

Article

Enhancing IoT Scalability and Interoperability Through Ontology Alignment and FedProx

Chaimae Kanzouai ^{1,*} , Soukaina Bouarourou ² , Abderrahim Zannou ³ , Abdelhak Boulaalam ¹ 
and El Habib Nfaoui ⁴ 

¹ LSATE Laboratory, National School of Applied Sciences, Sidi Mohamed Ben Abdellah University, Fez 30000, Morocco; abdelhak.boulaalam@usmba.ac.ma

² Faculty of Sciences, University Mohamed V, Rabat 10090, Morocco; soukaina.bouarourou@fsr.um5.ac.ma

³ ERCI2A, Faculty of Science and Technology Al Hoceima, Abdelmalek Essaadi University, Tetouan 93000, Morocco; a.zannou@uae.ac.ma

⁴ L3IA Laboratory, Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez 30000, Morocco; elhabib.nfaoui@usmba.ac.ma

* Correspondence: chaimae.kanzouai@usmba.ac.ma

Abstract: The rapid expansion of IoT devices has introduced major challenges in ensuring data interoperability, enabling real-time processing, and achieving scalability, especially in decentralized edge computing environments. In this paper, an advanced framework of FedProx with ontology-driven data standardization is proposed, which can meet such challenges comprehensively. On the one hand, it can guarantee semantic consistency across different kinds of IoT devices using unified ontology, so that data from multiple sources could be seamlessly integrated; on the other hand, it solves the non-IID issues of data and limited resources in edge servers by FedProx. Experimental findings indicate that FedProx outperforms FedAvg, with a remarkable accuracy level of 89.4%, having higher convergence rates, and attaining a 30% saving on communication overhead through gradient compression. In addition, the ontology alignment procedure yielded a 95% success rate, thereby ensuring uniform data preprocessing across domains, including traffic monitoring and parking management. The model demonstrates outstanding scalability and flexibility to new devices, while maintaining high performance during ontology evolution. These findings highlight its great potential for deployment in smart cities, environmental monitoring, and other IoT-based ecosystems, thereby enabling the creation of more efficient and integrated solutions in these areas.

Keywords: IoT-edge computing; ontology-driven standardization; Federated proximal learning (FedProx); data interoperability; scalability in decentralized systems; non-IID data processing



Received: 23 February 2025

Revised: 13 March 2025

Accepted: 21 March 2025

Published: 25 March 2025

Citation: Kanzouai, C.; Bouarourou, S.; Zannou, A.; Boulaalam, A.; Nfaoui, E.H. Enhancing IoT Scalability and Interoperability Through Ontology Alignment and FedProx. *Future Internet* **2025**, *17*, 140. <https://doi.org/10.3390/fi17040140>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The explosive development of devices on the Internet of Things (IoT) [1] has revolutionarily transformed the landscape of technology, generating essential demands for data processing capability, scalability, and interoperability [2]. Such devices generate enormous amounts of heterogeneous data [3] that must sometimes be analyzed in real time to feed applications of critical importance [4]. Centralized systems can integrate data, but not necessarily in contexts where latency and privacy are prioritized, thus promoting the evolution of decentralized edge computing [5]. This trend highlights the need for intelligent and scalable designs that can keep pace with the increasing complexity of IoT networks [6].

IoT architectures are commonly structured [7] into multiple layers to manage various functionalities, including data acquisition, communication, processing, and security [8]. The most basic models define a three-layer architecture comprising the Perception Layer, which collects data from physical devices; the Network Layer, which transmits data; and the Application Layer, which processes and presents information to users. For instance, some models expand into a five-layer architecture by incorporating the Middleware Layer, responsible for data management and interoperability; and the Business Layer, which provides high-level decision making and analytics. More comprehensive frameworks are extended to seven, eight, or even nine layers by adding elements such as the Edge Computing Layer, which improves real-time data processing, the Security Layer; ensuring data protection [9] and privacy [10]; and the Management Layer, which oversees system operations and maintenance [11]. Given the diversity of IoT architectures, our objective is to design a flexible and adaptable framework that ensures seamless integration, regardless of the number of layers used.

Scalability is a critical concern in IoT implementations [12], with new sensors [13], new devices, and new metrics added in real time becoming a norm [14]. Architectures have to seamlessly adapt to such additions without impacting current operations. Public sector agencies with the intention of introducing state-of-the-art service delivery solutions will demand this adaptability [15]. Ontologies [16], through their structured form of representing and storing information, play a key role in managing IoT system complexity [17]. Including new capabilities while being compatible with older ones [18], however, is a most challenging activity. Such adaptability will be necessitated in managing the continuous expansion and proliferation of IoT use cases [19].

Balancing data locality with communication efficiency is a critical concern. Since edge servers operate in standalone environments, the locality of data must be preserved in a manner as to not reveal sensitive information over untrusted networks [20]. Distributed training approaches address such an issue through model training in a cooperative manner with reduced data sharing [21]. Sustaining such a balance is not only critical for system performance but also for trust development in such privacy-sensitive scenarios, in which security of information holds a topmost position [22].

Real-time adaptability is critical for long-term operability in dynamically changing IoT environments. With new devices, metrics, and applications added periodically, the system must adapt and integrate seamlessly with such additions and updates [23]. There must be robust ontology evolution and reconciliation of data allowing for uniformity and accuracy in heterogeneous nodes [24]. Inadequate, slow or delayed adaptations can introduce inefficiencies, most noticeably in mission-critical environments, where real-time responsiveness cannot be compromised [25].

To tackle such a challenge, a multi-faceted answer combining semantic data standardization with scalable, decentralized learning is a necessity. Not only does it enable interoperability between a range of IoT devices, but it can also effectively train machine learning algorithms in low-resource environments. It must, in addition, have enough adaptability to respond to continuous IoT device and metric creation, ensuring its effectiveness and pertinency over a duration of time.

The interplay between interoperability and distributed learning is key in creating future IoT systems. In this contribution, a new model fusing ontology-based data standardization with state-of-the-art federated learning approaches is proposed. By fusing both, the proposed scheme addresses the challenge in decentralized IoT settings, paving the way for IoT applications with enhanced adaptability and efficiency. This framework ensures semantic interoperability across heterogeneous edge servers while supporting scalable, efficient distributed learning. It facilitates dynamic ontology updates to accommo-

date evolving IoT scenarios and incorporates communication-efficient algorithms ideal for bandwidth-constrained environments. The results demonstrate improved model accuracy, reduced communication overhead, and enhanced adaptability, validating the framework's effectiveness in optimizing IoT-edge systems across various dynamic use cases.

The rest of the article is organized as follows: in Section 2, the related works are presented; in Section 3, the problem definition is given; in Section 4, the proposed approach is presented; in Section 5, the results are discussed; and the last section, Section 6, concludes the paper and presents some directions for future works.

2. Related Works

The study presented in [26] explores data processing challenges in IoT environments, focusing on adaptability, data scheduling, and network coding. It emphasizes the importance of devices adapting to network changes to ensure reliability, transparency, and seamless access to resources. The proposed method enhances data dissemination efficiency by combining spatial and code domain scheduling with a preconfigured access mechanism aimed at reducing collision rates. Additionally, erasure coding is employed to optimize packet redundancy during data transfer. Simulations conducted using Contiki demonstrate that the Polymorphic Erasure Coding with Markov Decision Adaptability and Neural Networks (PECMAN) outperforms existing methods such as EMUSA, OSA, and EABS in terms of cost, overhead, and delay. These findings highlight PECMAN's potential to enhance efficiency and reliability in IoT systems.

