

# Minimizing the Carbon Footprint in LoRa-Based IoT Networks: A Machine Learning Perspective on Gateway Positioning

FRANCISCO-JOSE ALVARADO-ALCON , RAFAEL ASOREY-CACHEDA  (Senior Member, IEEE),  
ANTONIO-JAVIER GARCIA-SANCHEZ , AND JOAN GARCIA-HARO  (Member, IEEE)

Department of Information and Communication Technologies, Universidad Politécnica de Cartagena, 30202 Cartagena, Spain

CORRESPONDING AUTHOR: RAFAEL ASOREY-CACHEDA (e-mail: rafael.asorey@upct.es).

This work was supported in part by MICIU/AEI/10.13039/501100011033 and FEDER, EU, under Grant PID2023-148214OB-C21, in part by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR under Grant TED2021-129336B-I00, and in part by Fundación Séneca under Grant 22236/PDC/23. The work of Francisco-Jose Alvarado-Alcon was supported by the Spain's Ministry of Universities under Grant FPU22/00316.

**ABSTRACT** The Internet of Things (IoT) is gaining significant attention for its ability to digitally transform various sectors by enabling seamless connectivity and data exchange. However, deploying these networks is challenging due to the need to tailor configurations to diverse application requirements. To date, there has been limited focus on examining and enhancing the carbon footprint (CF) associated with these network deployments. In this study, we present an optimization framework leveraging machine learning techniques to minimize the CF associated with IoT multi-hop network deployments by varying the placement of the required gateways. Additionally, we establish a direct comparison between our proposed machine learning method and the integer linear program (ILP) approach. Our findings reveal that placing gateways using neural networks can achieve a 14% reduction in the CF for simple networks compared to those not using optimization for gateway placement. The ILP method could reduce the CF by 16.6% for identical networks, although it incurs a computational cost more than 250 times higher, which has its own environmental impact. Furthermore, we highlight the superior scalability of machine learning techniques, particularly advantageous for larger networks, as discussed in our concluding remarks.

**INDEX TERMS** Carbon footprint (CF), Internet of Things (IoT) networks, LPWAN, multilayer perceptrons, neural networks.

## I. INTRODUCTION

As the world transitions to an increasingly interconnected era, the IoT has emerged as the cornerstone of future technological advancements. The deployment of a wireless IoT network entails distributing numerous autonomous, cost-effective devices, referred to as end devices or nodes, across a designated area. These end devices are tasked with monitoring various environmental parameters and transmitting collected data wirelessly to one or more gateways, also known as sink nodes. Subsequently, the gateways relay this information to network servers for further processing tailored to specific applications, spanning domains like precision agriculture, home automation, smart cities, e-health, and Industry 4.0, among many

others. Extensive research has focused on evaluating the overall performance of this architecture, encompassing metrics such as economic cost [1], [2], [3], maximum lifetime [4], [5], packet loss [2], [3], [6], energy efficiency [1], [5], [6], [7], and the Human Toxicity Parameter [1]. However, the planning phase of IoT networks before deployment is crucial to determining the performance and functionality of the end application. This task is extremely intricate given the myriad of configurations and scenarios that can be envisioned based on the desired objectives.

It is imperative to acknowledge the substantial environmental impact associated with IoT network deployments. The term “carbon footprint” refers to the collective direct

and indirect release of carbon dioxide or other greenhouse gases resulting from a specific activity or product [8]. Recently, this notion has attracted the attention of the scientific world due to its usefulness as a standardized metric for gauging environmental impact. Consequently, the CF serves as a straightforward means to comprehend the environmental ramifications of a particular process. The Information and Communications Technology industry accounts for approximately 2–3% of Global Greenhouse Gas (GHG) emissions, and forecasts suggest continued growth in the future [9], [10]. Although the exact contribution of IoT to these emissions remains unclear due to limited data, its potential impact cannot be dismissed [10], [11]. Despite this, literature addressing the sustainability, and specifically, the carbon emissions of IoT networks remains scarce. Our article focuses solely on CO<sub>2</sub> emissions as the equivalence of other GHGs depends on the chosen timeframe used to calculate their global-warming potential. Once GHGs have been converted into their CO<sub>2</sub>equivalence, the carbon footprint is measured in CO<sub>2</sub>eq. Since our article exclusively considers CO<sub>2</sub>, both units can be considered the same.

With networks continuously growing in size and complexity, approaches to optimizing their performance are failing to keep up. As a result, the scientific community is looking to machine learning as a promising approach due to its scalability and adaptability. Since a large portion of IoT networks collect real-world data from specific locations [4], [7], the placement of end devices (or nodes) is closely tied to the areas requiring sensing. One prevalent method of connecting IoT nodes involves Low Power Wide Area Networks (LPWANs). Essentially, these networks use gateways or sink nodes to gather data transmitted by end devices and relay them to data centers via long-range technologies for further processing. The number and strategic positioning of these gateways significantly influence network performance.

In this article, we propose two different neural networks (NNs) that generate a gateway position for a given network reducing its emissions. The first NN is a multilayer perceptron (MLP), a multilayer feedforward NN that receives the position of the end devices as input and provides a position for the gateway that leads to a reduced CF. The second NN can be an MLP or a convolutional neural network (CNN) that receives information about an already designed IoT network and proposes an alternative position for the gateway that further reduces network emissions, which can be used to improve the results from the first MLP. This workflow allows for the fast and efficient reduction of IoT's carbon footprint.

These LPWANs may employ various technologies and specifications, including LoRa, NB-IoT, and SigFox [1], [2], [12]. LoRaWAN is a well-established open network protocol that can provide a coverage area of tens of kilometers using unlicensed frequencies and low-cost devices. It has attracted much attention [12], [13], and its operational process will be explored in Section III-A. However, LoRaWAN, the dominant standard protocol for LoRa, only supports

star-of-stars topologies. This limitation hampers the scalability of LoRa wireless networks as deploying multiple gateways to extend the physical network range proves costly and power-intensive [13], [14]. This article employs the LoRa physical layer but assumes the possibility of a multi-hop topology, where communication between end devices is also available. This multi-hop configuration, while not part of the standard LoRaWAN protocol, has attracted significant attention recently mainly due to better scalability than the LoRaWAN standard [6], [13], [15], [16], with some topologies having been empirically tested [14]. The LoRa Alliance also published a technical specification that describes a relaying mechanism that can “transport LoRaWAN frames bi-directionally between an end-device and a gateway” [17]. The model of the network is developed in Section IV. In this context, the main contributions of this article are summarized below:

- A first multilayer perceptron that optimizes the carbon footprint of multi-hop IoT networks **prior to deployment** by obtaining the gateway position using LPWAN technology and applying renewable energy sources.
- A second neural network that minimizes the carbon footprint of **already designed or deployed** multi-hop IoT networks by obtaining the gateway position using LPWAN technology and applying renewable energy sources, which can be used to improve the results from the first multilayer perceptron.
- An improved method to use the end device position as input for an NN, offering better results and greater scalability.

The rest of the article is organized as follows. Section II reviews some relevant works on IoT network optimization using traditional methods and machine learning, and it represents some studies measuring the carbon footprint of an IoT network. Background on the technologies is provided in Section III. Section IV formulates the problem, and Section V introduces the NN design and its use to simulate a network. Section VI discusses the results obtained. Finally, in Section VII, we conclude the article and outline potential directions for future research to build upon this work.

## II. RELATED WORK

Given the complexity of network planning and design, identifying an exact solution is often infeasible for large networks. In [18], the benefits of optimization in network planning were demonstrated as they maximized throughput by strategically placing gateways based on a grid configuration. But the network demands kept growing, and this approach fell short. Due to the complexity of the process, machine learning techniques and heuristics are usually used nowadays to search for a suitable approximation to the optimum.

