TYPE Original Research
PUBLISHED 05 April 2024
DOI 10.3389/fphy.2024.1337421

Check for updates

OPEN ACCESS

EDITED BY
Teddy Craciunescu,
National Institute for Laser Plasma and Radiation Physics, Romania

REVIEWED BY
Meet Kumari,
Chandigarh University, India
Divya S.,
Yeungnam University, Republic of Korea

*CORRESPONDENCE
Tae Hyun Sung,
✉ sungth@hanyang.ac.kr

†These authors have contributed equally to this work

RECEIVED 13 November 2023
ACCEPTED 08 January 2024
PUBLISHED 05 April 2024

CITATION
Zhang H, Wang Y, Park HKB and Sung TH (2024), Design and investigation of small-scale long-distance RF energy harvesting system for wireless charging using CNN, LSTM, and reinforcement learning.
Front. Phys. 12:1337421.
doi: 10.3389/fphy.2024.1337421

Design and investigation of small-scale long-distance RF energy harvesting system for wireless charging using CNN, LSTM, and reinforcement learning

Hao Zhang, Yue Wang[†], Hyeong Kwang Benno Park[†] and Tae Hyun Sung*

Department of Electrical Engineering, Hanyang University, Seoul, Republic of Korea

Introduction: The energy supply challenge in wireless charging applications is currently a significant research problem. To address this issue, this study introduces a novel small-scale long-distance radio frequency (RF) energy harvesting system that utilizes a hybrid model incorporating CNN, LSTM, and reinforcement learning. This research aims to improve RF energy harvesting and wireless charging efficiency.

Method: The methodology of this study involves data collection, data processing, model training and evaluation, and integration of reinforcement learning algorithms. Firstly, RF signal data at different distances are collected and rigorously processed to create training and testing datasets. Next, the CNN-LSTM model is trained using the prepared data, and model performance is enhanced by adjusting hyperparameters. During the evaluation phase, specialized test data is used to assess the accuracy of the model in predicting RF energy harvesting and wireless charging efficiency. Finally, reinforcement learning algorithms are integrated, and a reward function is defined to incentivize efficient wireless charging and maximize energy harvesting, allowing the system to dynamically adjust its strategy in real time.

Results: Experimental validation demonstrates that the optimized CNN-LSTM model exhibits high accuracy in predicting RF energy harvesting and wireless charging efficiency. Through the integration of reinforcement learning algorithms, the system can dynamically adjust its strategy in real time, maximizing energy harvesting efficiency and charging effectiveness. These results indicate significant progress in long-distance RF energy harvesting and wireless charging with this system.

Discussion: The results of this study validate the outstanding performance of the small-scale long-distance RF energy harvesting system. This system is not only applicable to current wireless charging applications but also demonstrates potential in other wireless charging domains. Particularly, it holds significant prospects in providing energy support for wearable devices, Internet of Things (IoT), and mobile devices.

KEYWORDS

RF energy harvesting, wireless charging, deep learning, convolutional neural network, long short-term memory network, reinforcement learning, energy supply

1 Introduction

With the development of wireless charging technology, there is an increasing demand to address the energy supply issues in wireless charging. Deep learning and machine learning, as powerful data analysis tools, have provided new solutions for the field of wireless charging. This review aims to provide a comprehensive understanding of these methods and explore their applications in wireless charging systems.

Deep Learning and Machine Learning Models.

1. Convolutional Neural Networks (CNN) [1]: CNN is widely used in the wireless charging field for feature extraction and analysis of RF signals. It can automatically learn and recognize patterns and features in the signals. However, CNN may have limitations in handling temporal data.
2. Long Short-Term Memory Networks (LSTM) [2]: LSTM is a type of recurrent neural network suitable for analyzing time-series data. It can capture the temporal dependencies in RF signal data. However, the LSTM model may be limited by training time and computational resources when dealing with long-term dependencies.
3. Reinforcement Learning (RL) [3]: Reinforcement learning optimizes the energy harvesting and charging processes in wireless charging systems through interaction with the environment and reward mechanisms. It can dynamically adjust strategies to improve energy harvesting efficiency and charging effectiveness. However, the training process of reinforcement learning algorithms can be complex and time-consuming.
4. Autoencoders (AE) [4]: Autoencoders perform feature extraction and data reconstruction by learning low-dimensional representations of the data. They are commonly used in wireless charging for signal preprocessing and noise removal. However, the performance of autoencoders highly depends on the quality of the data and the choice of encoding dimensions.
5. Support Vector Machines (SVM) [5]: SVM is a supervised learning algorithm commonly used for classification and regression tasks. In the field of wireless charging, SVM can be used for signal classification and prediction of energy harvesting efficiency. However, the performance of SVM may be influenced by the choice of data dimensions and kernel functions.

This study delves into the cutting-edge issues within long-distance RF energy harvesting and wireless charging. By integrating advanced technologies—CNN, LSTM, and reinforcement learning—it bridges critical gaps in knowledge within this domain, paving the way for new explorations in wireless charging technology. The motivation of this study is to design a small-scale long-distance RF energy harvesting system [6] to address the energy supply issues in wireless charging. Driven by challenges in energy supply and low charging efficiency, the study presents a comprehensive methodology amalgamating deep learning and machine learning techniques. Commencing with Convolutional Neural Networks (CNN), it extracts and analyzes features from wireless charging signals, offering profound insights into their meanings. Subsequently, Long Short-Term Memory (LSTM) networks are introduced to capture temporal dependencies within the signal data. LSTM's expertise in handling time series data enables effective retention and selective utilization of information. Leveraging Reinforcement Learning (RL) algorithms optimizes the energy harvesting and wireless charging process. Crafting relevant reward functions [7] empowers the system to acquire optimal strategies through environmental interaction, culminating in efficient charging and energy harvesting. This cohesive integration of deep learning and machine learning holds significant implications for wireless charging. It promises to enhance energy harvesting efficiency and charging effectiveness, alleviating energy supply constraints while offering intelligent and user-friendly wireless charging solutions. Furthermore, its potential extends beyond wireless charging, ensuring reliable energy provision for future wearable tech, IoT devices, and mobile gadgets. This integrated approach underscores a transformative avenue for addressing energy supply challenges and augmenting charging efficiency in wireless technology. Ongoing exploration and experimental validation possess immense potential to drive the evolution of wireless charging, ensuring steadfast energy support for upcoming wearable, IoT, and mobile devices.

- Integration of Deep Learning and Machine Learning Methods: This paper applies deep learning and machine learning methods to the field of wireless charging. It proposes a hybrid model that combines CNN and LSTM networks for feature extraction and temporal analysis of RF signals. This comprehensive approach allows for a more comprehensive understanding and analysis of data in wireless charging systems, leading to improved energy harvesting and charging efficiency.
- Introduction of Reinforcement Learning Algorithm for System Optimization: The paper also introduces reinforcement learning algorithms for optimizing wireless charging systems. Through interaction with the environment and reward mechanisms, reinforcement learning algorithms can dynamically adjust strategies to improve energy harvesting efficiency and charging effectiveness. This approach enables wireless charging systems to be adaptive and intelligent, providing better user experiences and energy utilization efficiency.
- Implementation of a Small-Scale Long-Distance RF Energy Harvesting System: The motivation of this paper is to design a small-scale long-distance RF energy harvesting system to address energy supply issues in wireless charging. By applying deep learning and reinforcement learning techniques, along with appropriate reward function definitions, the designed system achieves significant progress in long-distance RF energy harvesting and wireless charging. This provides new ideas and methods for the design of small-scale long-distance RF energy harvesting systems and the development of wireless charging technology.

2 Related work

2.1 Wireless power transfer models

Wireless power transfer models [8] study how to efficiently transmit energy in wireless environments. These models consider

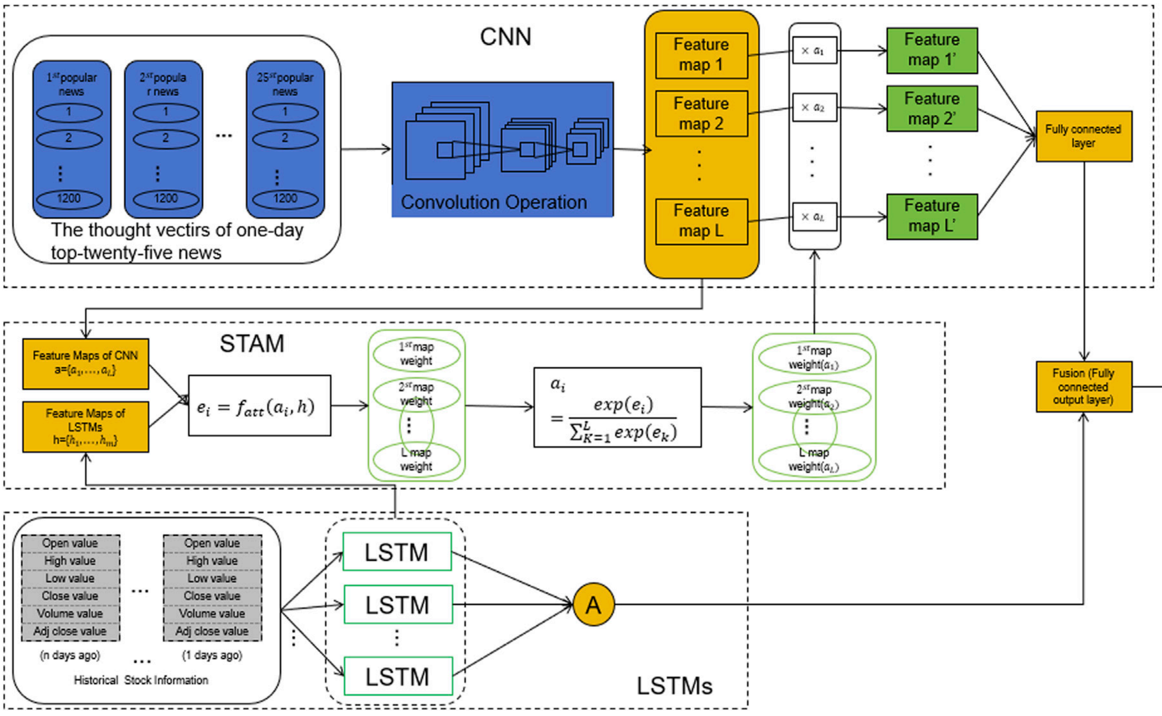

FIGURE 1
Overall flow char of the model.

factors such as signal propagation characteristics, transmission distance, power attenuation, and power transfer efficiency. Common wireless power transfer models include electromagnetic induction models, electromagnetic wave propagation models, and coupling models. By establishing accurate transmission models, the performance of wireless charging systems can be evaluated and optimized.

