An IoE-based Framework Supporting Human-Centric Industry

An IoE-based Framework Supporting Human-Centric Industry

Marco Arazzi*, Alberto Belli§, Claudio Cusano*, Marco Esposito§, Tullio Facchinetti*, Marco Ferretti*, Gabriele Galimberti‡, Monica Marconi Sciarroni§, Paolo Napoletano‡, Antonino Nocera*, Paola Pierleoni§, Emanuele Storti§, Domenico Ursino§

* Department of Industrial, Computer and Biomedical Engineering, University of Pavia, Pavia, Italy
§ Department of Information Engineering, Polytechnic University of Marche, Ancona, Italy
‡ Department of Informatics, Systems and Communication, University of Milan-Bicocca, Milan, Italy
Email: antonino.nocera@unipv.it, e.storti@univpm.it, paolo.napoletano@unimib.it

***Abstract*—Industry 5.0 envisions manufacturing systems that are human-centric, sustainable, and resilient. In this context, the Internet of Everything (IoE) enables integration of devices, people, and processes into a unified digital ecosystem. This paper presents a modular, semantically enriched framework that supports this transition by managing heterogeneous data sources—such as IoT sensors, wearable devices, and smart objects—through a layered architecture. The platform enables real-time data stream processing, semantic interoperability, and secure, context-aware access. Anomaly detection is enabled through a privacy-preserving mechanism based on behavioral fingerprinting and federated learning. The platform supports immersive human-machine interaction via gesture recognition, empowering workers to control and interact with industrial systems. Use cases demonstrate the system's ability to support gesture-based control and intelligent monitoring, highlighting its potential to enhance adaptability, security, and worker empowerment in Industry 5.0 environments.**

***Index Terms*—Data Stream, Big Data, Knowledge graphs, Metadata, Industry 5.0, IoT, Internet of Everything, Anomaly detection, Human-machine interaction**

I. Introduction

Industry 5.0 is a novel paradigm that describes a profound shift in manufacturing systems, highlighting human-centricity, sustainability, and resilience. In this setting, the Internet of Things and its evolution, named Internet of Everything (IoE), which extends traditional IoT by enabling seamless integration of devices, people, and processes into a unified digital ecosystem, play a crucial role. The heterogeneity of the entities involved and the complex interconnections among them require semantic alignment, the design of responsive data management solutions, and secure interaction modules.

This paper proposes a comprehensive IoE-based framework designed to support the development of human-centric applications in Industry 5.0. The framework is built on a modular and layered architecture that can manage and integrate heterogeneous data streams originating from IoT sensors, smart objects, wearable devices, and industrial processes. The semantical alignment of the involved data sources is obtained through a central Knowledge Graph (KG), which provides a uniform representation of core entities such as human agents, machines, objects, locations, activities, roles, and access rights. This semantic layer ensures interoperability and consistency of the different components producing and ingesting data and enables functionalities such as reasoning, querying, and context-aware data access. The proposed framework addresses the entire data lifecycle, targeting data stream acquisition, preprocessing, aggregation, transformation, storage, and semantic querying.

The heterogeneity of IoE smart components (such as sensors, smart devices, cobot, and so forth) exposes the framework to security threats that can be related to potential cyber attacks targeting such components or their malfunctioning. To address these issues and guarantee self-healing properties to the IoE backbone, the framework also includes a novel Anomaly Detection module that exploits the behavior analysis of the involved components to identify anomalies. Behavior modeling, also known as behavioral fingerprinting, and prediction is obtained by training suitable deep learning models based on message exchanges among peer components. The approach included in our framework advances traditional behavioral fingerprinting techniques by introducing a fully distributed solution that leverages Federated Learning to model the global behavior of each target component as observed collectively across the entire IoE ecosystem. Such a decentralized approach also supports privacy preservation, as generated data can be processed locally to train local behavioral models that are subsequently aggregated by designated components within the IoE infrastructure.

In parallel with these architectural, security, and semantic innovations, advanced Human-Machine Interfaces (HMIs) are gaining increasing relevance in Industry 5.0 scenarios.

This work has been partially supported by the PRIN 2022 Project "HOMEY: a Human-centric IoE-based Framework for Supporting the Transition Towards Industry 5.0" funded by the European Union - Next Generation EU, mission 4 component 1 (code: 2022NX7WKE, CUP: F53D23004340006).

DOI: 10.1109/ETFA65518.2025.11205771.

By leveraging wearable sensing technologies and real-time data processing, such interfaces enable intuitive, secure, and context-aware interaction between operators and smart environments. In this framework, gesture-based interaction via smartwatches and inertial sensors allows workers to control industrial processes and digital interfaces in a natural and contactless manner [1], [2]. Furthermore, wearable-enabled teleoperation facilitates seamless remote control of robotic equipment, enhancing flexibility and responsiveness in industrial setup [3], [4]. The proposed IoE infrastructure supports these HMI paradigms by integrating multimodal smart objects and providing a unified communication and processing layer for gesture recognition, human monitoring, and robot control.