Authors in [2] proposed PLACE, a methodology for optimizing the number and placement of LoRa gateways using the gap statistic method and Fuzzy C-Means, evaluated through QoS metrics. Simulations with 1000 Class-A devices showed

a 36% reduction in CAPEX and OPEX compared to the grid method. Similarly, [3] compared gateway positioning strategies using various algorithms, achieving cost reductions while maintaining Packet Delivery Ratio (PDR), though limited to single-hop topologies.

The study in [5] combined Fuzzy C-Means and Ant Colony Optimization (ACO) to optimize energy consumption in Underwater Wireless Sensor Networks. Fuzzy C-Means handled clustering and cluster head selection, while ACO determined the shortest route to the surface sink node. The method outperformed previous approaches in network lifetime, First Node Dies, and energy consumption. However, since the sink node was fixed at the water's surface, further improvements could be achieved by optimizing gateway placement alongside routing.

The study in [15] used mixed-integer non-linear programming to optimize gateway placement, spreading factor assignment, and power allocation relying on heuristics and greedy algorithms for scalability. While their approach yielded good results, our methodology could further enhance performance and coverage by leveraging a multi-hop topology.

In [19], authors reviewed machine learning techniques for optimizing various aspects of wireless mesh networks (WMNs), highlighting its suitability for real-time optimization. The study in [20] focused on network deployment, proposing an Artificial Intelligence driven framework for autonomous Wi-Fi deployment, where access points positions are adjusted based on environmental sensing. However, this iterative approach is impractical for networks with limited physical access since constantly moving them would be extremely laborious.

Section V in [12] reviewed various LoRa network planning strategies, primarily focusing on single-hop topologies without path optimization. Exceptions included [6], which examined relay installation for improved reliability, and [21], which proposed a modified slotted ALOHA protocol to reduce network delay using random-walk routing. Nevertheless, none of these studies addressed free gateway placement in a multi-hop environment.

Authors in [16] designed a multi-hop QoS aware LoRa topology with an ad-hoc routing protocol. Namely, they proposed a reinforcement learning for optimal transmission policy enforcement which learns the environmental parameters in order to allocate optimal transmission policies to improve the QoS in LoRa communication.

Concerning energy consumption by IoT end devices, there is extensive literature that includes various energy models. Studies such as [22], [23] have examined the functions of end devices and determined the electric current needed for each task. Manufacturers also provide online data [24]. Among these functions, transmission, reception, and idle states consume the most energy, while tasks like sensing and computing are relatively negligible.

Gateways have higher energy requirements due to their continuous listening. In [25], the authors investigated the power

**TABLE 1. Spreading Factor and Associated Settings for LoRa Devices Operating in the European 868 MHz Band**

| Spreading factor | Bitrate (bit/s) | Receiver sensitivity (dBm) |
|------------------|-----------------|----------------------------|
| SF7              | 5470            | -124                       |
| SF8              | 3125            | -127                       |
| SF9              | 1760            | -130                       |
| SF10             | 980             | -133                       |
| SF11             | 440             | -135                       |
| SF12             | 250             | -137                       |

consumption of LoRaWAN gateways. They proposed a simplified model for gateways, assuming they are always on with a fixed number of channels. In this model, power consumption is influenced by only two factors: the LoRaWAN gateway vendor and the backhaul technology.

Emissions from manufacturing IoT end devices were calculated in [11], but since their carbon footprint remains constant, these emissions were excluded from the ILP. Studies such as [26] and [27] estimated the carbon emissions of photovoltaic panels, considering factors like efficiency degradation and grid connection. Additionally, [28] highlighted that crystalline-silicon solar panels dominate 80% of the global market.

For batteries, [29] analyzed the life cycle emissions of lithium iron phosphate batteries, while [30] reported that lithium-ion cobalt-based batteries typically endure 500 recharge cycles. With this information, the CF can be predicted, as discussed in Section IV.

### III. BACKGROUND

#### A. LORA NETWORK

The environmental problem addressed in this study is based on LPWAN topologies, particularly focusing on the LoRaWAN specification. LoRaWAN technology operates on a one-hop radio system, facilitating long-range connectivity with minimal power consumption. Within a LoRaWAN network, a star-of-stars topology is employed that consists of three primary components: end devices, gateways, and a central network server. End devices, operating as sensors or actuators, communicate with the network server via gateway backhauls. Communication between end devices and gateways uses the LoRa physical layer, while gateway-to-network server communication employs an IP-based protocol stack. The spreading factor, a crucial parameter in LoRa, determines the bit rate and receiver sensitivity. Lower physical bit rates correspond to increased receiver sensitivity, as detailed in Table 1.

Despite certain limitations, such as single-packet-at-a-time and single-hop transmission, LoRaWAN is the current standard protocol for LoRa. To address the scalability issues inherent in LoRaWAN's topology, our proposed extended model considers the possibility of routing traffic not only directly to a gateway but also through intermediary end devices until reaching a gateway. However, to prevent potential network saturation, LoRaWAN imposes a maximum duty cycle:

for the selected frequency band and region, no end device can transmit packets for more than 1% of the time.

## B. NEURAL NETWORKS

An NN is a massively parallel distributed processor inspired by the human brain. It is made up of neurons, simple interconnected processing units with a propensity for storing experiential knowledge and making it available for use. To perform the learning process, a learning algorithm is used to modify the synaptic weights of the network in an orderly fashion to fulfill a specific design objective [31], [32].

Among different NN architectures, MLP is a multilayer feedforward network that stands out due to its versatility and ability to handle complex patterns. MLPs offer a balance between model complexity and computational efficiency, making them practical for a wide range of real-world scenarios. Their ability to learn nonlinear relationships and adapt to different input data [32], [33], [34] make them suitable for capturing the intricate spatial dynamics inherent in our problem. Conversely, CNNs use kernels to apply convolutions, which make them particularly useful for finding patterns in images and matrix-like data.

During network training, early stopping can be used to prevent overfitting, a common problem in machine learning tasks. Overfitting occurs when a model learns to memorize the training data excessively, leading to poor generalization performance on unseen data. Early stopping addresses this issue by monitoring the model's performance on a separate validation dataset during training. As training progresses, the model's performance on the validation set is continuously evaluated. If the validation error increases, training is halted prematurely. By stopping training at this point, the model's capacity to generalize to unseen data is preserved, thus mitigating the risk of overfitting and ensuring a more robust and reliable neural network model [34].

## C. POINT CLOUDS

Conventional machine learning algorithms were originally developed to handle structured data types, such as sequences and images. However, point clouds, which are unordered sets of vectors, have gained significant attention in the past decade as a form of input data for machine learning models. A unique challenge posed by point cloud data is the need for symmetry, meaning that the model must remain invariant to the order of the input. There are three main approaches to address this: sorting the input into a fixed canonical order, enhancing the training data with various permutations, or employing a symmetric function to aggregate information from each point [35]. Building on the third method, several specialized networks have been created for point cloud data, particularly focusing on 3D point sets, such as PointNet [35] and PointMLP [36]. While these studies target tasks like classification and segmentation, where local geometric details are critical, our network is specifically designed for regression, with an emphasis on the entire network's structure.

In our work, we use the data augmentation through permutation approach since it allows for the seamless combination of both vector and non-vector data as input while maintaining a simple and efficient network architecture. This method achieves a high degree of symmetry with a relatively small number of permutations, although for a network with  $n_{inputs}$  inputs,  $n_{inputs}!$  permutations could be applied. However, as seen in Section VII, increasing the number of permutations indiscriminately is not necessarily beneficial, but it improves network behavior up to a certain point.