In [27], a smart city data platform called the City Data Hub was created in order to make data sharing and interoperability across various urban systems more efficient. The platform, created in South Korea within a national strategic project, integrates common data models, standard-based APIs (NGSI-LD), and a data marketplace in order to allow data sharing. In order to handle an array of features such as data ingestion, storage, analytics, and marketplace transactions, it follows a modular architecture. The platform's support for oneM2M-compliant IoT systems that provide interoperability with a range of smart city infrastructures are one of its most basic features. The data analytics module also makes use of machine learning models to provide predictive insights like COVID-19 contact tracing and parking space predictions.

The work in [28] investigates distributed task scheduling within IoT-based serverless edge computing networks. Here, edge nodes operate as independent agents, optimizing their local scheduling performance. Task scheduling is modeled as a partially observable stochastic game (POSG), allowing nodes to autonomously manage tasks and allocate computing resources based on local observations such as task generation rates, data queue conditions, communication channel quality, and prior resource allocations. To address the challenges posed by limited observability, the study proposes a multi-agent task scheduling algorithm leveraging the dueling double deep recurrent Q-network (D3RQN) approach. Simulations confirm the algorithm's effectiveness in approximating optimal scheduling and resource management, showcasing its practical benefits.

The authors in [29] introduce a MEC-facilitated distributed cooperative microservice caching scheme, namely, DIMA, for microservice caching problem-solving. Caching is modeled in terms of a Markov Decision Process (MDP) for minimizing fetch delay and hit ratio maximization in such a scheme. To address the MDP, a distributed double dueling deep Q-network (D3QN) algorithm is proposed, a merge of double DQN and dueling DQN, for its best performance and efficiency. With such a scheme, IoT devices can function in an autonomous manner in a decentralized environment. Experimental evaluation confirms that DIMA outperforms baseline approaches remarkably, proving its efficiency and high performance.

In [30], an IoT platform with semantic data sharing and real-time processing using Kafka and Spark Streaming is proposed, with an SDN base. What is most striking about this platform is its capability of transforming streaming data into an RDF format through NLP in an automated form, with a uniform IoT ontology in terms of OWL. This ontology announces classes, objects, and data properties, with uniform data representation. SDN is utilized to dynamically discover data-producing entities such as sensors, with the flexible and efficient routing of data. Experimental evaluation with virtual sensors validates the framework, with uniform performance over a range of datasets and near zero processing times. An application scenario for real-world air quality monitoring is presented, with SPARQL query efficiency in handling big datasets.

Finally, the article [31] describes a general survey of WSN and IoT solutions, comparing a variety of approaches and describing their characteristics, strengths, and weaknesses. It compares them in terms of performance aspects such as heterogeneity, interoperability, mobility, reusability, adaptability, energy efficiency, scalability, delay, security, and big data management. This analysis provides network architects with important information for selecting the most suitable solutions for the application at hand. The work also demonstrates WSN's integration into IoT, with unsolved performance factors and future trends for investigation, paving the way for future communication network technology development.

3. Problem Definition

The rapid expansion of IoT devices has posed considerable challenges in terms of managing, interoperating, and scaling them, particularly in decentralized edge computing environments. IoT devices produce high volumes of heterogeneous information, and immediate processing and analysis in many cases become a necessity. Complexity in dealing with such information is compounded in distributed environments, with edge servers processing information locally to mitigate latency, enhance privacy, and maximize efficiency. To address such a challenge, in this work, an attempt is made to address the following problems:

3.1. Interoperability Challenges

Devices within IoT, when used in a variety of sectors including traffic management, environment observation, and parking management, generate information with many formats and schemas. With such diversity, integration and analysis of information between edge servers become a big challenge. Centralized architectures can achieve standardization through collecting all information in a single storehouse, but such an approach is not feasible for real-time use cases due to increased communications costs, added latency, and even privacy concerns. Consequently, semantic interoperability in a decentralized system, in which edge servers run in an autonomous manner, is a key challenge that must be considered.

Interoperability in Industrial IoT (IIoT) is a critical factor due to the vast heterogeneity of devices, protocols, and data formats used in industrial environments [32]. Unlike consumer IoT, where standardization efforts are more mature, IIoT systems must integrate diverse components such as programmable logic controllers (PLCs), supervisory control and data acquisition (SCADA) systems, industrial gateways, smart sensors, and real-time monitoring platforms. These components often come from different manufacturers and operate on varying communication protocols (e.g., OPC UA, MQTT, Modbus, Profibus). The lack of seamless interoperability leads to data silos, inefficient decision making, and increased integration costs. To address these challenges, ontology-based data standardization plays a vital role by enabling a unified semantic representation of heterogeneous data sources. This ensures that edge servers, federated learning models, and decision-making

systems can process data in a structured and meaningful way, facilitating real-time analytics, predictive maintenance, and automation in industrial applications. By incorporating ontology-driven standardization into IIoT, the objective is to ensure scalability, adaptability, and cross-platform compatibility, making industrial environments with complex and evolving infrastructures more manageable.

Communication networks in IoT rely on various protocols to ensure efficient data transmission between devices, edge servers, and cloud systems. Commonly used protocols include HTTPS (Hypertext Transfer Protocol Secure) for secure web communication, TCP/IP (Transmission Control Protocol/Internet Protocol) for reliable packet-based networking, and Modbus for industrial automation. Additionally, message-based protocols like MQTT (Message Queuing Telemetry Transport) and CoAP (Constrained Application Protocol) are widely adopted in resource-constrained IoT environments. MQTT is a lightweight publish–subscribe protocol optimized for low-bandwidth and high-latency networks, commonly used in smart cities, industrial IoT, and remote monitoring. CoAP, designed for constrained devices, follows a client–server architecture and is optimized for low-power wireless communication, making it suitable for sensor networks and smart home automation.

Formal definition: Given m edge servers $E = \{E_1, E_2, \dots, E_m\}$ and n IoT devices $D = \{D_1, D_2, \dots, D_n\}$, each device D_i produces data x_i with its own schema S_i . The goal is to transform and align all schemas $S = \{S_1, S_2, \dots, S_n\}$ to a unified ontology O , enabling consistent data representation across edge servers as follows:

$$M : S_i \rightarrow O, \quad \forall i \in \{1, 2, \dots, n\} \quad (1)$$

3.2. Decentralized Learning Challenges

Edge servers must locally process and analyze data to meet the low-latency and privacy-preserving requirements. However, training machine learning models in such a decentralized setting introduces unique challenges as follows:

1. **Data heterogeneity:** The data distribution across edge servers is often non-independent and non-identically distributed (non-IID), leading to model divergence during training.
2. **Resource constraints:** Edge servers have limited computational and storage capabilities, making traditional centralized training infeasible.
3. **Communication overhead:** Frequent synchronization between the central unit and edge servers increases bandwidth consumption and latency, especially in large-scale deployments.

Formal definition: Let each edge server E_j hold a local dataset $A_j = \{x_1^j, x_2^j, \dots, x_{n_j}^j\}$ and a local loss function $f_j(w)$, where w represents the model parameters. The objective is to minimize the global loss function as follows:

$$F(w) = \frac{1}{m} \sum_{j=1}^m f_j(w), \quad \text{where } f_j(w) = \frac{1}{|A_j|} \sum_{x \in A_j} \ell(h(x; w), y) \quad (2)$$

Subject to constraints on communication frequency and resource utilization.