II. Related Work

The emerging paradigm of Industry 5.0 marks a shift from traditional industrial models toward smart, sustainable, and human-centric manufacturing environments [5]. Within this context, conventional IoT technologies are being extended by the broader and more complex Internet of Everything (IoE) paradigm [6], which includes data generated not only by devices but also by humans, processes, and complex interactions among them. To support such scenarios, industrial infrastructures must evolve toward adaptive and interoperable architectures. These architectures should be capable of managing heterogeneous devices, diverse communication protocols, and a wide range of tasks, while ensuring both physical and semantic interoperability across applications and services [7]. Recent research has predominantly focused on IoT architectures encompassing data collection, processing, and storage, with sensors typically serving as the primary data publishers. Various architectural approaches have been proposed, including multi-tier storage models for Industry 4.0 applications [8] and AI-enhanced systems for advanced data analysis [9]. In parallel, the integration of Knowledge Graphs within IoT/IoE ecosystems is increasingly recognized as a pivotal strategy to enable broad interoperability among heterogeneous data producers and consumers, thereby facilitating flexible data management and analysis [10]. Semantic-based techniques have been employed for a range of purposes, including enriching data representation [11], enhancing communication efficiency, implementing a semantic monitoring [12], and supporting semantic data analytics through process-aware methodologies [13].

With the increasing complexity and widespread adoption of IoT-based systems, both the attack surface and the potential impact of threats in this domain are growing significantly [14], [15]. In recent years, the research community has proposed numerous countermeasures to address a wide range of threats targeting IoT devices [16]–[19], with more recent efforts incorporating Machine Learning (ML) and Deep Learning (DL) approaches [20]. A current trend involves designing ML and DL algorithms that learn distinctive traits of target devices to identify compromised nodes within a network. The collection of characteristics exhibited by an IoT device during its communication with other entities over a network forms what is known as its fingerprint. A more advanced approach to this is behavioral fingerprinting [21]–[24], which extracts high-level features based on device interactions to model their network behavior. For instance, [21] introduces a method that gathers features from a device's network traffic to train an ML model capable of recognizing similar device categories. Similarly, [24] presents a detection framework that applies behavioral fingerprinting along with ML techniques to identify anomalies and classify threats such as botnets, rootkits, backdoors, and ransomware targeting real-world IoT spectrum sensors.

Recent advances in wearable sensing and machine learning have enabled intuitive and immersive Human-Machine Interfaces for Industry 5.0 applications [25]. In particular, gesture recognition systems that leverage smartwatches and wearable inertial sensors enable operators to interact with industrial equipment in a natural, secure, and contactless manner [4], [26]. These interfaces enhance worker empowerment by enabling context-aware control and reducing cognitive and physical workload [3]. Current research explores gesture-driven control schemes integrated with IoE infrastructures, where sensor data streams are semantically annotated and processed in real time. Approaches such as those presented in [1] demonstrate how smartwatches can reliably detect arm gestures using supervised learning models, enabling remote actuation of robots or smart screens within cyber-physical environments.

III. Data model

The framework's data model is designed to: (i) represent the Internet of Everything (IoE), encompassing buildings, agents, roles, devices, smart objects, and activities, (ii) characterize smart objects and devices based on their technical specifications and functional capabilities, (iii) define rules and constraints for managing access, (iv) store data generated by smart devices and employees, including measurements, environmental parameters, and records of actions and activities. The first three aspects (i–iii) relate to the deployment, planning, and configuration of the IoE environment. The fourth aspect (iv) focuses on capturing and analyzing real-time task execution and performance, including the storage of data produced by both devices and personnel. In Industry 5.0, data management requires flexible, interoperable, and standards-compliant structures. For storing operational data (aspect iv), traditional database management systems (DBMS) are used. In contrast, aspects (i–iii) leverage a Knowledge Graph (KG) model, following best practices from existing literature. Knowledge Graphs offer a standardized way to integrate and represent diverse data sources, improving interoperability through shared vocabularies and structures. They also simplify querying and analysis by making relationships between entities more accessible, reducing the effort needed to retrieve and connect relevant information.

This approach enables a unified, comprehensive representation of both IoE knowledge and technical information supporting specific services.

A. IoE knowledge

The KG leverages SemIoE [27], an OWL2 lightweight ontology for defining IoE entities and their relationships, as its schema. The ontology incorporates and extends various external modules to represent specialized aspects (e.g., W3C Semantic Sensor Network (SSN) ontology [11]). Its main classes include *Agent*, representing an employee or a smart object, the *Site* in which it is located, and related containment relations (e.g., a lab is located on a floor which is part of a building), and the *Activity* it is executing. A *SmartObject* consists of one or more *System*s (e.g., CO2 sensors, ventilation actuators), each characterized by a set of technical properties and operating conditions. Furthermore, each agent is associated with a *Role* and corresponding *Rights* to access specific systems, smart objects, or entire sites, and a set of *Preferences* for environmental parameters.

We refer the interested reader to http://w3id.org/semioe for the ontology specifications.