## IV. PROBLEM FORMULATION

Based on the behavior of LoRa and building on our previous work in [37], we developed a mathematical model for the multi-hop network. This model serves as an ILP framework for generating training data and, later, as a method for calculating the CF of a designed network. We conceptualize the IoT network as a collection, denoted as  $\mathcal{N}$ , of  $n = 30$  interconnected end devices randomly distributed (i.e., independently and uniformly) within a  $3 \times 10^5 \text{m}^2$  square area. The detailed model of the network's behavior and the ILP are provided below.

The goal is to minimize the CF by optimizing gateway placement and link establishment. To achieve this, we define a set  $\mathcal{P}$  of size  $p$ , consisting of a finite number of potential gateway locations points (simply referred to as 'points' for clarity), and introduce a variable vector  $G$  of the same dimension  $p$ . One of the three optimization variables is this logical vector  $G$ , where element  $G_k$  is set to one if a gateway is present at point  $k$  and zero if it is absent.

All nodes transmit directly to either a network gateway or another end device. Gateways serve as the sole traffic sinks, as traffic must go through a gateway to reach the central network server. Consequently, when end device  $i$  transmits data to end device  $j$ , the latter must then relay it to a different device, which could be either a gateway or another end device. This means two additional optimization variables must be added. The first one, denoted as  $L$ , is represented by a logical matrix where  $L_{i,k}$  indicates the traffic units transmitted by end device  $i$  to gateway  $k$ . It assumes a value of 0 when there is no connection between them. We define a 'unit of traffic' as the amount of traffic generated by any end device, assuming that an equal amount of traffic originates from each one. Connections are limited to a maximum distance equal to the transmission range, with the transmission power set at 14dBm, adhering to the maximum allowable value in the LoRa specification for the European frequency band of 868 MHz [24]. However, the spreading factor can be adjusted to minimize energy consumption or increased for longer transmission ranges.

The last optimization variable,  $LN$ , is a logical matrix where  $LN_{i,j}$  represents the units of traffic transmitted from end device  $i$  to end device  $j$ . This representation allows for direct packet exchange between end devices, making a significant expansion from our prior research [37]. Similar to  $L$ , it

takes on a value of 0 when no connection exists between these end devices. The conditions for link establishment remain consistent with those outlined earlier. However, gateways are always listening, requiring no further action, whereas end devices are not listening by default. Thus, a receiving window must be programmed in the receiving end device to facilitate this communication. This differs from basic LoRaWAN behavior.

In an ideal scenario, perfect synchronization among devices results in a reception window that lasts only as long as the transmission. Conversely, in the worst-case scenario of unsynchronized devices, the receiver end device must remain continuously active, leading to increased energy consumption and a larger carbon footprint. For simplicity, this article assumes the perfect synchronization scenario. Even when no level of synchronization is achievable, only end devices that need to forward data must always be listening. Configuring these end devices to always listen increases the presented values of the CF by 5.3% on average. In this case, one could expect that the optimal topology would vary, reducing the number of end devices that must be listening, but an MLP can nonetheless be trained with this adjusted data.

Furthermore, in any practical multi-hop deployment, the necessity of a routing protocol arises to establish the next hop for devices. In our study, routing decisions are dictated by the optimization problem. As we have not designated a specific routing protocol, we have presumed that each end device possesses pre-programmed knowledge of its subsequent hop based on the outcomes of the ILP.

Both gateways and end devices are outfitted with photovoltaic panels to power their operation continuously, supplemented by batteries with enough capacity to sustain 24-hour operation. The photovoltaic panel is designed to ensure continuous power provision throughout the year, considering the total number of hours of solar incidence. Combining the battery and the panel enables seamless device operation round the clock. For the sake of simplicity in our analysis, we do not consider prolonged periods without sunlight beyond a single night or variations in daylight across seasons.

The CF can be calculated by summing up the carbon emissions associated with manufacturing the photovoltaic panel and the battery used in the end devices and gateways. It is noteworthy that end-of-life management of photovoltaic panels is an evolving field, requiring further research and development, as discussed in [28]. Therefore, assuming the average life expectancy of photovoltaic panels to be approximately 25 years [26], [27], [28], we omit the emissions associated with their end-of-life.

To characterize the complete carbon emissions of a generic device, which can be either an end device or a gateway, expression (1), extracted from [37], allows the computation of  $K_p$ , representing the carbon footprint of a photovoltaic panel per the kWh it generates annually. In this equation,  $C_p$  denotes the carbon footprint generated during the production of a 1kW photovoltaic panel.  $T_p$  and  $T_i$  indicate the panel's lifespan

TABLE 2. Problem Parameters

| Parameter     | Description                                                                                                                                                      |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $C_b$         | The carbon footprint generated when making a Li-ion rechargeable battery of 1 kW h in CO <sub>2</sub> eq.                                                        |
| $C_p$         | The carbon footprint generated when making a photovoltaic panel of 1 kW in CO <sub>2</sub> eq.                                                                   |
| $C_T$         | The total carbon footprint of the network in CO <sub>2</sub> eq/year.                                                                                            |
| $E_G$         | Energy consumption of a gateway for one year in kWh/year.                                                                                                        |
| $E_{i,k}$     | Energy consumed by the link between end device $i$ and point $k$ for one year and one unit of traffic in kWh or $\infty$ if the link cannot be established.      |
| $EN_{i,j}$    | Energy consumed by the link between end device $i$ and end device $j$ for one year and one unit of traffic in kWh or $\infty$ if the link cannot be established. |
| $G_k$         | Optimization variable representing whether a gateway is placed in point $k$ .                                                                                    |
| $K_b$         | Carbon footprint of 365 recharge cycles of a battery that provides a total of 1 kW h in CO <sub>2</sub> eq/kWh.                                                  |
| $K_p$         | Carbon footprint of a photovoltaic panel for every kW it produces in a year in CO <sub>2</sub> eq/kW.                                                            |
| $L_{i,k}$     | Optimization variable representing whether a link between end device $i$ and point $k$ exists and the number of traffic units it carries.                        |
| $LN_{i,j}$    | Optimization variable representing whether a link between end devices $i$ and $j$ exists and the number of traffic units it carries.                             |
| $\mathcal{N}$ | Set of end devices in the IoT network.                                                                                                                           |
| $n$           | Number of end devices in the IoT network.                                                                                                                        |
| $\mathcal{P}$ | Set of end potential gateway locations in the IoT network.                                                                                                       |
| $p$           | Number of points where a gateway could be placed in the IoT network.                                                                                             |
| $PTx_{i,k}$   | Percentage of time end device $i$ has to be transmitting to send one unit of traffic to point $k$ .                                                              |
| $PTNx_{i,j}$  | Percentage of time end device $i$ has to be transmitting to send one unit of traffic to end device $j$ .                                                         |
| $R$           | The number of battery recharge cycles.                                                                                                                           |
| $T_i$         | Number of hours of solar incidence in a year in h/year.                                                                                                          |
| $T_p$         | Lifetime of a photovoltaic panel.                                                                                                                                |
| $\eta_p$      | The efficiency of the solar panel, $0 < \eta_p < 1$ .                                                                                                            |

and the total hours of solar incidence, respectively, while  $\eta_p$  represents the panel's efficiency. All the parameters involved in these equations are specified in Table 2.

$$K_p = \frac{C_p}{T_p T_i \eta_p} \quad (1)$$

Expression (2) quantifies the CF associated with the daily use of a battery delivering a total of 1 kWh annually, designated as  $K_b$ . In this equation,  $C_b$  represents the emissions during the production of a battery, contingent on its capacity, and  $R$  symbolizes recharge cycles it can undergo.

$$K_b = \frac{C_b}{R} \quad (2)$$

Consequently, the overall emissions from a generic device with an annual energy consumption of 1 kWh comprise the sum of  $K_p$  and  $K_b$ .