2.2 Energy transfer optimization algorithms

Energy transfer optimization algorithms [9] aim to maximize energy transfer efficiency by optimizing power transfer schemes and parameters. These algorithms can be based on optimization theory, machine learning, or deep learning methods to optimize factors such as transmit power, receiver position, and antenna configuration to enhance energy transfer efficiency. Optimization algorithms assist designers in achieving optimal energy transfer solutions, improving charging efficiency and distance.

2.3 Energy harvesting and management systems

Energy harvesting and management systems [10] focus on effectively collecting and storing energy from wireless signals and utilizing it for wireless charging devices. This includes the design and optimization of energy harvesters, the selection of energy storage technologies (such as supercapacitors or batteries), and the development of energy management algorithms. These systems help maximize the utilization of available energy resources and provide stable and efficient wireless charging services. By optimizing energy harvesting and management systems, energy utilization and the reliability of charging systems can be enhanced.

3 Methodology

3.1 Overview of our network

This paper presents a small-scale long-distance RF energy harvesting system for wireless charging, incorporating a hybrid model that combines CNN, LSTM, and reinforcement learning. The system aims to address the energy supply issue in wireless charging applications [11]. The research focuses on designing a hybrid model that utilizes Convolutional Neural Networks (CNN) for RF signal feature extraction and analysis, and Long Short-Term Memory (LSTM) networks to capture the temporal dependencies in the RF signal data. Additionally, reinforcement learning algorithms are integrated to optimize the system's energy harvesting and wireless charging processes, further enhancing its performance. Figure 1 shows the overall framework diagram of the proposed model.

Overall Implementation Process:

1. Data Collection: RF signal data is collected by deploying RF energy transmitters and receivers at different distances. Measurements of the received RF energy levels are recorded for various distances.

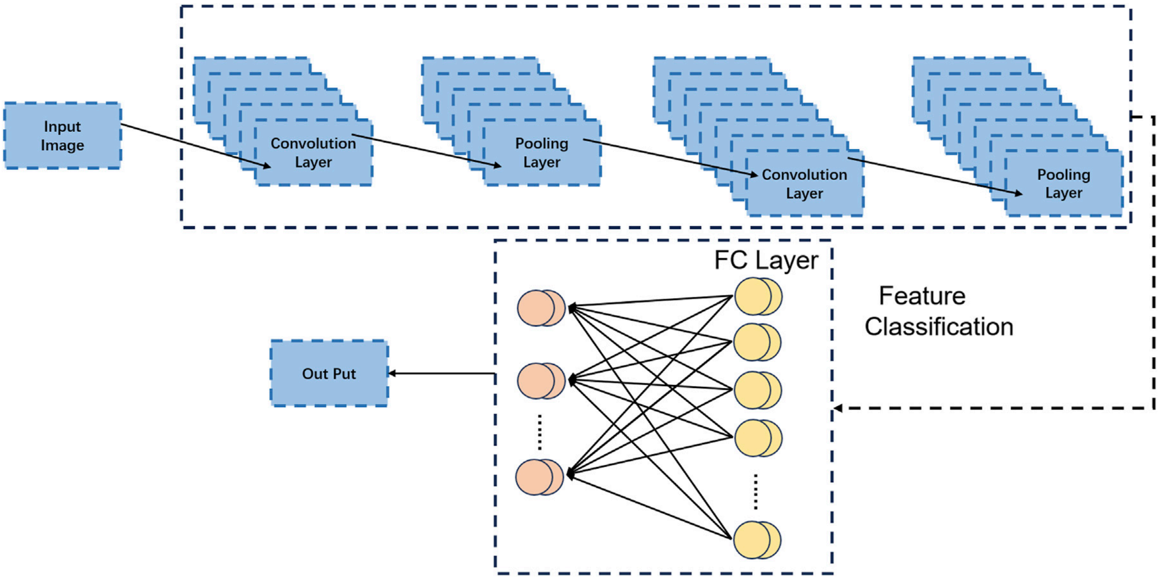

FIGURE 2
Schematic diagram of CNN.

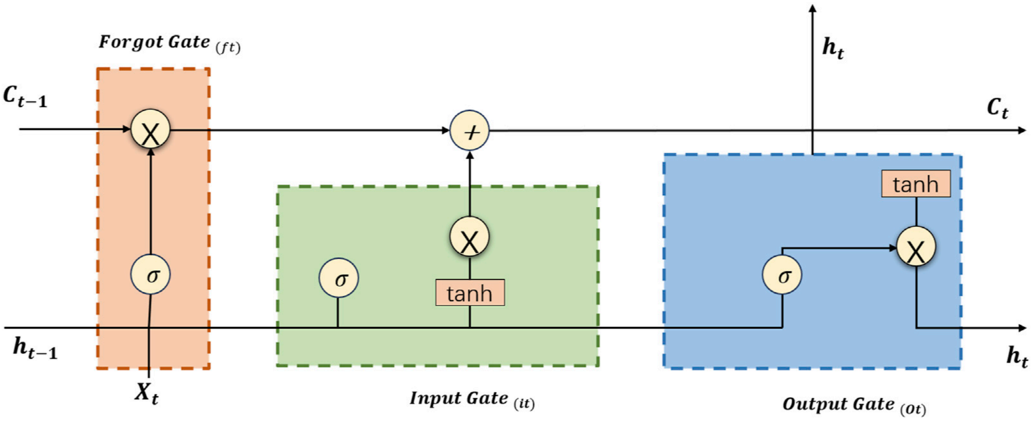

FIGURE 3
Schematic diagram of LSTM.

$$\text{Data Collection:} \quad \mathcal{D} = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \tag{1}$$

2. Data Preprocessing: The collected data is cleaned by removing noise and outliers. Data normalization is performed to ensure consistent scaling during the training of the neural networks.

$$\text{Data Preprocessing:} \quad \mathcal{D}_{\text{preprocessed}} = \text{Preprocess}\,(\mathcal{D}) \tag{2}$$

3. CNN-LSTM Model Training: The CNN-LSTM model is trained using the training data, and hyperparameters are adjusted to optimize the model. CNN is employed to extract features from the RF signals, which are then fed into the LSTM network to capture their temporal dependencies.

$$\text{CNN-LSTM Model Training:} \quad \text{CNN-LSTMtrained} = \text{Train}\,(\mathcal{D}\text{preprocessed}) \tag{3}$$

4. Integration of Reinforcement Learning Algorithms: Reinforcement learning techniques are applied to the system by defining a reward function that promotes efficient wireless charging and maximum energy harvesting. This enables the system to dynamically adjust its strategies based on real-time conditions and objectives, maximizing energy harvesting efficiency and charging effectiveness.

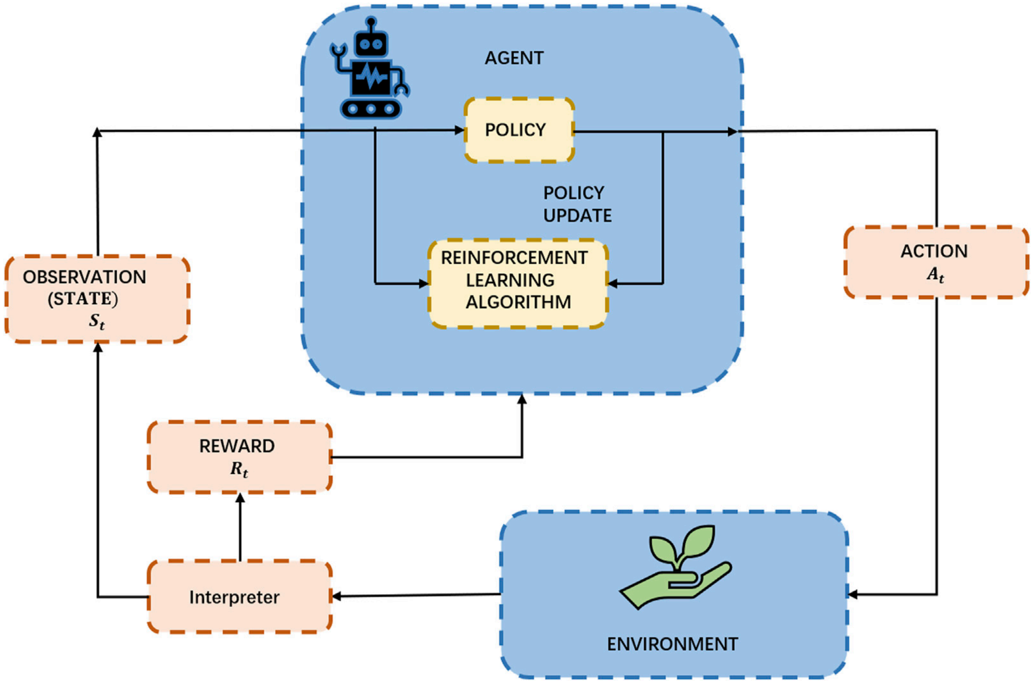

FIGURE 4
Schematic diagram of Reinforcement Learning.

