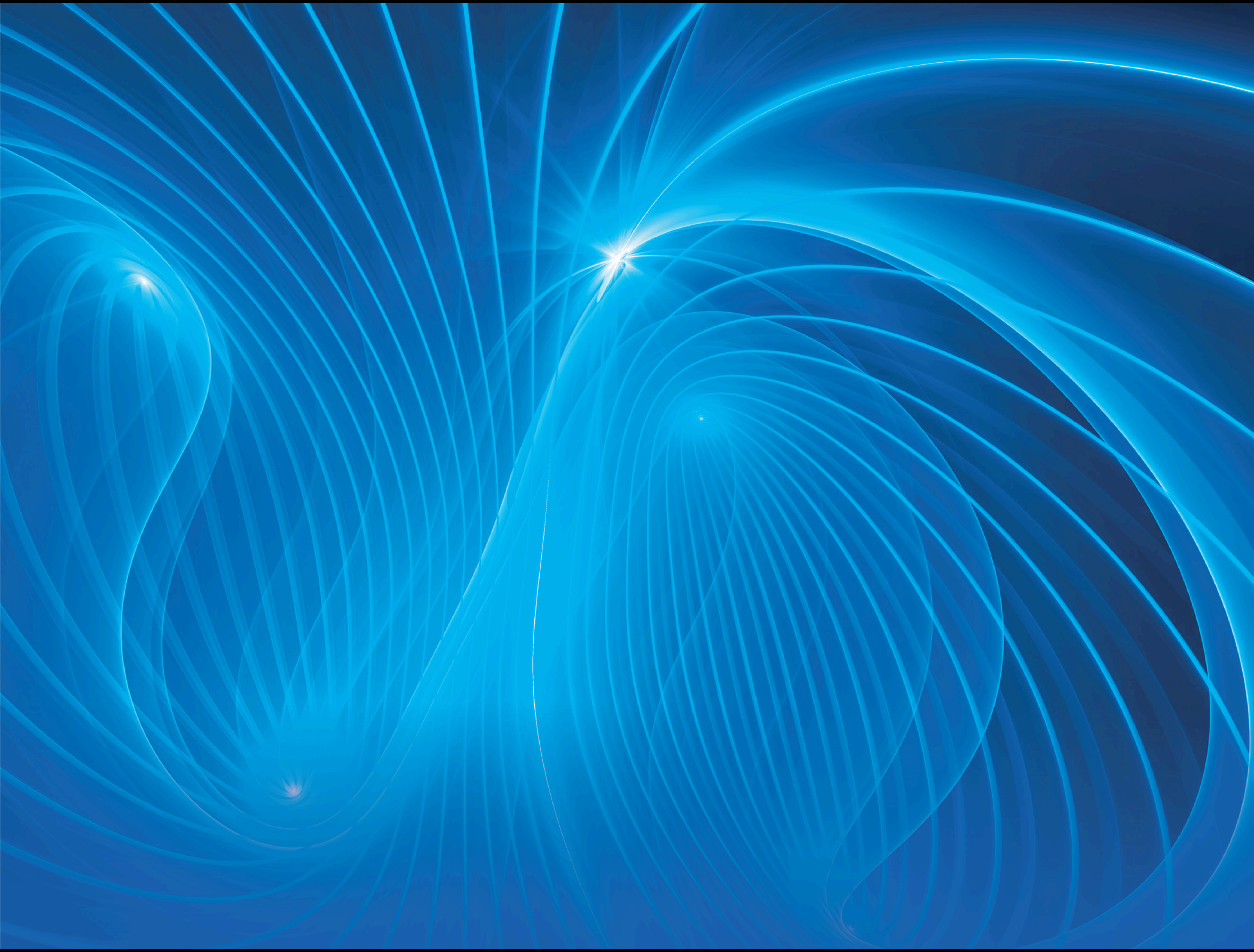
Wireless Power Transfer

Emerging Technologies and Applications of Wireless Power Transfer

Lead Guest Editor: N. Venkateswaran
Guest Editors: Narushan Pillay, Vinoth Babu Kumaravelu, Vijayakumar Ponnusamy, and Varun P. Gopi



Emerging Technologies and Applications of Wireless Power Transfer

Emerging Technologies and Applications of Wireless Power Transfer

Lead Guest Editor: N. Venkateswaran
Guest Editors: Narushan Pillay, Vinoth Babu Kumaravelu, Vijayakumar Ponnusamy, and Varun P. Gopi

Chief Editor

Academic Editors

Advisory Board Member(s)

Contents

Hindawi
Wireless Power Transfer
Volume 2022, Article ID 9921885, 10 pages
https://doi.org/10.1155/2022/9921885

CAMBRIDGE
UNIVERSITY PRESS

Research Article

Optimal Path Planning for Wireless Power Transfer Robot Using Area Division Deep Reinforcement Learning

Yuan Xing,[1] Riley Young,[1] Giaolong Nguyen,[1] Maxwell Lefebvre,[1] Tianchi Zhao,[2] Haowen Pan,[3] and Liang Dong[4]

[1]*Department of Engineering and Technology, University of Wisconsin-Stout, Menomonie, WI 54751, USA*
[2]*Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721, USA*
[3]*Changzhou Voyage Electronics Technology LLC, Changzhou, China*
[4]*Department of Electrical and Computer Engineering, Baylor University, Waco, TX 76706, USA*

Correspondence should be addressed to Yuan Xing; xingy@uwstout.edu

Received 26 October 2021; Accepted 31 January 2022; Published 4 March 2022

Academic Editor: Narushan Pillay

This paper aims to solve the optimization problems in far-field wireless power transfer systems using deep reinforcement learning techniques. The Radio-Frequency (RF) wireless transmitter is mounted on a mobile robot, which patrols near the harvested energy-enabled Internet of Things (IoT) devices. The wireless transmitter intends to continuously cruise on the designated path in order to fairly charge all the stationary IoT devices in the shortest time. The Deep Q-Network (DQN) algorithm is applied to determine the optimal path for the robot to cruise on. When the number of IoT devices increases, the traditional DQN cannot converge to a closed-loop path or achieve the maximum reward. In order to solve these problems, an area division Deep Q-Network (AD-DQN) is invented. The algorithm can intelligently divide the complete charging field into several areas. In each area, the DQN algorithm is utilized to calculate the optimal path. After that, the segmented paths are combined to create a closed-loop path for the robot to cruise on, which can enable the robot to continuously charge all the IoT devices in the shortest time. The numerical results prove the superiority of the AD-DQN in optimizing the proposed wireless power transfer system.

1. Introduction

The wireless power transfer technique is proved to be the most effective solution to the charging problem as the number of the IoT devices grows drastically, since it is impossible to replace the batteries of all IoT devices [1]. In recent years' Consumer Electronics Show (CES), a large number of wireless power transfer products have come into consumers' sights. There are two types of wireless power transmission products: near-field and far-field. In near-field wireless power transfer, the IoT devices, which are charged by resonant inductive coupling, have to be placed very close to the wireless transmitters (less than 5 cm) [2]. In far-field wireless power transfer, the IoT devices use the electromagnetic waves from transmitters as the power resource and the effective charging distance ranges from 50 centimeters to 1.5 meters [3–5]. Compared to the near-field transmitters, the far-field wireless power transmitters can charge the IoT devices (including the mobile IoT devices) that are deployed in a larger space.

However, the far-field wireless power transfer is still in its infancy for two reasons. First, the level of power supply is very low due to the long distance between the power transmitters and the energy harvesters. In [6], the authors mentioned that the existing far-field RF energy harvesting technologies can only achieve nanowatts-level power transfer, which is too tiny to power up the high-power-consuming electronic devices. In [3], the authors investigated the RF beamforming in radiative far field for wireless power transfer. The authors demonstrated that, with beamforming techniques, the level of the energy harvesting can be boosted. However, as the distance between the transceivers increases to 1.5 meters, the amount of the harvested energy is less than 5 milliwatts, which is still not ideal to power up the high-energy-consuming devices. Second, most of the existing wireless charging systems can only effectively charge stationary energy harvesters. In [7], a set of wireless chargers (Powercast [8]) are deployed on the square area. The Powercast transmitters can adjust the transmission strategies to optimize energy harvested at the stationary energy harvesters. In [9], the Powercast wireless charger is mounted on the moving robot. Therefore, the charger is a mobile wireless charger, which can adjust the transmission patterns of the stationary sensors while moving. However, the number of the IoT devices to be charged is too small. In order to wirelessly charge multiple IoT devices, some researchers proposed using Unmanned Aerial Vehicle (UAV) to implement the wireless power transfer [10–13]. The UAV is designed to plan the optimal path to charge the designated IoT devices. However, it is very inefficient to use UAV to charge the IoT devices, since UAV has very high power consumption and very short operational time. Installing the wireless power emitter on the UAV will further shorten the operational time of UAV.

In order to enhance the level of the energy harvesting and efficiency in charging a large number of energy-hungry IoT devices, in this paper, we assembled the wireless power transfer robot and applied deep reinforcement learning algorithm to optimize its performance. In the system, the wireless transmitter aims to find the optimal path for the wireless power transfer robot. The robot cruises on the path, which can charge each IoT device in the shortest time. DQN has been widely used to play the complicated games which have a large number of system states even when the environment information is not entirely available [14]. Lately, a lot of researchers have started to implement DQN in solving the complicated wireless communication optimization problems because the systems are very complicated and environment information is time-varying and hard to capture [15–18]. In particular, the researchers applied deep reinforcement learning to plan the optimal path for auto-drive robots [19–22] and the robots can quickly converge to the optimal path. Henceforth, we found that DQN is a perfect match to solve our proposed optimization problem. However, those papers either only proposed the theoretical model or could not implement wireless power transfer functions. To the best of our knowledge, we are the first ones to implement the automatic far-field wireless power transfer system in the test field and invent a DQN algorithm to solve it. In our system, the entire test field is evenly quantified into the square spaces. The time is slotted with the same interval. We consider the relative location of the robot in the test field as the system state, while we define the direction to move in the next time slot. At the beginning of each time slot, the wireless power transfer robot generates the system state and takes it as the input to DQN. The DQN can generate the Q values for each possible action and the one with the maximum Q value is picked to guide robot's move during the current time slot.

As the number of IoT devices increases and the testing field becomes more complicated, the traditional DQN cannot generate the close-loop path for the robot to cruise on, which does not satisfy the requirement of charging every regular time interval. In order to deal with this problem, area division deep reinforcement learning is proposed in this paper. At first, the algorithm divides the whole test field into several areas. In each area, DQN is utilized to calculate the optimal path. Next, the entire path is formulated with the paths of each separated area. In this way, a closed loop is guaranteed and the numerical results prove that the calculated path is also the optimal path.

2. System Model

The symbols used in this paper and the corresponding explanations are listed in Table 1.

As shown in Figure 1, a mobile robot that carries two RF wireless power transmitters cruises on the calculated path to radiate the RF power to K nearby RF energy harvesters. Both the power transmitter and the RF power harvesters are equipped with one antenna. The power received at receiver k, $k \in \mathscr{K} = \{1, 2, \ldots, K\}$, is

$$p_k = \frac{\eta G_{\mathrm{tx}} G_{\mathrm{rx}} (\lambda/4\pi)^2}{l_p (L + \mu)^2} p_{\mathrm{tx}}, \tag{1}$$

where p_{tx} is the transmit power; G_{tx} is the gain of the transmitter's antenna; G_{rx} is the gain of the receiver's antenna; L is the distance between the transmitter and harvester k; η is the rectifier efficiency; λ denotes the wavelength of the transmitted signal; l_p denotes the polarization loss; μ is the adjustable parameter due to Friis's free space equation. Since the effective charging area is critical in determining the level of energy harvesting and it is the parameter to be adjusted at the transmitter, equation (1) is reexpressed using the effective area:

$$p_k = \frac{\eta S_{\mathrm{tx}} S_{\mathrm{rx}} \cos \alpha}{l_p \lambda^2 (L + \mu)^2} p_{\mathrm{tx}}, \tag{2}$$

where S_{tx} is the maximum effective transmit area; S_{rx} is the effective received area; α is the angle between the transmitter and the vertical reference line.

Since we consider the mobile energy harvesters in the system, the distance and effective charging area may vary over the time; we assume that the time is slotted and the position of any mobile device within one time slot is constant. In time slot n, the power harvested at receiver k can be denoted as

$$p_k(n) = \frac{\eta S_{\mathrm{tx}} S_{\mathrm{rx}} \cos \alpha(n)}{l_p \lambda^2 (L(n) + \mu)^2} p_{\mathrm{tx}}. \tag{3}$$

For a mobile energy harvester, the power harvested in different time slots is determined by the angle between the transmitter and the vertical reference line $\alpha(n)$ together with the distance between the transmitter and the harvester $L(n)$ in the time slot.

In our model, the mobile transmitter is free to adjust the transmit angle $\alpha(n)$ and $L(n)$ as it can move around the IoT devices. We assume that the effective charging is counted only when $\alpha(n) = 0$ and $L(n) < \, = 45$ cm.

Table 1: Symbols and explanations.

