IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 17, 2020, accepted December 19, 2020, date of publication December 23, 2020,
date of current version January 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3046857

Predicting-Scheduling-Tracking: Charging Nodes With Non-Deterministic Mobility

YUANHAO LI, LINGYUN ZHONG, AND FENG LIN
College of Computer Science, Sichuan University, Chengdu 610017, China
Corresponding author: Feng Lin (linfeng@scu.edu.cn)

ABSTRACT In Wireless Sensor Networks (WSN), the energy problem plays the critical role in network performance and lifetime, because of the limited battery capacities of sensor nodes. Recently, emerging wireless charging technologies provide a promising approach to address the energy problem in WSN. Researchers construct Wireless Rechargeable Sensor Networks (WRSN), which introduce mobile chargers with high capacity batteries to charge sensor nodes. Most studies in WRSN have paid attention to charging static nodes or mobile nodes with deterministic trajectories. In this work, we explore how to charge nodes with non-deterministic mobility. We propose a novel approach, named Predicting-Scheduling-Tracking (PST), to perform charging tasks in this case. In the proposed scheme, different from the existing work, we guide the mobile charger to chase the sensor and recharge it. In our work, the base station runs an improved LSTM to predict the future locations of nodes periodically. Then, the mobile charger can select an appropriate node as the charging target by a charging scheduling algorithm. During the energy transferring, a Kalman-filter-based tracking algorithm is used to ensure the charging-required distance between the mobile charger and the target node. The simulation results show that the proposed charging scheme can fulfil the charging tasks in WRSN of nodes with non-deterministic mobility.

INDEX TERMS Wireless rechargeable sensor networks (WRSN), non-deterministic mobility, trajectory prediction, charging scheduling.

I. INTRODUCTION

Wireless sensor networks (WSN) comprise one or more base stations and many sensor nodes placed in a large area to monitor a physical environment cooperatively [1]. For its features of good self-organization and low cost, WSN have various application scenarios like environment surveying [2], [3], Smart City [4], Wildlife Tracking [5] and so on. The capacity of node battery often limits the working time of nodes, thus affecting the lifetime and performance of WSN, which has become a central issue restricting the development of WSN.

In recent years, there has been many researches on addressing the energy problem in WSN. We can divide most of the prior literature into three methods: energy conservation schemes [6], energy harvesting methods [7] and wireless charge approaches [8]. The energy conservation schemes can only improve the energy efficiency while not compensating for the energy consumed by nodes. Thus, the energy conservation schemes can only ease the energy problem in WSN but not address it. The energy harvesting methods prolong the lifetime of sensor nodes by converting nature energy into electricity with converters, such as solar panels [9]. Unfortunately, the efficiency of energy converting depends on the conditions of environment tightly in practice. In the past two decades, studies on wireless power transfer make it possible to recharge sensor nodes in a wireless approach. Specially, Magnetic Resonance Coupling [10] has attracted wide attention from researchers because of its long energy transmission distance and high transmission efficiency. As an application of wireless power transfer, researchers construct a new network architecture in which introduce chargers with high capacity batteries to charge sensor nodes wirelessly, namely Wireless Rechargeable Sensor Networks (WRSN).

Most of the researchers on WRSN pay particular attention to mobile charger (MC) scheduling algorithms that optimize the charging order of static sensors. Recently, some studies investigate the WRSN with mobile sensor nodes. To charge the mission-critical robots, He *et al.* [11] presented a tree-based schedule, which reduces the MC's travel distance without causing robot energy depletion. Cheng and Wang *al.* [12] explored the problem of static charging piles deployment so

The associate editor coordinating the review of this manuscript and approving it for publication was Moayad Aloqaily.

that mobile sensor nodes can get close to the piles for charging when their energy exhausts. These researches above generally assumed that the trajectories of sensor nodes are deterministic or the movements of nodes are controlled. However, this assumption is not always supported by the WSN applications. A well-known example is Wildlife Tracking [5], where sensor nodes are attached on each tracked animal. Thus, the nodal movement is out of control and non-deterministic, even the mobility model itself is the target of the network.