B. Technical knowledge

The technical KG is aimed to support data collection, monitoring, and access control, including operational metadata for systems, such as the notion of *Stream*, the related *Topic*, the *Field*s included in the schema of the generated messages, the target *Storage* system (e.g., a time-series database, relational database, or file storage), specific fields to monitor or store. It also documents the transformations applied to data streams, thereby supporting the stream processing steps managed by the Stream Management Platform discussed in the next section.

IV. Platform Architecture

In this section, we introduce the platform architecture [28], as illustrated in Figure 1. The platform is organized into multiple layers for data stream management, which are composed of modules for data stream acquisition, manipulation, real-time monitoring, persistent storage in DBMSs, and data access. The architecture is based on micro-services and also leverages the Knowledge Graphs to enhance data integration and accessibility.

Real-time data streams are produced by a variety of devices, including IoT sensors, wearable devices, and smart objects, related to a variety of environmental and operational parameters, with heterogeneous formats and different communication protocols. Depending on their deployment configuration, data can be sent either directly to the platform or can be routed through local gateways. These gateways collect and aggregate streams from multiple co-located sources before forwarding them to a broker, e.g., Mosquitto[1] for MQTT traffic, and then to the platform. For example, temperature and humidity sensors may adopt the MQTT protocol and send data to a gateway. Its agnostic payload format allows the use of different serialization schemes (e.g., JSON), while also supporting simplified integration of data compression and encryption through lightweight protocols such as JOSE (for JSON) and COSE

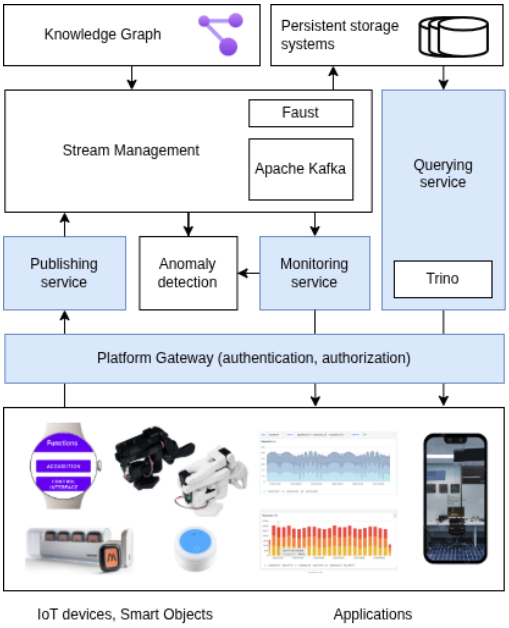


Fig. 1: Platform architecture. Modules in blue represent services callable through public APIs.

(for binary data). These protocols have demonstrated excellent performance in terms of latency and memory usage when used together with MQTT, making it possible to secure streams and application-level payloads with a very small overhead [29]. Conversely, some wearable devices, such as wristband health monitors, can send physiological data over Bluetooth Low Energy (BLE) to a local aggregator node such as a smartphone, which can transmit data directly to the stream management platform using HTTP.

The stream management platform collects incoming streams through a preprocessing bridge: a layer that applies a set of transformations before forwarding the enriched streams to the core management platform. The core of the stream management infrastructure is built upon Apache Kafka[2], a widely adopted open-source platform for distributed stream processing, specifically designed to handle real-time data workflows. Kafka logically organizes the incoming streams into structured topics, enabling parallel consumption and ensuring horizontal scalability. As such, it follows a publish/subscribe messaging paradigm that allows clients to subscribe to specific topics of interest. This leads the system to efficiently collect, process, and route high-throughput, heterogeneous data streams originating from multiple sources. Data streams are routed towards three main downstream services: stream monitoring, persistent storage, and querying.

A. Stream Monitoring service

The stream monitoring service enables real-time observation and analysis of incoming data streams. The platform leverages a semantic layer, which upgrades the basic monitoring

[1]https://mosquitto.org/

[2]https://kafka.apache.org/

approach to a semantically enhanced monitoring approach, decoupling data stream access from fixed topic naming. This approach allows users to formulate monitoring requests in semantic terms, as a set of high-level constraints derived from the KG. Constraints may refer to concepts such as measurement type, device location, or sensor class. These constraints are translated into SPARQL queries, possibly leveraging reasoning to infer additional knowledge. This process involves (1) identifying all devices in the KG that meet the given constraints, (2) retrieving their data schemas and Kafka topic identifiers, and (3) subscribing to the appropriate topics. A post-processing module built on Faust Streaming [3] is employed to support real-time stream aggregation. Faust Streaming is a Python-based stream processing library, inspired by the Kafka Streams [4] model, designed to support stateful and windowed stream transformations in real time. In our architecture, Faust agents consume messages from Kafka topics and apply aggregation functions over arbitrary sliding/tumbling time windows (e.g., compute the average of a monitored parameter within windows of length 30 seconds). This approach serves multiple purposes: firstly, it improves data interpretability for end-users by providing summarized views that are more intuitive for analysis, as it facilitates the identification of relevant trends and patterns. Secondly, in scenarios where sensors generate high-frequency data, storing raw streams may result in significant overhead in terms of both storage capacity and write throughput; by pre-aggregating measurements before persistence, the system reduces the data volume, thereby improving resource efficiency.