The next step involves modeling end device energy consumption due to radio communication.  $E_{i,k}$  denotes the energy consumption for one year and one unit of traffic (measured in kWh) associated with the link between the  $i$ -th end device and the  $k$ -th gateway. This consumption extends to  $EN_{i,j}$ , which similarly characterizes the energy consumption for one year and one unit of traffic between the  $i$ -th and  $j$ -th end devices.

Taking these factors into account, the expressions for end device and gateway energy consumption, expressions (3) and (4), respectively, can be derived. As previously mentioned, the number of end devices is constant. Moreover, since the energy consumption caused by the idle state and non-radio components of the end devices is invariable, it is also disregarded during the ILP. Therefore, energy consumption of end devices can be computed according to expression 3, while expression 4 measures the energy consumption of the gateways.

$$\sum_{i=1}^n \sum_{k=1}^p L_{i,k} E_{i,k} + \sum_{i=1}^n \sum_{j=1}^n LN_{i,j} EN_{i,j} \quad (3)$$

$$\sum_{k=1}^p G_k E_G \quad (4)$$

By integrating expressions (1) to (4), we can derive the network's overall carbon footprint, denoted as  $C_T$ , as outlined in (5). This revised equation quantifies the carbon emissions in kg per year ( $\text{CO}_2\text{eq/year}$ ) and serves as the optimization objective of the problem.

$$C_T = \left( \frac{C_p}{T_p T_i \eta_p} + \frac{C_b}{R} \right) \left( \sum_{i=1}^n \sum_{k=1}^p L_{i,k} E_{i,k} + \sum_{i=1}^n \sum_{j=1}^n LN_{i,j} EN_{i,j} + \sum_{k=1}^p G_k E_G \right) \quad (5)$$

Thus, the problem of minimizing the carbon footprint with known gateway positions is as follows:

$$\min_{G_k, L_{i,k}, LN_{i,j}} C_T \quad (6a)$$

$$\text{s.t.} \quad \sum_{k=1}^p L_{i,k} + \sum_{j=1}^n LN_{i,j} - \sum_{j=1}^n LN_{j,i} = 1, \quad \forall i \in \mathcal{N}, \quad (6b)$$

$$\sum_{i=1}^n L_{i,k} < \alpha G_k, \quad \forall k \in \mathcal{P}, \quad (6c)$$

TABLE 3. Problem Parameter Values

| Parameter | Value                               | References |
|-----------|-------------------------------------|------------|
| $E_G$     | 153.6504 kWh/year                   | [25]       |
| $T_p$     | 25 years                            | [26]–[28]  |
| $C_p$     | 2024.823 $\text{CO}_2\text{eq/kWh}$ | [26], [27] |
| $C_b$     | 0.0279 $\text{CO}_2\text{eq/kWh}$   | [29]       |
| $R$       | 500 recharge cycles                 | [30]       |
| $T_i$     | 1400 h                              | [25]       |
| $\eta_p$  | 0.8                                 | [26], [27] |

$$\sum_{k=1}^p L_{i,k} PT x_{i,k} + \sum_{j=1}^n LN_{i,j} PTN x_{i,j} \leq 0.01, \quad \forall i \in \mathcal{N}, \quad (6d)$$

$$L_{i,k} \in \mathbb{N}_0, \quad \forall i \in \mathcal{N}, \quad \forall k \in \mathcal{P}, \quad (6e)$$

$$LN_{i,j} \in \mathbb{N}_0, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{N}, \quad (6f)$$

$$G_k \in \{0, 1\}, \quad \forall k \in \mathcal{P}. \quad (6g)$$

Table 2 outlines the parameters used in the ILP, and Table 3 shows the values given to the problem constants, matching the values used in our previous research [37] with their corresponding relevant references. Expressions (6b) to (6g) represent the constraints for the carbon footprint problem:

- Expression (6b) denotes that every end device transmits one more unit of traffic than it receives. Given that the traffic generated by a single end device is one unit of traffic, they must forward all the packets they receive. And since every end device is transmitting more packets than it receives, only the gateways can behave as sinks.
- Expression (6c) enforces that every link to a potential gateway location point is made to a position where a gateway is located, therefore, there is no connection to an empty point, and no gateway receives more than  $\alpha$  connections.
- Expression (6d) ensures that no end device can transmit during more than 1% of the time.
- Expressions (6e) and (6f) model that problem variables  $L_{i,k}$  and  $LN_{i,j}$  can take zero or a natural value.
- Expression (6g) models that problem variable  $G_k$  can take zero or one logical value.

## A. STREAMLINED PROBLEM

In this article, an NN is used to provide the location of the gateway. However, this information alone is not sufficient to consider that a network is optimized. Link establishment is another crucial parameter for proper network operation and has a great impact on its performance. However, once the location of the gateway is known, link establishment is much easier. We have modified the previously proposed ILP to minimize the CF of a network when the placement of the gateway has been solved, so  $G$  has a single element (i.e.,  $p = 1$ ), with

value  $G_1 = 1$ . Thus, the simplified problem of minimizing the carbon footprint with known gateway positions is as follows:

$$\min_{L_{i,k}, LN_{i,j}} C_T \quad (7a)$$

$$\text{s.t.} \quad \sum_{k=1}^p L_{i,k} + \sum_{j=1}^n LN_{i,j} - \sum_{j=1}^n LN_{j,i} = 1, \forall i \in \mathcal{N}, \quad (7b)$$

$$\sum_{k=1}^p L_{i,k} PTx_{i,k} + \sum_{j=1}^n LN_{i,j} PTN_{i,j} \leq 0.01, \forall i \in \mathcal{N}, \quad (7c)$$

$$L_{i,k} \in \mathbb{N}_0, \forall i \in \mathcal{N}, \forall k \in \mathcal{P}, \quad (7d)$$

$$LN_{i,j} \in \mathbb{N}_0, \forall i \in \mathcal{N}, \forall j \in \mathcal{N}. \quad (7e)$$

However, not all of the proposed problems are solvable. For a given gateway position, it may be impossible to provide connectivity to all the end devices while complying with the constraints. The network surface and end device density have been chosen so that most cases are solvable but still challenging. For networks in which the ILP is infeasible, we increase the number of gateways from 1 to 2 (i.e.,  $p = 2$ ), place them in a grid, and try to solve the ILP again. If there is still no solution, we increase the number of gateways to a  $2 \times 2$  grid ( $p = 4$ ) and then to a  $3 \times 3$  grid ( $p = 9$ ) until the network restrictions can be satisfied. In this way, the problem always results in an operational network deployment with an associated CF.

## V. NEURAL NETWORK DEFINITION

Two different NNs were defined during different stages of this project. The data were generated using the ILP from Section IV. The 21,000 samples generated were divided into three subsets called Training ( $\sim 67\%$ ), Validation ( $\sim 14\%$ ), and Testing ( $\sim 19\%$ ). Every input and output were normalized between 0 and 1. The neuron activation function was fixed to a hyperbolic tangent function for all the neurons in the hidden layer (a historically prevalent bounded activation function that often perform wells in shallow networks [38]), whereas the output neurons used a linear activation function. The selected learning algorithm was Levenberg–Marquardt, and the maximum number of epochs was large enough for early stopping to always be the termination criteria: the training process is halted when the network’s performance fails to improve for six consecutive epochs.

The supervised NN is trained through pairs of inputs (data about the network) and outputs (coordinates for a gateway). This implies that a constant number of gateways can be placed using this NN, so all the topologies used for training have

a single gateway. The cases that could not be solved with a single gateway using the ILP from Section IV (0.13%) have been removed from the training dataset.

To extend the dataset used for network training, data augmentation techniques have been applied to both the training and validation subsets. Initially, the axes (X, Y, and both simultaneously) of the network have been flipped, a common data augmentation technique in image-based tasks, resulting in new inputs. Additionally, since the end device positions form a point cloud (non-vector data), altering their order as inputs does not affect the output. This property enabled us to generate multiple distinct samples from a single network topology. For this second technique, we expanded the dataset size by a factor of 10 by permuting the order of the end devices. Consequently, both the training and validation subsets were extended by a factor of 40, while the test subset remained unchanged.

