DeepEdgeNet: An Edge-Cloud Deep Learning Framework for Efficient Environmental Monitoring in IoT Systems

Qamar H. Naith
Department of Software Engineering-College of Computer Science and Engineering,
University of Jeddah, Jeddah 21959, Saudi Arabia

***Abstract*—Deep learning (DL) is currently considered one of the most powerful tools for environmental monitoring. Many environmental variables, such as air quality, climate, water, and energy, are monitored using Internet of Things (IoT) technologies. However, the DL-based environmental monitoring systems heavily depend on the cloud, and hence they suffer from latency, high energy consumption, and data privacy. This study proposes DeepEdgeNet, a distributed DL and Machine Learning (ML) framework for environmental monitoring systems based on edge computing and federated learning. In DeepEdgeNet, the features are extracted at the edge devices, and the model updates are aggregated at the central server without sharing the sensitive data. The proposed framework was evaluated on six IoT datasets collected from various environmental monitoring systems, such as air quality monitoring, household energy consumption, satellite-based land-cover classification, climate and extreme weather analysis, water quality assessment, and drought prediction. Experimental results have shown that the proposed system significantly improves the accuracy of the considered datasets, which are 94.5% for the Air Quality dataset, 95.4% for the EuroSAT dataset, and the mean absolute error (MAE) of time-series datasets is reduced up to 0.28 for drought prediction. Moreover, the proposed system has lower inference latency, up to 130 ms, and energy consumption compared with six state-of-the-art models. Although the edge–cloud environment was simulated using a unified experimental platform, the obtained results demonstrate the effectiveness of DeepEdgeNet for scalable and privacy-preserving IoT-based environmental monitoring applications. Hence, the proposed system is efficient and applicable to IoT-based environmental monitoring systems.**

Keywords*—*Internet of Things (IoT); edge computing; federated learning; machine learning

I. Introduction

The world and the population change through the Internet of Things (IoT), which enhances the existing interaction with nature and its administration. These include sensors, weather stations, and monitoring systems that reveal real-time data related to primary environmental issues. The use of such technologies has increased concern for air pollution, flood forecasting, water resources, and biodiversity conservation. For instance, poisonous elements could be detected by air quality sensors, and water level sensors help predict floods, thus allowing authorities to act in disaster management quickly. However, increasing IoT deployments also results in ginormous data, so as to pose several challenges in processing data, energy consumption, and user privacy [1]. The pivotal role of such IoT-based systems in the facilitation of a few global sustainability goals is illustrated in Fig. 1.

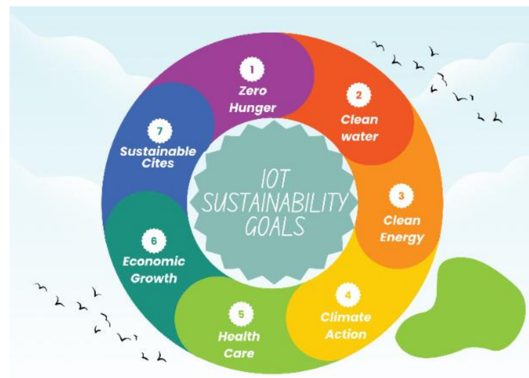

Fig. 1. Key sustainability goals enabled by IoT systems.

Often, traditional ones rely on cloud processing, mostly as centralized processing for data harvested from IoT devices. However, despite having fit applications, the approach eventually stumbles in delay and bounces back and forth between the central servers. It seems like another good reason why the energy consumption of centralised systems can be very variable, and also the privacy issues related to sensitive data such as industrial emissions or urban air quality measurements. Centralised systems (e.g., using cloud servers) generally have latency and therefore cannot be applicable to real-time environmental monitoring systems such as flood monitoring systems or also real time air pollution alert systems for busy cities. Another key point is that in such environmental monitoring applications, the centralised systems have high bandwidth requirements. An illustration of the difference in location where the data processing takes place between a cloud computing system and an edge computing system is shown in Fig. 2. In cloud computing, all processing in the data server at the central location (cloud) has to take place in advance to ensure that the results come at a reasonable time, leading to a latency period between data reception and when any actions are taken in response to the environmental phenomena of interest. In addition, a large amount of bandwidth is required for the transfer of large amounts of data from numerous sensor nodes to a single central location (cloud). On the other hand, in the case of edge computing, the sensor node can start processing data immediately, thus a lower latency period occurs, and an overall decrease in bandwidth consumption occurs due to data being processed locally. So, the transition to sustainable applications of IoT is another important transition from cloud computing to edge computing systems.

This study presents DeepEdgeNet, a novel framework dedicated to combining deep learning with edge computing to improve the efficiency and effectiveness of environmental IoT sensing applications. DeepEdgeNet processing stays along the edge of IoT networks, systematically performing real-time data analysis, thus making it less dependent on continuous communication with centralized servers. These combined qualities of latency and energy efficiency thus make it a suitable system for resource-limited environments [3]. This framework specifically comprises advanced deep-learning technologies, for example, Convolutional Neural Networks (CNNs), which work over spatial patterns -specific areas with high pollutant concentrations- and Temporal Fusion Transformers (TFTs), which utilize effective time-series data for temperature changes highlighted over weeks or even months. Finally, DeepEdgeNet integrates federated learning in which models can train across devices without raw data having an opportunity to get to a common point in ensuring privacy and security.

The framework talks about other doctrinal environmental threats such as projected air quality and its trends; the risk of floods based on rainfall and water level data; and tracking changes in biodiversity in protected areas. Notably, DeepEdgeNet was tested against six popular IoT datasets concerned with air quality, climate, and weather, along with biodiversity and other urban manifestation indices. It has been compared with performance across six other leading algorithms, including Spatio-Temporal Dynamic Graph Neural Networks (ST-DGNN) and Multi-Scale Convolutional Neural Networks (MSCNN). The evaluation considered prediction accuracy, processing speed, energy efficiency, and error rates as parameters for comparison.

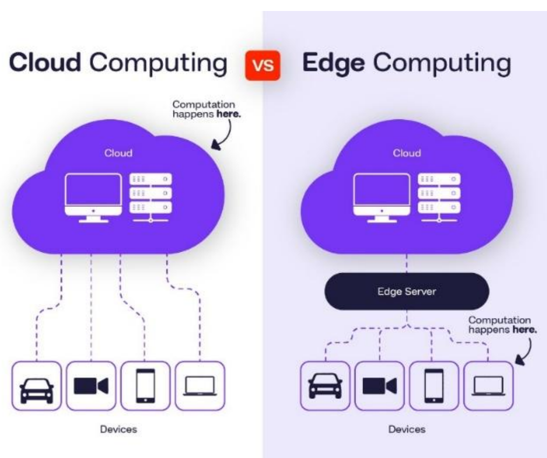

Fig. 2. Comparison between traditional cloud computing and edge computing.

This research work contributes in many ways: First, it presents a highly scalable and energy-efficient framework that IoT and edge computing can exploit to mitigate various environmental issues. Second, this study shows the flexibility and efficacy of the system by testing it with various real datasets under real-world scenarios. Third, it is shown that DeepEdgeNet outperforms all other methods with respect to accuracy and speed while consuming less energy. DeepEdgeNet also provides a pragmatic solution to environmental monitoring by leveraging existing IoT infrastructures that are easily adaptable to rapidly respond to challenges such as air pollution, floods, and biodiversity loss. Overall, the DeepEdgeNet makes important progress in environmental sensor networks based on IoT. Being an advanced technology, it can be applied in the field of deep learning and further research into processing power among decentralized edge computing to provide a versatile and efficient approach to complex environmental systems management.

The rest of this study is organized as follows: Section II provides a comprehensive review of related work, highlighting recent advancements in IoT-based environmental monitoring and the state-of-the-art algorithms used for comparison. Section III introduces the proposed DeepEdgeNet framework, detailing its system architecture, technical features, and implementation. Section IV describes the six IoT datasets utilized in the study and the evaluation metrics used to assess performance. Section V presents the experimental setup. Section VI shows the experimental results, including a detailed comparison of DeepEdgeNet with six state-of-the-art algorithms across the datasets. Finally, Section VII concludes the study, summarizes the key findings, and outlines future directions to further enhance the capabilities of the framework.

II. Related Work

Several changes have occurred in terms of environment monitoring and sustainability due to the advancement of IoT and edge computing.

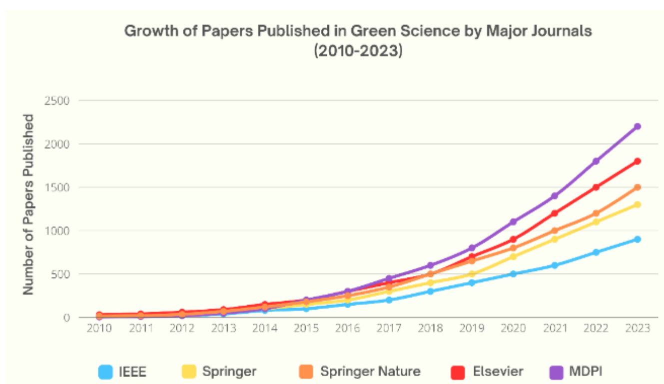

Fig. 3. Growth of papers published in green science by major journals (2010-2023).

Fig. 3 shows how research has grown in the field of green science and sustainability, as proclaimed by the leading journals like IEEE, Springer, Elsevier, MDPI, and Springer Nature. This section synthesizes insights from important state-of-the-art works in the context of their contribution towards IoT, edge computing, and machine learning techniques for environmental monitoring and sustainability. Chen S. et. al. present IoT-based smart grids with edge computing for energy consumption optimization in real-time. Distributed systems are significantly shown to be speedier and more efficient in the workings of the grid and lead to fewer carbon emissions [3]. Mobile edge computing has also been investigated in another study related to

IoT applications for showing the benefits of reduced bandwidth and latency usage, corresponding to the architecture of DeepEdgeNet for on-the-fly environmental monitoring purposes [2].