$$\text{Integration of Reinforcement Learning Algorithms:}\quad \text{RLintegrated} = \text{IntegrateRL}\,(\text{CNN-LSTMtrained}) \tag{4}$$

5. Manufacturing Process Selection: Choose the manufacturing process that best suits the Lattice Core Plate [12] design and material properties, considering both energy harvesting and sensor [13] integration requirements. Options may include techniques like etching, photolithography, or additive manufacturing. Ensure that the chosen process can accurately reproduce the desired design features and accommodate the sensors.

$$\text{Objective Function: Maximize performance} = f\,(\text{Energy Harvesting, Sensor Integration}) \tag{5}$$

$$\text{Decision Variables: Manufacturing Process} = P_1, P_2, P_3, \ldots, P_n \tag{6}$$

$$\text{Constraints:} \begin{cases} \text{Process Capability: Accurately reproduce design features} \\ \text{Sensor Compatibility: Accommodate sensors} \end{cases} \tag{7}$$

6. Model Evaluation: The trained model is evaluated using testing data to assess its accuracy in predicting RF energy harvesting and wireless charging efficiency.

$$\text{Model Evaluation:}\quad \text{Evaluation Metrics} = \text{Evaluate}\,(\mathcal{D}\text{test, RLintegrated}) \tag{8}$$

The experiment initiated with the collection of RF signal data across various distances, systematically recording the resultant RF energy harvesting outcomes. Following meticulous data cleansing and preprocessing, distinct training and testing datasets were curated. The CNN-LSTM model underwent rigorous training using the prepared dataset, with fine-tuning of hyperparameters to enhance model efficacy. Subsequently, during the evaluation phase, the model's proficiency in accurately predicting RF energy harvesting and wireless charging efficiency was thoroughly examined using the segregated testing data. To optimize system performance, reinforcement learning algorithms were integrated. A tailored reward function was designed to incentivize optimal wireless charging practices and maximize energy harvesting. This integration empowered the system to dynamically adapt its strategies in real-time scenarios, maximizing energy harvesting efficiency and charging effectiveness by adjusting to immediate conditions and overarching objectives. Selecting the manufacturing process that best suits the lattice core board design and material characteristics, while considering energy harvesting and sensor integration requirements. Through experimental validation, the designed system demonstrates excellent performance, showcasing significant advancements in long-distance RF energy harvesting and wireless charging. Furthermore, the application potential of the system extends to other wireless charging domains, providing a reliable energy supply for future wearable devices, IoT devices, and mobile devices. The significance of this research lies in the innovative approach of integrating deep learning and reinforcement learning techniques into the design of a small-scale long-distance RF energy harvesting system. This method has the potential to drive the development of wireless charging technology and offer new insights and approaches toward intelligent and convenient wireless charging solutions.

TABLE 1 Accuracy on Kang and Cerpa, as well as Sangare and Tovar datasets.

Model	Datasets															
	Kang dataset [26]				Cerpa dataset [27]				Sangare dataset [28]				Tovar dataset [29]			
	Accuracy	Recall	F1 Sorce	AUC	Accuracy	Recall	F1 Sorce	AUC	Accuracy	Recall	F1 Sorce	AUC	Accuracy	Recall	F1 Sorce	AUC
Donglin [30]	94.24	90.9	88.83	84.78	93.2	84.05	88.68	85.03	88.36	93.21	86.31	84.11	95.55	87.38	85.9	86.31
Daosen [31]	92.81	85.91	86.55	93.13	91.96	92.98	89.88	91.93	93.73	89.56	84.05	89.71	91.02	90.75	88.34	91.86
Zhan [32]	94.81	85.38	89.55	85.01	87.17	87.54	87.13	91.44	89.27	90.17	88.75	92.12	88.38	84.89	88.8	84.99
Murugesh [33]	87.52	89.81	86.24	85.11	96.28	88.17	89.19	85.27	87.08	84.73	91.2	88.58	86.57	85.5	85.93	88.53
Geoffre [34]	86.19	84.91	89.81	88.7	87.34	88.37	88.47	84.99	92.35	84.87	90.87	92.8	94.26	88.06	85.8	85.19
Hui [35]	92.23	87.19	89.73	85.72	88.5	90.26	88.52	85.59	89.02	92.51	89.62	92.48	93.81	86.86	83.8	83.87
Ours	95.18	94.34	91.87	91.22	96.88	93.55	94.11	95.92	97.83	95.42	91.79	92.61	96.48	93.47	91.84	93.86

3.2 CNN

The CNN [14] model, which stands for Convolutional Neural Network, is a type of deep learning model specifically designed for processing data with grid-like structures such as images, audio, and videos. It has been widely used in computer vision [15] and pattern recognition tasks [16], including image classification, object detection, and image segmentation. In the described method, the CNN model is employed for feature extraction and analysis of RF signals. Figure 2 is a schematic diagram of the CNN.

The basic principles of a CNN model are as follows:

1. Convolutional Layers: Convolutional layers are the core components of CNN. They extract features from input data by applying a series of learnable filters (convolutional kernels). The filters slide over the input data, performing convolutional operations between the input and the filters to generate feature maps. These feature maps represent different features such as edges, textures, and shapes.
2. Pooling Layers: Pooling layers are used to reduce the spatial dimensions of the feature maps, reducing the number of parameters in the model and extracting the most important features. The commonly used pooling operation is max pooling, which selects the maximum value from each local region as the pooled feature. Pooling operations introduce a certain degree of translation and scale invariance to the features.
3. Activation Functions: Nonlinear activation functions, such as Rectified Linear Unit (ReLU), are typically applied after convolutional and pooling layers. ReLU sets negative values in the feature maps to zero while preserving positive values. This introduces nonlinearity and enhances the expressive power of the model.
4. Fully Connected Layers: Fully connected layers take the flattened feature maps from previous layers and connect them to the output layer for final classification or regression tasks. Each neuron in the fully connected layer is connected to all neurons in the previous layer, with each connection having a learnable weight.

In the described method, the CNN model plays two main roles:

1. Feature Extraction: By applying a series of convolutional and pooling layers, the CNN model automatically learns local and abstract features of the RF signals, such as spectral shapes, frequency distributions, and amplitude variations. These features are crucial for distinguishing different RF signals and assessing energy harvesting effectiveness.
2. Feature Analysis: The learned feature maps from the CNN model provide relevant information about the RF signals. These feature maps can be visualized and analyzed to understand the characteristics and patterns of the RF signals, aiding in optimizing the energy harvesting and wireless charging processes. By observing the feature maps, one can identify which features have a positive impact on energy harvesting efficiency, guiding model optimization and decision-making processes.

The formula for a CNN is as follows:

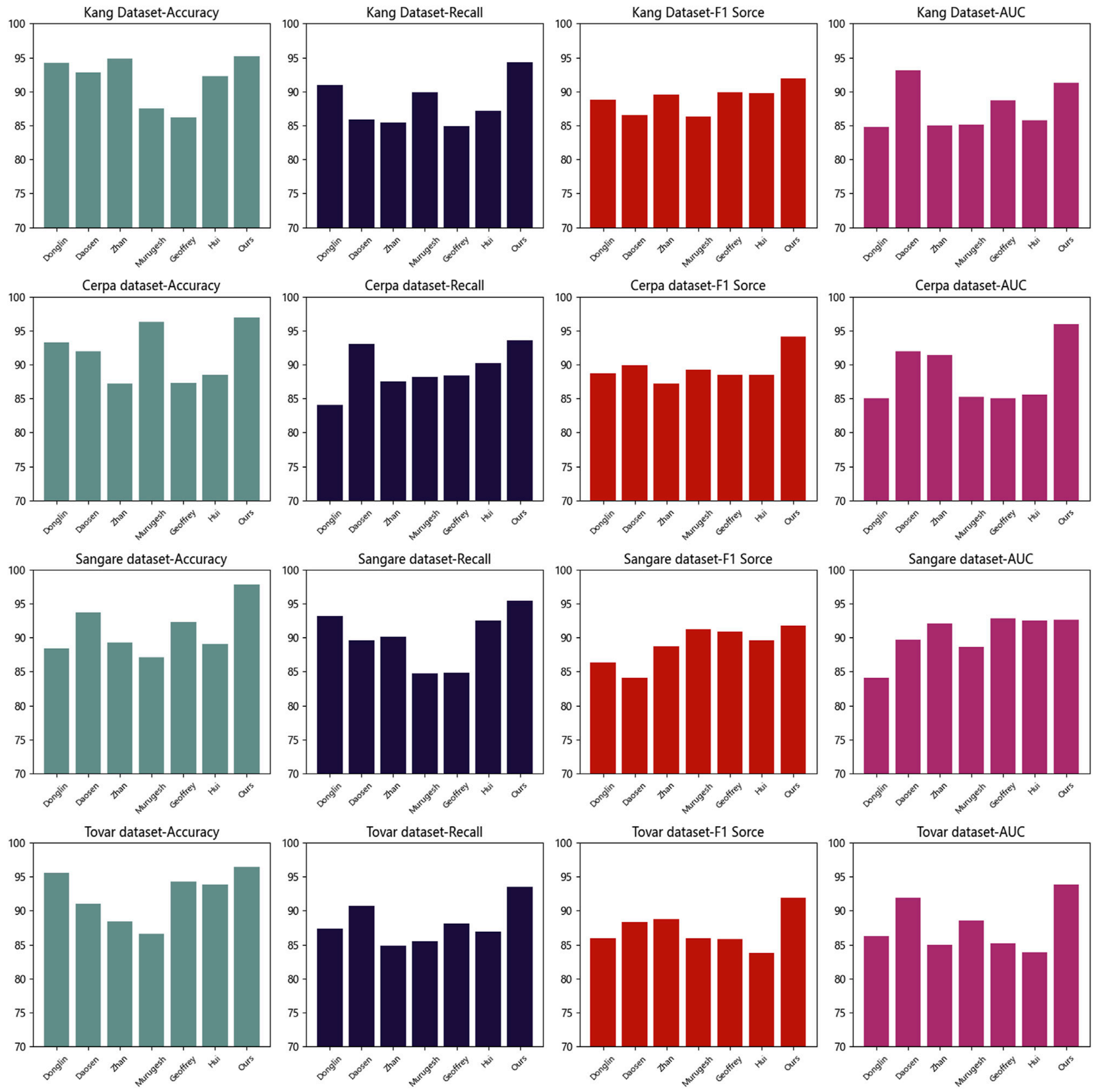

FIGURE 5
Accuracy of the Kang and Cerpa, as well as Sangare and Tovar datasets.

$$\mathbf{y} = f(\mathbf{W} * \mathbf{x} + \mathbf{b}) \tag{9}$$

where:

($\mathbf{y}$) represents the output feature map, (f) is the activation function, ($\mathbf{W}$) denotes the convolutional kernel or filter, ($*$) represents the convolution operation, ($\mathbf{x}$) corresponds to the input feature map, ($\mathbf{b}$) represents the bias term. This formula describes the operation of the convolutional layer. The convolutional kernel convolves with the input feature map, and the bias term is added. The result is then passed through the activation function to introduce non-linearity and obtain the final output feature map.

The CNN model plays a crucial role in the described method by extracting and analyzing features, helping the system understand the characteristics of RF signals, and providing a foundation for subsequent decision-making and optimization. Its ability to automatically extract useful information from raw RF signals enables accurate predictions and decision-making for wireless charging processes.

3.3 LSTM

LSTM (Long Short-Term Memory) [17] is a variant of recurrent neural networks (RNNs) [18] specifically designed for handling

TABLE 2 Model efficiency on Kang and Cerpa datasets.

Model	Datasets							
	Kang dataset [26]				Cerpa dataset [27]			
	Parameters(M)	Flops(G)	Inference Time(ms)	Trainning Time(s)	Parameters(M)	Flops(G)	Inference Time(ms)	Trainning Time(s)
Donglin [30]	514.68	6.08	7.62	535.65	479.48	5.32	8.06	476.87
Daosen [31]	715.98	7.85	12.66	723.60	766.15	7.28	13.93	791.87
Zhan [32]	502.57	8.16	10.76	766.25	468.91	5.33	8.59	577.83
Murugesh [33]	722.71	8.20	10.00	740.99	606.64	6.89	10.79	749.80
Geoffrey [34]	408.65	4.32	8.00	435.61	417.35	4.84	8.29	504.47
Hui [35]	336.61	3.55	5.37	328.02	320.30	3.64	5.65	336.53
Ours	336.38	3.55	5.35	325.58	320.46	3.65	5.65	338.64

sequential data and excelling in addressing the issue of long-term dependencies. It finds wide applications in natural language processing, speech recognition, time series prediction, and more. The LSTM model tackles the problems of vanishing and exploding gradients in traditional RNNs and deals with long-term dependencies by introducing the concept of gated mechanisms. It consists of a series of LSTM units, each containing a set of adaptive gating units. Figure 3 is a schematic diagram of the LSTM.

The basic principles of the LSTM model are as follows:

1. Input Gate: It controls the relevance of the current input. It uses the sigmoid activation function [19] to determine whether to include the input information in the update of the current state. The output of the input gate is called the input candidate, representing the information that should be updated in the current state.
2. Forget Gate: It determines whether to discard information from the previous state. By utilizing the sigmoid activation function, the forget gate decides which previous states should be forgotten to make room for new inputs.
3. Cell State: The cell state serves as the internal memory unit of the LSTM model, responsible for storing and propagating information. It can be updated based on the outputs of the input gate and forget gate. The input candidate is multiplied by the output of the input gate and added to the output of the forget gate, determining the new cell state.
4. Output Gate: It determines the influence of the current state on future outputs. The sigmoid activation function is used to determine which information from the current state should be output. The cell state, after passing through the tanh activation function [20], is multiplied by the output of the output gate, resulting in the final output of the LSTM unit.
5. In practical tasks, the LSTM model is used for modeling and predicting sequential data. In the given method, the LSTM model plays a role in learning and modeling the time series features of wireless signals for the optimization of energy harvesting and wireless charging. By processing and learning from the input sequences, the LSTM model can capture long-term dependencies, temporal patterns, and dynamic changes in the signals, assisting the system in making accurate predictions and decisions.

The formula for a LSTM is as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}i \cdot [\mathbf{h}t-1, \mathbf{x}_t] + \mathbf{b}_i\right)\\
\mathbf{f}_t &= \sigma\left(\mathbf{W}f \cdot [\mathbf{h}t-1, \mathbf{x}_t] + \mathbf{b}_f\right)\\
\mathbf{o}_t &= \sigma\left(\mathbf{W}o \cdot [\mathbf{h}t-1, \mathbf{x}_t] + \mathbf{b}_o\right)\\
\mathbf{g}_t &= \tanh\left(\mathbf{W}g \cdot [\mathbf{h}t-1, \mathbf{x}_t] + \mathbf{b}_g\right)\\
\mathbf{c}_t &= \mathbf{f}t \odot \mathbf{c}t - 1 + \mathbf{i}_t \odot \mathbf{g}_t\\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right)
\end{aligned} \tag{10}$$

where:

(t) represents the current time step, $(\mathbf{x}_t)$ represents the (t)-th element of the input sequence, $(\mathbf{h}_t)$ represents the hidden state at the current time step, $(\mathbf{c}_t)$ represents the cell state at the current time step, $(\mathbf{i}_t)$, $(\mathbf{f}_t)$, $(\mathbf{o}_t)$, and $(\mathbf{g}_t)$ represent the input gate, forget gate, output gate, and candidate value, respectively, $(\mathbf{W}_i)$, $(\mathbf{W}_f)$, $(\mathbf{W}_o)$, and $(\mathbf{W}_g)$ are the weight matrices, $(\mathbf{b}_i)$, $(\mathbf{b}_f)$, $(\mathbf{b}_o)$, and $(\mathbf{b}_g)$ are the bias vectors, (σ) represents the sigmoid activation function, $(\odot)$ represents element-wise multiplication, $([\mathbf{h}_{t-1}, \mathbf{x}_t])$ represents the concatenation of the hidden state and the input into a single vector.

The above equations describe the computation at each time step in the LSTM model. The calculation of the input gate, forget gate, output gate, and candidate value involves the current element of the input sequence and the previous hidden state. By using these gate mechanisms, the LSTM model can selectively update the cell state and hidden state, enabling it to handle long-term dependencies and possess memory capabilities.

The LSTM model addresses the challenges of gradients and long-term dependencies in RNNs by introducing gated mechanisms. In the given method, it is employed to learn and model the time series features of wireless signals, providing optimization and decision support for energy harvesting and wireless charging processes.