## A. NEURAL NETWORK FOR PLANNING

The first MLP receives the positions of the end devices as input and generates the gateway position as output. The position of the end devices can be codified in two different ways. The neural network can be given the Cartesian coordinates of every end device as an input. Alternatively, the network area can be divided into Spatial Sampling Regions (SSRs), and the number of end devices in every SSR is the input.

We define a Spatial Sampling Region as a region created by dividing the network area into partially overlapping circles, with the count of end devices within each circle serving as input data for the network. To ensure systematic coverage, the centers of these circles are uniformly spaced at distance  $r$ , corresponding to the radius of the circles. A circle is centered in the middle of the network area and, from that point, the grid of circles is expanded to the limits of the studied area with no center exceeding it. By leveraging this structured spatial sampling and neural network processing, the proposed methodology aims to achieve a nuanced understanding of the network topology.

Using SSRs as input allows the model to handle varying numbers of end devices without requiring retraining. When the coordinates of every end device are used as input for the NN, the MLP has been trained with a constant number of end devices, as there are two inputs per end device (or three inputs per end device if using three dimensions). However, if end device placement is only used through SSRs, the number of inputs remains constant for different end device counts and, therefore, the same MLP can be used for networks with different end device densities. However, data augmentation via permutation is not applicable if only the SSRs are used as input since changing the order of the end devices does not modify the inputs or outputs of the network. This leads to three possible neural network configurations based on different input combinations.

### B. NEURAL NETWORK FOR REFINEMENT

The second NN is provided with more intricate information regarding pre-existing network designs and suggests alternative gateway placements aimed at reducing the CF. This NN also receives the position of the end devices, along with the weight of every end device and the total CF of the network. The weight of an end device is calculated as the logarithm of the number of traffic units it transmits, so end devices that carry more traffic have more weight.

To obtain the dataset to train this network, the results of the best MLP from the previous stage have been used. Specifically, we have simulated the proposed networks using the gateway positioning provided by the first MLP and the results from the simplified ILP defined in Section IV-A.

Following our proposed workflow, this second NN can be used before the deployment to improve the results of the first MLP. We applied the first MLP to obtain the position of the gateway for the given end devices and, using the streamlined ILP, we calculated the optimal link establishment. Fig. 1 (top) contains an example network after this first optimization step. Then, for this new topology, we applied the second MLP, which gives us a better gateway position. Finally, we used the streamlined ILP again and obtained the optimized network, as shown in Fig. 1 (bottom). Alternatively, we proposed a CNN as this second NN. However, as it is discussed in Section VI, it proved less efficient, so our study focuses mainly on MLPs. The classical approach would comprise solving the ILP as described by (6a)–(6g), which, for the same end devices, would result in the network depicted in Fig. 2.

This NN can also be used to improve the CF of an already deployed network. It receives the end device positions, the traffic each end device carries, and the emissions of the network, and it offers a new placement for the gateway that further decreases the carbon emissions if possible. Therefore, it has the potential to help reduce the environmental factor of the numerous IoT networks already installed.

### C. SYMMETRY

The information in an MLP is propagated by multiplying the inputs by their respective weights and adding them at the next node. As a result, if two inputs are multiplied by the same or nearly identical weights, swapping these inputs will yield the same outcome. Applying this principle, if the weights associated with the 30 X-coordinate inputs and the 30 Y-coordinate inputs are very similar, the network is expected to produce similar outputs. The symmetry achieved by the proposed models will be discussed in the next Section.

## VI. RESULTS

For the scenarios comprising the test subset, the average CF obtained when the network emissions were minimized using the ILP from Section IV is  $11.149\text{CO}_2\text{eq}$ , and 99.9% of them were solved with a single gateway. Thus, we will consider this as the gold standard. In contrast, the CF when the gateway is simply placed in the center of the network

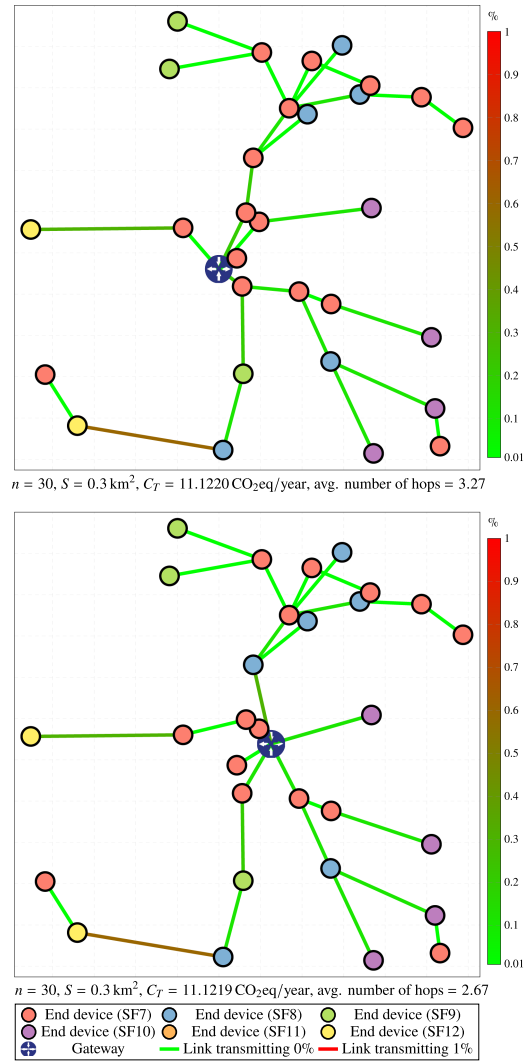


**FIGURE 1.** Designed network after the gateway position has been calculated with (top) the first MLP or (bottom) the complete proposed approach. Link establishment has been carried out according to the streamlined ILP from Section IV-A.

**TABLE 4.** Average Carbon Footprint When a Network is Deployed by Placing the Gateway According to the First MLP

| Inputs                       | Number of hidden nodes | Radius of the SSRs | CF ( $\text{CO}_2\text{eq}$ ) |
|------------------------------|------------------------|--------------------|-------------------------------|
| Cartesian coordinates        | 14                     | —                  | 13.395                        |
| Spatial Sampling Regions     | 16                     | 40.8 m             | 13.128                        |
| Cartesian coordinates & SSRs | 20                     | 50 m               | 13.082                        |

area is  $13.368\text{CO}_2\text{eq}$ , and 91.2% of them were solved with a single gateway. Placing the gateway in the center is a normal practice, so this value will represent the CF obtained when no optimization is employed for gateway placement. Table 7 contains a complete comparison of these results. In both cases, we are assuming that link establishment is carried out according to the ILP that minimizes the network emissions. In reality, link establishment in non-optimized networks is likely

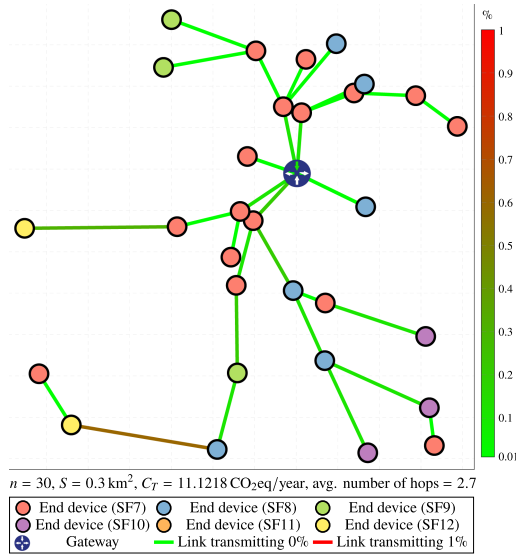


**FIGURE 2.** Designed network after the gateway position and link establishment have been calculated using the ILP from Section IV.