For this case, [13] proposed an interesting solution by exacting some fixed locations in the application area, called hotspots which are frequently visited by sensors and then scheduling the MC to wait and charge sensors at these hotspots based on a Reinforcement Learning (RL)-based algorithm. However, this approach can't adapt the changes of nodal mobility pattern, since the hotspots never change once they are selected. In addition, in this approach, the BS controls the MC to visit the hotspots, which means the MC has to wait for a certain time at a given hotspot even all the sensors that may access this hotspot have been charged.

In this work, we study how to charge nodes with non-deterministic mobility. Instead of making the MC wait for sensors at some fixed locations in [13], we try to guide the MC to chase the sensor directly in every charging task, to adapt the possible changes of nodal mobility patterns. To provide charging service in this case, we must address the following three problems.

1) Node Discovery Problem (NDP): Where are the sensor nodes? Before charging nodes, the MC has to find them first. Because of the non-deterministic mobility of sensors, the MC does not know the current location of each sensor. In addition, due to the limited resources in WRSN, the mobility management architecture [14] used in cellular networks can not be introduced in our work. The locations reported may lose efficacy in a short time, because sensor nodes may leave the locations when data packets transmit and the MC moves to the reported locations. No information on locations of nodes is the major challenge of our case. We have to find an approach to find the future locations of nodes so that the MC can meet nodes and charge them at these locations.
2) Node Selection Problem (NSP): Which node should be the current charging target? After obtaining the locations of sensor nodes, the MC has to select an appropriate node to charge under the limitation of residual capacity of batteries. NSP is like the scheduling problem in static WRSN. However, because of the non-deterministic movements of nodes, the MC only chooses one node to charge at a time for NSP in our case, while the BS controls the MC to access a batch of nodes at a time in static WRSN. In this way, the MC can begin a new charging task right after completing the current charging task.
3) Node Escort Problem (NEP): How does the MC track the moving node during the charging process? According to the studies on wireless power transfer, there are two fundamental requirements for wireless energy transferring: one is that the distance between the MC and the target node must be limited in some range, the other is that the energy transferring lasts for some time. To satisfy these two requirements, the MC has to escort the target node during the energy transferring. However, because of the non-deterministic mobility of nodes, the locations of nodes are dynamic and unknown. the MC has to track the moving node.

The problems above make to charge nodes with non-deterministic mobility become a very challenging task. NDP asks for a prediction approach to forecast the future locations of nodes, while NSP requires scheduling the MC to the optimal sensor for charging based on the prediction results. Furthermore, MC has to track the target node during the whole energy transferring. Therefore, the task to charge nodes with non-deterministic mobility has to be performed in a manner of ''predicting-scheduling-tracking''.

Based on the above analysis, we propose a charging scheme named Predicting-Scheduling-Tracking (PST) including three algorithms to address the three problems above, respectively. To the best of our knowledge, our work is the first approach to charge nodes with non-deterministic mobility by a chase method. To be specific, the main contributions of our work are as follows.

- We deduce and formalize the problems for charging nodes with non-deterministic mobility.
- To obtain the specific locations of sensor nodes, we first propose an improved LSTM algorithm to predict the future locations of each sensor node based on the previous trajectory. In addition to mining exclusive motion patterns from the unique trajectories of each sensor node, proposed algorithm corrects predicting result for each sensor node referring to the similar trajectory segments of other nodes in the same area.
- To maximize the performance of the network, we present a node selection scheme to select an appropriate node as the target node according to the predicting results and the energy status of each node.
- Finally, we introduce a Kalman filter based tracking algorithm to satisfy the requirements of wireless charging, which guides the MC to track the moving target nodes during the energy transferring.