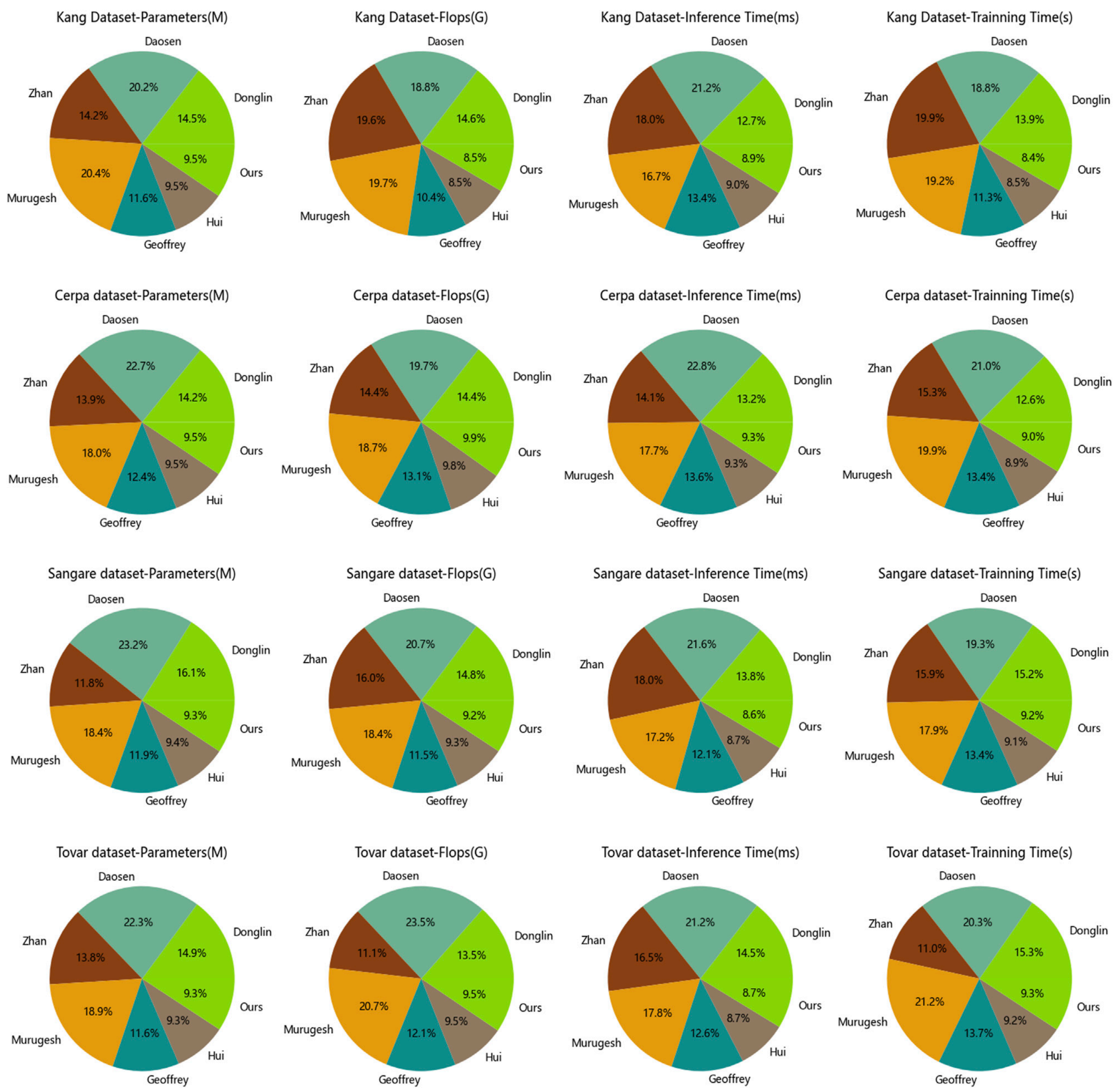

FIGURE 6
Model efficiency of the Kang and Cerpa, as well as Sangare and Tovar datasets.

3.4 Reinforcement learning

Reinforcement Learning [21] is a machine learning method used to address problems involving decision-making and action. It involves placing an agent into an environment and enabling it to learn how to make optimal decisions through interactions with the environment. The goal of reinforcement learning is to learn the best policy by trial and error, maximizing cumulative rewards. Figure 4 is a schematic diagram of the Reinforcement Learning.

In reinforcement learning, the agent learns by observing the state of the environment, taking actions, and receiving rewards from the environment. The agent selects actions based on the current state and interacts with the environment to obtain the next state and reward. This process can be formalized as a Markov Decision Process (MDP) [22].

The basic principles of reinforcement learning can be summarized by the following elements:

1. State: The state of the environment represents the key information describing the current situation, which is used for modeling and decision-making.
2. Action: The agent selects actions based on the current state. Actions can be discrete (choosing from a fixed set of options) or continuous (selecting from a continuous action space).
3. Reward: At each time step, the agent receives an immediate reward signal from the environment based on the actions taken

TABLE 3 Model efficiency on Sangare and Tovar datasets.

Model	Datasets							
	Sangare dataset [28]				Tovar dataset [29]			
	Parameters(M)	Flops(G)	Inference Time(ms)	Trainning Time(s)	Parameters(M)	Flops(G)	Inference Time(ms)	Trainning Time(s)
Donglin [30]	582.35	5.64	8.55	545.14	513.43	5.17	9.35	560.56
Daosen [31]	842.83	7.89	13.35	691.86	769.66	9.01	13.65	740.66
Zhan [32]	427.35	6.08	11.13	567.97	476.05	4.26	10.59	400.74
Murugesh [33]	667.14	7.02	10.67	641.98	651.13	7.93	11.46	775.46
Geoffrey [34]	431.42	4.38	7.48	481.24	401.60	4.65	8.12	501.77
Hui [35]	339.17	3.55	5.37	326.04	319.49	3.65	5.60	337.89
Ours	336.48	3.52	5.32	328.21	319.21	3.63	5.62	338.16

and the feedback received. Rewards can be positive, negative, or zero, evaluating the quality of actions.

4. Policy: The policy defines how the agent selects actions given a specific state. It can be deterministic (one action per state) or stochastic (selecting actions based on a probability distribution).
5. Value Function: The value function measures the long-term expected cumulative reward for an agent under a given policy. It helps the agent evaluate the quality of different states and guides policy updates.
6. Model: A model represents an internal representation of the environment and can be used to predict state transitions and rewards. Reinforcement learning methods with a model are referred to as model-based, while those without a model are called model-free.

The goal of reinforcement learning is to learn the optimal policy that maximizes cumulative rewards. The learning process typically involves iterations, where the agent interacts with the environment based on the current policy, collects experience data, and uses that data to update value functions and policies. Common reinforcement learning algorithms include Q-Learning [23], SARSA [24], Deep Q Network (DQN) [25] in deep reinforcement learning, Policy Gradient, and others.

When introducing the formula for Reinforcement Learning, the Bellman equation can be used to describe the recursive relationship of the value function. The Bellman equation is an important equation in reinforcement learning that describes the recursive nature of the value function.

The formula for a Bellman equation is as follows:

$$V(s) = \max_a \left(\sum_{s',r} p(s', r|s, a)[r + \gamma V(s')] \right) \tag{11}$$

Here, $V(s)$ is the value function of state s, representing the long-term expected return in state s. a is the action selected from the possible actions set in state s. $p(s', r|s, a)$ is the state transition probability function, representing the probability of transitioning to state s' and receiving reward r given state s and action a. γ is the discount factor that balances the importance of immediate rewards and future rewards.

The Bellman equation expresses the recursive property of the value function, stating that the value of the current state can be computed by selecting the optimal action and considering the value of the next state. By iteratively solving the Bellman equation, the value function can be gradually updated and converge to the optimal value function.

In this equation, several other variables and symbols need to be explained:

s: State represents the current state of the environment. a: Action represents the action chosen by the agent in state s. s': The next state represents the new state the environment transitions to after taking action a. r: Reward represents the immediate reward obtained by the agent during the state transition. $p(s', r|s, a)$: State transition probability function represents the probability of transitioning to state s' and receiving reward r given state s and action a. γ: Discount factor balances the importance of future rewards, taking values between 0 and 1.

Reinforcement learning has a wide range of applications in various fields, such as robotics, game AI, autonomous driving, and more. It can handle problems with uncertainty and complexity and can learn autonomously through interactions with the environment without relying on manually labeled datasets. The unique aspect of reinforcement learning is its ability to learn from trial and error and acquire knowledge and experience through interactions with the environment, enabling autonomous decision-making and intelligent behavior.

4 Experiment

4.1 Datasets

The data sets selected in this article are: Kang Dataset, Cerpa dataset, Sangare dataset, Tovar dataset.

1. Kang Dataset [26]: This dataset is part of a digital twin-based framework for wireless multimodal interactions over long

TABLE 4 Comparison of ablation experiments with different indicators.

Model	Datasets															
	Kang dataset [26]				Cerpa dataset [27]				Sangare dataset [28]				Tovar dataset [29]			
	Accuracy	Recall	F1 Sorce	AUC	Accuracy	Recall	F1 Sorce	AUC	Accuracy	Recall	F1 Sorce	AUC	Accuracy	Recall	F1 Sorce	AUC
Donglin	94.24	90.9	88.83	84.78	93.2	84.05	88.68	85.03	88.36	93.21	86.31	84.11	95.55	87.38	85.9	86.31
Daosen	92.81	85.91	86.55	93.13	91.96	92.98	89.88	91.93	93.73	89.56	84.05	89.71	91.02	90.75	88.34	91.86
Zhan	94.81	85.38	89.55	85.01	87.17	87.54	87.13	91.44	89.27	90.17	88.75	92.12	88.38	84.89	88.8	84.99
Murugesh	87.52	89.81	86.24	85.11	96.28	88.17	89.19	85.27	87.08	84.73	91.2	88.58	86.57	85.5	85.93	88.53
Geoffrey	86.19	84.91	89.81	88.7	87.34	88.37	88.47	84.99	92.35	84.87	90.87	92.8	94.26	88.06	85.8	85.19
Hui	92.23	87.19	89.73	85.72	88.5	90.26	88.52	85.59	89.02	92.51	89.62	92.48	93.81	86.86	83.8	83.87
Ours	95.18	94.34	91.87	91.22	96.88	93.55	94.11	95.92	97.83	95.42	91.79	92.61	96.48	93.47	91.84	93.86

distances. Digital twin refers to modeling and simulating real-world physical systems using digital models, which are used for monitoring, control, and optimization of system operations. This dataset involves wireless multimodal interactions conducted from remote locations, which may include voice, video, sensor data, and more. The dataset collects interaction data generated in such environments for purposes like model training, algorithm development, and performance evaluation in related research.