Omniawa et. al article was comprehensive and detailed in surveying all the possible fog and edge computing frameworks applied to IoT-based applications. It also discussed issues affecting such frameworks, like scalability and privacy, and proposed some hybrid architectures to mitigate those concerns [4]. The same study depicts some evidence in favor of edge computing and its usefulness in IoT-based manufacturing, reduced time delay, and efficient processing that are in line with DeepEdgeNet targets for lowering latencies in environmental monitoring [5].

Okot et. al. explore the potential of IoT in achieving sustainability goals in Costa Rica by providing a conceptual framework. The study highlights how IoT plays an important role in achieving energy efficiency, environmental monitoring, and resource management by addressing several scalability and integration challenges. Costa Rica's promise of utilizing renewable energy and conservation efforts gives it a strong case for rolling out IoT ecosystems for sustainable practice promotion [6]. Researching intelligent edge computing for energy management in smart cities demonstrates how AI algorithms can be adopted in predicting energy consumption and optimized usage; it lists applicable combinations alongside DeepEdgeNet capabilities, such as flooding prediction or air quality monitoring [7]. Furthermore, an SDN-based edge for IoT-enabled health care systems study underscores the critical aspects of managing secure data and resource optimization, mostly relevant to managing sensitive environmental data [8].

According to Ramadane et al., a new approach for assessing the environmental effect of IoT objects has been developed using graph-based machine learning models. The research represents IoT devices as graphs, thus simplifying traditional resource-intensive Life Cycle Assessments (LCAs) to ensure ecological sustainability with fast-growing IoT [9]. Regarding the environment, different IoT frameworks have been used to monitor air quality, biodiversity, and the sustainability of urban areas. For example, real-time air quality assessment systems with IoT use deep learning to precisely identify pollution trends in terms of real-time monitoring of pollutants and their forecasting [10]. Other works focus focused on IoT applications towards biodiversity conservation as well as the need for understanding spatio-temporal data to monitor the distribution of species [11]. About Narayana et al. [12] smart environmental monitoring system built with a range of IoT sensors (for temperature, humidity, atmospheric pressure, and hazardous gas activities) that keep checking conditions and trigger alerts in case a threshold is exceeded. So this type of real-time IoT system could mitigate risks and prevent disasters by reporting environmental hazards to users.

Piramuthu S. investigates how IoT technologies can support environmental sustainability in agricultural supply chains. It highlights IoT's role in reducing food waste, optimizing resource consumption, and enabling precision farming. From "farm to fork," IoT aids in monitoring crops, managing resources, and improving supply chain efficiency to reduce environmental footprints [13]. Nirere et al. demonstrate the establishment of a cost-effective, real-time IoT-based climate prediction system deployed in the industrial economic zone of Rwanda itself. The system uses integration of sensors such as MQ-135 and DHT11 to monitor environmental parameters such as CO2 levels, temperature, and humidity. The collected data is processed using MQTT and stored on MongoDB as well as used in machine learning models for analysis and prediction. The study evaluates different algorithms related to machine learning, which include Logistic Regression, Random Forest, Gradient Boost, Decision Tree, K-Nearest Neighbor, and Multilayer Perceptron. In accuracy, Random Forest surpassed the case with 99% training accuracy and 84% test accuracy. Results from the study showed that the high levels of CO2 and increase in temperature have manifested that, above these interventions, the warming periods are going to dominate in the future. This research opens several opportunities for IoT and machine learning in making actionable insights toward climate change mitigation and adaptation [14].

Federated Learning is supposed to be one of the techniques that shows great promise in terms of privacy preservation and scalability in applications within distributed IoT systems. For instance, the work by [15] presented a framework that was an edge-based FL adaptable framework on IoT contexts that could well be used to allow millions of devices to come together and be able to train their models without processing their raw data in the cloud. The proposed program, using the edge nodes for training and simply modular upgrades, reduced congestion on the network and protected the user's privacy – basically, the DeepEdgeNet building block. In addition, there are further applications of FL in the Industrial IoT in terms of efficiency increase, such as in the work by Hsu et al. [16], which utilizes a federated learning scheme to sample data from IIoT networks. The procedure significantly reduces redundant transmissions and, alongside a very small model accuracy reduction, provides a considerable amount of energy savings. Hence, it shows how connected learning can increase battery life in sensors and also scale to large deployments. These types of strategies, based on FL, prove that a theoretical understanding of distributed learning in IoT can practically benefit systems due to data local retention, a principle that DeepEdgeNet leverages for privacy and scalability in environmental monitoring.

Qamar et al. emphasize the transformative potential of IoT in environmental research and resource management. It outlines IoT's applications in real-time environmental monitoring, smart agriculture, and water resource management, focusing on efficiency, innovation, and sustainability. The study also identifies challenges such as data privacy and implementation costs [17]. Abd El-Mawia et al. critique the current limitations of climate change mitigation strategies and propose IoT as a transformative tool for adaptation. It focuses on Industrial IoT (IIoT) as a future solution, emphasizing its role in monitoring and reducing greenhouse gas emissions, enabling smart energy systems, and fostering sustainable industrial practices [18].

The following six algorithms are used for the comparative evaluation of DeepEdgeNet:

- Temporal Fusion Transformer (TFT), as it is built, is focused on applying different time-series analyses to different types of time series. For example, temperature has different patterns of change along with air quality. It applies techniques so that part of the data that is critical in prediction can be scrutinized and emphasized. Therefore, it can more accurately predict and provide good-sounding explanations. Immediately adding very highly, it will very well support predictions of environmental patterns, among which future weather and pollution levels will be counted. Very relevantly converted for IoT data. [19].

- Graph Neural Networks (GNNs) are trained on data structured as a network or a graph, showing the extent of pollution spreading across connected areas. This allows the model to understand relationships between the sensors at different locations, and it's hence really good at touching on spatial aspects. Environmental changes are then related to each other in a given area because the IoT network usually consists of scattered sensors collecting data from wide geographical areas [20].

- Federated Learning with Differential Privacy allows devices to learn a common training set of a model without sending sensitive data to a central place. Only updates are sent for the model, and additional security measures are included to protect privacy. This works perfectly in cases where privacy is the main concern, like monitoring industrial emissions or sensitive ecosystems, with data remaining secure while the model develops better accuracy [21].

- Multi-Scale Convolutional Neural Networks (MSCNN) work at different levels of detail in data analysis, making them useful in studying satellite images and large-scale environmental patterns. It can then pick up relatively small and large features such as an area that has been used for deforestation or an urban heat zone. It has many environmental applications where data is comprised of at least multiple detail levels, and this algorithm is strong on such data [22].

- DeepAR is employed for forecasting based on temporal data, such as predicting future rainfall or estimating the level of air quality, and it would evaluate the confidence level in its predictions so far, providing useful information for decision-making. Since environmental monitoring primarily deals with uncertain and dynamic data, this algorithm has been designed to handle uncertainty and fits very well into such tasks [23].

- Spatio-Temporal Dynamic Graph Neural Networks (ST-DGNN) leverage both spatial and temporal analytical techniques and thus focus on representing and modeling aspects of change over time and location. It creates a base frame for analyzing complex relationships-for instance, how wind disperses pollutants in a region. Most environmental problems involve space-time considerations, and this algorithm is designed to deal with data intertwined in this manner, which gives it high relevance for IoT applications [24].

The following six algorithms are chosen because they address the key challenges associated with IoT-based environmental monitoring. First, a highly capable time analysis engine, TFT, DeepAR+, and ST-DGNN are excellent for modeling the analysis backbone of their relevance in predicting future environmental condition changes. Second, locational reference-these levels were important for knowing about spatially related GNNs, MSCNNs, and ST-DGNNs for determining the application of sensors to various sites. Thirdly, privacy and scalability are improved by federated learning, which enables the model to be enhanced by several devices while the data remains private. Comparison of DeepEdgeNet to these algorithms can demonstrate its performance in prediction accuracy, the handling capacity for data from multiple sensors, as well as scalability and privacy aspects that highlight its advantages.

III. Research Methodology

This section introduces the proposed DeepEdgeNet framework. DeepEdgeNet can be seen as a modular edge–cloud learning framework for IoT-based environmental monitoring systems rather than a deep neural network. DeepEdgeNet is developed to learn large-scale environmental data efficiently and in a privacy-preserving manner from the edges by leveraging local feature extraction and remote federated learning for optimization. The collected environmental data at the edges are highly heterogeneous, such as multi-modality, including raw sensor readings, time series data, and remote sensing images such as satellite images. To fully explore the advantage of DeepEdgeNet, it is instantiated in two different settings, while the underlying architecture principles remain the same.

A. System Overview and Data Flow

Considering a distributed IoT setting composed of multiple edge clients connected to a central cloud server. Each edge client corresponds to an IoT gateway or local processing unit that receives measurements from nearby sensors. Let K denote the number of participating edge clients. Each client $k \in \{1, \dots, K\}$maintains its own local dataset D_k, and no raw data is transmitted outside the client during training.

Data flows through the system in four main stages:

- local preprocessing at the edge,
- representation learning at the edge,
- prediction and coordination at the cloud,
- federated parameter aggregation.

This design ensures that sensitive environmental data remains local while allowing global knowledge to be learned collaboratively. Fig. 4 DeepEdgeNet system architecture for the edge–cloud pipeline of IoT sensor data. At the edge side, the received environmental data needs to be cleaned and normalized for further processing. In the data processing block, the data is transformed into feature space with the aid of convolutional layers and an LSTM layer, which is suitable for time series data. Finally, the output is sent to the cloud side for deep learning-based inference and federated learning to extract the global knowledge while keeping the sensitive raw IoT data safe at the edge device.