**TABLE 5.** Average Carbon Footprint When a Network is Enhanced by Moving the Gateway According to the Second MLP

| Inputs                       | Number of hidden nodes | Radius of the SSRs | CF (CO <sub>2</sub> eq) |
|------------------------------|------------------------|--------------------|-------------------------|
| Cartesian coordinates        | 30                     | —                  | 13.124                  |
| Cartesian coordinates & SSRs | 22                     | 50 m               | 12.465                  |

**TABLE 6.** Average Carbon Footprint When a Network is Enhanced by Moving the Gateway According to the Second CNN. The Inputs are Both Cartesian Coordinates & SSRs

| Architecture                              | Number of filters | Kernel size | Hidden nodes | CF (CO <sub>2</sub> eq) |
|-------------------------------------------|-------------------|-------------|--------------|-------------------------|
| With hidden dense layer and no padding    | 6                 | 3×3         | 16           | 13.240                  |
| With hidden dense layer and same padding  | 10                | 3×3         | 18           | 13.232                  |
| Without hidden dense layer and no padding | 12                | 5×5         | —            | 13.196                  |

to differ from the optimum and, therefore, the achievable improvement is better than that shown in this article.

For the first MLP, different numbers of hidden nodes and radii for the SSRs (if applicable) were tested. The parameters offering the best solution for every input combination are included in Table 4, along with the CF obtained with said MLP for the test dataset. It is not surprising that offering the most information to the NN offers the best outcome at the price of a higher number of hidden nodes, and therefore, an increased computation cost.

Then, for the second MLP, different numbers of hidden nodes and radii for the SSRs (if applicable) were tested, and the parameters offering the best solution for each input combination have been included in Table 5, along with the CF obtained with said MLP for the test dataset. It is necessary to feed the neural network with end device positions using both

data structures to achieve an improvement from the previous stage.

Alternatively, the CNN processes the same data as the second MLP as three distinct inputs: the end devices per SSR as an  $11 \times 11$  matrix, the end device coordinates and weights as a  $30 \times 3$ , and the CF as a scalar feature. For the end device coordinates, the kernel size was fixed to  $1 \times 3$  with no padding, and the number of kernels was the same for both matrix inputs. These outputs are then flattened, concatenated, and optionally passed through a hidden dense layer. Finally, they pass through a dense layer with 22 nodes and the output layer. The parameters used (22 hidden dense nodes and 50m radius SSRs) were the best ones found for the second MLP. The CNN was tested with different kernel sizes and number of kernels, among other parameters. As seen on Table 6, the optimal configuration used  $12 \times 5$  filters for the SSRs, the same number of  $1 \times 3$  kernels for the coordinates and no hidden dense layer, but the obtained CF,  $13.196\text{CO}_2\text{eq}$ , was significantly worse than the offered by the MLP, leading to its dismissal.

After using input permutation as a data augmentation technique, the average standard deviation of the weights connecting the coordinates to each hidden layer node is 0.0103 for the first MLP and less than 0.002 for the second MLP. These values are over 100 times lower than the standard deviation of the non-permutable inputs, which indicates minimal variation in the output when the input entries are permuted.

The CF can be minimized when using the first MLP with the selected parameters during network planning. For the test dataset, the root mean square error (RMSE) was 0.0145 with normalized outputs, 92.4% of the cases could be resolved with a single gateway, and the average CF obtained when the network was deployed following this method was  $13.082\text{CO}_2\text{eq}$ . This outcome can be further improved by applying the second proposed NN, which reduces the emissions to  $12.465\text{CO}_2\text{eq}$ . The RMSE obtained for this second MLP is 0.0129 for the test dataset, and 94.9% of the cases were addressed with a single gateway,

As observed in Table 4, the MLP works better when the end device placement is formatted in SSRs rather than coordinates. Since the SSRs have a great impact on the performance of the network, some error in the results proportional to the SSRs' size can be expected. To find a better solution, we propose that not only the exact solution offered by the MLP should be considered. Instead, the gateway could also be placed adding a horizontal or vertical offset equal to the diameter used for the SSRs. Therefore, five possible locations for the gateway are considered, which greatly improves the results of the problem while increasing the computation time by approximately the same order of magnitude as the original runtime. The average CF obtained after selecting the best case for every network in the test subset is  $12.765\text{CO}_2\text{eq}$  for the first MLP and  $11.547\text{CO}_2\text{eq}$  for the second one. The scenarios resolved with a single gateway where 93.8% and 98.7%, respectively. This last value is referred to as "Multilayer Perceptron" in Table 7.

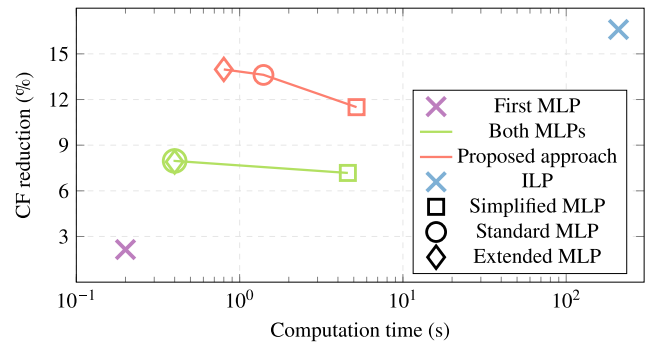
**TABLE 7.** Average Carbon Footprint Obtained by the Different Models With Their Average Required Time

| Model                            | Time per network      | Training time     | Training samples | CF (CO <sub>2</sub> eq) | Cases solved with 1 gateway | CF reduction per computation time (%/s) |
|----------------------------------|-----------------------|-------------------|------------------|-------------------------|-----------------------------|-----------------------------------------|
| Integer Linear Problem           | 209.0 s               | —                 | —                | 11.149                  | 99.9%                       | 0.08                                    |
| Multilayer Perceptron            | 0.2 s + 0.2 s + 1.0 s | 126.5 h + 93.8 h  | 560k             | 11.547                  | 98.7%                       | 9.73                                    |
| Simplified Multilayer Perceptron | 0.2 s + 4.4 s + 0.6 s | 126.5 h + 9.8 h   | 560k/56k         | 11.831                  | 97.7%                       | 2.21                                    |
| Extended Multilayer Perceptron   | 0.2 s + 0.2 s + 0.4 s | 126.5 h + 574.3 h | 560k/3360k       | 11.500                  | 98.8%                       | 17.47                                   |
| Gateway in the center            | 11.1 s                | —                 | —                | 13.368                  | 91.2%                       | —                                       |

It should be noted that there is no direct correlation between the RMSE of the MLP and its ability to predict a good position for the gateway. For example, when testing different radii and hidden nodes for the first MLP with only SSRs as inputs, the one that offered the lowest RMSE was the MLP with 28 hidden nodes and  $r = 78$ , but the carbon footprint obtained was  $13.306\text{CO}_2\text{eq}$ , worse than the  $13.128\text{CO}_2\text{eq}$  shown in Table 4. This may be due to the fact that the discrepancy between the anticipated position and that provided by the MLP does not inherently signify a negative outcome. There may be multiple optimal and near-optimal solutions for any given network, and they might not be close to one another. Different error functions, such as Mean Absolute Error, the distance between the expected solution and the obtained solution, Huber Loss, the correlation coefficient, and its square root, were not helpful either. The parameters allowing us to obtain the minimum error function are not the same as those that minimize the CF. Consequently, simulating the obtained positions is the best way to assess the CF reduction that can be obtained when using an MLP.