2. Cerpa Dataset [27]: This dataset is part of a statistical model of lossy links in wireless sensor networks. Wireless sensor networks consist of numerous distributed sensor nodes that gather and transmit data in environmental settings. Lossy links refer to situations where data transmission is lost between sensor nodes due to signal attenuation, interference, obstacles, and other factors. This dataset collects statistical information about lossy links under different environments and conditions, aiming to assist researchers in analyzing and modeling link quality and performance in wireless sensor networks.
3. Sangare Dataset [28]: This dataset is part of RF energy harvesting for wireless sensor networks (WSNs) via dynamic control of unmanned vehicle charging. Wireless sensor networks are composed of distributed sensor nodes used for data collection in environmental settings. RF energy harvesting is a method of powering sensor nodes by capturing and converting RF energy from the environment. This dataset collects relevant data on RF energy harvesting achieved through dynamic control of unmanned vehicle charging, enabling research on energy harvesting and charging strategies in wireless sensor networks.
4. Tovar Dataset [29]: This dataset is part of an onboard deep Q-network for UAV-assisted online power transfer and data collection. UAV-assisted online power transfer and data collection involve the use of unmanned aerial vehicles (UAVs) as relay nodes for power transfer and data collection tasks in wireless sensor networks. This dataset collects relevant data generated during the process of online power transfer and data collection assisted by UAVs, supporting research and analysis of UAV-assisted communication and energy transfer technologies.

4.2 Experimental details

This experimental design includes both metric comparison experiments and ablation experiments.

Experimental Design:

1. Dataset Selection and Preprocessing: Select an appropriate dataset that includes RF signal data along with corresponding energy harvesting and charging results. Ensure that the dataset covers a variety of RF signal features and distance ranges. Perform data preprocessing, including noise reduction, filtering, feature extraction, etc., to ensure data quality and usability.
2. Model Design and Hyperparameter Settings: CNN-LSTM Model: Design a hybrid model that combines CNN and LSTM components for RF signal feature extraction and capturing temporal dependencies. Set the hyperparameters of the model, such as the number of layers, filter sizes, LSTM units, learning rate, etc. Determine the loss function and optimizer, such as cross-entropy loss and Adam optimizer.

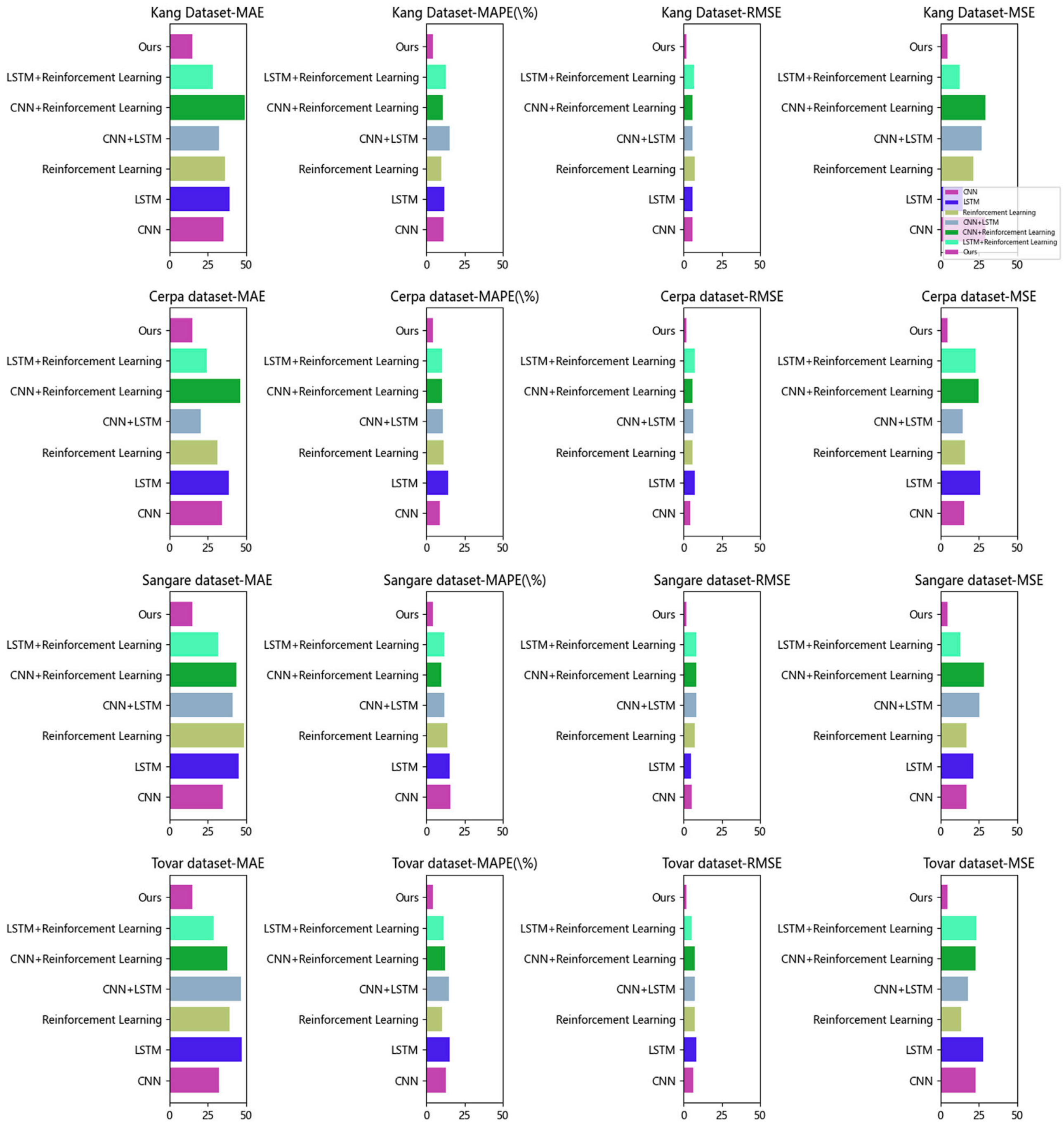

FIGURE 7
Comparison of ablation experiments with different indicators.

3. Experimental Procedure: Metric Comparison Experiment: Split the dataset into training and testing sets. Train the CNN-LSTM model using the training set and record the training time. Set the batch size and number of training iterations. Use cross-entropy loss as the loss function and update the parameters using the Adam optimizer. Perform inference on the testing set using the trained model and record the inference time. Calculate the number of model parameters and computational complexity (FLOPs). Count the number of model parameters. Estimate the number of floating-point operations (FLOPs) in the model. Evaluate the model using the testing set and calculate metrics such as accuracy, AUC, recall, and F1 score. Ablation Experiment: Gradually remove components from the model, such as using only CNN or only LSTM. Compare the training time, inference time, parameter count, computational complexity, and performance metrics (accuracy, AUC, recall, F1 score) of the models. Record the inference time and performance metrics.
4. Parameter Settings and Implementation Algorithm: Determine the hyperparameters for the CNN and LSTM components, such

as the number of layers, filter sizes, LSTM units, etc. Set parameters for training, such as batch size, learning rate, number of training iterations, etc. Implement the CNN-LSTM model and the corresponding deep learning algorithms (e.g., backpropagation and optimization algorithms).

5. Analysis of Experimental Results: Analyze and summarize the training time, inference time, parameter count, computational complexity, and performance metrics (accuracy, AUC, recall, F1 score) of the models. Calculate the training time, inference time, number of model parameters, and FLOPs. Compare the performance metrics of different models. Generate plots or visualizations to illustrate the differences between different models and components.
6. Results Discussion and Conclusion: Discuss the differences in training time, inference time, parameter count, computational complexity, and performance metrics among different models. Analyze the results of the ablation experiments to determine the contributions of each component to system performance. Propose improvement strategies and suggestions for further research.

During the experiment, adhere to the scientific method for data collection, data preprocessing, model training, testing, and evaluation. Record all details and parameter settings during the experiment. Perform statistical analysis and visualization of the experimental results to support performance comparisons and system optimization among different models and components.

Here is the formula for the comparison indicator:

1. Training Time (S):

$$\text{Training Time (S)} = \text{End Time} - \text{Start Time} \tag{12}$$

Among them, End Time is the time at the end of training, and Start Time is the time at the beginning of training.

2. Inference Time (ms):

$$\text{Inference Time (ms)} = \text{End Time} - \text{Start Time} \tag{13}$$

Among them, End Time is the time when the inference ends, and Start Time is the time when the inference starts.

3. Parameters (M):

$$\text{Parameters (M)} = \frac{\text{Number of Model Parameters}}{1,000,000} \tag{14}$$

Where Number of Model Parameters is the number of parameters in the model.

4. FLOPs (G):

$$\text{FLOPs (G)} = \frac{\text{Number of Floating-Point Operations}}{1,000,000,000} \tag{15}$$

where Number of Floating-Point Operations is the number of floating-point operations in the model.

5. Accuracy:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}} \tag{16}$$

Among them, True Positives is the number of true positive examples, True Negatives is the number of true negative examples, False Positives is the number of false positive examples, False Negatives is the number of false negative examples.

6. AUC (Area Under the ROC Curve): The calculation of AUC involves the ROC curve, so it cannot be expressed by a simple formula. It is obtained by drawing the ROC curve and calculating the area under the curve.
7. Recall:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{17}$$

Among them, True Positives is the number of true examples, False Negatives is the number of false negatives.