Symbol	Explanation
K	The number of energy harvesters
η	Rectifier efficiency
G_{tx}	Gain of transmitter's antenna
G_{rx}	Gain of receiver's antenna
λ	Wavelength of transmitted signal
l_p	Polarization loss
μ	Friis's free space adjustable parameter
L	Distance between transmitter and harvester
p_{tx}	Transmit power
p_k	Received power
α	Angle between transmitter and the vertical reference line
S_{tx}	Maximum effective transmit area
S_{rx}	Effective received area
n	Time instant
$\mathbf{pos}(h, v)$	Position h and v units to left and upmost edges
$\mathbf{o}_k$	Position of kth energy harvester
$\mathbf{eff}_k$	Effective charging area for kth IoT devices
s	Present system state
s'	Next system state
$\mathbf{a}^n$	Action taken at n
T	Total time consumption
$p_{s,s'}(\mathbf{a})$	Transition probability from state s to state s' taking action $\mathbf{a}$
$w(s, \mathbf{a}, s')$	Reward function at state s taking action $\mathbf{a}$
$\mathbf{acc}_{\mathbf{o}_{k-1}}$	Indicator whether $k-1$ harvesters have been charged
ζ	Unit price for reward function
π	Optimal strategy
$Q(s, \mathbf{a})$	Cost function at state s taking action $\mathbf{a}$
γ	Reward decay
$\sigma(s', \mathbf{a})$	Learning rate for Q-learning
$\mathbf{p}_i$	Selected location for ith area
$\mathscr{W}_i$	ith area

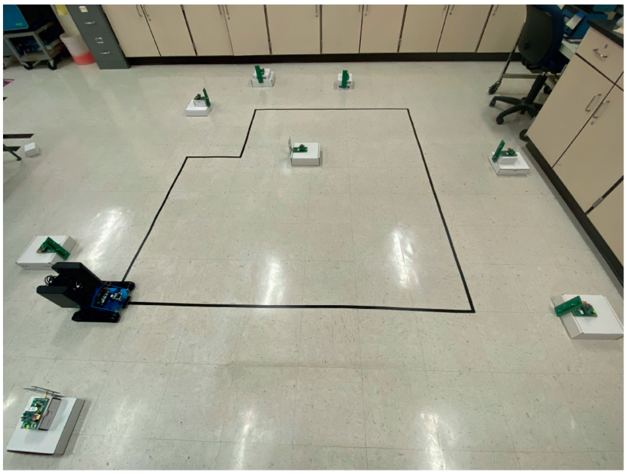


Figure 1: Mobile wireless power transmitter cruises on the calculated path to charge multiple harvested energy-enabled IoT devices.

3. Problem Formulation

In this paper, the optimization problem is formulated as a Markov Decision Process (MDP) and reinforcement learning (RL) algorithm is utilized to solve the problem. Furthermore, DQN algorithm is applied to address the large number of system states.

3.1. Problem Formulation. In order to model our optimization problem as an RL problem, we define the test field consisting of same area unit square, whose side length is 30 cm. $K = 8$ harvested energy-enabled IoT devices are deployed in the test field, whose orders are 0, 1, 2, 3, 4, 5, 6, and 7, respectively. The map is shown in Figure 2. The system state s^n at time slot n is defined as the position of a particular square where the robot is currently located at in the test field, which is specified as $s^n = \mathbf{pos}(h, v)$, where h is the distance between the present square and the leftmost edge, which is counted by the number of squares. v indicates the distance between the present square and the upmost edge, which is counted by the number of squares. For example, the No. 5 IoT device can be denoted as $\mathbf{o}_5 = \mathbf{pos}(2, 0)$. The shadow area adjacent to No. k IoT devices indicates the effective charging area for the respective IoT devices, which is denoted as $\mathbf{eff}_k$. For example, the boundary of effective charging areas for No. 6 IoT device is highlighted in red. We define the direction of movement in a particular time slot n as the actions $\mathbf{a}^n$. The set of possible actions $\mathscr{A}$ consists of 4 different $\mathscr{A} = \{\mathbf{U}, \mathbf{D}, \mathbf{L}, \mathbf{R}\}$, where $\mathbf{U}$ is moving upward one unit, $\mathbf{D}$ is moving downward one unit, $\mathbf{L}$ is moving left one unit, and $\mathbf{R}$ is moving right one unit.

Given the above, the mobile wireless charging problem can be formulated as minimizing the time duration T for the robot to complete running one loop at the same time the robot has to pass through one of the effective areas of each IoT device.

$$\mathscr{P}_1: \quad \begin{array}{ll} \underset{\{\mathbf{a}^n\}}{\text{minimize}} & T, \\ \text{subject to} & s^0 = s^T, \\ & \exists s^n \in \mathbf{eff}_k, \forall k \in \mathscr{K}, n = 1, 2, \ldots, T. \end{array} \tag{4}$$

Time duration for the robot to complete running one loop is defined as T. The starting position is the same as the last position, since the robot cruises in a loop. In the loop, the robot has to pass through at least one of the effective charging areas of each IoT device.

Adapting to the different positions, the agent chooses different action at each time slot. Henceforth, we can model our proposed system as a Markov chain. In the system, we use the current position to specify a particular state s. $\mathscr{S}$ denotes the system state set. The starting state s^0 and final state s^T are the same, since the robot needs to move and return to the starting point. The MDP process can be described as the agent chooses an action $\mathbf{a}$ from $\mathscr{A}$ at a specific system state s. After that, a new system state s' will be transit into. $p_{s,s'}(\mathbf{a})$, $s, s' \in \mathscr{S}$ and $\mathbf{a} \in \mathscr{A}$, denotes the probability that system state transits from s to s' with $\mathbf{a}$.

The reward of the MDP is denoted as $w(s, \mathbf{a}, s')$, which is defined for the condition that system state transits from s to state s'. The optimization problem is formulated as reaching s^T in the fewest transmission time slots; henceforth, the reward has to be defined to motivate the mobile robot that

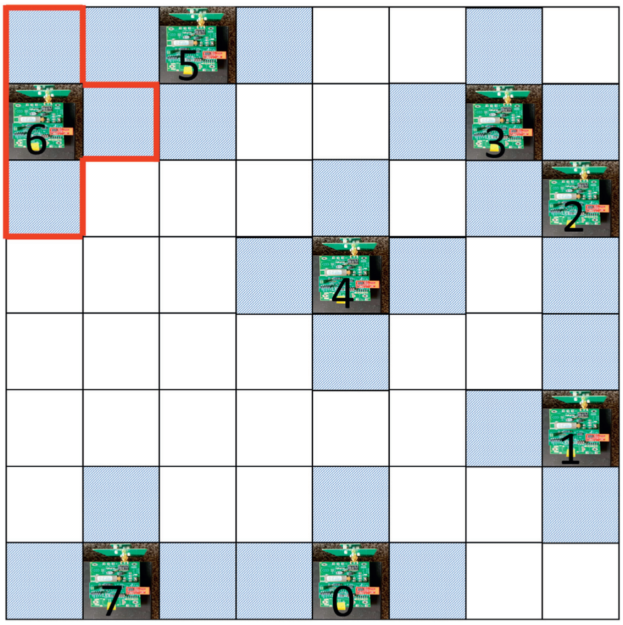


Figure 2: The entire test field consists of same space unit square. $K = 8$ harvested energy-enabled IoT devices are deployed in the test field. The shadow area adjacent to each IoT device indicates the effective charging area for the respective IoT devices. For example, the boundary of effective charging areas for No. 6 IoT device is highlighted in red.

does not repeatedly pass through any effective charging area of any IoT devices. Besides, the rewards at different positions are interconnected with each other, since the goal of the optimization is to pass through the effective charging areas of all the IoT devices. We assume that the optimal order to pass through all the IoT devices is defined as $\mathbf{o}_0, \mathbf{o}_1, \ldots, \mathbf{o}_7$. $\mathbf{o}_k = 0, 1, \ldots, 7$. Specifically, the reward function can be expressed as

$$w\left(s, \mathbf{a}, s'\right) = \begin{cases} \mathbf{o}_k \zeta, & s' \in \mathbf{eff}_{\mathbf{o}_k}, \mathbf{acc}_{\mathbf{o}_{k-1}} = 1, \\ -1, & \text{otherwise}. \end{cases} \quad (5)$$

In the above equation, $\mathbf{acc}_{\mathbf{o}_{k-1}} = 1$ if the robot has already passed through as effective area of the $\mathbf{o}_{k-1}$th IoT device; and ζ denotes the unit price of the harvested energy.

As we have defined all the necessary elements for MDP, we can characterize the formulated problem as a stochastic shortest path search that starts at s^0 and ends at s^T. At each system state s, we derive the best action $\mathbf{a}^*(s)$ which can generate the maximum reward. The optimal policy sets are defined as $\pi = \{\mathbf{a}(s): s \in \mathcal{S}\}$.

3.2. Optimal Path Planning with Reinforcement Learning. If the systematic dynamics obey a specific transition probability, reinforcement learning will be the perfect match to solve the optimization problems. In this section, Q-learning [23] is first introduced to solve the proposed problem. After that, to address the large states and actions sets, the DQN algorithm [14] is utilized to determine the optimal action for each particular system state.

3.2.1. Q-Learning Method. The traditional Q-learning method is widely used to solve the dynamic optimization problem provided that the number of the system states is moderate. Corresponding to each particular system state, the best action can be determined to generate the highest reward.

$Q(s, \mathbf{a})$ denotes the cost function, which uses a numerical value to describe the cost of taking action $\mathbf{a}$ at state s. At the beginning of the algorithm, all the cost function is zero since no action has ever been taken to generate any consequence $Q(s, \mathbf{a}) = 0$. All the Q values are saved in the Q table. Only one cost function is updated in each time slot as the action is taken and the reward function is calculated. The cost function is updated as

$$\begin{aligned} Q(s, \mathbf{a}) &= (1 - \sigma(s, \mathbf{a})) Q(s, \mathbf{a}) \\ &\quad + \sigma(s, \mathbf{a}) \left[w\left(s, \mathbf{a}, s'\right) + \gamma f\left(s', \mathbf{a}\right) \right], \end{aligned} \quad (6)$$

where

$$f\left(s', \mathbf{a}\right) = \max_{\mathbf{a} \in \mathcal{A}} Q\left(s', \mathbf{a}\right). \quad (7)$$

The learning rate is defined as $\sigma(s', \mathbf{a})$.

When the algorithm initializes, the Q table is empty since no exploration has been made to obtain any useful cost function to fill the Q table. Since the agent has no experience about the environment, the random action selection is implemented at the beginning of the algorithm. A threshold $\epsilon_c \in [0.5, 1]$ is designed to start the exploration. In each time slot, a numerical value $p \in [0, 1]$ is generated and compared with the threshold. If $p \geq \epsilon_c$, action $\mathbf{a}$ is picked as

$$\mathbf{a} = \max_{\mathbf{a} \in \mathcal{A}} Q(s, \mathbf{a}). \quad (8)$$

However, provided that $p < \epsilon_c$, an action is randomly selected from the action set $\mathcal{A}$.

After iteratively updating the value in the Q table, the Q value converges. We can calculate the best action corresponding to each action and state by

$$\pi^*(s) = \arg \max_{\mathbf{a} \in \mathcal{A}} Q^*(s, \mathbf{a}), \quad (9)$$

which corresponds to finding the optimal moving direction for each system state explored during the charging process.

3.2.2. DQN Algorithm. The increase in the number of IoT devices has led to an increase in the number of system states. Suppose that Q-learning algorithm is used; a very large Q table has to be created and the convergence speed is too slow. DQN algorithm is more compatible since there is a deep neural network in the structure that can be well trained and take immediate action to determine the best action that is going to be taken.

The deep neural network in the structure has the system state as the input and the Q value for each action is defined as the output. Henceforth, the function of the neural network is to generate the cost function for particular state and action. We can describe the cost function as $Q(s, \mathbf{a}, \theta)$, where θ is the weight on the neuron nodes in the structure. As we collect the data when different actions are taken in different time slot, the neural network is trained to update the weight of the neural network, which can output a more precise Q value:

$$Q(s, \mathbf{a}, \theta) \approx Q^{*}(s, \mathbf{a}). \tag{10}$$

There are two identical neural networks existing in the structure of DQN [24]: one is called the evaluation network eval_net, and the other is called the target network target_net. Since these two deep neural networks have the same structure, multiple hidden layers are defined for each network. We use the current system state s and the next system state s' as the input to eval_net and target_net, respectively. We use $Q_e(s, \mathbf{a}, \theta)$ and $Q_t(s, \mathbf{a}, \theta')$ to define the output of two deep neural networks eval_net and target_net. In the structure, in order to update the value of the weight of neuron nodes, we only continuously train the evaluation network eval_net. The target network is not trained. It periodically duplicates the weights of the neurons from the evaluation network (i.e., $\theta' = \theta$). The loss function is described as follows, which is used to train eval_net:

$$\mathrm{Loss}(\theta) = E\left[\left(y - Q_e(s, \mathbf{a}, \theta)\right)^2\right]. \tag{11}$$

We use y to represent the real Q value, which can be expressed as

$$y = w(s, \mathbf{a}, s') + \epsilon \max_{\mathbf{a}' \in \mathscr{A}}, Q_t(s', a', \theta'). \tag{12}$$

We denote the learning rate as ϵ. The idea of backpropagation is utilized to update the weight of eval_net; as a result, the neural network is trained.

The experience reply method is utilized to improve the training effect, since it can effectively eliminate the correlation among the training data. Each single experience includes the system state s, the action $\mathbf{a}$, and the next system state s', together with the reward function $w(s, \mathbf{a}, s')$. We define the experience set as $\mathrm{ep} = \{s, \mathbf{a}, w(s, \mathbf{a}, s'), s'\}$. In the algorithm, D individual experiences are saved and, in each training epoch, only D_s (with $D_s < D$) experiences are selected from D. As the training process is completed, target_net copies the weight of the neurons from the evaluation network (i.e., $\theta' = \theta$). D different experiences are generated from ep, while only D_s are picked to train the evaluation network eval_net. The total number of training iterations can be denoted as U. Both evaluation network and target network share the same structure, in which the deep neural networks have N_l hidden layers.

3.2.3. Dueling Double DQN. In order to leverage the performance of DQN, which can effectively select the optimal action to charge multiple harvesters in a time-varying channel conditions, we redesign the structure of the deep neural network by using Dueling Double DQN. Doubling DQN is an advanced version of DQN which can prevent the overestimating problem appearing throughout the training [24]. Dueling Double DQN can efficiently solve the overestimating problem throughout the training process. In the same training epochs, Dueling Double DQN is proved to outperform the original DQN in learning efficiency.

In traditional DQN, as shown in equation (12), the target network target_net is designed to derive the cost function for a particular system state. Nevertheless, because we do not update the weight of the target network target_net in each training epoch, the training error will increase while training, hence prolonging the training procedure. In Doubling DQN, both the target network target_net and the evaluation network eval_net are used to calculate the cost functions. We use evaluation network eval_net to calculate the best action for system state s'.

$$y = w(s, \mathbf{a}, s') + \epsilon \max_{\mathbf{a}' \in \mathscr{A}} Q_e\left(s', \arg_{\mathbf{a} \in \mathscr{A}} \max Q(s', \mathbf{a}, \theta), \theta'\right). \tag{13}$$

The latest research proves that the training error can be dramatically reduced using the Doubling DQN structure [24].