B. Storage service

A persistent Storage service is designed to store both raw and post-processed data streams into appropriate DataBase Management Systems (DBMSs). The choice of the DBMS depends on the data type, data schema, and intended use case (e.g., time-series databases for sensor logs, document stores for semi-structured data, relational databases for structured records). Metadata about storage configuration is included in the KG. For each stream, the service is responsible for subscribing to the corresponding topic, consuming messages, and executing the writing process to the designated databases. Once stored, data is available for historical analysis and business intelligence tools. This allows users to perform complex time-based queries, and in order to enable unified access across a heterogeneous set of storage back-ends, a dedicated Querying service is provided.

C. Querying service

Considering the variety of data storage systems and the heterogeneity of the stored data, the Querying service implements a semantic-based mechanism to uniformly extract information based on a user request. Users are required to specify at least one mandatory input parameter, which can be either the name of a sensor or the name of a measured parameter, and a set of optional parameters, such as time window, aggregate functions, and sorting preferences. Based on these inputs, the platform leverages the technical graph to retrieve mappings between sensors and their storage backends. In order to write the right query, the service is powered by Trino [5], a distributed SQL query engine, which enables federation across multiple DBMSs, allowing data stored in heterogeneous DBMSs to be accessed through a unified SQL interface. A structured SQL query is, then, dynamically constructed and executed via Trino.

V. Securing the IoE

The platform adopts a multi-faceted approach to security, combining authentication and authorization mechanisms with context-aware access control and anomaly detection.

A. Role-based authentication, authorization and access

First, a multi-level, role-based access control (RBAC) module ensures that only authenticated and authorized *Agents* can interact with platform services: upon authentication, the system identifies the Agent's current role and possibly enables access to specific APIs, e.g., the Monitoring service.

Second, access to content, e.g., to specific data streams or to data stored in databases, is governed by context-aware policies defined in the platform's Knowledge Graph, based on the agent's role and corresponding rights to access specific systems, smart objects, or entire sites. These policies are dynamically evaluated based on the agents' contexts, including their location and possible delegation relations, enabling fine-grained control over data streams and service interactions.

B. Anomaly detection strategy

Due to the heterogeneity of smart objects that compose an IoE network in Industry 5.0, the risk of potentially maliciously controlled or malfunctioning devices can compromise the system's fulfillment of its tasks. To overcome this potential issue, the proposed framework includes a fully distributed anomaly detection technique, originally presented in [22], [30], [31], based on the fingerprinting of the normal behavior of the object, showing an advantage since it does not require knowing any malicious behavior but relies on the normal activity of the devices. The basic idea behind this approach is to teach a Neural Network to predict the next possible packet given a short sequence of the preceding ones. Subsequently, the predicted packets are compared to the packets actually received. If, within a specified window, over 50% of the predicted packets differ from the received packets, this is considered an anomaly. In particular, the selected model is a Gated Recurrent Unit (GRU) neural network composed of two recurrent layers, each with 512 and 256 neurons, respectively, and a fully connected layer with 128 hidden neurons. In order to build a behavioral fingerprint model for a specific object, a device connects to the monitoring service to retrieve the packets queued in the stream management service or directly from the MQTT broker. To trigger this procedure, a given device can exploit the delegation mechanism offered by the

[3]https://github.com/faust-streaming/faust

[4]https://kafka.apache.org/documentation/streams/

[5]https://trino.io/

Knowledge Graph, based on the SemIoE ontology, to find the best candidates that can train a model using their data.

C. Distributed Anomaly Detection

As described in [30], the training of the anomaly detection model can also be performed in a distributed fashion using Federated Learning. Each device provides different services to different objects, which makes it difficult to obtain a functional model capable of forecasting the packets generated by different services. The advantage of using Federated Learning is to leverage the data collected by the different devices that use different services of a single device to build a global model capable of predicting packets generated by different services. To make the process privacy-preserving, the framework proposed in this paper provides different solutions. First of all, the devices that take part to the training do not have to know about the others involved; they need to know only which device will perform the aggregation of the single local updates. In line with the previous section, the device that wants to be fingerprinted can leverage the delegation feature offered by the KG to identify an appropriate aggregator. According to [30], rather than relying on a blockchain, the target device may utilize the proposed framework as a public ledger by publishing a homomorphically encrypted secret on the Kafka stream. This allows collaborating devices to determine the designated aggregator. In this scenario as well, the KG facilitates the process by offering collaboration logic. Concurrently, updates to local models can be shared via a dedicated queue in Kafka and collected by the aggregator. In this instance, disclosing the clear updates within the stream management system increases the risk of information leakage from worker devices. To prevent this, each device employs homomorphic encryption on its updates, ensuring the preservation of the updates' mathematical properties while simultaneously safeguarding against information disclosure.