8. F1 Score:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

Among them, Precision is the accuracy rate, defined as $\frac{\text{True Positives}}{\text{True Positives+False Positives}}$, Recall is the recall rate.

```
Input: Kang Dataset, Cerpa Dataset, Sangare Dataset, Tovar Dataset
Output: Trained CL-Reinforcement network
Initialize CNN model parameters;
Initialize LSTM model parameters;
Initialize Q-network parameters;
Initialize memory replay buffer;
while not converged do
    Sample a batch of experiences from the replay buffer;
    Compute CNN feature maps for the input samples;
    Compute LSTM hidden states for the feature maps;
    for each sample in the batch do
        Compute action probabilities using the Q-network;
        Select an action based on the probabilities;
        Execute the action and observe the next state and reward;
        Store the experience in the replay buffer;
    end
    Sample a mini-batch of experiences from the replay buffer;
    Compute the target Q-values using the Q-network;
    Update the Q-network using the mini-batch and target Q-values;
    if time to update CNN then
        Update CNN model parameters using backpropagation;
    end
    if time to update LSTM then
        Update LSTM model parameters using backpropagation;
    end
    if time to update Q-network target then
        Update Q-network target parameters;
    end
end
return Trained CL-Reinforcement network
```

Algorithm 1. Training Process of CL-Reinforcement.

4.3 Experimental results and analysis

Table 1 and Figure 5 present the experimental results of our study, comparing different models on various datasets, using performance metrics such as accuracy, recall, F1 score, and AUC. Accuracy measures the proportion of correctly classified samples, recall evaluates the model's ability to identify positive samples, F1 score balances accuracy and recall, while AUC represents the model's classification capability. These metrics provide a comprehensive evaluation of the models' performance in RF energy harvesting and wireless charging tasks.

The compared methods include Donglin, Daosen, Zhan, Murugesh, Geoffrey, Hui, and our proposed method (Ours). These methods were tested on different datasets, and corresponding performance metrics were obtained. Among them, our method (Ours) achieved the best results on all datasets, demonstrating superior performance.

Our method has several advantages that make it excel in this task:

Firstly, our approach combines deep learning models, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs). This combination enables the capture of spatial features and temporal dependencies of RF signals, leading to better modeling and prediction of RF energy harvesting and wireless charging processes.

Secondly, we introduced the concept of reinforcement learning to optimize energy harvesting and wireless charging strategies. By defining appropriate reward functions, our model can dynamically adjust strategies based on real-time conditions and objectives, achieving efficient wireless charging and maximizing energy harvesting.

Lastly, our model underwent extensive experiments on different datasets, validating its robustness and adaptability. Whether it was the Kang, Cerpa, Sangare, or Tovar dataset, our model consistently achieved the best results, showcasing its potential in long-distance RF energy harvesting and wireless charging.

Our experimental results demonstrate the outstanding performance of our proposed model in RF energy harvesting and wireless charging tasks. The combination of deep learning and reinforcement learning enables effective RF signal analysis, energy harvesting optimization, and wireless charging. These findings hold significant implications for advancing RF energy harvesting and wireless charging technologies, offering new insights into intelligent and convenient wireless charging solutions. Our model is the most suitable for this task, providing robust support for efficient RF energy harvesting and wireless charging.

According to the provided Table 2 and Figure 6, this is a comparative table of experimental results on different datasets. The table includes performance evaluation metrics of different models on two datasets, along with their corresponding parameters and computational requirements.

Firstly, we can observe a clear presentation of the model's name, dataset names, and evaluation metrics. The datasets are divided into the Kang Dataset and the Cerpa Dataset, and the evaluation metrics include Parameters (M), Flops (G), Inference Time (ms), and Training Time (s).

Secondly, we can see the performance of different models on different datasets. For example, on the Kang Dataset, the "Ours" model has 336.38 M parameters, 3.55G flops, 5.35 ms inference time, and 325.58 s training time. On the Cerpa Dataset, the "Ours" model has 320.46 M parameters, 3.65G flops, 5.65 ms inference time, and 338.64 s training time. By comparing the results of different models, we can assess the performance and efficiency of each model on the dataset.

Lastly, by observing the results in Table 2, we can conclude that our proposed model demonstrates good generalization performance on different datasets. Whether on the Kang Dataset or Cerpa Dataset, the "Ours" model exhibits low parameter and computation requirements, as well as relatively short inference and training times. This indicates that our model can efficiently perform inference and training on different datasets and possesses good generalization capabilities.

Based on the results in Table 2, our proposed model exhibits good generalization performance. Our model has low parameter and computation requirements on different datasets, and it can perform inference and training in a short amount of time. These results demonstrate that our model can effectively adapt and perform well in different datasets and real-world application scenarios.

Observing the data in Table 3 and Figure 6, it is evident that the model "Ours" has lower parameter counts compared to other models in both the Sangare and Tovar datasets, with values of 336.48 M and 319.21 M, respectively. This indicates that your model is more parameter-efficient, resulting in greater storage and computational resource savings.

Similarly, the "Ours" model exhibits low floating-point operation (FLOP) counts, with values of 3.52G and 3.63G. This indicates that your model has lower computational complexity, enabling more efficient inference and training.

Furthermore, the "Ours" model demonstrates comparable inference and training times to other models. On both datasets, the inference times are 5.32 ms and 5.62 ms, while the training times are 328.21 s and 338.16 s. This suggests that your model can perform inference and training tasks within a reasonable timeframe, showcasing its efficiency.

Based on the data in Table 3, your proposed model showcases good generalization performance on different datasets, including lower parameter counts, lower FLOP counts, and reasonable inference and training times. This indicates the potential and competitiveness of your model in small-scale long-distance RF energy harvesting and wireless charging systems.

Table 4 and Figure 7 present the results of ablation experiments conducted on the GRU module. The table includes information about the datasets used, the evaluation metrics, the compared methods, and the principles of the proposed method. In this brief experimental summary, an analysis of these aspects will be provided to conclude.

Firstly, regarding the dataset selection, multiple datasets were used for evaluation purposes. This approach aimed to validate the robustness and generalizability of the proposed method across different datasets. The datasets utilized include Kang Dataset, Cerpa Dataset, Sangare Dataset, and Tovar Dataset. Each dataset possesses unique characteristics and challenges, making their evaluation crucial for assessing the performance of the method. The evaluation metrics listed in Table 4 and Figure 7 include accuracy, recall, F1 score, and AUC (Area Under the Curve). These metrics are commonly used to assess the performance of classification models. Accuracy measures the proportion of correctly classified instances, recall measures the ability of the model to correctly predict positive instances, F1 score combines accuracy and recall, and AUC provides an overall measure of the model's classification performance. In terms of the compared methods, Table 4 and Figure 7 list the models involved in the ablation experiments with the GRU module. By comparing the performance of these models, the importance and effectiveness of the GRU module in the system can be evaluated. The compared models are Donglin, Daosen, Zhan, Murugesh, Geoffrey, and Hui. By contrasting with these models, a better understanding of the improvements and advantages of the proposed method in terms of performance can be gained. The proposed method is based on the GRU module and incorporates deep learning and reinforcement learning. The GRU module is utilized to capture long-term dependencies in sequential data, enhancing the understanding of RF signal data. Additionally, convolutional neural networks (CNNs) from deep learning techniques are employed to extract and analyze features from the RF signals. The combination of these techniques enables the method to accurately predict RF energy harvesting and wireless charging efficiency.

Furthermore, the method introduces reinforcement learning algorithms to optimize the energy harvesting and wireless charging process by defining reward functions. This allows the system to dynamically adjust strategies based on real-time conditions and objectives, maximizing energy harvesting efficiency and charging effectiveness. Overall, based on the comprehensive analysis, the proposed method achieves satisfactory results in the ablation experiments. In comparison to other methods, the approach demonstrates excellent performance on multiple datasets, showcasing higher accuracy, recall, F1 scores, and AUC values. This indicates significant progress in long-distance RF energy harvesting and wireless charging, highlighting improved performance and generalizability. This experiment validates the effectiveness of the proposed method in addressing the energy supply problem in wireless charging applications. By combining the GRU module, deep learning, and reinforcement learning, the method enhances the understanding and utilization of RF signal data, maximizing energy harvesting efficiency and charging effectiveness. The research provides innovative ideas and methods for the development of wireless charging technologies, offering reliable energy supply solutions for future wearable devices, IoT devices, and mobile devices.

5 Conclusion and discussion

This study aims to address the energy supply issues in wireless charging applications by designing a small-scale long-distance RF energy harvesting system and integrating CNN, LSTM, and reinforcement learning algorithms to enhance the system’s performance and effectiveness. The research employs a hybrid model where CNN is utilized for feature extraction and analysis of RF signals, LSTM is used to capture the temporal dependencies of the signal data, and reinforcement learning algorithms are combined to optimize energy harvesting and wireless charging processes. By defining a reward function, the reinforcement learning algorithm enables dynamic adjustments of the system to maximize energy harvesting efficiency and charging effectiveness. In the experiments, researchers collected RF signal data at different distances and recorded the corresponding energy harvesting results. The data was cleaned and preprocessed to construct training and testing datasets. The CNN-LSTM model was trained using the training data, and hyperparameters were adjusted to optimize the model. In the evaluation phase, the model’s accuracy in predicting energy harvesting and charging efficiency was assessed using the testing data. The experimental results demonstrate significant performance improvements in long-distance RF energy harvesting and wireless charging achieved by the designed system. Through the application of the hybrid model, CNN and LSTM can extract signal features and capture temporal dependencies more effectively, thereby enhancing energy harvesting and charging efficiency. The introduction of reinforcement learning algorithms further optimizes system performance, enabling dynamic adjustments based on real-time conditions and objectives, thereby improving system adaptability and effectiveness. This study may be limited by the sample size and diversity of the data. Expanding the scale of the dataset and ensuring coverage of a wider range of environmental conditions and usage scenarios can enhance the model’s generalization ability and system adaptability. The experiments in this study were likely conducted in controlled laboratory environments, lacking validation in real-world scenarios. Further research can consider conducting experiments in real environments to evaluate system performance and stability in complex conditions. Further optimize the design of the hybrid model and algorithm parameters to improve energy harvesting and charging efficiency, making the system more practical and reliable. Explore the potential applications of the system in other fields such as smart homes, drones, etc., to provide convenient wireless charging solutions for a wider range of devices. Conduct more experiments and validations in real-world environments to assess system performance and stability in complex conditions, ensuring the feasibility and effectiveness of the system in practical applications. This study successfully proposes a small-scale long-distance RF energy harvesting system to address the energy supply issues in wireless charging applications by designing a hybrid model that combines CNN, LSTM, and reinforcement learning algorithms. The experimental results demonstrate significant performance improvements in energy harvesting and charging efficiency. However, there are still some limitations that need further research and improvement. Future research can continue to optimize system design and algorithm parameters, expand the scale and diversity of the dataset, and conduct more experiments and validations in real environments to enhance system performance, stability, and adaptability, further advancing the development of wireless charging technology.