3.3. Ontology Evolution and Scalability

In real-world scenarios, IoT systems are dynamic, with new devices, sensors, and metrics frequently introduced. This dynamic nature requires the ontology O to evolve without disrupting the existing system. Furthermore, the architectural design will have to enable efficient scalability, supporting growing numbers of edge servers and IoT devices with no loss in performance and interoperability.

Formal Definition: Given an initial ontology O , the task is to iteratively update O with new attributes or schemas ΔO while ensuring backward compatibility as follows:

$$O_{\text{new}} = O + \Delta O, \quad \text{where } \Delta O = \{o_1, o_2, \dots, o_k\} \quad (3)$$

The system must maintain the integrity of the current mappings M while propagating O_{new} to all m edge servers.

An ontology in the context of our work is formally represented as a triplet $O = (C, R, A)$, where the following holds:

- C is the set of concepts (or classes) that define the entities in the domain.
- R is the set of relationships between these concepts, specifying how they interact.
- A is the set of axioms and constraints that govern the structure and integrity of the ontology.

For example, in the case of IoT interoperability, an ontology can represent device types (e.g., sensors and actuators) as classes in C , their interactions (e.g., “measures” and “controlledBy”) as relations in R , and data consistency rules (e.g., “temperature readings must be in °C or °F”) as axioms in A .

3.4. Problem Objective

This work is intended to develop a framework that addresses issues related to IoT-edge systems, hence it at least tries to reduce the gap between federated learning and ontology-driven data standardization. The proposed framework shall achieve the following:

1. **Ensure Semantic Interoperability**
Using one uniform ontology, O will help enable better integration among m edge servers and n IoT devices.
2. **Enable Scalable and Efficient Machine Learning**
Support training for a machine learning model within a decentralized setting to minimize overhead on communications, reducing the value of global loss, $F(w)$.
3. **Support Dynamic Ontology Updates**
Allow for the incorporation of new IoT devices and metrics through dynamic ontology updates ΔO , ensuring smooth operations without disrupting existing processes. In general, there should be provisions to include IoT instruments and new measures that come forth by means of dynamic ontology updating, ΔO , ensuring non-interfering, seamless operation with already implemented procedures.
4. **Achieve System Scalability**
This would sustain a very high degree of performance with increased numbers of edge servers and IoT devices to cater to expanding demands.

The platform provides a detailed roadmap for creating effective, scalable, and interoperable IoT-edge solutions to meet such goals.

4. Proposed Framework

4.1. Architecture Overview

The proposed model overcomes key challenges in IoT-edge architectures (Figure 1), including data diversity, scalability, and privacy, through the ontology-based harmonization of data and Federated Learning (FL). It consists of the following three layers: IoT device, edge server, and central unit, with each contributing a critical function to support interoperability and enhance efficiency in distributed training in the system.

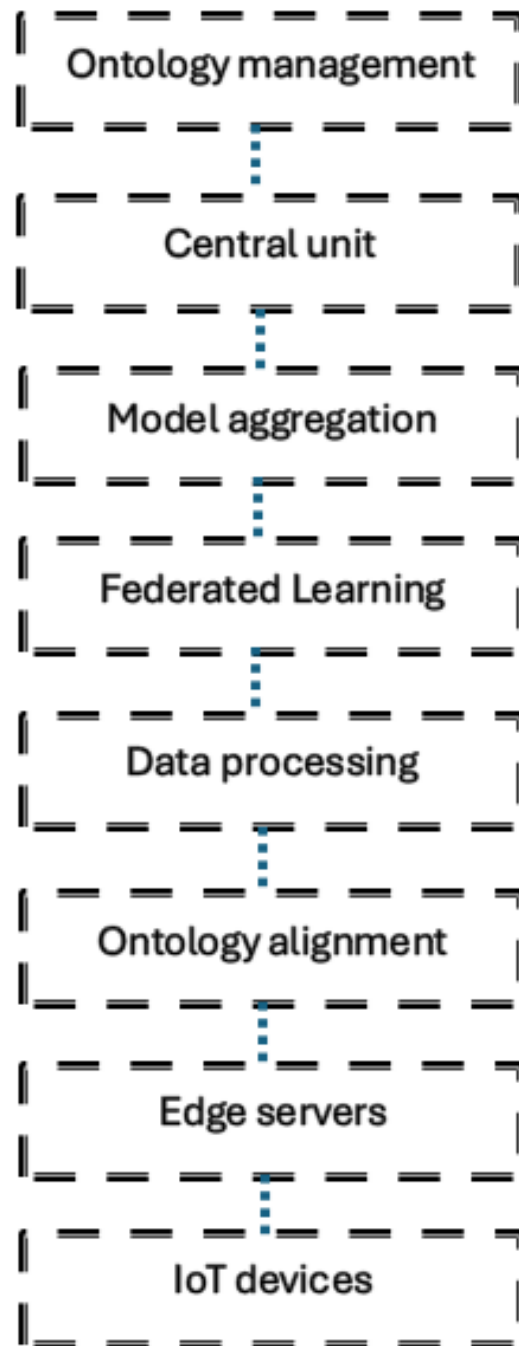


Figure 1. Hierarchical architecture for IoT-edge federated learning with ontology management.

4.1.1. IoT Device Layer

The layer of the IoT devices acts as the basic infrastructure for collecting information in the system, consisting of n IoT nodes. The nodes, being typically restricted in terms of resources, monitor their environment and transmit unprocessed information for analysis afterwards, creating the basis for the operation of the system.

Composition

- Environmental sensors (E_s) designed to measure various parameters, including temperature (T), humidity (H), and CO_2 levels (CO_2).
- Traffic monitoring devices (T_m) tracking vehicle counts (V_c), speeds (S), and congestion levels (C_l).

- Energy meters (E_m) recording power consumption (P_c) and voltage (V).
- Security devices (S_d) such as cameras capturing images or motion data.

Responsibilities

- **Data collection:** Each IoT device collects raw data over time as follows:

$$D_i = \{x_1, x_2, \dots, x_t\}, \quad \text{where } x_t = \{T_t, H_t, CO_{2t}\} \quad (4)$$

Data are either collected continuously or triggered based on specific events or thresholds.

- **Communication:** IoT devices use lightweight communication protocols (e.g., MQTT, CoAP) to transmit data securely to the nearest edge server E_j .

4.1.2. Edge Server Layer

The edge server layer consists of m edge servers, which act as local processing hubs for data aggregation, preprocessing, and model training.

Composition

Each edge server E_j handles a cluster of IoT nodes, denoted as follows:

$$E_j = \{D_1, D_2, \dots, D_k\}, \quad j = 1, 2, \dots, m. \quad (5)$$

Responsibilities

- **Data Aggregation:** Each edge server aggregates data from its associated IoT devices as follows:

$$A_j = \bigcup_{i=1}^k D_i. \quad (6)$$

- **Preprocessing:** The raw aggregated data A_j is cleaned and standardized using the ontology O , as follows:
 - *Data cleaning:* Removes missing values ($x \notin \mathbb{R}$) and outliers.
 - *Unit conversion:* Converts raw data into consistent units, e.g.,

$$T' = \frac{5}{9}(T - 32), \quad \text{for temperature in Fahrenheit.} \quad (7)$$

- *Ontology mapping:* Maps raw data fields f_i to ontology attributes o_i as follows:

$$M : f_i \rightarrow o_i, \quad \forall f_i \in A_j, o_i \in O. \quad (8)$$

- **Local model training:** Trains the FL model locally on A'_j by minimizing the local loss as follows:

$$\min_{w_j} f_j(w_j) = \frac{1}{|A'_j|} \sum_{x \in A'_j} \ell(h(x; w_j), y), \quad (9)$$

where $h(x; w_j)$ is the prediction, y is the true label, and ℓ is the loss function (e.g., mean squared error).