D. Feature Extraction for the Anomaly Detection

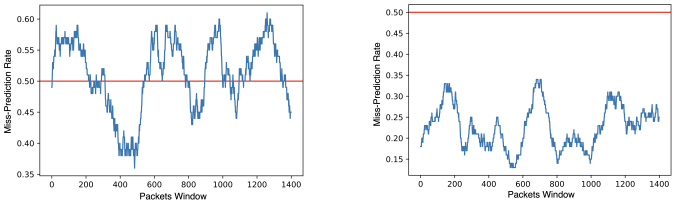
As anticipated in the previous sections, the proposed behavioral fingerprinting model is intended to be used to verify if a new received packet is expected or not by the normal behavior of the monitored device. To do so, it is fundamental to train the model using meaningful features that must not be altered or lost in the delivery through the proposed framework from the device that generates them to the final device that receives them through the monitoring service. For this experiment, we used the ayyoob dataset[6]. In particular, the main selected features extracted from each packet are: *Source port type*, *TCP flags*, *Encapsulated protocol types*, *Inter-arrival times between packets*, *Packet length*, and *Payload Value*.

Inspecting the single feature, most of them are interdependent on possible alterations caused by the throttling of the system under exceptionally heavy loads. The only exceptions are for the inter-arrival time between the packets. This is a fundamental feature for the proposed model, so possible delays in the arrival of the packets can negatively affect the reliability of the model. In this sense, stream preprocessing plays a crucial role in extracting the features before feeding the packets into the stream management service, helping to avoid the bottlenecks that could delay or alter the normal flow of packet arrival. To prove the importance of this preprocessing, we simulated four different scenarios. In three of them, the extraction is done by the final recipient under varying levels of throttling: $Light$, $Medium$ or $Heavy$. In the last scenario, instead, the feature extraction is performed by the preprocessing unit. In this experiment, we considered three different target devices to train the model against the 4 considered scenarios. Results are reported in Table I.

TABLE I: Model accuracy under different levels of throttle compared to the scenario in which the features are extracted by the Stream Management Platform during preprocessing.

Scenario	Model Accuracy		
	Device 1	Device 2	Device 3
Heavy Throttle	49.3%	40.2%	46.7%
Medium Throttle	59.9%	53.3%	57.1%
Light Throttle	64.2%	55.6%	63.1%
Preprocessing	75.1%	75.2%	72.2%

As expected, the performance of the final model is highly affected by the extraction of good features, especially in this case, the inter-arrival time between packets. As we can see, under conditions of heavy throttling, the accuracy of the model drops by 25%, making it unusable. As illustrated in Figure 2, without the preprocessing of the features under Heavy throttling, the model detects the normal traffic as malicious, exceeding the detection threshold of misclassifications of the packets in a given window (the red line), set at 0.5. This result confirms the importance of a preprocessing step to ensure good performance of the behavioral fingerprinting model.



(a) Heavy throttle without Preprocessing

(b) Heavy throttle with Preprocessing

Fig. 2: Anomaly Detection based on a behavioral fingerprinting model under the scenario *heavy throttle* causing delay in the packets delivery (a) and the results of the detection in the same setting when the preprocessing stream service is used to avoid feature alterations (b).

VI. Enabling Advanced HCI in the IoE Infrastructure

Our proposed IoE infrastructure not only integrates heterogeneous data sources within an environment composed of *smart objects*, but also enables advanced human-machine interactions through them. These smart objects—such as wearable

[6]The dataset is available at https://iotanalytics.unsw.edu.au/attack-data.html

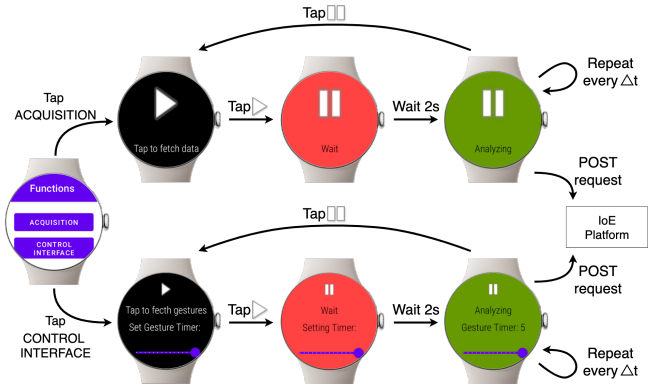


Fig. 3: Smartwatch Wear OS application for gesture recognition and data acquisition.

devices, robotic systems, and intelligent interfaces—serve as both data producers and interaction endpoints, allowing users to actively engage with the IoE ecosystem in a natural and context-aware manner. While various types of smart objects can be integrated within an IoE infrastructure, for experimental purposes, we included the following representative components: wearable inertial measurement units (IMUs based on Movella XSens DOT), a robotic arm (LeRobot SO100), and a smartwatch (Google Pixel 2). These elements were selected to experiment with several Human-Computer Interaction (HCI) scenarios representative of Industry 5.0 environments. Specifically, the infrastructure supports: (i) the logging of heterogeneous data streams within the IoE ecosystem; (ii) the control of visual interfaces through arm gestures recognized by wearable devices; and (iii) the teleoperation of robotic arms using both inertial sensors and smartwatch-based input. Each smart object formats data as a JSON document, as described in Subsection VI-D, which is then published to the platform.