In traditional DQN, we only define the cost function Q value as the output of the deep neural network. The Dueling DQN is invented to speed up the convergence of the deep neural network by designing 2 individual streams of the output for the deep neural network. We use the output value $V(\mathbf{s}, \theta, \beta)$ to represent the first stream of the neural network. It denotes the cost function for a specific system state. We name the second stream of the output as advantage output $A(s', \mathbf{a}, \theta, \alpha)$, which is utilized to illustrate the advantage of using a specific action to a system state [25]. We define α and β as the parameters to correlate the output of two streams and the neural network. The cost function can be denoted as

$$\begin{aligned} Q(s, \mathbf{a}, \theta, \alpha, \beta) &= V(s, \theta, \beta) \\ &\quad + \left(A(s', \mathbf{a}, \theta, \alpha) - \frac{1}{|A|}\sum_{a'} A(s', \mathbf{a}, \theta, \alpha)\right). \end{aligned} \tag{14}$$

The latest research proves that Dueling DQN can speeds up the training procedure by efficiently annihilating the additional freedom while training the deep neural network [25].

3.3. Area Division Deep Reinforcement Learning. In this paper, the optimization problem can be seen as calculating the optimal close-loop path which generates the maximum accumulated reward. However, the traditional DQN shows the difficulty converging to the optimal path because of the complicated experimental field. In order to leverage the performance of traditional DQN, we invent an AD-DQN in this paper. At first, the experimental field is divided into multiple separate parts. DQN is run on each part individually to obtain the optimal path for the robot, respectively. Finally, the entire close-loop path is formulated using the path on each part. In area division, the whole area is defined as $\mathscr{W}$. The whole area is divided at multiple specific locations. $\mathbf{p}_i \in \mathscr{P}$.

The criterion to pick $\mathbf{p}_i$ is finding the squares, which exist in more than one effective charging area of the IoT devices.

$$\begin{aligned} &\forall \mathbf{p}_i \in \mathscr{P}, \\ &\quad \mathbf{p}_i \in \mathbf{eff}_m, \\ &\quad \mathbf{p}_i \in \mathbf{eff}_n, \\ &m, n = 0, 1, \ldots, K-1, m \neq n. \end{aligned} \tag{15}$$

For each $\mathbf{p}_i$, we define $\mathscr{N}_i = \{\mathbf{p}_i\}$. We define set $\mathscr{K}_e = \left\{\mathbf{o}_{\arg \mathbf{p}_i \in \mathbf{eff}_j,\, j=0,1,\ldots,K-1}\right\}$. In the clockwise direction, we find that the IoT device $\mathbf{o}_i$ has the shortest distance to $\mathbf{p}_i$, and then add both $\mathbf{o}_i$ and the effective charging area of $\mathbf{o}_i$ to $\mathscr{N}_i$. The new area can be expressed as

$$\mathscr{N}_i = \mathscr{N}_i \cup \{\mathbf{o}_i\} \cup \{\mathbf{eff}_i\}. \tag{16}$$

Next, we find the IoT device having the shortest distance to the IoT device $\mathbf{o}_i$ that is just added to set $\mathscr{N}_i$, and then add both the new IoT device and the effective charging area of it to $\mathscr{N}_i$. Iteratively, we find that all the IoT devices besides the ones in $\mathscr{K}_e$ are included in one $\mathscr{N}_i$. Finally, classify all the rest squares to the nearest $\mathscr{N}_i$. $\{\mathscr{N}_i\} = \mathscr{W}$.

In each area, the DQN is run to determine the optimal path for the robot. In each area, the starting point is the same as the position of $\mathbf{p}_i$; the end point is one of the effective charging squares of the furthest IoT device from the starting point in the same area. After the optimal path is calculated for each individual area, the close-loop optimal path for the entire area can be synthesized. The algorithm is shown in Algorithm 1.

```
(i) Define $\mathscr{E} = \{\mathbf{eff}_k, k = 0, 1, \ldots, K-1\}$. Among $\mathscr{E}$, find all the area division points $\mathbf{p}_i$ by $\{\mathbf{p}_i\} = \{\mathbf{pos}(h, v) | \mathbf{pos}(h, v) \in \mathbf{eff}_m, \mathbf{pos}(h, v) \in \mathbf{eff}_n, m, n \in \mathscr{K}\}$
(ii) The number of area divison points is defined as $|\mathscr{P}|$.
(iii) $i = 1, \ldots, |\mathscr{P}|$. The number of the area to be divided is $|\mathscr{P}| + 1$.
(iv) $\mathscr{K}_e = \left\{\mathbf{o}_{\arg \mathbf{p}_i \in \mathbf{eff}_j,\, j=0,1,\ldots,K-1}\right\}$.
(v) for $i = 1, \ldots, |\mathscr{P}|$:
(vi) $\mathbf{r}_1 = \mathbf{p}_i$. $\mathbf{r}_2 = \mathbf{p}_i$.
(vii)     while $\nexists \mathbf{o}_g \in \mathscr{N}_i, \mathbf{o}_g \in \mathscr{N}_{\mathscr{I} \setminus \{i\}}$
(viii)        if $i < \; = |\mathscr{P}|$
(ix) In the clockwise direction, find the the IoT devices, that has the shortest distance to $\mathbf{r}_1$. The order of the IoT device is: $g = \operatorname{argmin}_{\mathbf{o}_i \notin \mathscr{K}_e} |\mathbf{o}_i - \mathbf{r}_1|$. $\mathscr{N}_i$ is updated as: $\mathscr{N}_i = \mathscr{N}_i \cup \{\mathbf{o}_g\} \cup \{\mathbf{eff}_g\}$. $\mathbf{r}_1 = \mathbf{o}_g$.
(x)        else
(xi) In the counterclockwise direction, find the the IoT devices, that has the shortest distance to $\mathbf{r}_2$. The order of the IoT device is: $g = \operatorname{argmin}_{\mathbf{o}_i \notin \mathscr{K}_e} |\mathbf{o}_i - \mathbf{r}_2|$. $\mathscr{N}_i$ is updated as: $\mathscr{N}_i = \mathscr{N}_i \cup \{\mathbf{o}_g\} \cup \{\mathbf{eff}_g\}$. $\mathbf{r}_2 = \mathbf{o}_g$.
(xii)        en d
(xiii)    en dw hile
(xiv) en d
(xv) for $i = 1, \ldots, |\mathscr{P}|$:
(xvi) Define set $\mathscr{W}_i$
(xvii) $\mathscr{W}_i = \mathscr{N}_i \cup \{\mathbf{pos}(h, v) | \arg_{h,v} \min_{\mathbf{pos}(h,v) \in \mathscr{W}} |\mathbf{pos}(h, v) - \mathbf{o}_k|, k \notin \mathscr{K}_e\}$
(xviii) en d
(xix) for $i = 1, 2, \ldots, |\mathscr{P}| + 1$:
(xx)     for $j = 1, 2, \ldots, |\mathscr{J}|$:
(xxi) The starting point is defined as $\mathbf{p}_i$. The end point is defined as $e_j \in \mathbf{eff}_c$, $j \in \mathscr{J}$.
(xxii) The weight of the neuron nodes $\theta$ are randomly generated for the eval_net and the weights are copied by target_net$\theta' = \theta$. $u = 1$. $D = d = 1$.
(xxiii)    while $u < U$ $\mathbf{s} = s^0$. $t = 1$.
(xxiv) A probability is generated as a numerical parameter $p \in [0, 1]$.
(xxv)      if $D > 200$ an d $p \geq \epsilon_{ch}$
(xxvi) $\mathbf{a} = \max_{\mathbf{a} \in \mathscr{A}} Q(s, \mathbf{a})$
(xxvii)    else
(xxviii) Randomly choose the action from action set $\mathscr{A}$.
(xxix)     en d
(xxx)      while $s' \neq s^T$
(xxxi) The state transit into $s'$ after taking the action. $d = d + 1$. $\mathrm{ep}(d) = \{s, \mathbf{a}, w(s, \mathbf{a}, s'), s'\}$. Suppose $D$ keeps unchanged if it goes over the experience pool's limitation, $d = 1$; otherwise, \$ $D = d$\$. $t = t + 1$. $s = s'$. After enough data has been collected in experience pool, eval_net is trained using $D$ of $D_s$ experiences. Minimize the loss function $\mathrm{Loss}(\theta)$ using Back-propagation. target_net copies the weight from eval_net periodically.
(xxxii)    en dw hile
(xxxiii) en dw hile
(xxxiv) The optimal path of the entire test field is synthesized with the optimal path in each $\mathscr{W}_i$.
```

4. Experimental Results

The implementation of the proposed wireless power transfer system is shown in Figure 3.

In the test field, 8 harvested energy-enabled IoT devices are placed as Figure 2 indicates. The top view of the test field can be seen as a 2D map. Henceforth, the map is modeled and inputted into the computer. Then the AD-DQN algorithm is implemented in computer using Python and the optimal charging path can be derived. At the same time, a wireless power transfer robot is assembled. Two Powercast RF power transmitters TX91501 [8] are mounted on two sides of the Raspberry Pi [26] enabled intelligent driving robot. Each transmitter is powered by 5 V power bank and continuously emits 3 Watts RF power. The infrared patrol

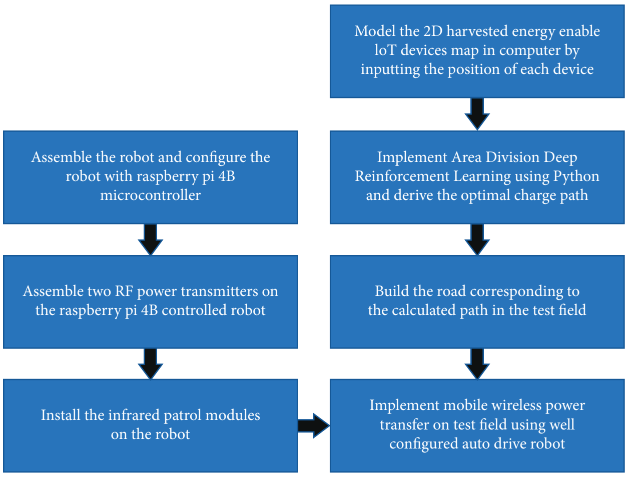

Model the 2D harvested energy enable IoT devices map in computer by inputting the position of each device
Assemble the robot and configure the robot with raspberry pi 4B microcontroller
Implement Area Division Deep Reinforcement Learning using Python and derive the optimal charge path
Assemble two RF power transmitters on the raspberry pi 4B controlled robot
Build the road corresponding to the calculated path in the test field
Install the infrared patrol modules on the robot
Implement mobile wireless power transfer on test field using well configured auto drive robot

Figure 3: Flowchart of wireless power transfer implementation.

module is installed on the robot to implement the autodrive on the test field; henceforth, the robot can automatically cruise on along the path and continuously charge the multiple IoT devices, as shown in Figure 1. To the best of our knowledge, we are the first ones to implement the automatic wireless power transfer system in the test field and invent AD-DQN algorithm to design the optimal path for the wireless power transfer robot. Since we are the first ones to design and implement the mobile far-field wireless power transfer system, there is no hardware reference design we can refer to and use for validation. So the validation of our work is done in the software aspect. But referring to the flowchart, our mobile wireless power transfer system can be replicated.

For the software, we use TensorFlow 0.13.1 together with Python 3.8 in Jupyter Notebook 5.6.0 as the software simulation environment to train the AD-DQN. The number of hidden layers is 4 and each hidden layer owns 100 nodes. The learning rate is less than 0.1. The mini-batch size is 10. The learning frequency is 5. The training starting step is 200. The experience pool is greater than 20000. The exploration interval is 0.001. The target network replacement interval is greater than 100. Reward decay is 0.99.

First, different reward functions are tested for the optimal one. Reward one $\mathbf{reward}_1$ is defined using equation (5). The unit price is defined as $\zeta = 4$. Reward two $\mathbf{reward}_2$ is defined as

$$w_2(s, \mathbf{a}, s') = \begin{cases} \zeta, & s' \in \mathrm{eff}_{\mathbf{o}_k}, \mathbf{acc}_{\mathbf{o}_{k-1}} = 1, \\ -1, & \text{otherwise}, \end{cases} \tag{17}$$

where $\zeta = 4$. Reward three $\mathbf{reward}_3$ is defined with equation (5); however, $\zeta = 2$. Two factors are observed for the performance of different rewards, which are average reward during the training and average time consumption during the training.

Based on the procedures of AD-DQN in Algorithm 1, the experimental field is divided into two areas along the only shared effective charging area for both device 2 and device 3. In area I, IoT devices 2, 3, 4, 5, and 6 are included, while in area II, IoT devices 0, 1, 2, 6, and 7 are included.

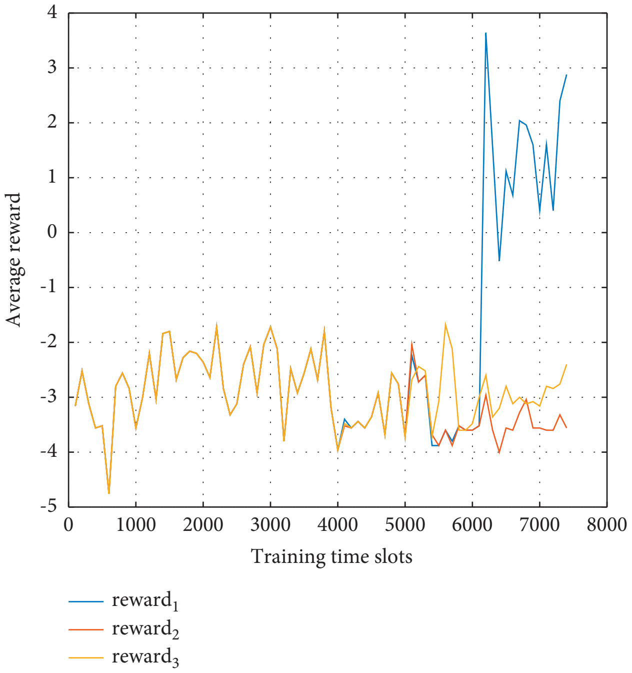


Figure 4: The average rewards of $\mathbf{reward}_1$, $\mathbf{reward}_2$, and $\mathbf{reward}_3$ versus the training episodes in area I of the experimental field.