We have assessed the CF reduction achieved by the second MLP when used in conjunction with the first MLP during the deployment of an IoT network following the proposed workflow in Section V-B. However, it can also be used to optimize other networks that have not been planned with the proposed method and offer improved results. We have selected the scenarios without optimization mentioned at the beginning of this Section, as the base CF. The average emissions of the network were  $13.368\text{CO}_2\text{eq}$  (see Table 7), but after moving the gateway according to the MLP, it was reduced to  $12.431\text{CO}_2\text{eq}$ . If we also consider the precision error induced by the radius of the SSRs, the achieved CF is  $11.544\text{CO}_2\text{eq}$ , and 98.6% of the scenarios were solved with a single gateway. Therefore, a 13.6% carbon reduction can be obtained when unoptimized networks are modified according to the second MLP.

The proposed methodology in this article has shown the potential to reduce the CF for both new and already deployed networks. It has been proven that it can offer similar results to the ILP, considered as the ground truth. Even if the results are not as optimal, the amount of time and computational resources required by the ILP to find the solution can be more than 250 times longer. Table 7 contains a complete comparison of the final results obtained for the different approaches. The ILP is compared to our proposed approach (using both MLPs sequentially, followed by the analysis of the offset) referred to as “Multilayer Perceptron.” We also compare this

**FIGURE 3.** Carbon footprint reduction per computation time using different methods. For solutions where MLPs of different sizes have been considered, these have been grouped in the graph as a single plot.

approach to a simplified second stage multilayer perceptron (without modifying the first MLP) trained with ten times fewer samples and an extended MLP that uses six times more samples through the use of permutation. We have also added the gateway in the center approach, which is considered the base point from which the CF reduction is calculated.

Table 7 shows the time the proposed methods require to find a solution for a single network, the training time for the NNs, number of training samples (if applicable), and the average CF estimated for each one. The time per network for the MLPs is computed as the sum of the time needed to solve the first MLP, the second MLP, and the analysis of the offset error, whereas the training time for the MLP is computed as the sum of the training for both NNs. It can be observed that the ILP offers a 16.6% CF reduction, 3 percentage points more than the 13.6% reduction achieved by the method developed in this article, at the cost of a resolution time 150 times higher, which entails greater environmental impact from the optimization process. The simplified and extended MLPs require, respectively, ten times less and six times more time to train than the second MLP, and they achieve a CF reduction of 11.5% and 14.0%, respectively.

Table 7 also contains the carbon reduction attained per second of computation time, which is represented graphically in Fig. 3, where the ILP and the different steps followed during our approach are compared in terms of carbon reduction (with respect to the gateway in the center) and computation time. It can be seen that using only the first MLP offers little reward, as discussed in Section V-A. Using both MLPs offers a more significant reduction, and the improvement is even greater with our proposed approach, which includes the

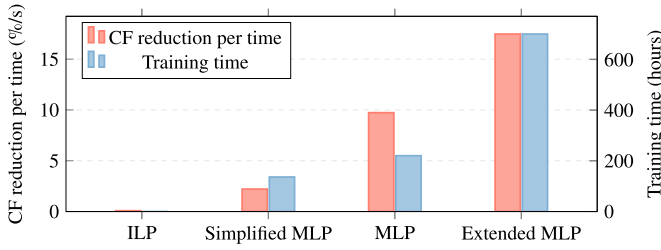


**FIGURE 4.** Carbon footprint reduction per computation time for each method opposed to the training time of both MLPs together. Note that no training is required for the ILP.

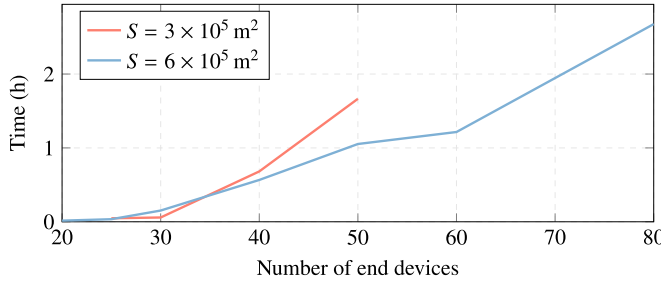


**FIGURE 5.** Time required to solve the ILP that minimizes the CF for networks of varying size.

offset analysis. Although the ILP offers the best results in terms of CF reduction, it does so at the cost of longer computation time, which entails its own computation cost. Our proposed approach results in the best option for CF reduction per computation time—once the neural networks have been trained. Additionally, the more complex MLPs offer the best behavior in terms of CF reduction and computation time. A complete comparison of the three training sizes for our proposed approach is offered in Fig. 4. It can be seen that, as expected, the models requiring the most effort to train are the ones that offer the best CF reduction per computation time.

Finally, to contextualize the relevance of our proposed method, we have analyzed ILP resolution time as a function of network size. The network size used in this article (30 end devices in  $3 \times 10^5 \text{ m}^2$ ) results in still manageable times for the ILP. However, only by increasing the number of end devices by 20, the time needed to solve the ILP increases by a factor of 30. Fig. 5 shows the average solving time as the number of end devices increases for two different configurations,  $3 \times 10^5 \text{ m}^2$  (the one used in this article) and  $6 \times 10^5 \text{ m}^2$ . The rapid growth of the ILP required time is undeniable in both cases, making the machine learning approach more attractive for larger networks. Comparatively, the number of weights of a MLP is  $(n_{inputs} + 1) \times n_{hidden} + (n_{hidden} + 1) \times n_{output}$ . Doubling the number of end devices increases the number of inputs by 33% for the first MLP and 42% for the second, resulting in the number of weights increasing from 3682 to 4882 and 4732 to 6712, respectively. Consequently, the time expected for a single forward pass grows at a linear rate relative to the number of end devices.

## VII. CONCLUSION

The optimal planning of IoT networks poses a multifaceted challenge due to the flexibility needed to meet specific application goals. In this study, we introduce and advocate an optimization framework aimed at minimizing the carbon footprint in IoT network deployments through the application of machine learning techniques. This proposed workflow implies the sequential use of two different MLPs. Additionally, we conduct a comparative analysis between the machine learning approach and the more conventional ILP optimization method. Our methodology achieves a 14.0% CF reduction, a reasonable improvement compared to the 16.6% obtained with the ILP, the ground truth.

Although the MLPs do not achieve the same level of carbon minimization as the ILP, they offer similar values with distinct advantages. As shown in Section VI, MLPs can calculate the position of the gateway and the estimated CF in seconds, facilitating the rapid analysis of a variety of scenarios. This can be crucial in dynamic environments, e.g., if we want to analyze the behavior of a network with moving end devices. Another of the greatest advantages of machine learning techniques over traditional optimization is its scalability: the amount of time and computational resources required to find an appropriate position for the gateway using ILP increases rapidly with network size. Note that, in our previous study [37], with a simpler optimization problem, the larger networks proved too computationally complex. We have found similar limitations when trying to double the number of end devices for the networks used in this work. All things considered, this machine learning approach offers the best results in terms of CF reduction per computation time, which will improve for larger networks.

The use of machine learning has proven a reliable solution for solving complex problems. However, this article only considers the placement of a single gateway. Future work will extend the model to multiple gateways, thereby enabling problems to be solved in larger networks where the advantages of machine learning will be even more remarkable. A potential approach could involve combining this study with methods discussed in the Related Work, such as dividing end devices into clusters using algorithms like Fuzzy C-Means. Each cluster, representing a group of end devices that can be covered by a single gateway, could then be optimized using the methodology proposed in this article. Furthermore, there are additional venues of machine learning approaches that merit exploration. Future efforts will be directed towards more complex deep learning architectures (e.g., graph neural networks and transformers) and non-supervised learning methods (e.g., reinforcement learning).