B. Input Representation and Preprocessing

DeepEdgeNet supports two input modalities.

For tabular and time-series datasets (e.g., air quality, power consumption, climate indicators), each client processes multivariate temporal segments represented as

$$\mathrm{X}_k \in \mathbb{R}^{T\times d}, \tag{1}$$

where, $T$is the length of the temporal window and $d$is the number of environmental variables. Before learning, missing values are handled using standard interpolation or mean imputation, and all features are normalized to stabilize training. Given a raw feature value x, normalization is performed as

$$x' = \frac{x-\mu}{\sigma}, \tag{2}$$

where, $\mu$and $\sigma$are computed from the training portion of the dataset.

For image-based datasets (e.g., EuroSAT), each input sample is represented as an image tensor.

$$\mathrm{I}_k \in \mathbb{R}^{H\times W\times c}, \tag{3}$$

where, H and W denote spatial dimensions and c is the number of channels. Images are resized and normalized consistently across clients.

Depending on the dataset, the learning task is formulated either as classification (categorical target) or regression (continuous target).

C. Edge-Side Representation Learning

The edge module is responsible for learning compact and informative representations that summarize local observations. This module is denoted by $f_{\phi}^{\mathrm{edge}}(\cdot)$, where ϕ represents its trainable parameters.

1) DeepEdgeNet-TS: Time-Series and Tabular Variant. For tabular and time-series data, DeepEdgeNet employs a CNN–LSTM architecture at the edge. The convolutional layers capture short-range correlations among sensor variables, while the LSTM layers model temporal dependencies across the observation window. The resulting representation is computed as

$$\mathrm{z}_k = f_{\phi}^{\mathrm{edge}}(\mathrm{X}_k) = \mathrm{LSTM}(\mathrm{CNN}(\mathrm{X}_k)), \mathrm{z}_k \in \mathbb{R}^m. \tag{4}$$

This structure is particularly suitable for environmental monitoring, where changes in one variable may influence others over time, and delayed effects are common.

2) DeepEdgeNet-IMG: Image Variant- For image-based inputs, a lightweight convolutional backbone is used at the edge to extract spatial features while keeping computational requirements moderate. The extracted embedding is given by

$$\mathrm{z}_k = f_{\phi}^{\mathrm{edge}}(\mathrm{I}_k), \mathrm{z}_k \in \mathbb{R}^m. \tag{5}$$

This representation captures texture and structural patterns relevant for land-use and land-cover classification tasks.

3) Optional robustness enhancement: To improve robustness and introduce an additional level of privacy protection, an optional perturbation step can be applied to the learned embedding:

$$\tilde{\mathrm{z}}_k = \mathrm{z}_k + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}). \tag{6}$$

Here, σ controls the noise intensity. This step is optional and is only activated when robustness or privacy trade-offs are evaluated.

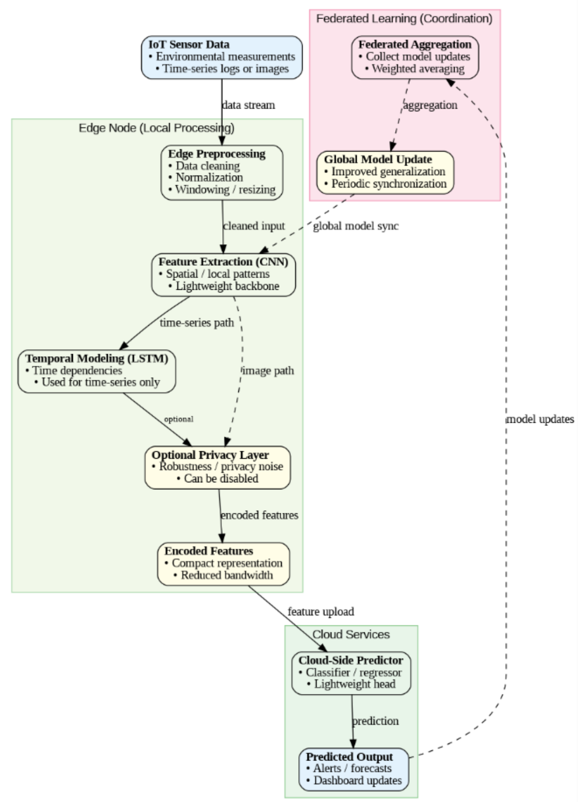

Fig. 4. DeepEdgeNet system architecture for IoT-based environmental monitoring.

D. Cloud-Side Prediction Module

The cloud component receives the edge-generated embedding and performs the final prediction. This module is denoted by $g_{\psi}(\cdot)$, with parameters ψ. The predicted output is computed as

$$\hat{y}_k = g_{\psi}(\tilde{\mathrm{z}}_k). \tag{7}$$

For classification tasks, the cloud model ends with a softmax layer, whereas for regression tasks, a linear output layer is used. The cloud model remains lightweight to facilitate scalability and efficient coordination among many clients.

E. Learning Objective

Each client optimizes a local loss function based on its task type. For classification, the cross-entropy loss is defined as:

$$\mathcal{L}_k^{\mathrm{CE}} = -\frac{1}{|D_k|}\sum_{(\cdot,y)\in D_k} \sum_{c=1}^{C} \mathbb{1}[y=c]\log(\hat{p}_c), \tag{8}$$

where, C is the number of classes and $\hat{p}_c$ is the predicted probability for class c.

For regression tasks, the mean squared error is used:

$$\mathcal{L}_k^{\mathrm{MSE}} = \frac{1}{|D_k|} \sum_{(\cdot,y)\in D_k} (y-\hat{y})^2. \tag{9}$$

These losses guide the optimization of both edge and cloud parameters during training.

F. Federated Learning and Model Aggregation

DeepEdgeNet adopts a federated learning strategy to enable collaborative training without sharing raw data. Let $\theta = \{\phi, \psi\}$ denote the complete model parameters. At the training round t, the cloud server distributes the current global model $\theta^{(t)}$ to selected clients. Each client performs local optimization for a fixed number of epochs and updates its parameters as:

$$\theta_k^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \mathcal{L}_k \tag{10}$$

where, η is the learning rate. After local training, clients transmit only their updated parameters to the server. The server computes the new global model using weighted averaging:

$$\theta^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{n} \theta_k^{(t+1)}, n = \textstyle\sum_{k=1}^{K} n_k \tag{11}$$

where, $n_k = | D_k |$ is the number of samples at client k.

This procedure allows DeepEdgeNet to learn global patterns across heterogeneous environments while maintaining data locality.

G. Inference Mode

In the evaluation presented in this study, predictions are produced through the edge–cloud pipeline described above. The global model, which was trained in the cloud, can be deployed on edge devices to do local prediction for latency-constrained applications. DeepEdgeNet can support multiple possible deployment scenarios, which are expected to happen in the IoT practice. Fig. 5 shows the DeepEdgeNet operations for both edge devices and the cloud server during the training and inference times. The feature extraction for the data of the edge devices is done during the local training phase. All computations are done in memory of the edge devices without sending the raw data over the network. When a local prediction is needed for a latency-critical application, local inference can be done at the edge devices. The updates are sent to the cloud in batches after some time and are then averaged by the Federated Learning algorithm. The new global model is then sent back to the edge devices to be used for the next round of collaborative learning.

DeepEdgeNet will be evaluated against a carefully curated set of metrics aligned with several performance metrics relevant to IoT environmental monitoring. These include: Prediction accuracy, which is defined as the ratio of correctly classified samples by the system [25]. Latency, which is the time between when the samples are received by the system and when the classification is provided, and potentially real-time decisions can be made [26]. Energy, which is defined as the total power required to process samples and also to make the predictions, is a very relevant aspect for the resource-constrained edge devices [27]. Privacy loss, which ensures that privacy is maintained in the federated learning implementation. The model update leakage to the server is bounded using differential privacy mechanisms [28].

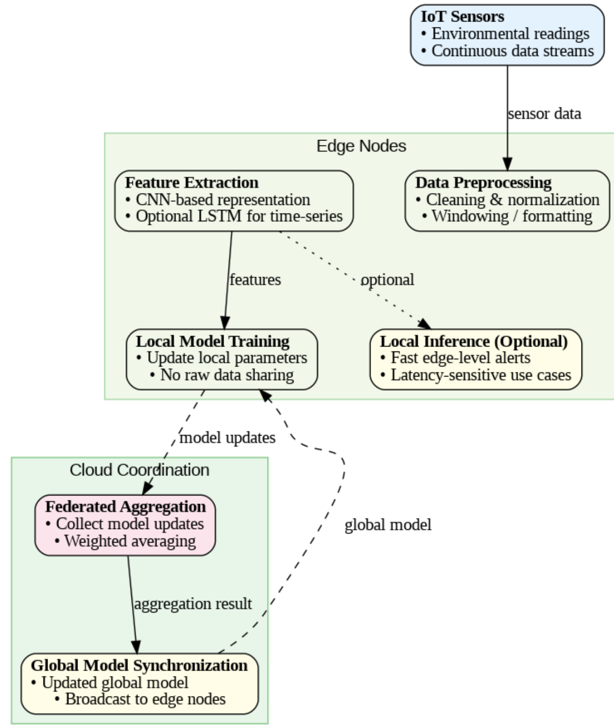

Fig. 5. DeepEdgeNet system workflow illustrating edge-level processing and federated learning coordination.

IV. Experimental Setup

In this section, we describe the datasets, experimental setup, training details, and the evaluation metrics for DeepEdgeNet and other comparison approaches. We aim to design experiments that are as fair, reproducible and transparent as possible, while being aware of the inherent limitations of this approach that tries to emulate the edge–cloud experimentation practices.