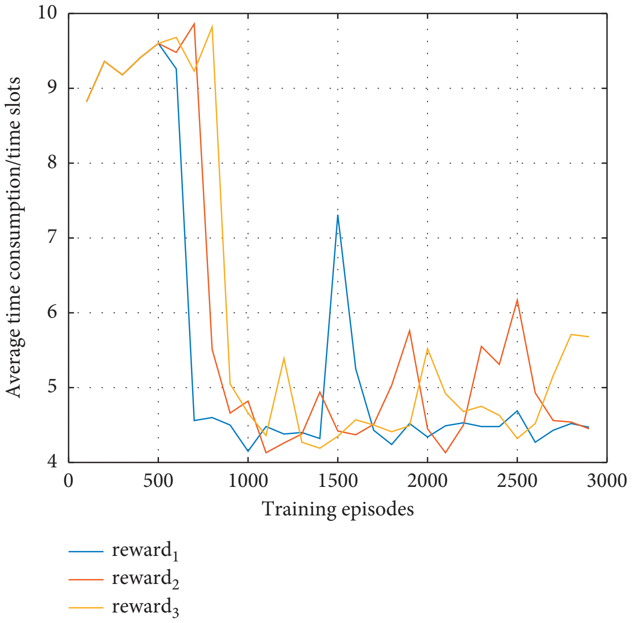


Figure 5: The average time consumption achieved by $\mathbf{reward}_1$, $\mathbf{reward}_2$, and $\mathbf{reward}_3$ versus the training episodes in area I of the experimental field.

In area I, the performances of three different rewards are compared in Figures 4 and 5.

In area II, the performances of three different rewards are compared in Figures 6 and 7.

From Figures 4 and 5, we can observe that $\mathbf{reward}_1$ is optimal. Since all three rewards perform similarly on the time consumption, $\mathbf{reward}_1$ is the highest reward among all,

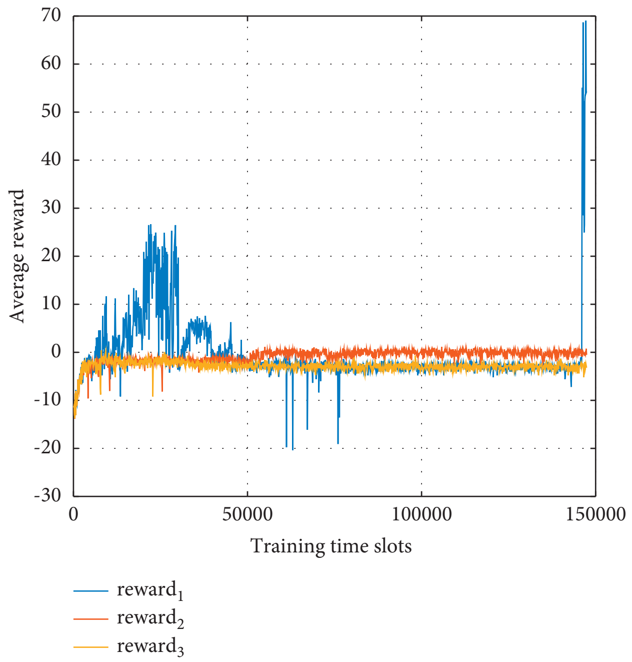


Figure 6: The average rewards of **reward**$_1$, **reward**$_2$, and **reward**$_3$ versus the training episodes in area II of the experimental field.

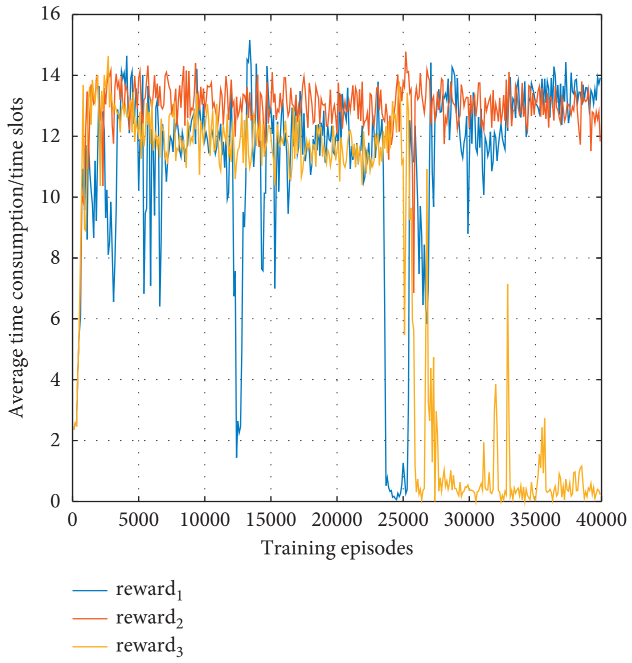


Figure 7: The average time consumption achieved by **reward**$_1$, **reward**$_2$, and **reward**$_3$ versus the training episodes in area II of the experimental field.

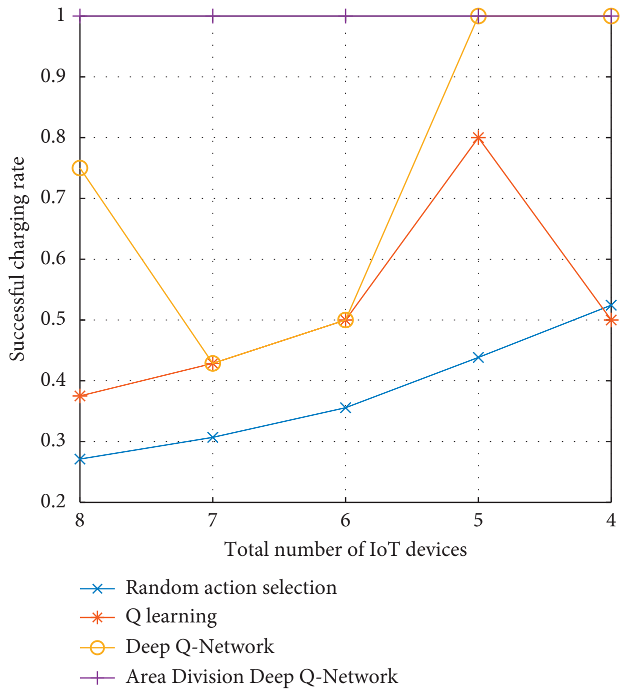


Figure 8: The effective charging rate of random action selection, Q-learning, DQN, and AD-DQN versus the total number of IoT devices.

which means that **reward**$_1$ can effectively charge most of the IoT devices compared with the other two rewards.

From Figures 6 and 7, we can observe that **reward**$_3$ performs best on the time consumption to complete one episode; however, **reward**$_1$ is much more average reward than **reward**$_3$. That can be explained as follows: compared with **reward**$_1$, **reward**$_3$ can only effectively charge fewer number of the IoT devices.

Overall, **reward**$_1$ has optimal performance in both areas I and II; henceforth, **reward**$_1$ is used to define the reward for AD-DQN.

In Figures 8 and 9, the performances of four different algorithms are compared. The random action selection randomly selects the action in the experimental test field. Same as AD-DQN, **reward**$_1$ is used as the reward of Q-learning and DQN.

We define the successful charging rate as the number of IoT devices that can be successfully charged in one complete charging episode over the total number of the IoT devices. From Figure 8, we can observe that random action selection has the worst successful charging rate. That can be explained as follows: random action selection never converges to either suboptimal or optimal path. Q-learning has a better performance than random action selection; however, it is outperformed by the other two algorithms, since Q-learning can only deal with the simple reinforcement learning model. DQN performs better than Q-learning and random action selection; however, it is outperformed by the AD-DQN, since the rewards for different states are defined as interconnected; even the reward decay is 0.99; DQN still cannot learn the optimal solution. When the total number of the IoT devices decreases, both DQN and AD-DQN perform the same since the decrease of the number of the IoT devices degrades the interconnections between different system states. From Figure 9, we can observe that, compared with the other algorithms, AD-DQN is not the one consuming

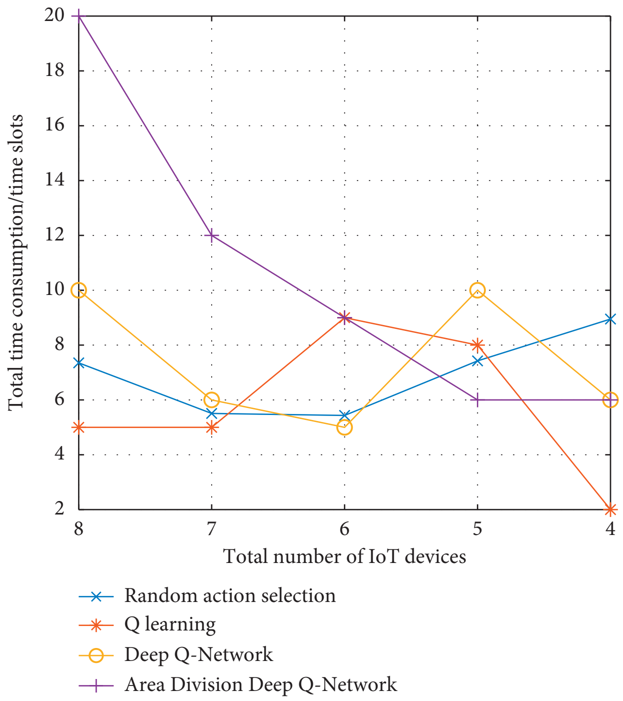


Figure 9: The average time consumption of random action selection, Q-learning, DQN, and AD-DQN versus the total number of IoT devices.

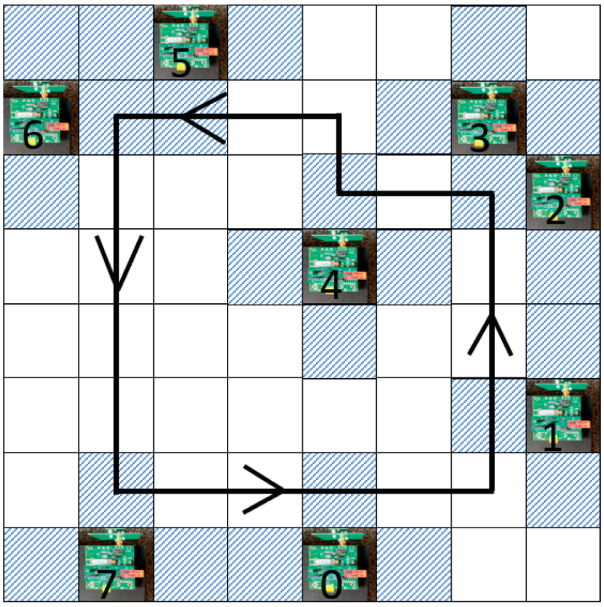


Figure 10: The optimal path determined by AD-DQN. Bold black line indicates the path for the wireless power transfer robot.

the least time slots to complete one charging episode; however, AD-DQN is still the optimal algorithm, since all the other algorithms cannot achieve 100% effective charging rate; hence they consume fewer time slots to complete one charging episode.

In Figure 10, the optimal path determined by AD-DQN is shown as the bold black line. The arrows on the path show the direction of the robot to move as we assume that the robot is regulated to cruise on the path in the counterclockwise direction. In this way, the robot can continuously charge all the IoT devices. The experimental demonstration is shown in Figure 1.

5. Conclusions

In this paper, we invent a novel deep reinforcement learning algorithm AD-DQN to determine the optimal path for the mobile wireless power transfer robot to dynamically charge the harvesting energy-enabled IoT devices. The invented algorithm can intelligently divide a large area into multiple subareas and implement the individual DQN in each area, finally synthesizing the entire path for the robot. Compared with the state of the art, the proposed algorithm can effectively charge all the IoT devices on the experimental field. The whole system can be used in a lot of application scenarios, like charging IoT devices in the dangerous area and charging medical devices.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Yuan Xing designed the proposed wireless power transfer system, formulated the optimization problem, and proposed the innovative reinforcement learning algorithm. Riley Young, Giaolong Nguyen, and Maxwell Lefebvre designed, built, and tested the wireless power transfer robot on the wireless power transfer test field. Tianchi Zhao optimized the performance of the proposed deep reinforcement learning algorithm. Haowen Pan implemented the comparison on the system performance between the proposed algorithm and the state of the art. Liang Dong provided the theoretical support for far-field RF power transfer technique.

Acknowledgments

This work was supported by WiSys Technology Foundation under Spark Grant.

References

[1] F. Giuppi, K. Niotaki, A. Collado, and A. Georgiadis, "Challenges in energy harvesting techniques for autonomous self-powered wireless sensors," in *Proceedings of the 2013 European Microwave Conference*, pp. 854–857, Nuremberg, Germany, October 2013.

[2] A. M. Jawad, R. Nordin, S. K. Gharghan, H. M. Jawad, and M. Ismail, "Opportunities and challenges for near-field wireless power transfer: a review," *Energies*, vol. 10, no. 7, p. 1022, 2017.

[3] P. S. Yedavalli, T. Riihonen, X. Wang, and J. M. Rabaey, "Far-field rf wireless power transfer with blind adaptive

beamforming for internet of things devices," *IEEE Access*, vol. 5, pp. 1743–1752, 2017.

[4] Y. Xing and L. Dong, "Passive radio-frequency energy harvesting through wireless information transmission," in *Proceedings of the IEEE DCOSS*, pp. 73–80, Ottawa, ON, Canada, June 2017.