## REFERENCES

- [1] M. Rady, J.-P. Georges, and F. Lepage, “Can energy optimization lead to economic and environmental waste in LPWAN architectures?,” *ETRI J.*, vol. 43, no. 2, pp. 173–183, 2021.
- [2] N. Matni, J. Moraes, D. Rosário, E. Cerqueira, and A. Neto, “Optimal gateway placement based on fuzzy C-means for low power wide area networks,” in *Proc. IEEE Latin-Amer. Conf. Commun.*, 2019, pp. 1–6.

- [3] C. N. da Silva, P. F. F. de Abreu, J. D. F. da Silveira, and J. V. dos R., "Estimating the number of gateways through placement strategies in a LoRaWAN network," in *Proc. 12th Braz. Symp. Comput. Syst. Eng.*, 2022, pp. 1–6.
- [4] R. Asorey-Cacheda, A. J. García-Sánchez, F. García-Sánchez, J. García-Haro, and F. J. González-Castaño, "On maximizing the lifetime of wireless sensor networks by optimally assigning energy supplies," *Sensors*, vol. 13, no. 8, pp. 10219–10244, 2013.
- [5] H. Panchal and S. Gajjar, "Fuzzy and ant colony optimization based clustering and routing protocol in underwater wireless sensor networks," in *Proc. IEEE Region 10th Conf.*, 2022, pp. 1–5.
- [6] S. Borkotoky, U. Schilcher, and C. Bettstetter, "Cooperative relaying in LoRa sensor networks," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–5.
- [7] M. Altuwairiqi, "An optimized multi-hop routing protocol for wireless sensor network using improved honey badger optimization algorithm for efficient and secure QoS," *Comput. Commun.*, vol. 214, pp. 244–259, 2024.
- [8] T. Wiedmann and J. Minx, "A definition of 'carbon footprint,'" *Ecological Econ. Res. Trends*, vol. 1, pp. 1–11, 2008.
- [9] A. Abdelli, L. Mokdad, J. B. Othman, and Y. Hammal, "Dealing with a non green behaviour in WSN," *Simul. Modell. Pract. Theory*, vol. 84, pp. 124–142, 2018.
- [10] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. Blair, and A. Friday, "The climate impact of ICT: A review of estimates, trends and regulations," 2021, *arXiv:2102.02622*.
- [11] T. Pirson and D. Bol, "Assessing the embodied carbon footprint of IoT edge devices with a bottom-up life-cycle approach," *J. Cleaner Prod.*, vol. 322, 2021, Art. no. 128966.
- [12] F. S. D. Silva et al., "A survey on long-range wide-area network technology optimizations," *IEEE Access*, vol. 9, pp. 106079–106106, 2021.
- [13] M. A. M. Almuahaya, W. A. Jabbar, N. Sulaiman, and S. Abdulmalek, "A survey on LoRaWAN technology: Recent trends, opportunities, simulation tools and future directions," *Electronics*, vol. 11, no. 1, 2022, Art. no. 164.
- [14] V. Kulik, V. D. Pham, and R. Kirichek, "Methods and models for using heterogeneous gateways in the mesh LPWANs," in *Proc. Distrib. Comput. Commun. Networks*, 2020, pp. 137–148.
- [15] B. Ousaf and M. Ghaderi, "LoRa network planning: Gateway placement and device configuration," in *Proc. IEEE Int. Congr. Internet Things*, 2019, pp. 25–32.
- [16] M. S. A. Muthanna et al., "Deep reinforcement learning based transmission policy enforcement and multi-hop routing in QoS aware LoRa IoT networks," *Comput. Commun.*, vol. 183, pp. 33–50, 2022.
- [17] "Relay/D2D task force of the LoRa alliance technical committee TS011-1.0.0 Relay," 2022. Accessed: Mar. 18, 2025. [Online.] Available: <https://resources.lora-alliance.org/technical-specifications/ts011-1-0-0-relay>
- [18] F. Li, Y. Wang, and X.-Y. Li, "Gateway placement for throughput optimization in wireless mesh networks," in *Proc. IEEE Int. Conf. Commun.*, 2007, pp. 4955–4960.
- [19] S. Karunaratne and H. Gacanin, "An overview of machine learning approaches in wireless mesh networks," *IEEE Commun. Mag.*, vol. 57, no. 4, pp. 102–108, Apr. 2019.
- [20] R. Atawia and H. Gacanin, "Self-deployment of future indoor Wi-Fi networks: An artificial intelligence approach," in *Proc. IEEE Glob. Commun. Conf.*, 2017, pp. 1–6.
- [21] V. Loginov, I. Kargin, E. Khorov, and A. Lyakhov, "Study of fast multi-hop ALOHA with instant forwarding," in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2018, pp. 1–6.
- [22] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, "Energy consumption model for sensor nodes based on LoRa and LoRaWAN," *Sensors*, vol. 18, no. 7, 2018, Art. no. 2104.
- [23] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the energy performance of LoRaWAN," *Sensors*, vol. 17, no. 10, 2017, Art. no. 2364.
- [24] "MultiTech mDot long range LoRa module's datasheet," 2019. Accessed: Apr. 18, 2022. [Online.] Available: <https://www.multitech.com/documents/publications/data-sheets/86002171.pdf>
- [25] H. H. R. Sherazi, G. Piro, L. A. Grieco, and G. Boggia, "When renewable energy meets LoRa: A feasibility analysis on cable-less deployments," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5097–5108, Dec. 2018.
- [26] Y. Fu, X. Liu, and Z. Yuan, "Life-cycle assessment of multi-crystalline photovoltaic (PV) systems in China," *J. Cleaner Prod.*, vol. 86, pp. 180–190, 2015.
- [27] X. Guo, K. Lin, H. Huang, and Y. Li, "Carbon footprint of the photovoltaic power supply chain in China," *J. Cleaner Prod.*, vol. 233, pp. 626–633, 2019.
- [28] M. S. Chowdhury et al., "An overview of solar photovoltaic panels' end-of-life material recycling," *Energy Strategy Rev.*, vol. 27, 2020, Art. no. 100431.
- [29] Y. Liang et al., "Life cycle assessment of lithium-ion batteries for greenhouse gas emissions," *Resour. Conservation Recycling*, vol. 117, pp. 285–293, 2017.
- [30] D. Deng, "Li-ion batteries: Basics, progress, and challenges," *Energy Sci. Eng.*, vol. 3, no. 5, pp. 385–418, 2015.
- [31] D. D. Oliveira, M. Rampinelli, G. Z. Tozatto, R. V. Andreão, and S. M. T. Müller, "Forecasting vehicular traffic flow using MLP and LSTM," *Neural Comput. Appl.*, vol. 33, pp. 17245–17256, 2021.
- [32] S. S. Haykin, *Neural Networks and Learning Machines*. Hoboken, NJ, USA: Pearson, 2009.
- [33] K. Tyagi, C. Rane, and M. Manry, "Chapter 1 - supervised learning," in *Artificial Intelligence and Machine Learning for EDGE Computing*, R. Pandey, S. K. Khatri, N. Kumar Singh, and P. Verma, Eds. Orlando, FL, USA: Academic Press, 2022, pp. 3–22.
- [34] M. Bentoumi, M. Daoud, M. Benaouali, and A. Taleb Ahmed, "Improvement of emotion recognition from facial images using deep learning and early stopping cross validation," *Multimedia Tools Appl.*, vol. 81, pp. 29887–29917, 2022.
- [35] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.
- [36] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," 2022, *arXiv:2202.07123*.
- [37] F.-J. Alvarado-Alcon, R. Asorey-Cacheda, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "Carbon footprint vs energy optimization in IoT network deployments," *IEEE Access*, vol. 10, pp. 111297–111309, 2022.
- [38] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," *Neural Netw.*, vol. 138, pp. 14–32, 2021.