A. Smartwatch

The IoE infrastructure interacts with a Google Pixel Watch 2 smartwatch equipped with a Wear OS custom application to support two possible applications: arm gesture recognition and acquisition of human data. Once authenticated with the infrastructure, the user accesses the Wear OS application containing two buttons, each corresponding to a different smart object configuration. Upon selecting one of the options, the smartwatch is configured to operate accordingly. Figure 3 shows a visual sequence of possible uses of the application.

1) Arm-gesture: Following this approach, arm gestures are recognized using a deep learning model based on a Long Short-Term Memory (LSTM) network developed by Colombo et al. [1]. The algorithm processes a time window of approximately 2 seconds of IMU data and classifies one of 25 predefined arm gestures, including a rejection class that indicates the absence of any meaningful gesture. The smartwatch application allows the user to select the duration of the acquisition window via a slider so that it is possible to recognize multiple consecutive gestures.

2) Data acquisition: This mode enables continuous sampling of sensor data from the smartwatch. After the user presses the Play button, the application waits for 2 seconds before initiating a loop that collects data from the IMU and other onboard sensors at fixed intervals of approximately 5 ms. Sensor update rates depend on the sensor type: IMU sensors and derived values (e.g., gravity vector, compass heading, and device pose) are updated at frequencies of up to 200 Hz, while context-dependent sensors—such as humidity, temperature, barometer, altimeter, and ambient light—are sampled at a much lower rate (0.1 Hz, or once every 10 seconds).

B. Robotic Arm

A robotic arm interacts with the IoE infrastructure through a computer desktop to support two possible applications: robotic arm gesture recognition and acquisition of motor data. After that, the robotic arm is authenticated with the infrastructure, and the user accesses the functions of the robotic arm using a Graphical User Interface (GUI) on a computer desktop containing two buttons, each corresponding to a different smart object configuration. The computer desktop is configured to select one of the options, and the robotic arm receives or sends data to control motors movement. In Section VI-F, we will discuss a possible use of the application.

1) Robotic arm gesture: A deep learning model can be trained on the motor values of a robot using imitation learning. In this approach, the algorithm outputs an array of motor values that guide the robotic arm to imitate human gestures and perform tasks. During these tasks, a deep learning model based on a Long Short-Term Memory (LSTM) network can be used to recognize which gesture the robotic arm is doing. The algorithm processes a time window of approximately 10 seconds of motor values and classifies one of 25 predefined human arm gestures, including a rejection class that indicates the absence of any meaningful gesture.

2) Motor data acquisition: This application enables continuous sampling of motor values from the robotic arm teleoperated by another device (eg, Movella XSens Dot or a different robotic arm). After the user presses the Recording button, the application waits for a warm-up time of 5 seconds before initiating a loop that collects motor values at fixed intervals of approximately 8 ms (∼128 Hz).

C. Wearable IMU unit

The IoE infrastructure interacts with a computer desktop connected to a set of Movella XSens Dot through an application. Once authenticated with the infrastructure, the application allows sampling data from the set based on a configuration at fixed intervals of approximately 50 ms.

D. IoE platform communication

Once a smart object is connected to a network, the smart object can send an information packet to the IoE Infrastructure containing a JSON document specifically structured for this environment. An example of a JSON document used by the *arm-gesture* object is shown in code List 1. This

JSON document is composed of two main keys: the key "Identifier" is used to recognize the smart object within the IoE Infrastructure; the other key describes the smart object's "type", which contains the set of measurements performed by that specific smart object. The identifier consists of three keys, each one associated with an alphanumeric value: a unique identifier "id" for the smart object, a "device" name indicating the name of the smart object, and the "model" of the smart object device. Instead, the smart object *type* defines which registered smart object within the IoE Infrastructure the measurements are being attributed to. This key also defines, for each measurement, which data values are acquired and uploaded to the infrastructure. Each smart object type uses distinct keys to indicate updates to specific data within the JSON file. In the example shown in code List 1, *arm-gesture* object uses, as value of the type "GestureData", the key pair "gesture" and "timestampGestureData" to denote both the newly sampled gesture and the exact time (timestamp) at which it is sent to the IoE Infrastructure. Consequently, each smart object may employ a distinct set of key-value pairs, depending on its data type and functional role. Figure 4 shows an example of the Knowledge Graph constructed by the IoE platform related to the Smartwatch and its functionalities, arm gesture, and data acquisition. Each color of the graph refers to a given type of node, for instance, smart objects are colored as light purple, properties are colored as cian etc.

Listing 1: JSON for the smart object *arm-gesture*.

```
{
  "Identifier": {
    "id": String,
    "device": String,
    "model": String
  },
  "GestureData": {
      "timestampGestureData": Long,
      "gesture": String
   }
}
```

The information packet header includes an authentication *token* issued by the IoE Infrastructure during the authentication phase (see Subsection V-A).