[5] Y. Xing, Y. Qian, and L. Dong, "A multi-armed bandit approach to wireless information and power transfer," *IEEE Communications Letters*, vol. 24, no. 4, pp. 886–889, 2020.

[6] Y. L. Lee, D. Qin, L.-C. Wang, and G. H. Sim, "6g massive radio access networks: key applications, requirements and challenges," *IEEE Open Journal of Vehicular Technology*, vol. 2, 2020.

[7] S. Nikoletseas, T. Raptis, A. Souroulagkas, and D. Tsolovos, "Wireless power transfer protocols in sensor networks: experiments and simulations," *Journal of Sensor and Actuator Networks*, vol. 6, no. 2, p. 4, 2017.

[8] Powercast, https://www.powercastco.com/documentation/, 2021.

[9] C. Lin, Y. Zhou, F. Ma, J. Deng, L. Wang, and G. Wu, "Minimizing charging delay for directional charging in wireless rechargeable sensor networks," in *Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1819–1827, IEEE, Paris, France, April-May 2019.

[10] H. Yan, Y. Chen, and S.-H. Yang, "Uav-enabled wireless power transfer with base station charging and uav power consumption," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12883–12896, 2020.

[11] Y. Liu, K. Xiong, Y. Lu, Q. Ni, P. Fan, and K. B. Letaief, "Uav-aided wireless power transfer and data collection in rician fading," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3097–3113, 2021.

[12] W. Feng, N. Zhao, S. Ao et al., "Joint 3d trajectory design and time allocation for uav-enabled wireless power transfer networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9265–9278, 2020.

[13] X. Yuan, T. Yang, Y. Hu, J. Xu, and A. Schmeink, "Trajectory design for uav-enabled multiuser wireless power transfer with nonlinear energy harvesting," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1105–1121, 2020.

[14] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing atari with deep reinforcement learning," arXiv:1312.5602, 2013.

[15] Y. He, Z. Zhang, F. R. Yu et al., "Deep reinforcement learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10433–10445, 2017.

[16] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Proceedings of the 29th Conference on Advances in Neural Information Processing Systems*, pp. 2137–2145, Barcelona, Spain, May 2016.

[17] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud rans," in *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Paris, France, May 2017.

[18] Y. Xing, H. Pan, B. Xu et al., "Optimal wireless information and power transfer using deep q-network," *Wireless Power Transfer*, vol. 2021, Article ID 5513509, 12 pages, 2021.

[19] C. Chen, J. Jiang, N. Lv, and S. Li, "An intelligent path planning scheme of autonomous vehicles platoon using deep reinforcement learning on network edge," *IEEE Access*, vol. 8, pp. 99059–99069, 2020.

[20] S. Wen, Y. Zhao, X. Yuan, Z. Wang, D. Zhang, and L. Manfredi, "Path planning for active slam based on deep reinforcement learning under unknown environments," *Intelligent Service Robotics*, vol. 13, pp. 1–10, 2020.

[21] S. Koh, B. Zhou, H. Fang et al., "Real-time deep reinforcement learning based vehicle navigation," *Applied Soft Computing*, vol. 96, Article ID 106694, 2020.

[22] R. Ding, F. Gao, and X. S. Shen, "3d uav trajectory design and frequency band allocation for energy-efficient and fair communication: a deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7796–7809, 2020.

[23] A. G. Barto, S. J. Bradtke, and S. P. Singh, "Learning to act using real-time dynamic programming," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 81–138, 1995.

[24] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, vol. 2, p. 5p. 5, Phoenix, AZ, USA, February 2016.

[25] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," arXiv:1511.06581, 2015.

[26] RaspberryPi, "Tx91501b-915mhz powercaster transmitter," 2021, https://www.raspberrypi.org/documentation/computers/raspberry-pi.html.

Hindawi
Wireless Power Transfer
Volume 2021, Article ID 8192582, 12 pages
https://doi.org/10.1155/2021/8192582

Research Article

Intelligent Power Grid Video Surveillance Technology Based on Efficient Compression Algorithm Using Robust Particle Swarm Optimization

Hongyang He, Yue Gao, Yong Zheng, and Yining Liu

Chongqing Fuling Electric Power Industry Co., Ltd, Chongqing, China

Correspondence should be addressed to Hongyang He; 0000003@yzpc.edu.cn

Received 19 September 2021; Revised 21 October 2021; Accepted 2 November 2021; Published 30 December 2021

Academic Editor: N. Venkateswaran

Companies that produce energy transmit it to any or all households via a power grid, which is a regulated power transmission hub that acts as a middleman. When a power grid fails, the whole area it serves is blacked out. To ensure smooth and effective functioning, a power grid monitoring system is required. Computer vision is among the most commonly utilized and active research applications in the world of video surveillance. Though a lot has been accomplished in the field of power grid surveillance, a more effective compression method is still required for large quantities of grid surveillance video data to be archived compactly and sent efficiently. Video compression has become increasingly essential with the advent of contemporary video processing algorithms. An algorithm's efficacy in a power grid monitoring system depends on the rate at which video data is sent. A novel compression technique for video inputs from power grid monitoring equipment is described in this study. Due to a lack of redundancy in visual input, traditional techniques are unable to fulfill the current demand standards for modern technology. As a result, the volume of data that needs to be saved and handled in live time grows. Encoding frames and decreasing duplication in surveillance video using texture information similarity, the proposed technique overcomes the aforementioned problems by Robust Particle Swarm Optimization (RPSO) based run-length coding approach. Our solution surpasses other current and relevant existing algorithms based on experimental findings and assessments of different surveillance video sequences utilizing varied parameters. A massive collection of surveillance films was compressed at a 50% higher rate using the suggested approach than with existing methods.

1. Introduction

As discussed by Memos et al. [1], the number of Switch-Mode Power Supply is increasing, as are incentive-based switching activities at the end-user level. A high-resolution time-resolution monitoring system will be required for future smart grids' operational stability to properly examine the state of the electricity grid. When it comes to power grid measurement applications, kilohertz frequencies are used, but the degree of aggregation and the reporting rate is not the same. Instead of using a second rate for instantaneous data like the smart meters, they utilize a day or more rate for the cumulative consumption data. As a result of the consolidation, communication lines and storage space needs have been significantly reduced. Assessing power quality (PQ) as well as disaggregating loads necessitates more data, as Gao et al. have shown [2]. Several features can be added on top of Harmonics; however, they can only provide partial information. Changes in grid operating approaches, demand-side control, and the rise of decentralized generation have led to an unknown number of combinations of interruptions. Features-based approaches may be unreliable due to the fact that data gets destroyed, particularly when exciting short-lived occurrences. Some thresholds may be adjusted by the user in commercially available equipment for PQ measurement at sample rates ranging from 10 kHz to 100 megahertz; raw data is captured when an event happens. A future smart grid, on the other hand, will have hard-to-predict threshold values. There may be further insights to be gained by examining raw data from synchronized measurements at different locations—even if not all scattered sensors were able to classify events simultaneously and hence did not capture them at a high resolution was depicted by Tsakanikas et al. [3]. Deploying a continuous storage system for raw data will assist data-driven research that attempts to improve event classification and smart grid analysis algorithms; for example, when using lossless data compression, compressing and transmitting large volumes of data is considerably easier. Uncommonly, the raw data stream of a recording device has three voltage readings (from the 3 phases) and 4 current measurements (3 for the phase currents and 1 for the neutral conductor). Nominally sinusoidal voltage curves that are 120° out of phase make up a three-phase power system. A strong relationship has been established between the various modes of communication. The same applies to current lines, and leveraging this interaction allows for a particular minimization in data volume. As a result of their unique distortion, the waveforms are less connected. Conceived as a way to decrease correlation in current channels owing to phase-load distortions. It is important to note that waveforms only change at the equipment's contact state or load alterations in terms of distortions. In general, these operations are slow compared to the length of time. With only one load connection, waveform changes are rare. Because load currents are increased whenever a massive number of demands get linked to the sensing field of the grid, variations exist rapidly. Waveform compression is therefore conceivable. It is known that lossless compression methods exist for certain applications, such as music and video. However, no method has been identified that is specifically designed to take advantage of the periodicity and multichannel nature of electrical signals encountered in a stream compression methodology. An overview of lossy and nonlossy techniques is included in the book, as are the CR values from trials. Applications that focus on PQ-event compression are listed; these implementations were developed by Shidik and his colleagues [4] among themselves. There is no statistical analysis of lengthy original data. These models focused on extremely precise incident data to validate the applicability of algorithmic changes in their own unique contexts. In the majority of cases, data sources are not referenced or provided at all. We find the researchers do not have a benchmark against which to assess the feasibility of compression algorithms for grid wave information, regardless of whether they are using known techniques or new ones that have yet to be found. We have chosen to focus on the development of compression with no degradation techniques and performances for grid data at a lot of sampling to address these issues in the present contribution. When utilizing input data with a variety of ideas, we are considering new growth ideas made of natural time-series analysis. New lossless compression algorithms could be developed by using testing data and comparison parameters for the first thorough accessible standard. They can be used as a decision assistance tool by researchers dealing with data-intensive smart grid measures. The preprocessing phase entails changing the color space, after which the features may be retrieved using pseudo-component analysis. Then, utilizing Robust Particle Swarm Optimization, the encoding and decoding process may be completed. The main contribution of the research work is as follows:

(i) To design and develop a compression-based video surveillance technology based on the optimization approach

(ii) For the purpose of the authentication, run-length encoding and decoding were performed

The following is how the rest of the article is organized. In Section 2, a literature survey is being reported on strategies to reduce loss during video compression. The issue of lossless video compression mechanisms was then addressed in Section 3. Section 4 then poses the proposed mechanism over lossless video compression. The results of the suggested method and the conclusions were examined in Sections 5 and Section 6.

2. Related Works

In [1], the article looks into wireless sensor networks (WSNs) alongside the most recent research on social confidentiality and protection in WSNs. While adopting High-Efficiency Video Coding (HEVC) as a new media compression standard, a novel EAMSuS in the IoT organization is presented (HEVC). In [5], complete situational awareness is provided via real-time video analysis and active cameras. In [6], a new section of MPEG standards called Video Compression Modulation (VCM) has been suggested by the author. Video Coding for Machine Vision seeks to bridge the gap between machine vision feature coding and human vision video coding to accomplish collaborative compression and intelligent analytics. VCM's definition, formulation, and paradigm are provided first, corresponding with Digital Retina's rising compress instance. This is why they analyze video compression and features from the unique perspective of MPEG standards, which offers both academics and industry proof to accomplish the collaborative compression of the video shortly. In [7], using MapReduce, the author has developed UTOPIA Smart Video Surveillance for smart cities. From their end, we were able to incorporate smart video surveillance into our middleware platform. With the help of this article, we show that the system is scalable, efficient, dependable, and flexible. In [8], here when it comes to edge computing capabilities, the cloud object tracking and behavior identification system (CORBIS) was demonstrated. To increase distributed video surveillance systems' resiliency and intelligence, network bandwidth and reaction time between wireless cameras and cloud servers are being reduced in the Internet-of-things (IoT). In [9], an effective cryptosystem is used to create a safe IoT-based surveillance system. There are three parts to it. An automated summary technique based on histogram clustering is used to extract keyframes from the surveillance footage in the first stage. To compress the data, a discrete cosine transform is applied to it (DCT). Not to mention, a discrete fractional random transform is used to develop an efficient picture encryption approach in the suggested framework (DFRT). In [10], the author proposes a novel approach for compressing video inputs from surveillance systems. There is no way to reduce visual input redundancy using outdated methods that do not meet the demands of modern technologies. Video input storage needs to increase as a result, as does the time required to process the video input in real-time. To compress video inputs from surveillance systems, a unique technique is presented in this research paper. Visual input redundancy cannot be reduced using obsolete approaches that do not match the expectations of contemporary technology. This raises the storage requirements for video input and the processing time as a result. In [11], by using compressed sensing (CS), the author suggests creating security keys from the measurement matrix elements to secure your identity. Assailants cannot reconstruct the video using these. They are designed to prevent this. A WMSN testbed is used to analyze the effectiveness of the proposed security architecture in terms of memory footprint, security processing overhead, communication overhead, energy consumption, and packet loss, for example. In [12], a new binary exponential backoff (NBEB) technique was suggested by the author to "compress" unsent data that can preserve important information but recover the electronic trend as much as feasible. Data coming in may be temporally chosen and dumped into a buffer, while fresh data can be added to the buffer as it is received. As a result of the algorithm, the incoming traffic rate can be reduced in an exponential relationship with the transmitting failure times. In [13], the author suggested the lossless compression technique to handle the problem of managing huge raw data amounts with their quasiperiodic nature. The best compression method for this sort of data is determined by comparing the many freely accessible algorithms and implementations in terms of compression ratio, calculation time, and operating principles as well as algorithms for audio archiving; there are other algorithms for general data compression (Lempel–Ziv–Markov chain algorithm (LZMA), Deflate, Prediction by partial matching (PPMd), Burrows–Wheeler algorithm (Bzip2), and GNU zip (Gzip)) that are put to the test against one other. Deal with the challenge of managing enormous raw data quantities with their quasiperiodic nature by using lossless compression. Compression ratio, computation time, and operating principles are all taken into account when comparing publicly available algorithms and implementations to decide which is the most efficient. Additionally, generic data compression techniques such as LZMA, Deflate, PPMd, Bzip2, and Gzip are also put to the test. In [14], an efficient embedded image coder based on a reversibly discrete cosine transform is proposed for lossless. ROI coding with a high compression ratio (RDCT) was suggested. To further compress the background, a hierarchical (SPIHT) partitioning technique is used to combine the proposed rearranged structure with a lost zero tree wavelet coding. Results of the coding process indicate that the new encoder outperforms many state-of-the-art methods for still photo compression. In [15–17], the focus was based on the loss of video compression. Even at lower bit rates, the novel loss-compression method improves contourlet compression performance. Along with SVD, compression efficiency is improved by standardization and prediction of broken subband coefficients (BSCs) [18]. We measure the computational complexity of our solution with a better video quality. HCD uses DWT, DCT, and genetic optimization to improve the performance of transformed coefficients, among other techniques. This method works well with MVC to get the best possible rate distortion. The simulation results are produced using MATLAB Simulink R2015 to examine PSNR, bit rate, and calculation time for various video sequences using various wavelet functions, and the performance results are evaluated [19]. To solve the optimization issue of trajectory combination while producing video synopses, a new approach has been devised. When dealing with the optimization issue of motion trajectory combination, the technique makes use of the genetic algorithm's temporal combination methods (GA) [20]. The evolutionary algorithm is utilized as an activation function within the hidden layer of the neural network to construct an optimum codebook for adaptive vector quantization, which is proposed as a modified video compression model. The context-based initial codebook is generated using a background removal technique that extracts motion items from frames. Furthermore, lossless compression of important wavelet coefficients is achieved using Differential Pulse Code Modulation (DPCM), whereas lossy compression of low energy coefficients is achieved using Learning Vector Quantization (LVQ) neural networks [21]. This paper presents a rapid text encryption method based on a genetic algorithm. It is possible to use genetic operators Crossover and Mutation to encrypt data. By splitting up the plain text characters into pairs and using a crossover operation to obtain the encrypted text from the plain text, this encryption approach uses mutations to get its encrypted message.