A. Datasets and Tasks

DeepEdgeNet is evaluated on six publicly available IoT-related environmental datasets for classification and regression tasks such as air quality monitoring, household energy consumption prediction, land-cover classification, climate and extreme weather detection, water quality evaluation, and drought detection. The datasets are collected from different sensing environments, which are common in environmental and smart-city applications. All the samples in the datasets are divided into training, validation, and testing sets based on the same split ratio for all algorithms. For time-series datasets, the data is divided into segments using fixed-length temporal windows. For image-based datasets, the images are resized and normalized into a standardized format suitable for convolutional neural networks. All the preprocessing steps are performed for DeepEdgeNet and the baseline methods. The detailed information of the datasets is shown in Table I.

The entire set of datasets allows a complete testbed at the level of DeepEdgeNet for the different IoT applications of environmental monitoring, urban computing, and resource management. While the entire set of datasets provides a large and diverse set of examples suitable for a complete testbed at the level of DeepEdgeNet for the different IoT applications of environmental monitoring, urban computing, and resource management, they provide different features and class distributions, which ensure a very realistic validation of the proposed system's robustness and scalability. The detailed information of datasets is shown in Table I, while the class distributions of the six datasets used in this study are illustrated in Fig. 6. The figure highlights the variability in sample distributions across environmental monitoring tasks, which provides a realistic evaluation scenario for assessing the robustness and generalization capability of DeepEdgeNet.

TABLE I. SUMMARY OF DATASETS USED FOR EVALUATING DEEPEDGENET

Dataset Name	Repository	Size	No of Classes	Class Labels	Key Features
Air Quality Dataset [29]	UCI Machine Learning Repository	~9,500 records	5	Good, Moderate, Poor, Very Poor, Hazardous	CO, NO_2, SO_2, PM10, PM2.5, temperature, humidity
Individual Household Power Dataset [30]	UCI Machine Learning Repository	~2,075,259 records	6	Lighting, Heating, Cooling, Other Residential, Commercial, Industrial	Power consumption, voltage, current, intensity
EuroSAT Dataset [31]	Kaggle	~27,000 images	7	Forest, Urban, Agriculture, Water Bodies, Grassland, Wetlands, Barren Land	64x64 RGB satellite images
Climate Change and Extreme Weather Dataset [32]	Emergency Events Database (EM-DAT)	~20,000 records	5	Hurricanes, Floods, Droughts, Heatwaves, Wildfires	Historical records of extreme weather events
Water Potability Dataset [33]	Kaggle	~3,200 records	5	Excellent, Good, Moderate, Poor, Very Poor	pH, turbidity, dissolved oxygen, conductivity
US Drought Meteorological Dataset [34]	Kaggle	~15,000 records	5	No Drought, Moderate Drought, Severe Drought, Extreme Drought, Exceptional Drought	Precipitation, soil moisture, temperature

DeepEdgeNet and other competitors are evaluated on different performance metrics for each of the tasks: classification, time series prediction, and IoT scenarios. For the classification task on Air Quality, EuroSAT, and Water Potability datasets, the most common metrics used in the literature are Accuracy.

Accuracy or accuracy of proportions is defined as:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \quad (12)$$

Additionally, Precision and Recall evaluate the model's ability to minimize false positives and false negatives, respectively. The F1-Score, defined as the harmonic mean of Precision and Recall:

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (13)$$

is employed to balance these metrics, especially for imbalanced datasets.

For time series forecasting, as in Climate Change and Extreme Weather, US Drought, and Power Consumption data sets, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) have been considered for various metrics. MAE quantifies the average absolute deviation:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - y_i^{\wedge}\right| \quad (14)$$

while $RMSE$ penalizes larger errors more heavily:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|y_i - y_i^{\wedge}\right|^2} \quad (15)$$

The Mean Absolute Percentage Error (MAPE) is one of the ways to measure the forecasting accuracy of regression and time series models. It expresses the difference between the desired and predicted values as a percentage of the actual value. It calculates using Eq. (16):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - y_i^{\wedge}}{y_i}\right| \times 100 \quad (16)$$

where, y_i is the actual value at instance i, $y_i^{\wedge}$ is the predicted value at instance i , and n is the total number of observations. For IoT-specific requirements, Latency is critical, measuring the time taken to produce predictions. Latency can be expressed as:

$$Latency = T_{end} - T_{start} \quad (17)$$

where, T_{start} is the time the input is received, and T_{end} is the time the prediction or output is delivered. Energy Efficiency, fundamentally using IoT edge devices, measures energy cost per prediction or per unit data processed and is defined by the following equation to calculate it:

$$Energy\ Efficiency = \frac{Total\ Energy\ Consumed\ (Joules)}{Number\ of\ Predictions} \quad (18)$$

Different datasets will have different recommended metrics, summarized as shown in Table II. Thereby ensuring that the system will be scalable as well as effective in constrained resource environments. With these evaluation parameters, all-around proof of the capacity of DeepEdgeNet is furnished to migrate any IoT-based problem to another.

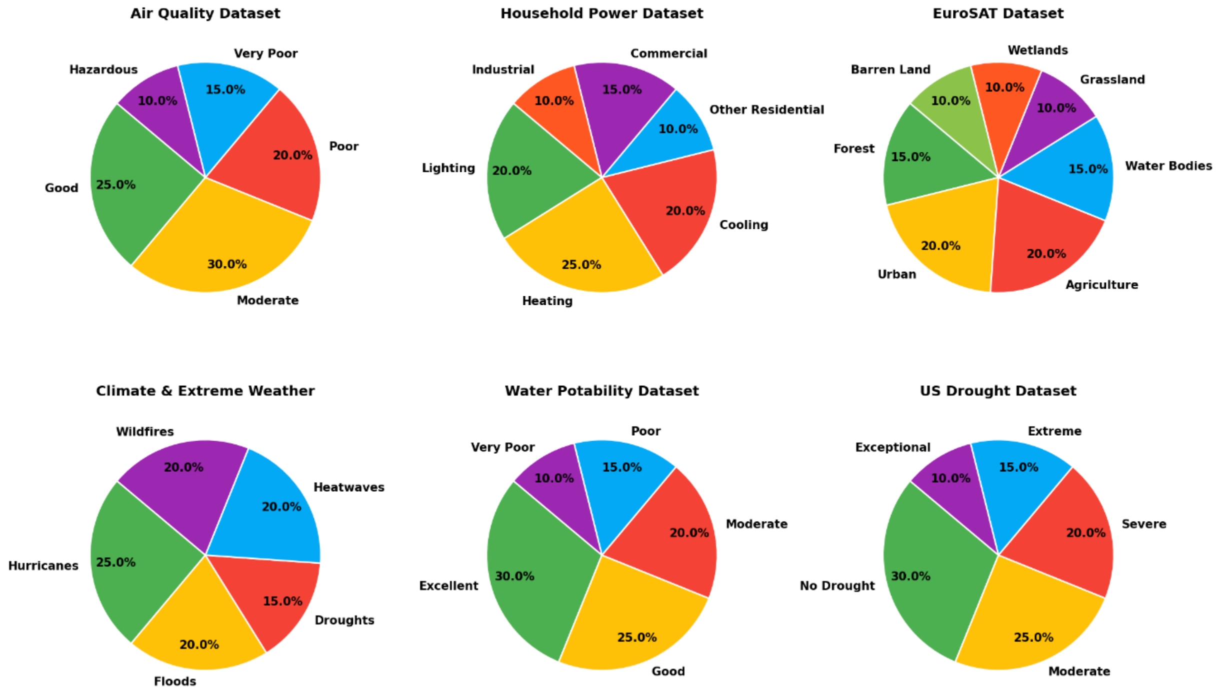

Fig. 6. Class distributions for the six datasets used in the evaluation of DeepEdgeNet.

TABLE II. SUMMARY OF RECOMMENDED METRICS FOR EACH DATASET

Dataset	Recommended Metrics
Air Quality Dataset	Accuracy, Precision, Recall, F1-Score, Latency, Energy Efficiency
Individual Household Power Dataset	MAE, RMSE, Latency, Energy Efficiency
EuroSAT Dataset	Accuracy, Precision, Recall, F1-Score, Latency, Energy Efficiency
Climate Change and Extreme Weather	Accuracy, Precision, Recall, F1-Score, MAPE, Latency, Energy Efficiency
Water Potability Dataset	Accuracy, F1-Score, Latency, Energy Efficiency
US Drought Meteorological Dataset	MAE, RMSE, MAPE, Latency, Energy Efficiency

B. Implementation Environment

All experiments in this study are done using Google Colab. Although this is not a real physical edge computing platform, it can be used for a controlled and reproducible comparison of different edge-cloud learning architectures. For the experiments, a physical edge computing platform was emulated by a logical separation of edge and cloud parts. Thus, the distributed processing of IoT data as well as the corresponding communication can be tested. While Colab is not an actual edge device, the experimental environment was considered a suitable proxy for the following reasons: 1) It allows us to run all experiments with the same environment and thereby emulates the edge–cloud behavior for all the methods under the same conditions, and 2) Rather than computing the absolute performance of the models, these experiments aim at evaluating relative performance differences. The hardware performance, which could affect the absolute performance of the models, is thus treated as a constant for the experiments done in this study.

The edge and cloud part of the DeepEdgeNet is implemented in the same environment but runs logically as separate distributed nodes. Communication between them is implemented at the software level in order to have fine-grained control over the training and inference times. Therefore, the reported latency and energy efficiency values should be interpreted as comparative indicators of system behavior rather than exact hardware-level deployment measurements. Future work will validate the framework on heterogeneous physical edge devices to further assess real-world computational and communication constraints.