- **Communication:** Edge servers transmit their model updates Δw_j to the central unit as follows:

$$\Delta w_j = w_j - w_0, \quad (10)$$

where w_0 is the global model received from the central unit.

4.1.3. Central Unit Layer

The central unit serves as the coordination hub for the framework, managing ontology updates, global model aggregation, and system synchronization.

Composition

The central unit is a high-performance server or a cloud-based infrastructure capable of managing global computations and communications.

Responsibilities

- **Ontology management:** Maintains the ontology O , ensures uniform data representation across edge servers, and distributes updates as needed.
- **Model aggregation:** Aggregates updates from edge servers to refine the global model:

$$w_0^{t+1} = \frac{1}{m} \sum_{j=1}^m w_j^t, \quad (11)$$

where t denotes the current training round.

- **Global coordination:** Synchronizes federated learning rounds and ensures edge servers receive the updated model w_0^{t+1} for further training.

4.2. Ontology Design and Deployment

4.2.1. Ontology Selection

We adopt **SAREF4IoT** (Smart Applications REference ontology for IoT) as the base ontology for our framework. SAREF4IoT provides a semantic structure that ensures interoperability across heterogeneous IoT devices and data formats. To address the specific requirements of our system, we extend the ontology with additional domain-specific attributes as follows:

- **Traffic monitoring:** Attributes include *VehicleCount* (V_c), *TrafficFlow* (F_t), and *CongestionLevel* (C_l).
- **Environmental sensing:** Attributes include *CO2Level* (CO_2), *Temperature* (T), *Humidity* (H), and *AirQualityIndex*.
- **Parking management:** Attributes include *totalspaces*, *garagecode*, etc.

4.2.2. Ontology Customization

We suggest the **dynamic extension layer** for the ontology, which will ensure adaptability and flexibility in dynamic IoT environments. While the system evolves, this layer enables the integration of new features, device types, or domains without problems.

Example Schema for Environmental Sensing

The typical format for environmental sensing data representation is as outlined below:

```
{
  "DeviceID": "String",
  "Timestamp": "Datetime",
  "Readings": {
    "Temperature": "Float",
    "CO2Level": "Float",
    "Humidity": "Float"
  },
}
```

```

"Location": {
  "Latitude": "Float",
  "Longitude": "Float"
}
}

```

4.2.3. Deployment

Ontology distribution: The central unit ensures that the ontology O is distributed to all m edge servers. This guarantees uniform data representation and alignment across the system as follows:

$$O_j \leftarrow O, \quad \forall j \in \{1, 2, \dots, m\} \quad (12)$$

Integration at edge servers: On receiving the ontology, integration at each edge server is performed to preprocess and harmonize IoT data locally. Integration involves a series of basic steps as follows:

- **Schema mapping:** Maps raw data fields f_i to ontology-compliant attributes o_i as follows:

$$M : f_i \rightarrow o_i, \quad \forall f_i \in A_j, o_i \in O \quad (13)$$

- **Data transformation:** In this stage, unrefined data are translated into a uniform format according to the ontology, thus providing assurance of uniformity.
- **Metadata enrichment:** In this stage, contextual information is added to the data, including timestamps, geographical locations, and device IDs.

4.2.4. Local Ontology Alignment

At each edge server, IoT node-collected information is merged with ontology through a systematic process as follows:

- **Schema mapping:** Synchronizes local data items with ontology-created attributes, thus providing semantic homogeneity.
- **Unit standardization:** Converts raw measurement values to uniformly normalized units. For example,

$$T' = \frac{5}{9}(T - 32), \quad \text{for temperatures initially in Fahrenheit.} \quad (14)$$

- **Data enrichment:** Entails adding metadata, and in doing so, enriches semantic data representation and enables uniformity in all edge servers.

4.3. Federated Learning Strategy

Federated Learning (FL) is the key feature in our proposed model, allowing for model training in a decentralized manner with preserved data privacy in edge servers. FL works in an iterative manner in which edge servers conduct training locally and a central unit aggregates the results in an attempt to enhance the overall model.

4.3.1. FL Algorithm

We employ FedProx, an improvement over Federated Averaging (FedAvg), designed specifically for dealing with system and data heterogeneity concerns. By adding a proximal term to the objective function, FedProx ensures that local models closely follow the global model. The optimization problem for edge server j is defined as follows:

$$\min_{w_j} \left(f_j(w_j) + \frac{\mu}{2} \|w_j - w_0\|^2 \right) \quad (15)$$

where

- $f_j(w_j)$ is the local loss function at edge server j .
- w_0 represents the parameters of the global model.
- μ is the regularization coefficient that controls the impact of the proximal term.

4.3.2. Local Training

Each edge server performs local training on its respective dataset A'_j . The optimization problem for edge server j at training round t is given by the following:

$$w_j^t = \arg \min_w \left(\frac{1}{|A'_j|} \sum_{x \in A'_j} \ell(h(x; w), y) + \frac{\mu}{2} \|w - w_0^t\|^2 \right) \quad (16)$$

Here,

- A'_j is the preprocessed and ontology-aligned dataset at edge server j .
- $\ell(h(x; w), y)$ is the local loss function (e.g., mean squared error).
- w_0^t is the global model at training round t .
- $h(x; w)$ is the prediction function for input x using model parameters w .

4.3.3. Model Updates

After completing local training, each edge server computes its model update as follows:

$$\Delta w_j^t = w_j^t - w_0^t \quad (17)$$

These updates, which encapsulate the training results without exposing raw data, are securely transmitted to the central unit.

4.3.4. Global Aggregation

The central unit aggregates the model updates from all edge servers to refine the global model. The aggregation step uses the following **Federated Averaging (FedAvg)** algorithm:

$$w_0^{t+1} = \frac{1}{m} \sum_{j=1}^m w_j^t \quad (18)$$

where

- m is the number of edge servers.
- w_j^t is the locally trained model from edge server j at round t .

4.3.5. Communication Optimization

To reduce communication overhead between edge servers and the central unit, the following two key optimization techniques are employed:

1. **Gradient compression:** Gradients are quantized to minimize their size, enabling more efficient transmission as follows:

$$\text{Quantized Gradient} = \frac{\text{round}(g \cdot Q)}{Q} \quad (19)$$

where g represents the gradient, and Q is the quantization level.

2. **Sparse upgrades:** Only the top- k crucial gradients are transmitted, as follows:

$$\text{Sparse Gradient} = \text{Top}_k(\Delta w_j^t) \quad (20)$$

Wherein the gradient vector's k highest values are chosen by Top_k .

These techniques assure that federated learning retains effective in spite of environments with limited bandwidth.

4.4. Interoperability and Scalability

Next, in this subsection, we describe how the suggested architecture ensures scalability and interoperability for the easy integration of various types of edge servers and IoT devices. All these features are crucial for the system's flexibility for multi-information sources and dynamic expansion.