From the literature survey, reviewed images and videos are compressed using transform-based and fractal approaches, along with other lossless encoding algorithms, which are now the most frequently used methods for still and video compression. Each technique has its own set of pros and downsides like breaking of the wavelet signal and low compression ratio; hence, it is important to choose the right one. It is most common for video-based images to be compressed using transform-based compression (TBC). In order to achieve compression, the signal or values are altered. Using various transformations, they convert a spatial domain representation into a picture. Brushlet is an example of an adaptive transformation (Verdoja and Grangetto 2017); bandelet (Raja 2018) (Erwan et al. 2005) and directionlet (Jing, et al. 2021) give information about the picture in advance. After applying these modifications to a picture, its essential function is altered. Hence, we are motivated to develop a methodology that overcomes all the existing video compression issues.

3. Problem Statement

Rapid advances are being made in compressing technology. As a challenging and essential topic, real-time video compression has sparked a lot of studies. This corpus of information has been included in the motion video standards to a large extent. Unanswered are several significant questions. According to the point of view of a compression algorithm, eliminating various redundancies from certain types of video data is a compression challenge. Thorough knowledge of the problem is needed, as well as a novel approach to solve all of the existing research gaps with irreversible video compression. Progress in other fields, such as artificial intelligence, has contributed to the breakthroughs in compression. A compression algorithm's success depends on the acceptance of a new generation of algorithms in addition to its technological excellence.

4. Proposed Work

As a result of the smart grid's usage of ICTs, the generation, distribution, and consumption of electricity are all more efficient (Information and Communication Technologies). For example, the transmission system and the medium-voltage level distribution system are monitored by Supervisory Control and Data Acquisition (SCADA) and wide-area monitoring systems (WAMS). It is important to remember that the primary objective of compression is to minimize the amount of data. That is if the compressed data retains most of its original content. Various scholars are currently involved in proposing effective techniques of data compression. Listed below are some of the most prevalent data compression techniques. With this analysis, we are focusing on compressing the PQ-event data in a video context in each successive frame to save space. To accomplish this, we must first identify the video frame object. Robust Particle Swarm Optimization is used to create a lossless video compression method. This is a diagram of the recommended technique shown in Figure 1.

4.1. Dataset. They used the UK Domestic Appliance-Level Electricity (UK-DALE) Dataset to conduct the experiments. A smart distribution system collects data on three-phase voltage, current, active and reactive power, and power factor from transformers at 54 substations as well as estimations of current and voltage at the inlets of three homes. The data is then analyzed and compared with the raw data from three homes. A 16 kHz sampling rate and a 24-bit vertical resolution were employed in the acquisition. There was a random selection of six FLAC-compressed recordings from 2014-8-08 to 2014-05-15, each having an hour of recordings. In a proprietary format, these data are recorded as four-byte floating-point numbers with timestamps at a sampling rate of 15 kHz. Voltage and current values are included in phase 2 of house 5. Every one of the four files contains 266 s. Large-scale databases hold all data transferred via a network. Raw data for three-phase voltages need 8.4 GB per day, whereas three-phase currents (including neutral) require 19.35 GB per day. To transmit the data, you need 0.8 Mbit/s and 1.8 Mbit/s in turn. This dataset was compiled in the following locations: as our institution's main power supply in Karlsruhe, Germany, we also have power outlets in our practical room and a substation transformer there. A total of seven channels consisting of four currents and three voltages are sampled at 12.8 and 25 kS/sec, respectively. There are seven channels, with four currents and three voltages sampled at 12.8 and 25 kS/sec, for a total of seven channels. Single-channel testing and dual-channel testing include measuring the current and/or voltage of a single phase in both situations, depending on which method is used. To save the data, raw 16-bit integers are stored in blocks of 60 s.

Electricity generation, transmission, and distribution in smart power systems are all affected by the analysis of the data. As a result, data exchange and memory requirements are expected to grow considerably, and data storage and bandwidth requirements for communication links in smart grids are also expected to increase. It is necessary to raise the sampling frequency to receive reliable and real-time information from the intelligent grid. There will be a greater emphasis on smart grid data compression in the future. Figure 1 illustrates the proposed compression approach. In areas of the grid with significant data volume, this approach can be used successfully.

4.1.1. Preprocessing. There are several steps to video compression, the first being preprocessing. Preprocessing is essential for a database's longevity and usefulness. For this reason, each stage in the video data processing workflow appears to be crucial. The procedure involves preprocessing, such as error detection or any other conversions that are not essential. Power grid video can cause picture frames to be split. The Bayesian motion subsampling approach may be used to create the video frame. This is a common method for removing frames from a movie. As the name implies, it is a computerized method used to enhance the frame creation process. For the most common sensitivities, the picture frame intensity range has been expanded, which results in a better image frame sensitivity value.

Let p denote the subsampled of each possible frame illustrated as

$$p^{y} = \frac{(\text{Number of the pixel frames with the } y \text{ intensity})}{\text{Total number of the pixel frames}}. \tag{1}$$

Here, $y = 0, 1, \ldots, y - 1$.

The separated pixel frames can be defined as depicted in [22]:

$$H_{i,j} = \text{base}\left((Y - 1) \sum_{Y=0}^{b_{i,j}} p^{Y} \right), \tag{2}$$

where base represents the nearest integer. This is equivalent to transforming the pixel intensity [23]:

$$\frac{\partial N}{\partial x}\left(\int_{0}^{N} pN(x)\mathrm{d}z \right) = \left(\partial N(N)\left(x^{-1}\right)(N) \right) \frac{\mathrm{d}}{\mathrm{d}N}. \tag{3}$$

Here, finally, the probability distributed uniformity function can be represented as $\partial N/\partial x$.

When it comes to histograms, the equalization procedure can soften and enhance them. However, even though the histogram produced by the equalization is perfectly flat, it will be softened. After reducing the pictures' superfluous

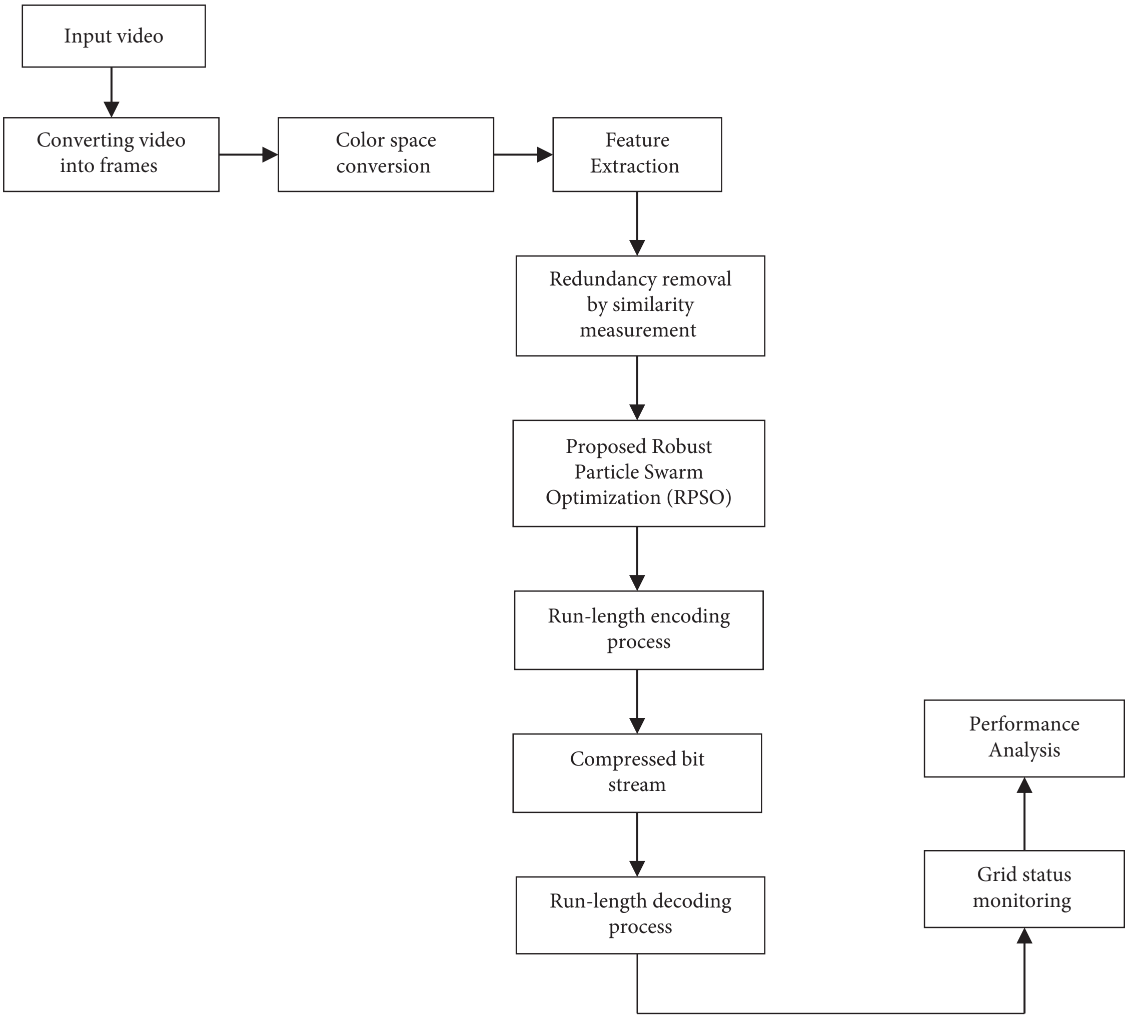

Input video
Converting video into frames
Color space conversion
Feature Extraction
Redundancy removal by similarity measurement
Proposed Robust Particle Swarm Optimization (RPSO)
Run-length encoding process
Compressed bit stream
Run-length decoding process
Grid status monitoring
Performance Analysis

FIGURE 1: Schematic representation of the suggested methodology.

noise, we apply a threshold technique to improve the refined frame acquired from the context. Thereafter, binary images are created, which streamlines the image processing process. As a result of the color space conversion, we see a shading effect in the majority of pictures. The picture contains three channels in most cases (red, green, blue). In the blue channel, there is no more information, but there is a great deal of contrast. Preprocessed green channel was deleted next. For example, here is how to extract the green channel [24]:

$$\begin{aligned} I_{\text{org}} &= f(\sigma, \mu, \beta), \\ I_{\text{red}} &= f(1, \mu, \beta), \\ I_{\text{Green}} &= f(\sigma, 2, \beta), \end{aligned} \tag{4}$$

where σ denotes the Red channel, μ denotes the Green channel, and β denotes the Blue channel.

Translation of color representation from one basis to another is called color space conversion (CSC). In most cases, this occurs while converting a picture from one color space to another. The use of a single threshold value for converting the color space is thus not recommended.

$$\theta \propto \text{Threshold}\,(E) \approx j * \left(\frac{u}{\left|v^{1}/3\right|}\right)\left(j_{\text{best}} - j_i\right), \tag{5}$$

where E represents converted the color space.

The color space is transformed to grayscale by keeping the brightness information. A grayscale picture frame can be represented as a collection of grayscale images by D_2.

$$ED_2 = \text{GS}\left(D_1\right) = \left\{d_2(1), d_2(2), \ldots, d_2(i), \ldots, d_2(|D|)\right\}. \tag{6}$$

After the frame gets preprocessed, the data can undergo the step of feature extraction.

4.1.2. Feature Extraction. We implemented the pseudo-component analysis in the feature extraction module to improve the compression performance and concentrate the image's information. The method for decreasing the size and complexity of data sets involves converting huge numbers of variables into smaller ones that retain the majority of the information contained in the large set. Naturally, limiting the number of parameters in sets of data lowers the information's accuracy, but the trick is to give up just a little precision for convenience. It is simpler to examine and interpret smaller data sets. Machine learning algorithms can also examine data more easily and quickly without dealing with extraneous issues. Each pseudo-redundancy component must be selected as a first stage in the process of feature extraction. In this module, the main goal is to extract the highlighted characteristics. Below are the configurations of this mechanism.

$$\begin{aligned} y_{\text{input}} &= \left[V_c^T y_c + B_c, V_s^T y_s + B_s\right], \\ \beta &= f_2\left(V_{\text{int}}^T f_1\left(y_{\text{input}}\right) + B_{\text{int}}\right). \end{aligned} \tag{7}$$

Here, $[V_c^T y_c + B_c, V_s^T y_s + B_s]$ Error! Bookmark not defined denotes the overall feature level; $V_c^T \in \mathbb{Z}^{C_c \times C_{\text{int}}}, V_s^T \in \mathbb{Z}^{C_s \times C_{\text{int}}}$, and $V_{\text{int}}^T \in \mathbb{Z}^{2C_{\text{int}} \times 1}$ represent feature weights; B_c, B_s, and B_{int} depict the associated features; C_c and C_s correspond to the sizes of the input medium of the categorization and feature sections, accordingly; and C_{int} denotes the internal input. Operations $f_1(y) = \max(y, 0)$ and $f_1(y) = 1/(1 + \exp(-y))$ associate to sigmoid activation operation, accordingly. The attention map is further standardized to [0, 1]. The outcome of the feature extraction is represented as depicted in [25]:

$$y_{\text{out}} = f_3([\beta \times y_c, y_s]). \tag{8}$$

Here, f_3 consists of a sequence of the feature components.