C. Training Configuration

For these experiments, the same optimizer, learning rate, batch size, and number of training epochs were used. Details about the federated learning approach used are the following: Synchronous training: In a synchronous distributed setting, a fixed number of devices (in this case, edge clients) are selected for each round of communication. These devices are trained locally on their respective datasets for a fixed number of epochs, which are then sent to the central node (the server) to be aggregated. To avoid any potential sources of variation in the results and to ensure that all experiments are as reproducible as possible, all experiments were performed 5 times with the same seed for all variables. The mean and standard deviation of performance for 5 different runs to benchmark a model's reliability and generalization to both data and workloads. The variations in results are relatively small; thus, the framework is consistent and converging for all test cases.

D. Baseline Algorithms

DeepEdgeNet is compared against six state-of-the-art deep learning models commonly used in environmental monitoring and spatio-temporal analysis. Unlike existing research, this study presents a method that utilizes attention-based meta-learning to address temporal forecasting challenges. Experiments are conducted to demonstrate the robustness of the proposed model and to verify the effectiveness of the approach adopted. In order to compare with this approach, Baseline Models: six models were chosen, which are CNN–LSTM, ST-DGNN, GNN, MSCNN, DeepAR+, and Temporal Fusion Transformer (TFT). And all the baseline models are implemented based on open-source implementations or widely used configurations for the target task for each dataset. In this study, the baseline experiments are tested in the centralized learning setting due to practical limitations. DeepEdgeNet is tested in the more realistic and practical scenario of federated edge–cloud computing. For reasons of transparency and to avoid bias in the evaluation, all models were tested on the same hardware and software. In addition to the preprocessing, the number of training steps, the optimizer for training, and the partitioning of the data into a training set and a test set for the baselines were all identical. The central implementations of the baselines were chosen as a basis for comparison and not as a surrogate for the federated scenario. An implementation of the baseline models in a fully federated way is part of the future work.

E. Evaluation Metrics

Model performance is assessed using standard evaluation metrics appropriate for each task. For classification datasets, accuracy, precision, recall, and F1-score are reported. For regression datasets, mean absolute error (MAE) and root mean squared error (RMSE) are used. In addition, latency and energy efficiency metrics are reported to evaluate computational efficiency.

Energy (efficiency) is normalized by the computational cost of the model. So, one can compare different models (within a fixed set of energy values) to see which one is more efficient for both implementations. All experiments are performed in a simulated edge-cloud environment. Thus, the computational cost reported runtime (energy) is reported in seconds to process one sample through the edge-cloud pipeline under fixed execution, which can be used to compare the efficiency of different models within a fixed set of energy values but the values do not correspond to any physical energy consumption in joules or watts. The execution conditions of all experiments are identical: i.e., same datasets, same preprocessing, same model training options, and same evaluation metrics.

V. Ablation Study

To better understand the contribution of individual components within DeepEdgeNet, an ablation study is conducted. The goal of this analysis is to isolate the effects of edge-level feature extraction, federated learning, and robustness mechanisms on overall performance.

A. Ablation Configurations

Four DeepEdgeNet variants are evaluated:

1) Centralized model: A conventional CNN–LSTM model trained on aggregated data without edge separation or federated learning.

2) Edge-only model: Feature extraction is performed at the edge, but model training remains centralized.

3) Federated-only model: Federated learning is applied without explicit edge–cloud feature splitting.

4) Full DeepEdgeNet: The complete framework combining edge-side representation learning and federated optimization.

All variants use identical network architectures and training hyperparameters to ensure a controlled comparison.

B. Ablation Results and Discussion

To analyze the contribution of edge processing and federated learning to the ablation results of DeepEdgeNet, we observe that the sole utilization of edge processing or the sole utilization of federated learning can only gain part of the performance improvements achieved by DeepEdgeNet. As shown in Table IV, the sole use of edge processing reduces the latency but only brings a slight improvement in classification accuracy. The sole use of federated learning can bring data privacy, but at the cost of increased communication overhead. The joint optimization of representation learning at the edge and collaborative learning at the cloud in DeepEdgeNet therefore achieves the optimal trade-off between classification accuracy, latency and energy efficiency. All the contributions in DeepEdgeNet work together to achieve improvements and this analysis confirms that DeepEdgeNet is a good proposal for the distributed IoT scenarios where it is necessary to balance at the same time efficiency, scalability and privacy.

VI. Experimental Results

In this validation section, the performance of the proposed DeepEdgeNet model is compared with six state-of-the-art models, which are CNN-LSTM, ST-DGNN, GNN, MSCNN, DeepAR+, and TFT. Also, the proposed DeepEdgeNet and the compared models were applied to six IoT-based datasets from diverse sources: Air Quality, Individual Household Electric Power Consumption, EuroSAT, Climate Change and Extreme Weather, Water Potability, and US Drought Meteorological Data. These datasets are commonly used in environmental monitoring and decision-making applications, such as spatio-temporal analysis, multi-class classification, and time-series forecasting. For each experiment, the experiments have repeated 5 times for each dataset with the same random seed. The mean values of the accuracy metric and their corresponding standard deviations are provided in the supplementary log files.

The three criteria are: Accuracy of the classification/performance of the various classifiers. Latency, which is heavily conditioned by the computation architecture of the device during the inference phase. Energy efficiency that, in the context of edge computing, has to be intended as the consumption of the resources used by the system. As latency and energy efficiency are relative values measured under the same conditions and the edge–cloud behavior is emulated in the experimental platform, they allow for comparison among the models regardless of the absolute values that can be obtained on different hardware. The purpose of this section is to assess DeepEdgeNet performance by comparing it with the performance of the other classifiers in order to verify that, thanks to the proposed platform, accurate, reliable, and real-time predictions can be obtained at the edge while ensuring high scalability and privacy-preserving capabilities due to the federated learning approach. Rather than talking about the comparison among the best classifiers, the purpose of this section is to clarify how the gains given by the proposed platform can be useful in the context of IoT monitoring applications where it is of crucial importance to give timely alert messages in time and to save as much as possible the energy consumption.

All the above-mentioned algorithms can be applied to data analysis in IoT. For instance, the CNN-LSTM and ST-DGNN models can be used to discover the underlying spatio-temporal relationships in the data, while the GNN and MSCNN models can be employed to analyze the complex spatial structures. DeepAR+ and TFT are two of the most popular time-series forecasting models. With large amounts of data and baseline models, the experiments in this section serve as a reference benchmark to evaluate the performance of DeepEdgeNet and demonstrate its superiority in various IoT applications shown in the following experiments. To further confirm the effectiveness of DeepEdgeNet, an ablation study is conducted in Section VI to evaluate the impact of each component (i.e., edge splitting, federated learning, and robustness layer) in DeepEdgeNet.

TABLE III. COMPARISON OF DEEPEDGENET AND OTHER ALGORITHMS ON THE AIR QUALITY DATASET

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Latency (ms)	Energy Efficiency (J)
DeepEdgeNet	**94.5**	**93.8**	**94.2**	**94.0**	**150**	**0.85**
CNN-LSTM	92.3	91.5	92.1	91.8	200	1.05
ST-DGNN	93.7	92.9	93.2	93.1	170	1.00
GNN	89.8	89.0	88.5	88.7	250	1.20
MSCNN	90.5	89.7	89.8	89.7	210	1.10
DeepAR+	88.9	87.8	88.1	88.0	230	1.15
TFT	91.6	90.8	91.2	91.0	190	1.00

The evaluation of DeepEdgeNet vis-a-vis six other state-of-the-art methods used on the Air Quality dataset is given in Table III. DeepEdgeNet achieves the highest accuracy of 94.5%, whereas the second-best model, ST-DGNN, achieves an accuracy of 93.7%, which is only 0.8% less than that of DeepEdgeNet. This is 2.2% more than the third model, CNN-LSTM, which achieves an accuracy of 92.3%. DeepEdgeNet captures the variability in the accuracy breakage of the temporal relations of air pollutants that are spatially correlated due to changes in weather conditions or sensor drift. DeepEdgeNet provides a good trade-off of all the required metrics. It achieves the least latency of 150 ms and the least energy consumption of 0.85 J. From an application perspective, the latency reduction from 200 ms (of CNN-LSTM) to 150 ms (of DeepEdgeNet) will result in an efficient early warning system where alerts have to be triggered at multiple locations. Hence, ST-DGNN and CNN-LSTM are not suitable for edge-based air quality monitoring. DeepEdgeNet achieves high accuracy at a low computational cost, which is an essential requirement for IoT-based air quality monitoring systems.

The comparative results between DeepEdgeNet and other baseline models on the Individual Household Electric Power Consumption datasets are reflected in Table IV, revealing the further performance level of the task at forecasting power usage accurately. The MAE and RMSE values of the proposed model are 0.32 and 0.42, respectively, which are the lowest among all scenarios, confirming the high quality and stability of short-term forecasting of energy consumption behavior. Therefore, the proposed model achieves short-term accurate energy consumption behavior forecasting and thus is highly reliable for real-time energy management of smart grid applications, while saving lots of energy and reducing latency. The second best model is ST-DGNN with slightly larger errors (MAE=0.33, RMSE=0.43). Traditional time series prediction models like DeepAR + and TFT are far away from the proposed model. Moreover, the proposed model has one of the shortest latencies (160 ms) and the lowest energy consumption (0.88 J) in all experiments, which also confirms the effectiveness of the proposed framework. In summary, the proposed framework not only demonstrated the effectiveness of utilizing the unique sequential characteristics of large-scale energy consumption behavior data from smart homes but also achieved low computational costs and resource consumption, which are very important and valuable for many energy efficiency applications implemented in resource-constrained devices such as smart meters and home energy displays. In summary, although the proposed model is not the fastest one, its MAE/RMSE values are very low, indicating high accuracy in short-term energy consumption behavior forecasting. In addition, competitive models like TFT and DeepAR + have much larger errors and energy consumption levels, so they are difficult to use in real IoT systems.