E. Use case - Interaction with Smart Front-End

A use case example of our IoE-based platform consists of controlling a smart front-end, typically an adaptive dashboard, through the gestures recognized by a wearable unit of a human operator. Figure 5 shows a graphic representation of this use case. Our framework enables the integration of gestures detected by smart objects, such as a `smart_watch_gesture` event from a smartwatch, with the functionalities of a dynamic front-end interface. Once captured by the device, the gesture is propagated through the IoE infrastructure and made available to the front-end via the *Stream Monitoring Service*. The dynamic front-end interprets the gesture to adapt the visual content accordingly, aligning the graphic interface with the intent encoded in the user's interaction.

Fig. 4: Fragment of the KG related to the smartwatch used for arm gesture recognition. Nodes are colored according to their type: classes (purple, on the right), sites (brown), smart objects (light purple), systems (green), property (cian), streams (red), message schema (yellow), and literals (khaki).

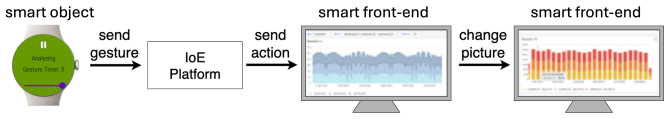


Fig. 5: Use case: sending an action to a smart front-end.

F. Use case - Remote Control Robots

Another use case example using our IoE platform consists of controlling a remote robot through wearable sensors located far from the robot. The Figure 6 shows the example of this use case. The IoE infrastructure detects a new message event (i.e., the reception of a JSON document) sent by a smart object (e.g., "`smart_watch_acquisition`" from a smartwatch or wearable sensor) which computes a movement. This message contains a required movement computed by a smartwatch. The gesture is stored by the IoE Infrastructure. The IoE Infrastructure subsequently sends an input message to a desktop controlling a robot. The message contains the movement required by the robot.

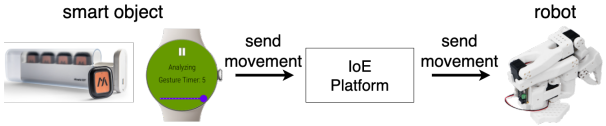


Fig. 6: Use case: sending a movement to a robot.

VII. Discussion and Future Work

This work has introduced a modular, semantically enriched IoE-based framework designed to support the development of human-centric applications in Industry 5.0. The proposed platform integrates heterogeneous data sources through a layered architecture, facilitating real-time data processing, semantic interoperability, and secure interaction between humans and machines. The implementation of multiple use cases, such as gesture-based control via smartwatches and robotic teleoperation using wearable IMUs, shows the framework's adaptability and effectiveness in enabling natural and context-aware interactions in an Industry 5.0 context, particularly in terms of worker empowerment and seamless integration between cyber and physical systems. Future developments will focus on several directions. On the one hand, the integration of additional smart objects and interaction modalities, including Augmented Reality interfaces, could broaden the spectrum of human-machine interactions. A particularly relevant enhancement under exploration is also the integration of a smartwatch-based application for safe task assignment, based on the methodology discussed in [32]. The app receives the worker's assigned activities from the enterprise workflow system, then automatically rejects a task if its estimated effort exceeds the current stress level of the worker, estimated from physiological parameters measured by the smartwatch. Rejected activities will then be reassigned, enhancing well-being, load balancing, and reducing occupational risk.

References

[1] A. Colombo, L. Celona, S. Bianco, A. Nocera, and P. Napoletano, "Arm gesture recognition with smartwatches," in *2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI)*. IEEE, 2024, pp. 625–629.

[2] S. Bianco, P. Napoletano, A. Raimondi, and M. Rima, "U-wear: User recognition on wearable devices through arm gesture," *IEEE transactions on human-machine systems*, vol. 52, no. 4, pp. 713–724, 2022.

[3] A. Fornaro, D. D'Auria, H. Amrani, and P. Napoletano, "Responsive teleoperation of a robotic arm via wearable inertial sensors," in *2024 IEEE Gaming, Entertainment, and Media Conference (GEM)*. IEEE, 2024, pp. 1–6.

[4] I. E. Stan, H. Amrani, P. Napoletano, and D. D'Auria, "Authenticated robotic teleoperation with task recognition," *IEEE Consumer Electronics Magazine*, 2025, accepted for publication.

[5] P. K. R. Maddikunta, Q.-V. Pham, B. Prabadevi, N. Deepa, K. Dev, T. R. Gadekallu, R. Ruby, and M. Liyanage, "Industry 5.0: A survey on enabling technologies and potential applications," *Journal of Industrial Information Integration*, vol. 26, p. 100257, 2022.

[6] J. Leng, W. Sha, B. Wang, P. Zheng, C. Zhuang, Q. Liu, T. Wuest, D. Mourtzis, and L. Wang, "Industry 5.0: Prospect and retrospect," *Journal of Manufacturing Systems*, vol. 65, pp. 279–295, 2022.

[7] H. Rahman and M. I. Hussain, "A comprehensive survey on semantic interoperability for internet of things: State-of-the-art and research challenges," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 12, p. e3902, 2020.

[8] K. Villalobos, V. Ramírez-Durán, B. Diez, J. Blanco, A. Goñi, and A. Illarramendi, "A three level hierarchical architecture for an efficient storage of industry 4.0 data," *Computers in Industry*, vol. 121, p. 103257, 2020.

