[38] M. Al-Rubaie, J. M. Chang, Privacy-preserving machine learning: Threats and solutions, IEEE Security & Privacy 17 (2) (2019) 49–58.
[39] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Transactions on Intelligent Systems and Technology (TIST) 10 (2) (2019) 1–19.
[40] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan, et al., Towards federated learning at scale: System design, arXiv preprint arXiv:1902.01046.
[41] T. Li, A. K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Processing Magazine 37 (3) (2020) 50–60.
[42] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, arXiv preprint arXiv:1610.05492.
[43] A. Mitra, R. Jaafar, G. J. Pappas, H. Hassani, Federated learning with incrementally aggregated gradients, in: Proceedings of the 2021 60th IEEE Conference on Decision and Control (CDC), IEEE, 2021, pp. 775–782.
[44] A. Nagar, Privacy-preserving blockchain based federated learning with differential data sharing, arXiv preprint arXiv:1912.04859.
[45] A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, S. Yin, Industrial cyberphysical systems: A backbone of the fourth industrial revolution, IEEE Industrial Electronics Magazine 11 (1) (2017) 6–16.
[46] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet of Things Journal 3 (5) (2016) 637–646.
[47] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. Rodrigues, M. Guizani, Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay, IEEE communications magazine 56 (2) (2018) 44–51.
[48] F. Firouzi, B. Farahani, E. Panahi, M. Barzegari, Task offloading for edge-fog-cloud interplay in the healthcare Internet of Things (IoT), in: Proceedings of the 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), IEEE, 2021, pp. 1–8.
[49] W. Dai, H. Nishi, V. Vyatkin, V. Huang, Y. Shi, X. Guan, Industrial edge computing: Enabling embedded intelligence, IEEE Industrial Electronics Magazine 13 (4) (2019) 48–56.
[50] F. Firouzi, K. Chakrabarty, S. Nassif, Intelligent Internet of Things: From Device to Fog and Cloud, Springer, Cham, 2020.
[51] Data sovereignty–requirements analysis of manufacturing use cases, https://www.internationaldataspaces.org/, 2022 (accessed 2 August 2022).
[52] Collaborative data-driven business models: Collaborative condition monitoring – how cross-company collaboration can generate added value, https://www.bmwi.de/redaktion/en/artikel/digital-world/gaia-x-use-cases/collaborative-conditionmonitoring.html, 2022 (accessed 14 March 2022).
[53] Collaborative condition monitoring, https://www.bmwi.de/redaktion/en/artikel/digital-world/gaia-x-use-cases/collaborative-condition-monitoring.html, 2022 (accessed 14 March 2022).
[54] M. Redeker, J. N. Weskamp, B. Rössl, F. Pethig, Towards a digital twin platform for industrie 4.0, in: Proceedings of the 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), IEEE, 2021, pp. 39–46.
[55] Schwan dataset, https://github.com/deeptichevvuri/predictive-maintenance-modelling-datasets, 2022 (accessed 1 January 2022).
[56] A. Ghosh, J. Chung, D. Yin, K. Ramchandran, An efficient framework for clustered federated learning, Advances in Neural Information Processing Systems 33 (2020) 19586–19597.
[57] C. Briggs, Z. Fan, P. Andras, Federated learning with hierarchical clustering of local updates to improve training on non-iid data, in: Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–9.
[58] V. Smith, C.-K. Chiang, M. Sanjabi, A. Talwalkar, Federated multi-task learning, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, p. 4427–4437.
[59] S. Caldas, V. Smith, A. Talwalkar, Federated kernelized multi-task learning, in: Proceedings of the SysML Conference, 2018, pp. 1–3.