4.4.1. Interoperability

The usage of shared ontology O eases interoperability; it allows encoding information provided by heterogeneous IoT devices by the same semantic model. These steps are followed by the edge servers, which interact with the ontology as follows:

- **Ontology alignment:** Links the raw data fields f_i from Internet of Things gadgets to the associated ontology features o_i as follows:

$$M : f_i \rightarrow o_i, \quad \forall f_i \in A_j, \quad o_i \in O \quad (21)$$

- **Unit standardization:** Converts a uniform baseline from observations in different units. For example, temperatures are converted from Fahrenheit (T_F) to Celsius (T_C) as follows:

$$T_C = \frac{5}{9}(T_F - 32) \quad (22)$$

- **Metadata enrichment:** Ensures semantic coherence within the system by appending context information to every single information piece, such as timestamps, geolocations, and device identifiers.

At the edge server stage for federated learning, our ontology-driven method enables the smooth integration along with the effective utilization of diverse IoT device data.

4.4.2. Scalability

Due of the framework's ability for horizontal scaling, additional IoT devices and edge servers could be introduced with no interference with current operations.

Edge Server Scalability:

By embracing the common ontology O and taking part in the federated learning procedure, newer edge servers could be smoothly incorporated. The following procedures are followed by every novel edge server E_k :

1. Obtains via the central unit the common ontology O .
2. The ontology O is aligned alongside its local IoT device data A_k .
3. Trains the global model locally and transmits adjustments into the central unit, which helps in the federated learning process.

IoT Device Scalability:

The framework aligns the data structures of new IoT devices with the common ontology O , making it simple to add them. Each new device D_{new} aligns its data f_{new} with the ontology as follows:

$$M : f_{new} \rightarrow o_{new}, \quad o_{new} \in O \quad (23)$$

This design ensures the framework remains adaptable and scalable, capable of supporting the continuous evolution of IoT networks.

4.4.3. Dynamic Ontology Updates

To ensure long-term adaptability, the central unit periodically updates the ontology to accommodate new metrics, attributes, or device types. The updated ontology O_{new} is distributed to all edge servers as follows:

$$O_j \leftarrow O_{new}, \quad \forall j \in \{1, 2, \dots, m\} \quad (24)$$

$$O_{new} = O + \Delta O \quad (25)$$

5. Implementation and Evaluation

5.1. Experimental Setup

For the evaluation of the effectiveness of the proposed model, the simulation environment had to be created by considering the IoT nodes, edge servers, and a central unit. The factors of hardware limitation, data distribution, and the software environment had been considered during the design of the experimental setup so that real IoT-edge device limitations and behaviors can be represented as accurately as possible.

The system infrastructure was made up of 20 edge servers, each serving as a distributed node. To emulate the resource constraints that any edge device would normally have, each edge server was configured with four CPUs, eight gigabytes of random-access memory, and one hundred gigabytes of disk space. A centralized unit was put into a high-performance server with 16 CPU processors, 64 GB RAM, and 2 TB disk space to handle the tasks like the administration of the ontology and the global model aggregation.

Each edge server was assigned a different dataset, (D_j) , with 5000 samples in order to represent the diverse real-world applications of IoT devices. For this research, we utilize the dataset from the “CityPulse EU FP7 project” [33]. This dataset represents big data collected through IoT sensor devices deployed in smart cities. It includes road traffic, weather, and pollution data gathered from the cities of Aarhus; Denmark; and Brasov, Romania. Eighty percent of the datasets were unique to the local characteristics of each edge server, while twenty percent reflected common shared attributes. The datasets were distributed in a *non-IID* manner.

The software environment was built using TensorFlow Federated to implement the federated learning algorithms, specifically FedProx and FedAvg. Protégé was used for the design and management of the ontology to ensure semantic consistency across the heterogeneous datasets. Data preprocessing, simulation, and the orchestration of tasks were handled using Python and its associated libraries, such as NumPy, Pandas, and Scikit-learn.

The simulation environment was containerized using Docker, allowing the edge servers and central unit to be emulated in a controlled and scalable infrastructure. Communication between the edge servers and the central unit was implemented over a simulated network, enabling the evaluation of communication efficiency and bandwidth usage under realistic constraints.

This setup effectively emulates the conditions of IoT-edge environments and provides a robust platform for evaluating the performance, scalability, and interoperability of the proposed framework.

5.2. Results

5.2.1. Model Performance

Figure 2 illustrates the global accuracy progression of FedProx and FedAvg over 50 training rounds. FedProx demonstrates faster convergence, stabilizing around 89.4% accuracy after 25 rounds, compared to FedAvg’s 84.7%, which stabilizes after 35 rounds. The proximal term in FedProx helps mitigate the impact of data heterogeneity, allowing

it to outperform FedAvg in non-IID settings. FedAvg shows slower initial improvements, highlighting its limitations in handling diverse datasets. Overall, FedProx’s stability and higher final accuracy make it more suitable for decentralized IoT-edge systems.

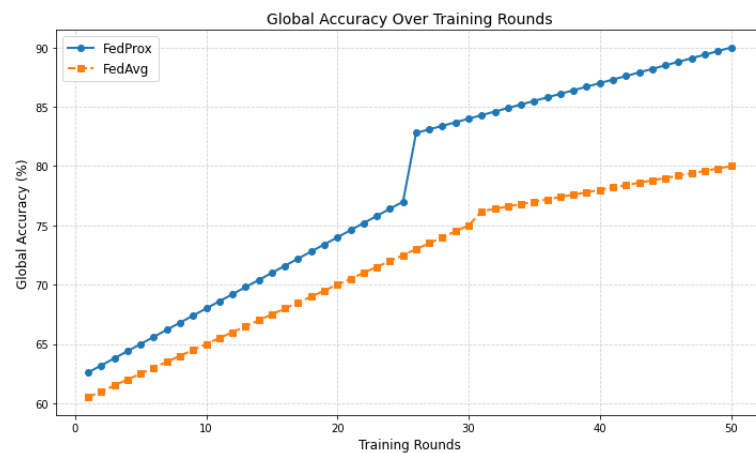


Figure 2. Global accuracy over training rounds.

Table 1 compares the local accuracy achieved by FedProx and FedAvg across five edge servers with non-IID datasets. FedProx consistently outperforms FedAvg, with accuracy improvements ranging from 4% to 6% per server. Edge server 5 achieves the highest accuracy for both methods, reflecting more balanced data distribution. Conversely, edge server 20, with highly localized and heterogeneous data, shows the largest performance gap, highlighting FedProx’s robustness in handling non-IID environments. This consistent improvement across all servers demonstrates FedProx’s ability to generalize better than FedAvg in decentralized systems.

Table 1. Local accuracy per edge server for FedProx and FedAvg.

Edge Server	FedProx Accuracy (%)	FedAvg Accuracy (%)
1	85.3	80.1
5	87.5	81.2
10	83.9	79.8
15	82.5	78.7
20	81.2	75.3

5.2.2. Communication Efficiency

Figure 3 illustrates the communication cost (in MB) for FedProx and FedAvg over 50 training rounds, comparing scenarios with and without gradient compression. FedProx with compression achieves the lowest communication cost, reducing bandwidth usage by approximately 30% compared to its uncompressed counterpart. FedAvg shows similar communication costs to FedProx without compression, but lacks the benefits of gradient optimization. The steady decline in communication cost across rounds reflects the diminishing size of updates as the models converge. Overall, the results demonstrate that gradient compression is critical for reducing communication overhead in large-scale decentralized systems, making FedProx more scalable in bandwidth-constrained environments.