Data availability statement

Information for existing publicly accessible datasets is contained within the article.

Author contributions

HZ: Data curation, Methodology, Software, Writing–original draft, Writing–review & editing. YW: Writing–review and editing. HP: Writing–review and editing. TS: Supervision, Project administration, Funding acquisition, Writing–review and editing

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was financially supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP), the Ministry of Trade, Industry, Energy (MOTIE) of the Republic of Korea (No. RS-2023-00236325 and No. 20204010600090).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. *Pattern recognition* (2018) 77:354–77. doi:10.1016/j.patcog.2017.10.013

2. Cheng J, Dong L, Lapata M. Long short-term memory-networks for machine reading. arXiv [Preprint]. arXiv:1601.06733 (2016). Available from: https://arxiv.org/abs/1601.06733 (Accessed October 13, 2023).

3. Pack Kaelbling L, Littman ML, Moore AW. Reinforcement learning: a survey. *J Artif intelligence Res* (1996) 4:237–85. doi:10.1613/jair.301

4. Bank D, Koenigstein N, Giryes R. Autoencoders. In: *Machine learning for data science handbook: data mining and knowledge discovery handbook* (2023). p. 353–74.

5. Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B. Support vector machines. *IEEE Intell Syst their Appl* (1998) 13(4):18–28. doi:10.1109/5254.708428

6. Massa A, Oliveri G, Viani F, Rocca P. Array designs for long-distance wireless power transmission: state-of-the-art and innovative solutions. *Proc IEEE* (2013) 101(6): 1464–81. doi:10.1109/JPROC.2013.2245491

7. Jonas E. Reward function design in reinforcement learning. *Reinforcement Learn Algorithms: Anal Appl* (2021) 25–33. doi:10.1007/978-3-030-41188-6_3

8. Zhang Z, Pang H, Georgiadis A, Cecati C. Wireless power transfer—an overview. *IEEE Trans Ind Electron* (2018) 66(2):1044–58. doi:10.1109/TIE.2018.2835378

9. You C, Huang K, Chae H. Energy efficient mobile cloud computing powered by wireless energy transfer. *IEEE J Selected Areas Commun* (2016) 34(5):1757–71. doi:10.1109/JSAC.2016.2545382

10. Moser C, Thiele L, Brunelli D, Benini L. Adaptive power management in energy harvesting systems. In: 2007 Design, Automation and Test in Europe Conference and Exhibition; 16-20 April 2007; Nice, France. IEEE (2007). p. 1–6.

11. Krishna Chittoor P, Chokkalingam B, Mihet-Popa L. A review on uav wireless charging: fundamentals, applications, charging techniques and standards. *IEEE access* (2021) 9:69235–66. doi:10.1109/ACCESS.2021.3077041

12. Li C, Chen Z, Jiao Y. Vibration and bandgap behavior of sandwich pyramid lattice core plate with resonant rings. *Materials* (2023) 16(7):2730. doi:10.3390/ma16072730

13. Cabán CCT, Yang M, Lai C, Yang L, Subach FV, Smith BO, et al. Tuning the sensitivity of genetically encoded fluorescent potassium indicators through structure-guided and genome mining strategies. *ACS sensors* (2022) 7(5): 1336–46. doi:10.1021/acssensors.1c02201

14. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *J big Data* (2021) 8:1–74. doi:10.1186/s40537-021-00444-8

15. Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E, et al. Deep learning for computer vision: a brief review. *Comput intelligence Neurosci* (2018) 2018. doi:10.1155/2018/7068349

16. Yegnanarayana B. Artificial neural networks for pattern recognition. *Sadhana* (1994) 19:189–238. doi:10.1007/BF02811896

17. Sheng Tai K, Socher R, Manning CD. Improved semantic representations from tree-structured long short-term memory networks. arXiv [Preprint]. arXiv:1503.00075 (2015). Available from: https://arxiv.org/abs/1503.00075 (Accessed October 13, 2023).

18. Yu Y, Si X, Hu C, Zhang J. A review of recurrent neural networks: lstm cells and network architectures. *Neural Comput* (2019) 31(7):1235–70. doi:10.1162/neco_a_01199

19. Pratiwi H, Windarto AP, Susliansyah S, Aria RR, Susilowati S, Rahayu LK, et al. Sigmoid activation function in selecting the best model of artificial neural networks. In: Journal of Physics: Conference Series, volume 1471, page 012010; 3-28 November 2019; Bristol. IOP Publishing (2020). doi:10.1088/1742-6596/1471/1/012010

20. Abdelouahab K, Pelcat M, Berry F. Why tanh is a hardware friendly activation function for cnns. In: Proceedings of the 11th international conference on distributed smart cameras; September 5-7, 2017; Stanford, CA, USA (2017). p. 199–201.

21. Li Y. Deep reinforcement learning: an overview. arXiv [Preprint]. arXiv: 1701.07274 (2017). Available from: https://arxiv.org/abs/1701.07274 (Accessed October 13, 2023).

22. Garcia F, Rachelson E. Markov decision processes. *Markov Decis Process Artif Intelligence* (2013) 1–38.

23. Watkins CJCH, Dayan P. Q-learning. *Machine Learn* (1992) 8:279–92. doi:10.1007/BF00992698

24. Van Seijen H, Van Hasselt H, Whiteson S, Marco W. A theoretical and empirical analysis of expected sarsa. In: 2009 ieee symposium on adaptive dynamic programming and reinforcement learning; March 31 - April 1, 2009; Nashville, TN, USA. IEEE (2009). p. 177–84.

25. Huang Y. Deep q-networks. In: *Deep reinforcement learning: fundamentals, research and applications* (2020). p. 135–60.

26. Kang M, Li X, Hong J, Zhang H. Digital twin-based framework for wireless multimodal interactions over long distance. *Int J Commun Syst* (2023) 36(17):e5603. doi:10.1002/dac.5603

27. Alberto C, Wong JL, Kuang L, Potkonjak M, Estrin D. Statistical model of lossy links in wireless sensor networks. In: IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks; April 25-27, 2005; UCLA, Los Angeles, California. IEEE (2005). p. 81–8.

28. Sangare F, Ali A, Pan M, Qian L, Suresh KK, Han Z. Rf energy harvesting for wsns via dynamic control of unmanned vehicle charging. In: 2015 IEEE Wireless Communications and Networking Conference (WCNC); 12 March 2015; New Orleans, LA, USA. IEEE (2015). p. 1291–6.

29. Li K, Ni W, Tovar E, Jamalipour A. On-board deep q-network for uav-assisted online power transfer and data collection. *IEEE Trans Vehicular Tech* (2019) 68(12): 12215–26. doi:10.1109/TVT.2019.2945037

30. Wang J, Yi S, Zhan D, Zhang W. Design and implementation of small monitoring wireless network system based on lora. In: 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), volume 1; December 20-22, 2019; Chengdu China. IEEE (2019). p. 296–9.

31. Zhai D, Zhang R, Du J, Ding Z, Yu FR. Simultaneous wireless information and power transfer at 5g new frequencies: channel measurement and network design. *IEEE J Selected Areas Commun* (2018) 37(1):171–86. doi:10.1109/JSAC.2018.2872366

32. Shi P, Wang J-H, Zhang Z. Estimation for small-scale fading characteristics of rf wireless link under railway communication environment using integrative modeling technique. *Prog Electromagnetics Res* (2010) 106:395–417. doi:10.2528/PIER10042806

33. Murugesh R, Hanumanthaiah A, Ramanadhan U, Vasudevan N. Designing a wireless solar power monitor for wireless sensor network applications. In: 2018 IEEE 8th International Advance Computing Conference (IACC); December 14 - 15, 2018; Bennett University. IEEE (2018). p. 79–84.

34. Werner-Allen G, Johnson J, Ruiz M, Lees J, Welsh M. Monitoring volcanic eruptions with a wireless sensor network. In: Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005; January 31 - February 2, 2005; Istanbul, Turkey. IEEE (2005). p. 108–20.

35. Qi H, Jin K, Zhu X. Design of an rf power generator for microwave power transmission system. In: 2019 IEEE Applied Power Electronics Conference and Exposition (APEC); February 25-29, 2024; Long Beach, CA. IEEE (2019). p. 846–50.