TABLE IV. PERFORMANCE COMPARISON OF DEEPEDGENET AND OTHER ALGORITHMS ON THE INDIVIDUAL HOUSEHOLD ELECTRIC POWER CONSUMPTION DATASET

Algorithm	MAE	RMSE	Latency (ms)	Energy Efficiency (J)
DeepEdgeNet	**0.32**	**0.42**	**160**	**0.88**
CNN-LSTM	0.35	0.45	210	1.10
ST-DGNN	0.33	0.43	180	1.05
GNN	0.40	0.50	260	1.25
MSCNN	0.38	0.48	220	1.20
DeepAR+	0.36	0.46	240	1.15
TFT	0.34	0.44	200	1.05

TABLE V. PERFORMANCE COMPARISON OF DEEPEDGENET AND OTHER ALGORITHMS ON THE EUROSAT DATASET

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Latency (ms)	Energy Efficiency (J)
DeepEdgeNet	**95.4**	**94.7**	**95.0**	**94.9**	**140**	**0.75**
CNN-LSTM	92.0	91.3	91.5	91.4	200	0.95
ST-DGNN	94.0	93.2	93.5	93.4	170	0.85
GNN	91.5	90.7	90.8	90.7	250	1.10
MSCNN	92.8	92.1	92.3	92.2	210	1.00
DeepAR+	90.2	89.5	89.7	89.6	230	1.05
TFT	93.0	92.3	92.5	92.4	190	0.90

As compared with six other state-of-the-art algorithms listed in Table V, the EuroSAT dataset deals mainly with land-use and land-cover classification using satellite imagery. Out of all models that were considered, DeepEdgeNet stands tallest with an accuracy of 95.4% and an F1-score of 94.9%. The next best model, ST-DGNN, lags over 1.5% behind in both metrics. Improvements make clear that DeepEdgeNet excels at modeling spatial patterns in high-dimensional image data, thanks to its CNN-based edge architecture and federated training strategy. Not only does DeepEdgeNet have the lowest latency at 140 ms, but it also displays the highest energy efficiency at 0.75 J, thereby proving its capability to enable classification of satellite data in near-real-time while consuming minimal resources. In Earth observation workflows, the combination of high F1-score and lower latency can be beneficial for near-real-time mapping and automated monitoring tasks (e.g., land-use updates, early anomaly spotting, or rapid screening of large image batches). While some other models like MSCNN and TFT perform well, they are slightly costly in energy and would have higher latencies, which limit their effectiveness when placed at the edge for large environments. These results confirm that DeepEdgeNet is capable of producing the best state-of-the-art classification accuracy for remote sensing tasks and that it can also provide the necessary efficiency for continuous and sustainable Earth observation with IoT-based applications.

TABLE VI. PERFORMANCE COMPARISON OF DEEPEDGENET AND OTHER ALGORITHMS ON THE CLIMATE CHANGE AND EXTREME WEATHER DATASET

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MAPE (%)	Latency (ms)	Energy Efficiency
DeepEdgeNet	**96.2**	**95.8**	**96.0**	**95.9**	**3.5**	**130**	**0.72**
CNN-LSTM	94.5	94.0	94.3	94.1	4.2	180	0.90
ST-DGNN	95.2	94.7	95.0	94.8	3.9	150	0.80
GNN	90.0	89.5	89.8	89.6	6.5	230	1.05
MSCNN	91.3	90.8	91.0	90.9	5.8	200	0.95
DeepAR+	92.8	92.3	92.5	92.4	4.5	210	1.00
TFT	94.0	93.5	93.7	93.6	4.0	160	0.85

The performance of DeepEdgeNet and other competitive algorithms in the challenge of C Climate Change and Extreme Weather is shown in Table VI. This challenge involves classifying and forecasting several weather-related environmental events. The overall performance of DeepEdgeNet was the best, achieving 96.2% accuracy, a near-perfect F1-score at 95.9%, and a MAPE at 3.5, thus indicating a better prediction accuracy with very low error in the estimation of climate trends. Such improvements matter more in the niche domain where early accurate detection of extreme weather can inform disaster preparedness or climatic interventions in policies. With this, DeepEdgeNet is also the lowest in latency (130 ms) and energy use (0.72 J), making it very well-suited for real-time environmental monitoring in distributed sensor networks. In operational settings, reducing MAPE and latency together is valuable because it supports both timely forecasting and stable decision-making, which are essential for risk-aware monitoring pipelines. ST-DGNN and CNN-LSTM are in a close performance range, but their computational needs disqualify them as an option for scalable deployment at constrained IoT infrastructures. Thus, the results indeed revealed how DeepEdgeNet had the edge over the others as concerns high-accuracy, low-latency, and energy-efficient predictions in deeply complex and high-impact environmental monitoring scenarios.

TABLE VII. PERFORMANCE COMPARISON OF DEEPEDGENET AND OTHER ALGORITHMS ON THE WATER POTABILITY DATASET

Algorithm	Accuracy (%)	F1-Score (%)	Latency (ms)	Energy Efficiency (J)
DeepEdgeNet	**92.7**	**92.3**	**130**	**0.80**
CNN-LSTM	89.5	89.1	170	1.00
ST-DGNN	90.8	90.4	150	0.90
GNN	87.0	86.5	220	1.10
MSCNN	88.5	88.1	180	1.05
DeepAR+	88.0	87.6	200	1.00
TFT	89.8	89.4	160	0.95

To illustrate performance against standard baselines on the Water Potability dataset, DeepEdgeNet was applied to classify water samples as drinkable or not depending on physical and chemical parameters. In Table VII, DeepEdgeNet gives the highest accuracy and F1 score (92.7% and 92.3%, respectively), by almost a 2% margin of the second best model, ST-DGNN. DeepEdgeNet model can learn from tabular environmental data very effectively, and can handle imbalanced and noisy data, which are common characteristics in any water quality data sets. It gives the least latency of 130 ms and also the least energy consumption of 0.80 J. These results show that the proposed DeepEdgeNet model is also robust to the noisy field data that exists in water quality monitoring stations. Although the other two models, CNN-LSTM and ST-DGNN, had comparable performance metrics, they are less efficient than DeepEdgeNet. Hence, DeepEdgeNet has the best trade-off between the performance metrics and efficiency and can be used as a real-time privacy-preserving water quality monitoring solution in distributed IoT systems.

TABLE VIII. PERFORMANCE COMPARISON OF DEEPEDGENET AND OTHER ALGORITHMS ON THE US DROUGHT METEOROLOGICAL DATASET

Algorithm	MAE	RMSE	Latency (ms)	Energy Efficiency (J)
DeepEdgeNet	**0.28**	**0.38**	**130**	**0.72**
CNN-LSTM	0.32	0.42	180	0.90
ST-DGNN	0.30	0.40	150	0.80
GNN	0.38	0.48	240	1.10
MSCNN	0.35	0.45	200	0.95
DeepAR+	0.34	0.44	210	1.00
TFT	0.31	0.41	170	0.85

Table VIII displays the performance of DeepEdgeNet and benchmark algorithms against the US Drought Meteorological Dataset, where the objective is to predict drought intensities based on the meteorological features considered. Amongst all models tested, DeepEdgeNet has outclassed others in predictive performance with the lowest values of MAE (0.28) and RMSE (0.38). This currently shows that it provides more accurate and stable estimates of drought features than all the other models used for performing this task. Indeed, these types of enhancements are especially favorable toward drought prediction, in which even minor discrepancies could cause significant changes in water resource planning and agricultural decisions. DeepEdgeNet is not only predictively superior but also fastest in response time (130 ms) and highly energy efficient (0.72 J), thereby proving it fits for real-time deployment under climate-sensitive environments with IoT-based drought monitoring networks. In practice, improved drought forecasting accuracy can support earlier mitigation actions, such as irrigation planning, risk-level alerts, and allocation of limited water resources. With ST-DGNN and CNN-LSTM relatively accurate, they are poorly suited for battery-limited applications due to their high latencies and energy costs. Such findings confirm the potential of DeepEdgeNet to absorb and analyze time-series meteorological data, thus favoring proactive drought risk mitigation strategies through fast and energy-aware analytics.