Pseudo- and nonpseudo-component characteristics can be selected using a property calculation technique. To determine pseudo-component characteristics, Hong correlations approaches, which employ averaging techniques, and Leibovici correlations, which use mixing principles, are used. In this approach, the phase fraction values are collected from a compositional system to minimize the difference between them. Pseudo- and nonpseudo-redundancy characteristics can be retrieved, as shown as follows [26]:

$$Lo_{\text{Pseudo,nonpseudo}} = 1 - \frac{2|A \cap B|}{|A| + |B|} = 1 - \frac{\sum_j^N p_j g_j + sm}{\sum_j^N p_j + \sum_j^N g_j + sm}, \tag{9}$$

where p_j represents the pseudo features, g_j represents the nonpseudo features, and sm represents the empirical constant.

4.1.3. Optimized Compression Process. Video data may be compressed without losing any information using this method. Concerning Robust Particle Swarm Optimization (RPSO) and run-length coding (RLC), the common RLC can be optimized by using the optimization algorithm and has been employed for the compression stage. We analyze the properties of compressed data using this technique. To maximize compression-related parameters, it is advised to use this method as a population-based approach. RPSO is initialized with the sample particles and modified with the optimal answer in each cycle. The resulting answer is called fitness and is referred to as the $_{\text{best}}$. The best solution obtained by a particle in a population is considered the world's top value monitored by the particle swarm optimizer (g_{best}). By the two p_{best} and g_{best} solutions, the positions of each particle change to global optima. The individual speeds and location functions of each particle are as follows. In a dimension search space D, there is a swarm composed of particles where each particle is represented by 'i' in a vector of $X_i = x_{i1}, x_{i2}, \ldots\ldots, x_{\text{id}}$ and the particle bet solution p_{best} is denoted as $p_i = p_{i1}, p_{i2}, \ldots\ldots, p_{\text{id}}$. Then, the best solution of the subset swarm is given by $g_{\text{best}} p_g = \left\{p_{g1}, p_{g2}, \ldots\ldots, p_{g\,d}\right\}$. The i^{th} particle velocity is represented as $V_i = V_{i1}, V_{i2}, \ldots\ldots, V_{\text{id}}$. The particle velocity and location are updated based on equations (10) and (11).

The weights updates are given by [27]

$$\begin{aligned} v_{\text{id}}(n+1) = W(it) * v_{\text{id}}^n + C_1 * \text{rand} * \left(p_{\text{id}} - x_{\text{id}}^n\right) \\ + C_2 * \text{rand} * \left(p_{g\,d} - x_{\text{id}}^n\right), \end{aligned} \tag{10}$$

where W represents the weighed features, C represents the cross features, n represents the constant, rand represents the random number, x_{id} represents the particle velocity, and p_{id} represents the particle motion. Here, depending on the feature extracted, the details can be updated depending upon the weightage, where $V_{\text{id}} = V_{\max}$, and it reflects the number of iterations between 1 and 10, where 10 is the maximum number of iterations. The random value of 0 to 1 is represented by the rand. C_1 and C_2 normally signify a nonnegative amount of an acceleration constant; here, C_1 and $C_2 = 1.05$. The particle orientation is also modified with [28]

$$x_{\text{id}}^{n+1} = x_{\text{id}}^n + v_{\text{id}}^{n+1}. \tag{11}$$

Any swarm obtains the health or objective f and each iteration provides the best solution; then if $f(x_i) < f(p_{\text{best}})$ and $f(g_{\text{best}})$, then p_{best} and $g_{\text{best}} = x_i$. The optimal measurement is obtained to maximize the curve transformation coefficients.

Video reconstruction can be done via run-length encoding once the optimized values are acquired. Using sequential data, this is a fairly simple operation to perform on your computer. For redundant data, it is a great tool. Running symbols are replaced by shorter ones in this technique. There are two ways to express the run-length code in grayscale images: V and R, where V represents the character count and R represents the run length. For optimized run-length optimal run-length encoding (ORLE), the following steps are required:

Step 1: Coefficient optimization

Step 2: Enter the string

Step 3: Give a unique value from the very first symbol or letter

Step 4: Otherwise, leave if the character or symbol is the final one in the string

Step 4: Additional symbols can be read and counted

Step 5: Until the preceding symbol subband has a nonmatching value, move on to step 3

Step 6: This will give you a count of the number of times a certain symbol appears in a given sentence

The suggested methodology uses a vector that contains a variety of scales to convert subbands that are optimized minimum and maximum to achieve the best result.

$$\text{Compressed}_{\text{Fitness value}} = \frac{-40 * q\left(-3 * \sqrt{\sum S_v}\right)}{2 - \exp\left(\sum \cos\left(3\pi * S_v\right)/d_b\right)} + 10 \text{ exp} , \tag{12}$$

where q denotes the compressed reconstructed value and S_v is the compressed score value that is obtained. Finally, the best rate of compression can get obtained. The RPSO reconstructs the data by using run-length decoding after refining the transforming Algorithm 1 curvelet parameters are as follows.

Finally, after compression, the status of the grid can get sorted out and it can get monitored and the irregular grid distribution can get identified.

5. Performance Analysis

Increasingly, data is being exchanged across smart grid sectors. Many types of data are created every day. For example, meteorological data such as the amount of sun or wind, humidity, or temperature are essential for optimal performance in many industries. Two phases in the data interchange procedure are encoding and decoding (or decryption). Numerous operations take place during the encoding phase to prepare data for transmission; when the data is encoded and decoded, it will be returned to its original form. In this section, you will learn about the complete process of performing experiments for performance evaluation. It is written in MATLAB, which is a programming language. Measurement data was collected over 24 hours in 1-minute, 5-second, 10-second, and 20-second intervals to assess the proposed compression methods. Readings from multiple meters were collected for each period in a data matrix.

Table 1 illustrates the effect of truncating small singular values on the compression ratio (CR) and percentage residual root difference. It can be seen that the minimum root mean square distance is obtained when eight singular values are considered. This leads to a reduction in the signal length. Compared to other sets of data, the calculated CR values for the 5-second time interval data are closer to the Total compression ratios (TCR) values in Table 1. Data obtained at 1-minute, 10-second, and 20-second intervals have generated CRs that deviate somewhat from the TCRs. Figure 2 illustrates the relationship between the number of significant singular values and TCR. According to the plotted data, the size of the data matrix has an impact on the ratio of compression (r), the number of singularly significant values. There are two different sizes of a data matrix: 5 seconds and 1 minute. A greater number of significant single values were required to match the TCR in 5-, 10-, and 20-second datasets than in the 1-minute dataset, as can be observed in Figure 2. As an alternative, selecting a shorter time interval, such as five seconds, will offer a better approximation on the number of significant singular values, resulting in the computed CRs being closer to the TCRs.

The mean error is a colloquial phrase that refers to the average of all mistakes in a collection. In this context, an "error" refers to a measurement uncertainty or the difference between the measured value and the correct/true value. Measuring error, often known as observational error, is the more formal word for error. According to Figure 3, there is a relationship between the related mean error for different time interval data and TCRs. As shown in Figure 3, the data consisting of measurements per 1-minute interval has the lowest mean error. The MAE found for greater matrix sizes is larger when the TCRs are higher.

According to Figure 4, there is a correlation between the number of significant singular values and the rate of mistake. For the first 100 single values, the 5-second dataset has the greatest MAE, followed by 10-seconds, 20-seconds, and 1-minute time interval dataset that has the lowest MAE. There is practically no inaccuracy in any dataset after the first 100 single values. A dataset's size has a substantial impact on singular values and the correctness of reconstructed data.

A smart distribution system's data is compressed in this part to see how well the approach works. To sum up, more singular values are required to fulfill TCR as a data set grows in size, as shown by the experimental findings. Nevertheless, increasing the number of singular values will reduce the amount of data that has to be compressed. As a result, there are fewer errors when the data is rebuilt after it has been compressed. As a result, a greater amount of data must be transferred through a wider range of communication channels. By compressing information with a high number of singular values to fulfill the TCR, you will have to send more data. The TCR must be matched to the quantity of data to be compressed to maximize the connection bandwidth when transferring the compressed data. The data reconstruction error can be calculated between the reconstructed data $g(i, j, s)$ and the original data $F(i, j, s)$ using

$$P(s) = \frac{1}{3MN} \sum_{i=0}^{a-1} \sum_{j=0}^{b-1} \sum_{s=0}^{2} \left\| g(i, j, s) - F(i, j, k) \right\|. \tag{13}$$

In addition, the Mean Average Error (MSE) (calculated by averaging squared error) is another way to assess reconstruction accuracy.

The MAE is defined as [29]

$$\text{MAE} = \sum_{i=0}^{a-1} \sum_{j=0}^{b-1} \sum_{s=0}^{2} \left\| g(i, j, s) - f(i, j, s) \right\|^2. \tag{14}$$

A measure of the quality of compression and reconstruction is the signal-to-noise ratio (SNR). There are two ways to define the peak SNR [30]:

$$\text{PSNR}(\text{dB}) = 10 \log_{10} \frac{(\text{Max}_i)}{\sqrt{\text{MAE}}}, \tag{15}$$

where Max_i is the maximum possible pixel value.

MD quantifies the greatest difference between original and reconstructed values. The average difference between original and reconstructed values is denoted as SSIM. For each of the formulas [31],

```
Input: Extracted features
Output: Compressed data $C_d$
To compute compressed value,
For $i = 1$: size$(D_{n\_\text{parameters}}, 1)$
    For $j = 1$: size$(D_{n\_\text{features}}, 1)$
    Weighed updates
$v_{\text{id}}(n+1) = W(it) * v_{\text{id}}^{n} + C_1 * \text{rand} * (p_{\text{id}} - x_{\text{id}}^{n})$
$+C_2 * \text{rand} * (p_{g\,d} - x_{\text{id}}^{n})$,
End
End
Data compressed features $d_{n\_\text{fea}} = [d_{n\_\text{fea}}\text{rand}]$
To compute, Run length encoding
Class label = unique(target)
$K$ = length(class label)
For $d = 1$: $k$
    Temp = total class mean($I$, :)
Run length decoding
Data grouping
Compressed$_{\text{Fitness value}}$ =
$(-40 * q(-3 * \sqrt{\sum S_v})/2 - \exp(\sum \cos(3\pi * S_v)/d_b))$
+10 exp,
End
```

Algorithm 1: (RPSO)

Table 1: Computed CR, and number of singular values, r of compression.

TCR	1 minute		20 seconds		10 seconds		5 seconds	
	r	CR	r	CR	r	CR	r	CR
100	6	96.68	8	103.62	9	101.89	10	96.84
80	7	82.87	10	82.90	11	83.36	12	80.70
60	10	58.01	14	59.21	15	61.13	16	60.53
50	12	48.34	17	48.76	18	50.94	19	50.97
30	19	30.53	28	29.60	31	29.58	32	30.26
10	58	10.00	83	9.99	92	9.97	97	9.98
5	116	5.00	166	4.99	183	5.01	194	4.99
4	145	4.00	207	4.00	229	4.00	242	4.00

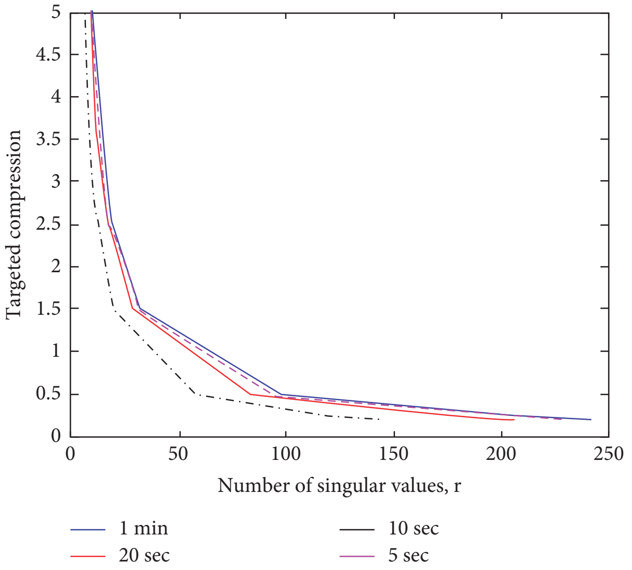


Figure 2: Plot of the number of singular values versus TCR for the different datasets.

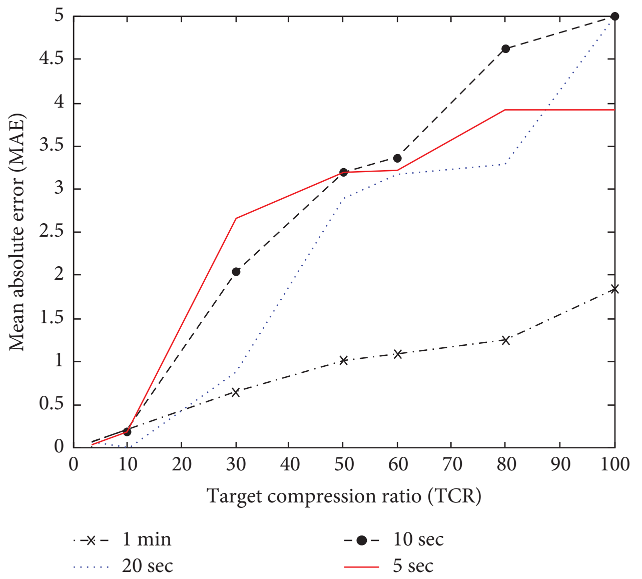


FIGURE 3: Plot of MAE versus TCR for different sampling rates.