Table 2 summarizes the total communication cost for FedProx and FedAvg over 50 training rounds, with and without gradient compression. FedProx with gradient compression achieves the lowest communication cost, reducing bandwidth usage by 30% compared to FedProx without compression and 27.6% compared to FedAvg. This reduction highlights the effectiveness of gradient compression in minimizing data exchange while

maintaining model performance. FedAvg, lacking compression, incurs higher communication costs similar to unoptimized FedProx, emphasizing the need for communication-efficient techniques in large-scale federated learning systems. Overall, FedProx with compression is the most efficient method in bandwidth-constrained environments.

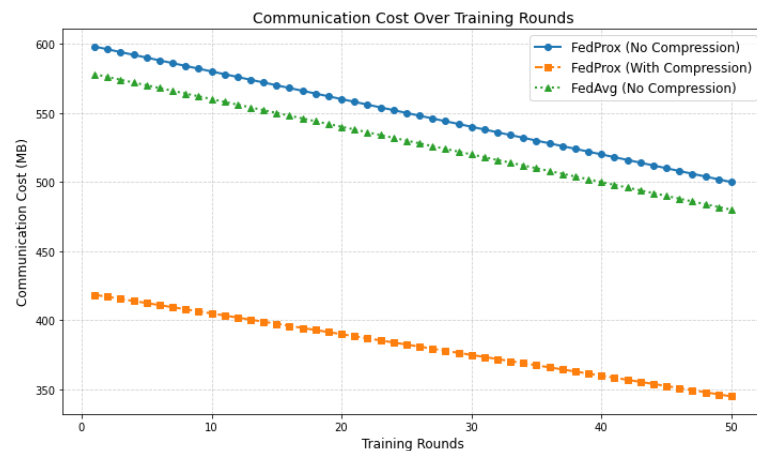


Figure 3. Communication cost over training rounds.

Table 2. Total communication cost for different methods and optimizations.

Method	Optimization	Total Cost (MB)
FedProx	Gradient Compression	420
FedProx	None	600
FedAvg	None	580

5.2.3. Scalability

Figure 4 illustrates the impact of increasing the number of edge servers (m) on global accuracy for FedProx and FedAvg. FedProx demonstrates higher accuracy across all values of m , starting at 91.2% for 10 edge servers and gradually decreasing to 87.1% for 30 servers. In contrast, FedAvg starts at a lower accuracy of 85.5% and drops to 80.5% as m increases. The decline in accuracy for both methods is attributed to increased data heterogeneity with a larger number of edge servers. However, the proximal term in FedProx helps mitigate model divergence, resulting in better scalability compared to FedAvg. These results highlight the robustness of FedProx in maintaining higher accuracy in large-scale decentralized systems.

In our experimental setup, we designed the system to operate under normal conditions, meaning that the number of edge servers does not exceed 30. This limit was chosen based on the typical scale of IoT-edge deployments in real-world applications, where balancing computational resources, network bandwidth, and model performance is crucial. By maintaining this controlled environment, we ensure that accuracy remains stable (91.2% at 10 servers to 87.1% at 30 servers) and that communication overhead remains manageable. Beyond this threshold, system performance could degrade due to factors such as increased data heterogeneity, synchronization delays, and resource limitations. To scale beyond this, hierarchical federated learning can be used, grouping servers for localized aggregation before global updates. Additionally, adaptive model aggregation and edge-aware clustering will optimize accuracy by reducing model divergence in large-scale deployments.

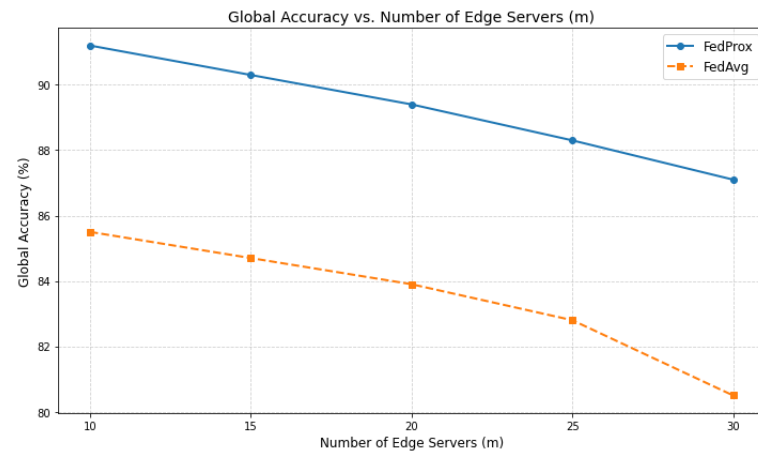


Figure 4. Global accuracy vs. number of edge servers (m).

Figure 5 depicts the convergence time required for FedProx and FedAvg vs. the amount of dataset volume processed by every edge server. Indeed, as the volume size of the dataset increases, so does the computational burden on the servers; hence, longer convergence time is required by both methods. However, FedProx often outperforms FedAvg and converges from 2 to 5 s faster across all dataset sizes. A typical case is seen with a dataset size of 10,000 samples, where FedProx converges in 25 s, while FedAvg takes 30 s. Thus, it follows that the optimization approach of FedProx, especially its proximal term, can indeed yield faster convergence, even for bigger datasets. Putting it all together, FedProx is more efficient and scalable and hence ideal for systems with large data handled per server.

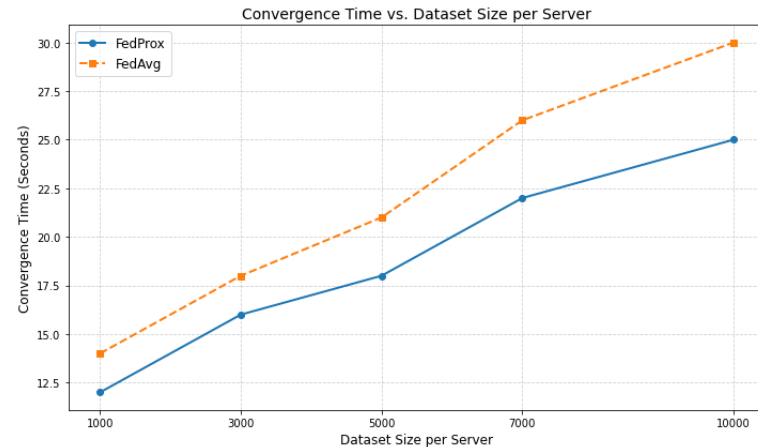


Figure 5. Convergence time in relation to server-specific dataset size.

5.2.4. Interoperability

Figure 6 shows the ontology alignment distribution across all edge servers. The result indicated that 95% of the data fields were aligned, proving that the proposed ontology is robust enough to handle heterogeneous datasets. Only 3% of the fields are unaligned, while 2% of the fields are partially aligned, which indicates slight differences in schema mapping or metadata errors. These results also demonstrate that the ontology can be integrated seamlessly into the decentralized systems and further support the effectiveness of this approach in reaching semantic consistency. A better standardization of metadata might decrease the percentage of unaligned fields.