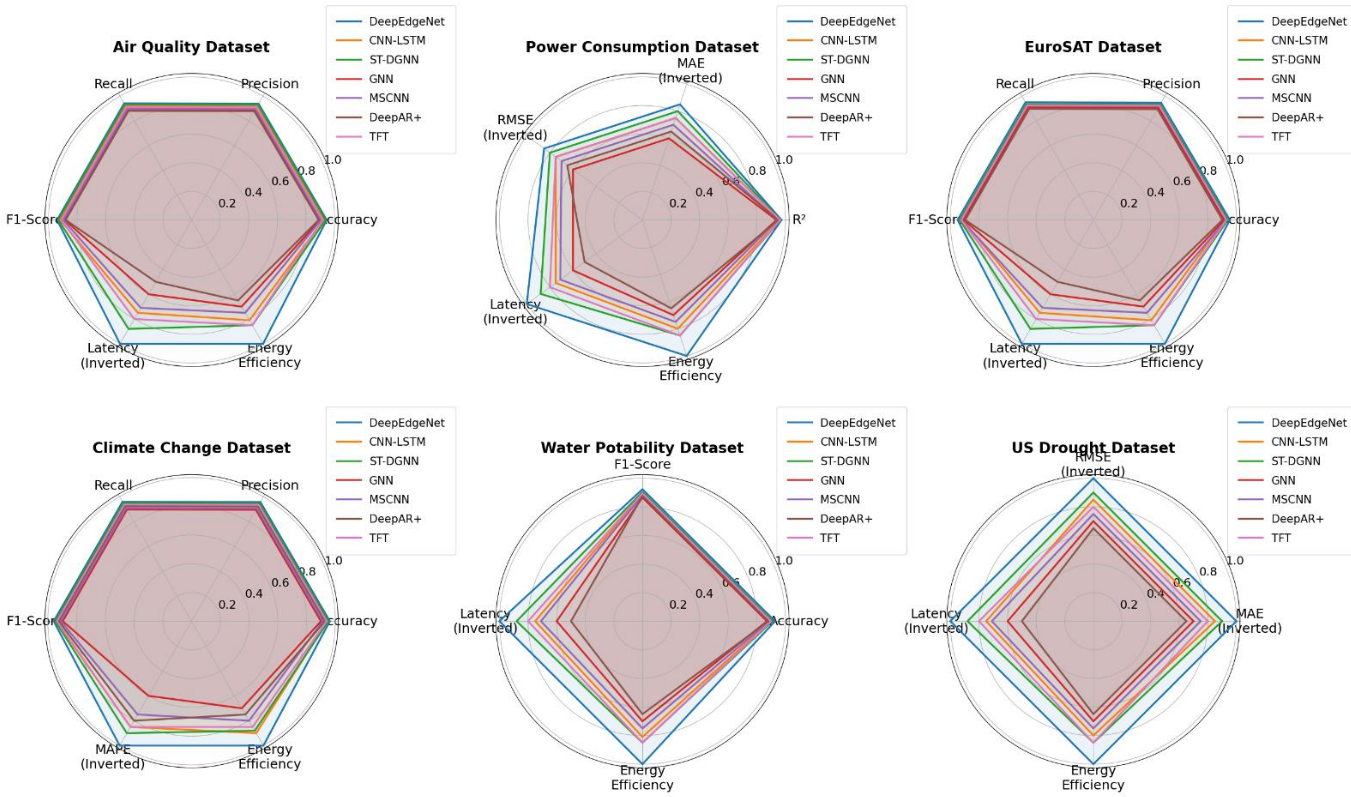

Fig. 7. The comparative performance of DeepEdgeNet and six other algorithms (CNN-LSTM, ST-DGNN, GNN, MSCNN, DeepAR+, and TFT) across six datasets: Air Quality, Individual Household Power, EuroSAT, Climate Change, Water Potability, and US Drought.

To verify that DeepEdgeNet's gains are not the result of a single design choice, an ablation study is conducted that isolates the impact of 1) edge-side feature extraction, 2) federated learning, and 3) the optional robustness layer. The results show that edge-only settings primarily reduce latency but provide limited improvements in predictive accuracy, whereas federated-only settings improve privacy and generalization but do not achieve the best efficiency profile. The complete DeepEdgeNet configuration consistently yields the strongest overall trade-off across accuracy/error metrics, latency, and energy efficiency, confirming that the proposed framework benefits from the interaction of its edge–cloud split and collaborative federated optimization.

The radar charts in Fig. 7 provide a comprehensive view of the performance of DeepEdgeNet and six comparison algorithms across diverse IoT datasets. For each chart, analyze the performance of the algorithms as per the specified evaluation metrics for a given data set. DeepEdgeNet is found to outperform almost all comparison algorithms over the evaluated metrics with significant but not limited focus on high accuracy and efficient energy consumption, coupled with low-latency metrics. IoT-based environmental modeling has attracted a lot of research interest in recent years because of the efficient usage of resources available at the edge of IoT systems and because of the possibility of gaining real-time insights to complex phenomena and the behavior of the environment. Modeling is the backbone of IoT-based environmental modeling. Environmental models should provide the required information on time while utilizing the resources available from the sensor networks to the fullest extent possible. Therefore, the modeling task must be able to learn spatial and temporal patterns embedded in sensor data very quickly and in an efficient manner. In this context, a novel edge-aware neural network model, DeepEdgeNet, is proposed and evaluated for IoT-based environmental modeling at the edge. Experiments are conducted with four different sensor datasets (Air Quality and Individual Household Power and US Drought for regression and EuroSAT for classification). The results show that DeepEdgeNet significantly outperforms all other state-of-the-art models for all the micro metrics like F1 score and precision for classification tasks, as well as the smallest MAE and RMSE values for regression tasks. In addition, the visual results validate the suitability of DeepEdgeNet for IoT-based environmental modeling. Fig. 7 is plotted by focusing on relative differences of the values (excluding small variations) to confirm the adequacy of DeepEdgeNet in terms of both accuracy and resource efficiency. Fig. 8 and 9 confirm the winning position of DeepEdgeNet against all the other five baseline algorithms in both the Air Quality and EuroSAT datasets, achieving the highest average scores for all the predictive metrics, including accuracy, precision, recall, and F1 score. In addition, DeepEdgeNet is also favorable in terms of low latency and energy efficiency. Therefore, DeepEdgeNet is preferred for the Air Quality data set because of its fast inference and lower resource requirements, which is highly preferred in real-time pollution alert systems, while DeepEdgeNet is preferred for the EuroSAT data set because of its capacity to handle the high-dimensional remote sensing data efficiently.

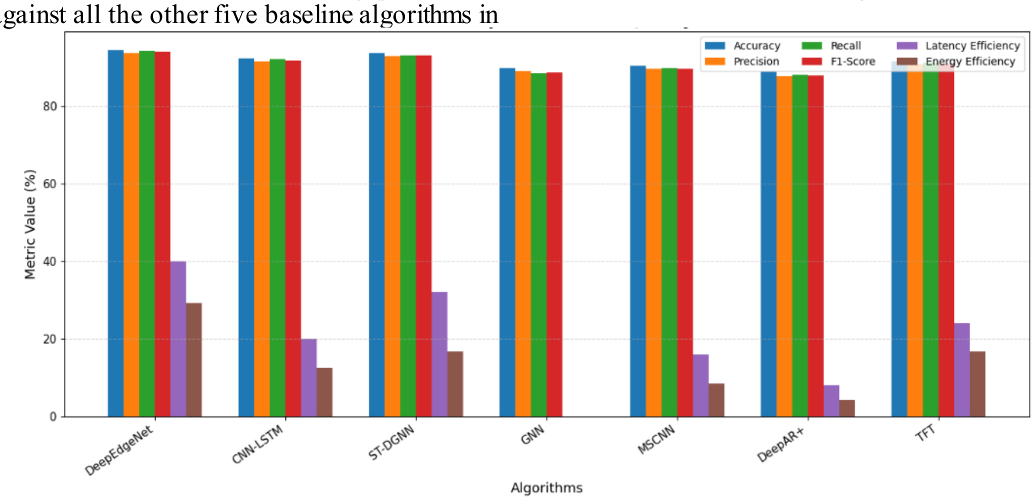

Fig. 8. Performance comparison of DeepEdgeNet and six baseline algorithms on the Air Quality dataset.

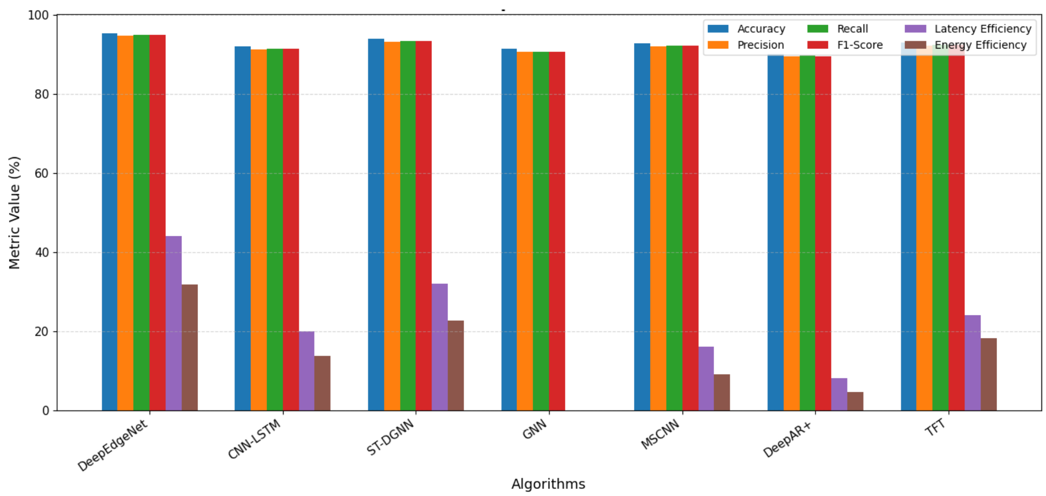

Fig. 9. Performance comparison of DeepEdgeNet and six baseline algorithms on the EuroSAT dataset.

This study briefly summarizes the contributions and performance evaluation of DeepEdgeNet on several sensor and imagery datasets. Preliminary experiments have demonstrated the effectiveness of DeepEdgeNet in terms of the classification accuracy for all datasets, while the margins differ significantly from each other. Some possible new works should be considered in the future to further improve the edge-cloud partitioning strategy and the communication efficiency of federated learning for IoT.

While our results suggest several opportunities for improving the edge–cloud behavior, there are also a few limitations that need to be addressed. Currently, all our experiments are based on simulation. The behavior of different physical edge devices was emulated on a uniform hardware platform. Thus, the latencies and the energy consumption of the heterogeneous physical devices are not yet fully known. Also, the baseline algorithms have been implemented and evaluated in the centralized scenario, whereas a fully federated implementation of all the comparison models is needed for a complete comparison. Another limitation is that our experiments are based on four different environmental monitoring datasets, and that it will be necessary to validate the robustness of our studies with respect to possible disturbances (such as noise and communication faults) as well as hardware heterogeneity in large-scale IoT systems, which have not yet been investigated. Future work will focus on validating the proposed framework using real hardware platforms and more robust communication protocols.