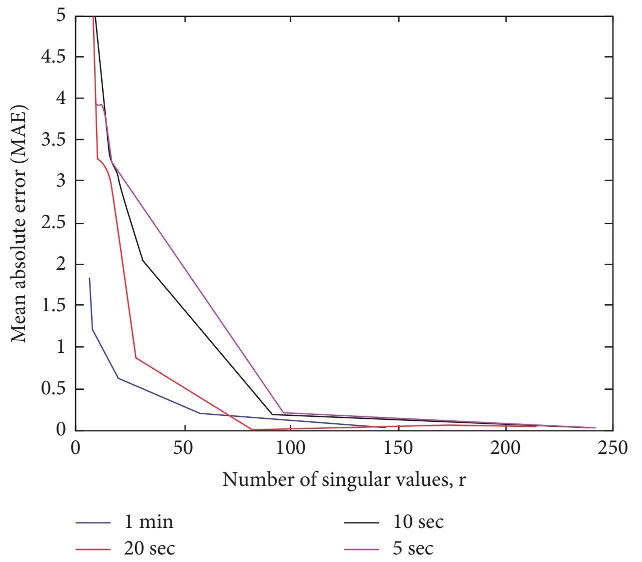


FIGURE 4: Plot of MAE versus the number of singular values, r for the dataset.

$$\mathrm{SSIM} = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\sum_{s=0}^{s-1}\left\|I\left(i,j,s\right)\right\|,$$
$$\mathrm{MD} = \max_{0\le i\le M}\left\|I\left(i,j,s\right)\right\|. \tag{16}$$

The video reconstruction error (MSE), signal-to-noise ratio (PSNR), matching distance (MD), and percent compression ratio (PCR) values are obtained as depicted in Figure 5. The satisfying results are obtained over the compression as depicted in Table 2. From Table 2 and Figure 5, the suggested methodology shows the highest performance over PSNR, MSE, and MD. As illustrated in the PSNR contours for the testing set in Figure 5, the PSNR improves as the compressed image bit rate increases. The results demonstrate a rising pattern in PSNR values, whereas MSE drops progressively as the compressed image bit rate improves. As a result, a higher compressed image bit rate means higher resolution images and fewer mistakes.

The advantages of the existing mechanism in which the high compression ratio was obtained but it takes more time for compression. Hence, it can be overcome by the proposed mechanism.

5.1. Complexity Analysis. In general, the total number of states is approximately equal to 2^N for computing nth RLE number ($F(^N)$). Notice that each state denotes a function call to ‘RPSO with RLE()′ which does nothing but makes another

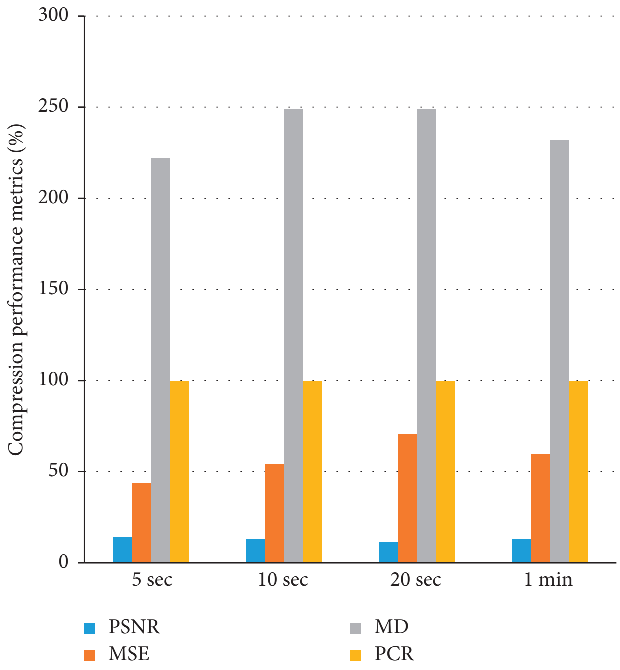


Figure 5: Image Quality metrics.

Table 2: Average Data Quality metrics.

Parameters	5 sec	10 sec	20 sec	1 min
PSNR	14.1913	13.2848	11.168	12.778
MAE	43.5359	54.1624	70.49	59.7
MD	222.1475	249.0821	249.05	232.1
PCR	100	100	100	100

Table 3: Average compression ratio.

Compression level	Number of images	Compression ratio
Discrete spatial multilayer perceptron (proposed)	100	10 : 1
Haar [17]	100	10 : 17
Cosine [17]	100	10 : 18

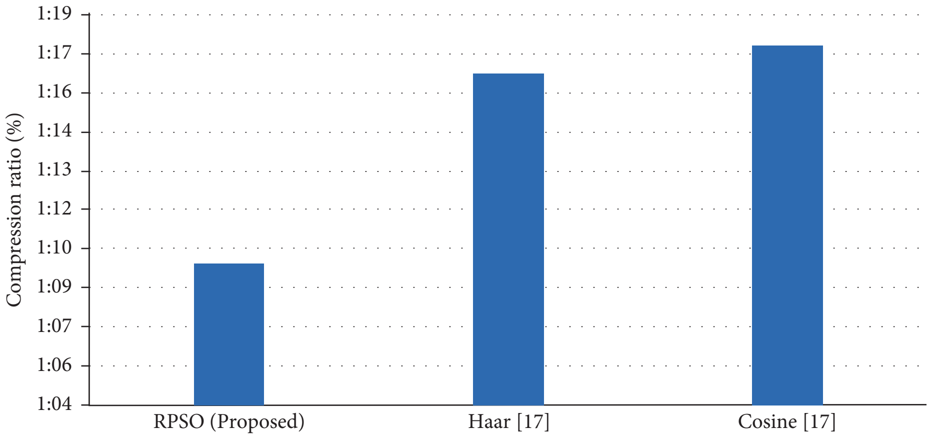


Figure 6: Compression ratio.

recursive call. Therefore, the total time taken to compute the nth number of the sequence is $O(2^N)$.

In digital file compression, duplication is the most important issue. If N_1 and N_2 signify the amounts of data holding units in the raw and encoded images, correspondingly, the compression ratio, CR, could be specified as $\mathrm{CR} = N_1/\mathrm{N}_2$ as well as the data duplication of the original image as $\mathrm{RD} = 1 - (1/\mathrm{CR})$. From Table 3 and Figure 6, the proposed methodology can acquire the exact ratio of compression (10 : 1) when compared to Haar [17] (10 : 16.5) and Cosine [17] (10 : 17.2) techniques.

6. Conclusion

Data compression techniques such as RPSO compression were examined and evaluated in this article. Data from a smart distribution system was used to evaluate the algorithm with 1-minute, 10-second, 20-second, and 5-second interval datasets. The results obtained demonstrate that the amount of the data has a considerable influence on the proposed approach. Larger datasets require more significant single values to achieve low error rates. When used to the smart grid, RPSO may be used as a simple and uncomplicated compression method. The significant singular values will provide a decent approximation when the compressed data has to be rebuilt using the recommended approach. Depending on the number of singular values used, RPSO compression can lower the volume of data. However, if you have a lot of data, you should consider using the proposed compression technique, which has a faster execution time and low error rates. Also, a lot of the pointed advantages exist. There will be some disadvantages also; in the proposed work, the order of bytes is independent. Compilation needs to be done again for compression. Errors may occur while transmitting data. We have to decompress the previous data. The disadvantages can be overcome in future work.

Data Availability

The data used to verify the study's findings can be obtained from the author on request.

Conflicts of Interest

The authors state that the publishing of this paper does not include any conflicts of interest.

References

[1] V. A. Memos, K. E. Psannis, Y. Ishibashi, B.-G. Kim, and B. B. Gupta, "An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework," *Future Generation Computer Systems*, vol. 83, pp. 619–628, 2018.

[2] Z. J. Gao and J. S. Wang, "Application of smart grid technology in the coalmine power system," *Applied Mechanics and Materials*, vol. 441, pp. 236–239, 2014.

[3] V. Tsakanikas and T. Dagiuklas, "Video surveillance systems-current status and future trends," *Computers & Electrical Engineering*, vol. 70, pp. 736–753, 2018.

[4] G. F. Shidik, E. Noersasongko, A. Nugraha, P. N. Andono, J. Jumanto, and E. J. Kusuma, "A systematic review of intelligence video surveillance: trends, techniques, frameworks, and datasets," *IEEE Access*, vol. 7, pp. 170457–170473, 2019.

[5] A. Hampapur, L. Brown, J. Connell et al., "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 38–51, 2005.

[6] L. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video coding for machines: a paradigm of collaborative compression and intelligent analytics," *IEEE Transactions on Image Processing*, vol. 29, pp. 8680–8695, 2020.

[7] C.-S. Yoon, H.-S. Jung, J.-W. Park, H.-G. Lee, C.-H. Yun, and Y. W. Lee, "A cloud-based UTOPIA smart video surveillance system for smart cities," *Applied Sciences*, vol. 10, no. 18, p. 6572, 2020.

[8] R. Rajavel, S. K. Ravichandran, K. Harimoorthy, P. Nagappan, and K. R. Gobichettipalayam, "IoT-based smart healthcare video surveillance system using edge computing," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2021.

[9] R. Hamza, A. Hassan, T. Huang, L. Ke, and H. Yan, "An efficient cryptosystem for video surveillance in the internet of things environment," *Complexity*, vol. 2019, Article ID 1625678, 11 pages, 2019.

[10] V. R. Prakash, "An enhanced coding algorithm for efficient video coding," *Journal of the Institute of Electronics and Computer*, vol. 1, pp. 28–38, 2019.

[11] S. A. Nandhini and S. Radha, "Efficient compressed sensing-based security approach for video surveillance application in wireless multimedia sensor networks," *Computers & Electrical Engineering*, vol. 60, pp. 175–192, 2017.

[12] T. Jiang, H. Wang, M. Daneshmand, and D. Wu, "Cognitive radio-based smart grid traffic scheduling with binary exponential backoff," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2038–2046, 2017.

[13] R. Jumar, H. Maaß, and V. Hagenmeyer, "Comparison of lossless compression schemes for high rate electrical grid time series for smart grid monitoring and analysis," *Computers & Electrical Engineering*, vol. 71, pp. 465–476, 2018.

[14] S. A. Elhannachi, N. Benamrane, and T.-A. Abdelmalik, "Adaptive medical image compression based on lossy and lossless embedded zero tree methods," *Journal of Information Processing Systems*, vol. 13, pp. 40–56, 2017.

[15] P. E. Sophia and J. Anitha, "Enhanced method of using contour let transform for medical image compression," *International Journal of Advanced Intelligence Paradigms*, vol. 14, no. 1/2, pp. 107–121, 2019.

[16] T. Kalidoss, L. Rajasekaran, K. Kanagasabai, G. Sannasi, and A. Kannan, "QoS aware trust based routing algorithm for wireless sensor networks," *Wireless Personal Communications*, vol. 110, no. 4, pp. 1637–1658, 2020.

[17] I. Yamnenko and V. Levchenko, "Video-data compression using wavelet analysis," in *Proceedings of the 2019 IEEE 20th International Conference on Computational Problems of Electrical Engineering (CPEE)*, pp. 1–4, Lviv-Slavske, Ukraine, September 2019.

[18] S. Rahimunnisha and G. Sudhavani, "Novel complexity reduction technique for multi-view video compression using HCD based genetic algorithm," *Design Engineering*, vol. 2021, no. 6, pp. 3219–3228, 2021.

[19] L. Xu, H. Liu, X. Yan, S. Liao, and X. Zhang, "Optimization method for trajectory combination in surveillance video synopsis based on genetic algorithm," *Journal of Ambient*

Intelligence and Humanized Computing, vol. 6, no. 5, pp. 623–633, 2015.

[20] S. M. Darwish and A. A. J. Almajtomi, "Metaheuristic-based vector quantization approach: a new paradigm for neural network-based video compression," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 7367–7396, 2021.

[21] R. B. Abduljabbar, O. K. Hamid, and N. J. Alhyani, "Features of genetic algorithm for plain text encryption," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, p. 434, 2021.

[22] B. Azam, S. Ur Rahman, M. Irfan et al., "A reliable auto-robust analysis of blood smear images for classification of microcytic hypochromic anemia using gray level matrices and gabor feature bank," *Entropy*, vol. 22, no. 9, p. 1040, 2020.

[23] D. Xiang, T. Tang, L. Zhao, and Y. Su, "Superpixel generating algorithm based on pixel intensity and location similarity for SAR image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 6, pp. 1414–1418, 2013.

[24] S. Kwon, H. Kim, and K. S. Park, "Validation of heart rate extraction using video imaging on a built-in camera system of a smartphone," in *Proceedings of the 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2174–2177, San Diego, CA, USA, August 2012.

[25] T. Tuncer, S. Dogan, F. Ertam, and A. Subasi, "A novel ensemble local graph structure based feature extraction network for EEG signal analysis," *Biomedical Signal Processing and Control*, vol. 61, Article ID 102006, 2020.

[26] H. Xie, Y. Ren, W. Long, X. Yang, and X. Tang, "Principal component analysis in projection and image domains—another form of spectral imaging in photon-counting CT," *Institute of Electrical and Electronics Engineers Transactions on Biomedical Engineering*, vol. 68, pp. 1074–1083, 2020.

[27] W. Liu, Z. Wang, N. Zeng, Y. Yuan, F. E. Alsaadi, and X. Liu, "A novel randomised particle swarm optimizer," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 2, pp. 529–540, 2021.

[28] Z. Yong, Y. Li-Juan, Z. Qian, and S. Xiao-Yan, "Multi-objective optimization of building energy performance using a particle swarm optimizer with less control parameters," *Journal of Building Engineering*, vol. 32, Article ID 101505, 2020.

[29] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, pp. 79–82, 2005.

[30] Q. Huynh-Thu and M. Ghanbari, "The accuracy of PSNR in predicting video quality for different video scenes and frame rates," *Telecommunication Systems*, vol. 49, no. 1, pp. 35–48, 2012.

[31] A. K. Moorthy and A. C. Bovik, "Efficient motion weighted spatio-temporal video SSIM index," *Human Vision and Electronic Imaging*, vol. XV, Article ID 75271I, 2010.