[9] M. Bolanowski, A. Paszkiewicz, T. Żabiński, G. Piecuch, M. Salach, and K. Tomecki, "System architecture for diagnostics and supervision of industrial equipment and processes in an ioe device environment," *Electronics*, vol. 12, no. 24, 2023.

[10] X. Li, M. Lyu, Z. Wang, C.-H. Chen, and P. Zheng, "Exploiting knowledge graphs in industrial products and services: A survey of key aspects, challenges, and future perspectives," *Computers in Industry*, vol. 129, p. 103449, 2021.

[11] M. Compton *et al.*, "The ssn ontology of the w3c semantic sensor network incubator group," *Journal of Web Semantics*, vol. 17, pp. 25–32, 2012.

[12] T. C. Piller and A. Khelil, "Semsub: Semantic subscriptions for the mqtt protocol," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020, pp. 1–6.

[13] C. Diamantini, A. Mircoli, D. Potena, and E. Storti, "Process-aware iiot knowledge graph: A semantic model for industrial iot integration and analytics," *Future Generation Computer Systems*, vol. 139, pp. 224–238, 2023.

[14] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[15] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Information systems frontiers*, vol. 17, pp. 243–259, 2015.

[16] D. Kozlov, J. Veijalainen, and Y. Ali, "Security and privacy threats in iot architectures." in *BODYNETS*, 2012, pp. 256–262.

[17] S. Tweneboah-Koduah, K. E. Skouby, and R. Tadayoni, "Cyber security threats to iot applications and service domains," *Wireless Personal Communications*, vol. 95, pp. 169–185, 2017.

[18] F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera, "A privacy-preserving localization service for assisted living facilities," *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 16–29, 2016.

[19] S. Sicari, C. Cappiello, F. De Pellegrini, D. Miorandi, and A. Coen-Porisini, "A security-and quality-aware system architecture for internet of things," *Information Systems Frontiers*, vol. 18, pp. 665–677, 2016.

[20] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (iot) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[21] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "Behavioral fingerprinting of iot devices," in *Proceedings of the 2018 workshop on attacks and solutions in hardware security*, 2018, pp. 41–50.

[22] A. Aramini, M. Arazzi, T. Facchinetti, L. S. Ngankem, and A. Nocera, "An enhanced behavioral fingerprinting approach for the internet of things," in *2022 IEEE 18th International Conference on Factory Communication Systems (WFCS)*. IEEE, 2022, pp. 1–8.

[23] M. Ferretti, S. Nicolazzo, and A. Nocera, "H2O: Secure Interactions in IoT via Behavioral Fingerprinting," *Future Internet*, vol. 13, no. 5, p. 117, 2021.

[24] A. H. Celdrán, P. M. S. Sánchez, M. A. Castillo, G. Bovet, G. M. Pérez, and B. Stiller, "Intelligent and behavioral-based detection of malware in iot spectrum sensors," *International Journal of Information Security*, pp. 1–21, 2022.

[25] K. R. Pyun, K. Kwon, M. J. Yoo, K. K. Kim, D. Gong, W.-H. Yeo, S. Han, and S. H. Ko, "Machine-learned wearable sensors for real-time hand-motion recognition: toward practical applications," *National Science Review*, vol. 11, no. 2, p. nwad298, 2024.

[26] H. Jeon, H. Choi, D. Noh, T. Kim, and D. Lee, "Wearable inertial sensor-based hand-guiding gestures recognition method robust to significant changes in the body-alignment of subject," *Mathematics*, vol. 10, no. 24, p. 4753, 2022.

[27] M. Arazzi, A. Nocera, and E. Storti, "The semioe ontology: A semantic model solution for an ioe-based industry," *IEEE Internet of Things Journal*, 2024.

[28] M. M. Sciarroni, M. Esposito, P. Pierleoni, and E. Storti, "Monitoring data streams in industry 5.0: a knowledge graph approach," in *2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI)*. IEEE, 2024, pp. 566–571.

[29] M. Esposito, M. M. Sciarroni, T. Fava, A. Belli, L. Palma, E. Storti, and P. Pierleoni, "Experimental evaluation of end-to-end security protocols for the internet of everything," in *2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI)*. IEEE, 2024, pp. 13–18.

[30] M. Arazzi, S. Nicolazzo, and A. Nocera, "A fully privacy-preserving solution for anomaly detection in iot using federated learning and homomorphic encryption," *Information Systems Frontiers*, pp. 1–24, 2023.

[31] M. ”Arazzi, S. Nicolazzo, and A. Nocera, “A novel iot trust model leveraging fully distributed behavioral fingerprinting and secure delegation,” *Pervasive and Mobile Computing*, vol. 99, p. 101889, 2024.

[32] C. Diamantini, O. Pisacane, D. Potena, and E. Storti, “Personalized task reassignment in industry 5.0: A milp-based solution approach,” in *Proceedings of the 27th International Conference on Enterprise Information Systems - Volume 2: ICEIS*, INSTICC. SciTePress, 2025, pp. 813–820.