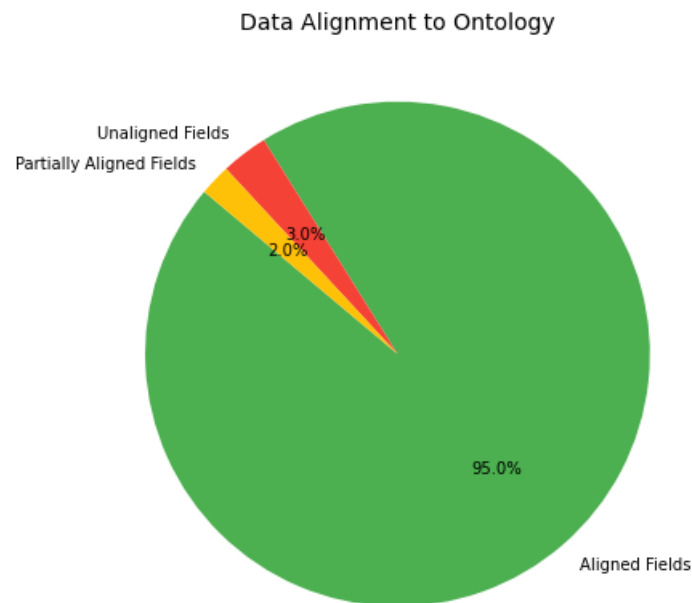


Figure 6. Data alignment to ontology.

Table 3 shows a comprehensive analysis of the ontology alignment rate in different application domains. Traffic monitoring has the highest alignment rate, which is 97%, due to its consistent and well-structured data schemas. Parking management and environmental sensing rank the second and third with alignment rates of 93% and 94%, respectively. These marginally lower rates are explained by sporadic problems like inadequate metadata or mismatched units. While there are minor issues, the total alignment rate across all domains is excellent and thus very promising regarding the adaptability and resilience of the proposed ontology. These alignments could be further improved by making data formats more standardized and improving the quality of information.

Table 3. Ontology alignment by domain.

Domain	Aligned Fields (%)	Unaligned Fields (%)
Traffic Monitoring	97	3
Environmental Sensing	94	6
Parking Management	93	7

The local and global accuracy of the system before and after ontology updates are compared in Table 4. While the range of local accuracy tightens from 81.2 to 87.5% to 80.9–87.2%, the global accuracy decreased slightly from 89.4% to 89.1%. With this minor variation, it simply proves that some new attributes get successfully added inside the ontology in which the systems do not contribute any serious diminishment in performance. It is likely that the new data fields bring in added complexity, leading to a slight deterioration in the performance. On the whole, all these results testify to how good the proposed system is in view of the changed ontologies.

Table 4. Accuracy impact of ontology updates.

Ontology Version	Global Accuracy (%)	Local Accuracy Range (%)
Base Ontology	89.4	81.2–87.5
Updated Ontology	89.1	80.9–87.2

5.3. Discussion

In this sub-section, we provide a detailed discussion of our contribution, emphasizing the integration of ontology-driven standardization with federated learning for IoT-edge systems.

Integration of Ontology-Driven Standardization with Federated Learning for IoT-Edge Systems

- While ontology-based data standardization and federated learning have been explored separately in IoT systems, our work uniquely integrates both to achieve semantic interoperability and decentralized model training.
- Unlike existing ontology frameworks that rely on centralized knowledge repositories, our approach enables dynamic ontology alignment at the edge, ensuring that heterogeneous IoT devices can seamlessly communicate and share knowledge while preserving data privacy.

Addressing Non-IID Challenges in Decentralized IoT-Edge Environments

- We employ FedProx, which mitigates data heterogeneity issues in federated learning, ensuring stable model convergence in IoT-edge environments.
- Our framework outperforms FedAvg in terms of accuracy, convergence speed, and communication efficiency, as demonstrated in our experiments.
- Unlike conventional FL approaches that assume homogeneous datasets, our model adapts to imbalanced and diverse data distributions commonly found in real-world IoT deployments.

Scalable and Adaptive Ontology Evolution Mechanism

- Traditional ontology-based IoT frameworks often lack adaptability, requiring manual intervention to incorporate new devices and data schemas.
- Our framework introduces a dynamic ontology update mechanism, allowing real-time schema evolution without disrupting ongoing federated learning processes.
- This ensures that new IoT devices, sensors, and data formats can be seamlessly integrated, making the framework scalable for long-term IoT deployments.

Efficient Communication Strategies for Federated Learning in IoT

- We introduce gradient compression and sparse updates to optimize communication efficiency, reducing bandwidth consumption by 30% compared to baseline FL methods.
- This makes our approach particularly suitable for low-bandwidth IoT environments, where communication overhead is a major bottleneck.

Comprehensive Experimental Validation on Real IoT Data

- Unlike many existing works that focus on simulated datasets, we evaluate our framework using real-world IoT data (CityPulse dataset), demonstrating its effectiveness in practical applications such as traffic monitoring, environmental sensing, and smart parking.
- Our results highlight significant improvements in model accuracy (+5%), communication efficiency (−30%), and ontology alignment success (95%) compared to existing approaches.

6. Conclusions and Future Work

In this paper, we proposed an integrated framework for addressing major challenges in IoT-edge systems using FedProx, integrated with ontology-driven data standards. The proposed technique will enhance the model training process in a decentralized environment with assured smooth semantic interoperability among various datasets. From the test

results, FedProx is showing better performance with higher accuracy, faster convergence, and lower transmission costs compared to FedAvg. Moreover, the ontology alignment process is successful in 95% of cases, while support for dynamic updates is possible without losing efficiency; data heterogeneity is also efficiently handled. The scalability of the framework has been tested by using larger datasets and an increasing number of edge servers, with positive results showing its resilience and appropriateness for extended IoT networks. These results allow us to consider its potentiality for real-time, privacy-preserving data analysis in applications like environmental monitoring and smart cities.

Future studies will focus on how to dynamically update mechanisms within the ontology, improve communication effectiveness of federated learning, and also optimize the framework for seamless real-time deployment across large-scale IoT networks. All these endeavors will definitely enhance the efficiency and adaptability of the suggested system, thus allowing the development of more creative and scalable IoT-edge solutions.