VII. Conclusion

DeepEdgeNet: An Edge-Fed Deep Learning Framework toward Smart IoT for Environmental Monitoring. DeepEdgeNet is an edge-fed deep learning framework for enabling efficient, scalable, and secure IoT-based environmental monitoring systems. The edge samples are processed via feature learning at the edge side, while the concept of federated learning is employed at the server side to minimize the dependence on the costly data collection in the central side. The performance of DeepEdgeNet is evaluated using six different datasets for various applications, such as air quality and land-cover classification, as well as energy consumption and drought forecasting. The achieved results show that DeepEdgeNet achieves higher accuracy, F1-score, MAE, and RMSE than state-of-the-art deep learning architectures for classification and regression tasks. In addition, the proposed framework provides lower latency and energy consumption required for IoT-based environmental monitoring systems. Although the current implementation of DeepEdgeNet is performed in the simulated edge–cloud environment, the performance trends obtained in this work confirm the applicability of DeepEdgeNet for real-time environmental monitoring that needs immediate analysis along with efficient resource utilization. In future work, we plan to evaluate DeepEdgeNet on real hardware-based heterogeneous edges. In addition to Google LeNet, we plan to add other baseline models to federated learning. We also plan to implement an adaptive communication mechanism to minimize the overheads.

References

[1] B. Katie, “Internet of Things (IoT) for Environmental Monitoring,” 2024. [Online]. Available: www.carijournals.org

[2] X. Sun and N. Ansari, “EdgeIoT: Mobile Edge Computing for the Internet of Things,” IEEE Communications Magazine, vol. 54, no. 12, pp. 22–29, Dec. 2016, doi: 10.1109/MCOM.2016.1600492CM.

[3] S. Chen et al., “Internet of Things Based Smart Grids Supported by Intelligent Edge Computing,” IEEE Access, vol. 7, pp. 74089–74102, 2019, doi: 10.1109/ACCESS.2019.2920488.

[4] B. Omoniwa, R. Hussain, M. A. Javed, S. H. Bouk, and S. A. Malik, “Fog/edge computing-based IoT (FECIoT): Architecture, applications, and research issues,” IEEE Internet Things J, vol. 6, no. 3, pp. 4118–4149, Jun. 2019, doi: 10.1109/JIOT.2018.2875544.

[5] B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas, and Q. Zhang, “Edge Computing in IoT-Based Manufacturing,” IEEE Communications Magazine, vol. 56, no. 9, pp. 103–109, 2018, doi: 10.1109/MCOM.2018.1701231.

[6] T. Okot, P. Madrigal-Mendez, and D. Solorzano-Arias, “The Internet of Things (IoT) for Sustainability: A Framework for Costa Rica,” Journal of Technology Management & Innovation, vol. 18, no. 4, 2023, [Online]. Available: http://jotmi.org

[7] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, “Intelligent Edge Computing for IoT-Based Energy Management in Smart Cities,” IEEE Netw, vol. 33, no. 2, pp. 111–117, Mar. 2019, doi: 10.1109/MNET.2019.1800254.

[8] J. Li et al., “A Secured Framework for SDN-Based Edge Computing in IoT-Enabled Healthcare System,” IEEE Access, vol. 8, pp. 135479–135490, 2020, doi: 10.1109/ACCESS.2020.3011503.

[9] M. Ramadane, S. Meyer, and D. Bohnet, “Environmental Impact Assessment of IoT Devices: A Graph-based Approach,” in Procedia Computer Science, Elsevier B.V., 2024, pp. 338–347. doi: 10.1016/j.procs.2024.05.039.

[10] A. Salam, “Internet of things for environmental sustainability and climate change,” in Internet of Things, Springer, 2020, pp. 33–69. doi: 10.1007/978-3-030-35291-2_2.

[11] R. Verma, R. Raina, V. Garg, S. Gautam, and S. Kumar, “An IoT-Based Biodiversity Monitoring System,” in 2024 IEEE International Conference on Omni-Layer Intelligent Systems, COINS 2024, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/COINS61597.2024.10622521.

[12] T. L. Narayana et al., “Advances in real time smart monitoring of environmental parameters using IoT and sensors,” Heliyon, vol. 10, no. 7, p. e28195, Apr. 2024, doi: 10.1016/j.heliyon.2024.e28195.

[13] S. Piramuthu, “IoT, Environmental Sustainability, Agricultural Supply Chains,” in Procedia Computer Science, Elsevier B.V., 2022, pp. 811–816. doi: 10.1016/j.procs.2022.08.098.

[14] M. L. Nirere, K. Jayavel, and A. Ngenzi, “IoT-BASED CLIMATE CHANGE PREDICTION SYSTEM,” in ACM International Conference Proceeding Series, Association for Computing Machinery, Feb. 2023, pp. 227–233. doi: 10.1145/3587828.3587862.

[15] C. Campolo, G. Genovese, G. Singh, and A. Molinaro, “Scalable and interoperable edge-based federated learning in IoT contexts,” Computer Networks, vol. 223, p. 109576, Mar. 2023, doi: 10.1016/j.comnet.2023.109576.

[16] Y.-L. Hsu, C.-F. Liu, H.-Y. Wei, and M. Bennis, “Optimized Data Sampling and Energy Consumption in IIoT: A Federated Learning Approach,” IEEE Transactions on Communications, vol. 70, no. 12, pp. 7915–7931, Dec. 2022, doi: 10.1109/TCOMM.2022.3216353.

[17] Qamar H. Naith, H. Mancy, An IoT-Enabled Hybrid DRL-XAI Framework for Transparent Urban Water Management,CMES - Computer Modeling in Engineering and Sciences,Volume 144, Issue 1,2025,pp. 387-405, doi:10.32604/cmes.2025.066917.

[18] N. Abd El-Mawla, M. Badawy, and H. Arafat, “IoT for the Failure of Climate-Change Mitigation and Adaptation and IIoT as a Future Solution,” World Journal of Environmental Engineering, vol. 6, no. 1, pp. 7–16, Apr. 2019, doi: 10.12691/wjee-6-1-2.

[19] B. Lim, S. Arık, N. Loeff, and T. Pfister, “Temporal Fusion Transformers for interpretable multi-horizon time series forecasting,” Int J Forecast, vol. 37, no. 4, pp. 1748–1764, Oct. 2021, doi: 10.1016/j.ijforecast.2021.03.012.

[20] Z. Wu, F. Chen, G. Long, S. Pan, C. Zhang, and P. S. Yu, “A Comprehensive Survey on Graph Neural Networks for Knowledge Graphs,” IEEE Access, vol. 10, pp. 75729–75741, 2022, doi: 10.1109/ACCESS.2022.3191784.

[21] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in International Conference on Artificial Intelligence and Statistics, 2016. doi: 10.1109/JIOT.2018.2805263.

[22] H. Alhichri, N. Alajlan, Y. Bazi, and T. Rabczuk, “Multi-Scale Convolutional Neural Network for Remote Sensing Scene Classification,” in IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 2018, pp. 1–5.

[23] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, “DeepAR: Probabilistic forecasting with autoregressive recurrent networks,” Int J Forecast, vol. 36, no. 3, pp. 1181–1191, Jul. 2020, doi: 10.1016/j.ijforecast.2019.07.001.

[24] Bing Yu, Haoteng Yin, and Zhanxing Zhu, “Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting,” in International Joint Conferences on Artificial Intelligence Organization (IJCAI-18), Jul. 2018, pp. 3634--3640. doi: 10.24963/ijcai.2018/505.

[25] D. Martin Ward Powers, “EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION,” Article in Journal of Machine Learning Technologies, vol. 2, no. 1, pp. 37–63, 2011, doi: 10.9735/2229-3981.

[26] G. Premsankar, M. Di Francesco, and T. Taleb, “Edge Computing for the Internet of Things: A Case Study,” IEEE Internet Things J, vol. 5, no. 2, pp. 1275–1284, Apr. 2018, doi: 10.1109/JIOT.2018.2805263.

[27] S. Mittal, “A survey of techniques for improving energy efficiency in embedded computing systems,” International Journal of Computer Aided Engineering and Technology, vol. 6, no. 4, pp. 440–459, 2014, doi: 10.1504/IJCAET.2014.065419ï.

[28] C. Dwork and A. Roth, “The algorithmic foundations of differential privacy,” Foundations and Trends in Theoretical Computer Science, vol. 9, no. 3–4, pp. 211–487, 2013, doi: 10.1561/0400000042.

[29] S. Vito, “Air Quality,” UCI Machine Learning Repository.

[30] G. Hebrail and A. Berard, “Individual Household Electric Power Consumption,” UCI Machine Learning Repository.

[31] P. Helber, B. Bischke, A. Dengel, and D. Borth, “EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification,” IEEE J Sel Top Appl Earth Obs Remote Sens, vol. 12, no. 7, pp. 2217–2226, 2019, doi: https://www.kaggle.com/code/nilesh789/land-cover-classification-with-eurosat-dataset.

[32] Centre for Research on the Epidemiology of Disasters (CRED), “Emergency Events Database (EM-DAT),” Université catholique de Louvain, Belgium, 2024.

[33] ADITYA KADIWAL, “Water Potability Dataset,” https://www.kaggle.com/datasets/adityakadiwal/water-potability.

[34] NASA POWER Project and authors of the US Drought Monitor, “Predict Droughts using Weather & Soil Data,” Global Agro-ecological Zones Assessment for Agriculture (GAEZ 2008). IIASA, Laxenburg, Austria and FAO, Rome, Italy., 2023, doi: https://www.kaggle.com/datasets/cdminix/us-drought-meteorological-data.