Author Contributions: Methodology, C.K.; Formal analysis, S.B.; Conceptualization, A.Z.; Writing—original draft, A.B.; Writing—review & editing, C.K and E.H.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sun, P.; Wan, Y.; Wu, Z.; Fang, Z.; Li, Q. A survey on privacy and security issues in IoT-based environments: Technologies, protection measures and future directions. *Comput. Secur.* **2025**, *148*, 104097. [\[CrossRef\]](#)
2. Greengard, S. *The Internet of Things*; MIT Press: Cambridge, MA, USA, 2021.
3. Kanzouai, C.; Bouarourou, S.; Zannou, A.; Nfaoui, E.H.; Boulaalam, A. IoT-Powered Big Data Processing for Intelligent Transportation: Enhancing Effectiveness and Forecasting. In *Big Data and Internet of Things*; Mahboub, O., Haddouch, K., Omara, H., Hefnawi, M., Eds.; Springer Nature Switzerland: Cham, Switzerland, 2024; pp. 913–923. [\[CrossRef\]](#)
4. Hazra, A.; Adhikari, M.; Amgoth, T.; Srirama, S.N. A Comprehensive Survey on Interoperability for IIoT: Taxonomy, Standards, and Future Directions. *ACM Comput. Surv.* **2023**, *55*, 9. [\[CrossRef\]](#)
5. Ferrer, A.J.; Marquès, J.M.; Jorba, J. Towards the Decentralised Cloud: Survey on Approaches and Challenges for Mobile, Ad hoc, and Edge Computing. *ACM Comput. Surv.* **2019**, *51*, 111. [\[CrossRef\]](#)
6. Shafique, K.; Khawaja, B.A.; Sabir, F.; Qazi, S.; Mustaqim, M. Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios. *IEEE Access* **2020**, *8*, 23022–23040. [\[CrossRef\]](#)
7. Zannou, A.; Nfaoui, E.H.; Boulaalam, A.; Allali, N.E.; Fariss, M. Data Gathering from IoT Networks. In Proceedings of the 2023 7th IEEE Congress on Information Science and Technology (CiSt), Essaouira, Morocco, 16–22 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 361–365. [\[CrossRef\]](#)
8. Lombardi, M.; Pascale, F.; Santaniello, D. Internet of Things: A General Overview between Architectures, Protocols and Applications. *Information* **2021**, *12*, 87. [\[CrossRef\]](#)
9. Marouan, A.; Badrani, M.; Zannou, A.; et al. E-Voting System Based on Blockchain for Enhanced University Elections. *SN Comput. Sci.* **2025**, *6*, 204. [\[CrossRef\]](#)
10. Marouan, A.; Badrani, M.; Kannouf, N.; Zannou, A.; Chetouani, A. Blockchain-based e-voting system in a university. *Indones. J. Electr. Eng. Comput. Sci.* **2024**, *34*, 1915–1923. [\[CrossRef\]](#)
11. Kanellopoulos, D.; Sharma, V.K.; Panagiotakopoulos, T.; Kameas, A. Networking Architectures and Protocols for IoT Applications in Smart Cities: Recent Developments and Perspectives. *Electronics* **2023**, *12*, 2490. [\[CrossRef\]](#)
12. Rahman, Z.; Yi, X.; Mehedi, S.T.; Islam, R.; Kelarev, A. Blockchain Applicability for the Internet of Things: Performance and Scalability Challenges and Solutions. *Electronics* **2022**, *11*, 1416. [\[CrossRef\]](#)
13. Bouarourou, S.; Zannou, A.; Nfaoui, E.H.; Boulaalam, A. An Efficient Model-Based Clustering via Joint Multiple Sink Placement for WSNs. *Future Internet* **2023**, *15*, 75. [\[CrossRef\]](#)
14. Kokila, M.; Reddy, K.S. Authentication, access control and scalability models in Internet of Things Security—A review. *Cyber Secur. Appl.* **2025**, *3*, 100057. [\[CrossRef\]](#)

15. Gilrein, E.J.; Carvalhaes, T.M.; Markolf, S.A.; Chester, M.V.; Allenby, B.R.; Garcia, M. Concepts and practices for transforming infrastructure from rigid to adaptable. *Sustain. Resilient Infrastruct.* **2021**, *6*, 213–234. [\[CrossRef\]](#)
16. Karabulut, E.; Pileggi, S.F.; Groth, P.; Degeler, V. Ontologies in digital twins: A systematic literature review. *Future Gener. Comput. Syst.* **2024**, *153*, 442–456. [\[CrossRef\]](#)
17. Tang, S.; Shelden, D.R.; Eastman, C.M.; Pishdad-Bozorgi, P.; Gao, X. A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends. *Autom. Constr.* **2019**, *101*, 127–139. [\[CrossRef\]](#)
18. Bouarourou, S.; Zannou, A.; Boulaalam, A.; Nfaoui, E.H. IoT Based Smart Agriculture Monitoring System with Predictive Analysis. In Proceedings of the 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 3–4 March 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–5. [\[CrossRef\]](#)
19. Alloui, H.; Mourdi, Y. Exploring the Full Potentials of IoT for Better Financial Growth and Stability: A Comprehensive Survey. *Sensors* **2023**, *23*, 8015. [\[CrossRef\]](#)
20. Xu, D.; Li, T.; Li, Y.; Su, X.; Tarkoma, S.; Jiang, T.; Crowcroft, J.; Hui, P. Edge Intelligence: Empowering Intelligence to the Edge of Network. *Proc. IEEE* **2021**, *109*, 1778–1837. [\[CrossRef\]](#)
21. Chen, M.; Gündüz, D.; Huang, K.; Saad, W.; Bennis, M.; Feljan, A.V.; Poor, H.V. Distributed Learning in Wireless Networks: Recent Progress and Future Challenges. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3579–3605. [\[CrossRef\]](#)
22. Liu, J.; Chen, C.; Li, Y.; Sun, L.; Song, Y.; Zhou, J.; Jing, B.; Dou, D. Enhancing trust and privacy in distributed networks: A comprehensive survey on blockchain-based federated learning. *Knowl. Inf. Syst.* **2024**, *66*, 4377–4403. [\[CrossRef\]](#)
23. Ariza, J.; Garcés, K.; Cardozo, N.; Rodríguez, Sánchez, J.P.; Jiménez Vargas, F. IoT architecture for adaptation to transient devices. *J. Parallel Distrib. Comput.* **2021**, *148*, 14–30. [\[CrossRef\]](#)
24. Hofer, M.; Obraczka, D.; Saeedi, A.; Köpcke, H.; Rahm, E. Construction of Knowledge Graphs: Current State and Challenges. *Information* **2024**, *15*, 509. [\[CrossRef\]](#)
25. Pasdar, A.; Koroniotis, N.; Keshk, M.; Moustafa, N.; Tari, Z. Cybersecurity solutions and techniques for Internet of Things integration in combat systems. In *IEEE Transactions on Sustainable Computing*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–20. [\[CrossRef\]](#)
26. Rathanasalam, R.; Kanagasabai, K. Data dissemination with interoperability in IoT network. *Int. J. Commun. Syst.* **2020**, *33*, e4513. [\[CrossRef\]](#)
27. Jeong, S.; Kim, S.; Kim, J. City Data Hub: Implementation of Standard-Based Smart City Data Platform for Interoperability. *Sensors* **2020**, *20*, 7000. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Tang, Q.; Xie, R.; Yu, F.R.; Chen, T.; Zhang, R.; Huang, T.; Liu, Y. Distributed Task Scheduling in Serverless Edge Computing Networks for the Internet of Things: A Learning Approach. *IEEE Internet Things J.* **2022**, *9*, 19634–19648. [\[CrossRef\]](#)
29. Tian, H.; Xu, X.; Lin, T.; Cheng, Y.; Qian, C.; Ren, L.; Bilal, M. DIMA: Distributed cooperative microservice caching for internet of things in edge computing by deep reinforcement learning. *World Wide Web* **2022**, *25*, 1769–1792. [\[CrossRef\]](#)
30. Marshoodulla, S.Z.; Saha, G. An approach towards removal of data heterogeneity in SDN-based IoT framework. *Internet Things* **2023**, *22*, 100763. [\[CrossRef\]](#)
31. Sharma, S.; Verma, V.K. An Integrated Exploration on Internet of Things and Wireless Sensor Networks. *Wirel. Pers. Commun.* **2022**, *124*, 2735–2770. [\[CrossRef\]](#)
32. Naidoo, P.; Sibanda, M. Emerging Trends and Future Directions of the Industrial Internet of Things. In *From Internet of Things to Internet of Intelligence*; Odhiambo, M.O., Mwashita, W., Eds.; Transactions on Computational Science and Computational Intelligence; Springer: Cham, Switzerland, 2024. [10.1007/978-3-031-55718-7_5](#). [\[CrossRef\]](#)
33. Ali, M.I.; Gao, F.; Mileo, A. CityBench: A Configurable Benchmark to Evaluate RSP Engines Using Smart City Datasets. In *The Semantic Web-ISWC 2015*; Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d’Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., et al., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 374–389. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.