The rest of this article is organized as follows. Section II reviews related work. Section III introduces the system model and formalizes the three problems in our case. Section IV demonstrates the proposed charging scheme. Section V shows the performance of our algorithm in the simulation, and Section VI concludes this article.

II. RELATED WORKS

According to sensor nodes' motion states in WRSN, we can divide existing charging schemes into two categories, one is

the WRSN of static sensor nodes, and the other is the WRSN of mobile sensor nodes.

Over the past decade, researchers have proposed a variety of charging schemes for WRSN of static sensor nodes, which try to find out the optimum sequence of the sensor nodes for MCs to recharge. For instance, [15] proposed a periodic scheme in scenarios where sensor nodes are uniformly or non-uniformly deployed. The MC calculates a shortest round path which links all the sensor nodes and then the MC patrols along the path and charges the sensor node whose energy is critical. Considering the deployment and status of sensor nodes, the authors in [16] modeled the charging problem as a TSP problem, a shortest Hamiltonian cycle could be constructed to be the solution. In another major study [17], Weifa Liang *et al*. presented a method of dispatching multiple MCs to maintain large-scale WRSN of life-critical sensor nodes. Liu *et al.* [18] introduced a novel concept called ''shuttling'' and an optimal charging algorithm, which achieved the minimum number of chargers and expanded the charging range by cooperation among chargers. In further work, aiming at maximizing the charging utility and minimizing the travel distance of MCs, [19] proposed a solution to charge multiple sensor nodes within the same energy transferring range simultaneously in a partial charging manner. Compared with traditional methods, this approach improves the charging efficiency and system performance of WRSNs. Guo *et al.* [20] studied the placement of wireless chargers to concurrently charge a given WSN and compared the charging efficiency of unique-frequency chargers and diverse-frequency chargers. According to [21], Tomar Abhinav *et al*. took the residual energy, distance to MC, and critical node density into consideration and proposed a fuzzy-logic-based algorithm to make an efficient charging schedule for on-demand charging in a dense WRSN. As the succeed study of [21], [22] introduced multiple MCs for on-demand WRSNs by partitioning the network in a load- balanced manner and calculating the adaptive recharging thresholds for sensor nodes.

Up to now, little attention has been paid to the research in WRSN of mobile sensor nodes. What's more, most of the existing studies assumed that the trajectories of sensor nodes are deterministic. For instance, [23] proposed a scheme to adapt the working power of the static charging pile to charge the sensor nodes moving nearby. The authors in [12] presented an approach to schedule mobile nodes with critical energy to the static charging piles for recharging, while redundant nodes take over the monitoring tasks of recharging nodes automatically. To maintain the performance of WRSN, [11] presented a tree-based schedule to charge the mission-critical robots, which arranges the MC waiting in the trajectory of exhausted sensor nodes without causing robot energy depletion. More recently, [13] was published as the first work to study the use of an energy-limited MC to charge mobile sensors with non-deterministic mobility. This work tried to exact some fixed locations that are frequently visited by sensors and then send the MC to wait and charge sensors at these locations. However, this approach requires a large quantity of history trajectory data to find out these locations and can not adapt to the changes of the nodal mobility model.

III. PRELIMINARIES

In this section, we first introduce the system model and then present the formulation of problems for charging nodes with non-deterministic mobility.

A. SYSTEM MODEL

We consider a WRSN with N mobile sensor nodes, a base station (BS), and a MC. Let S denote the set of N sensor nodes. The sensor node $s_i \in S$ is powered by a rechargeable battery with energy capacity E_{sensor}. The MC is also powered by a rechargeable battery with high capacity E_{mc}.

The sensor node s_i is equipped with GPS to record its location $l_i(t)$ at time t and sends this location data including current coordinate and timestamp and its residual energy to the BS every ΔT via a long distance communication tech like LPWAN. We call ΔT as the time unit in our case. The energy consumption of sensors consists of the following parts: the energy consumed by travelling, the energy consumed by sensing, and the energy consumed by data transferring and receiving. To be simplified, we use the static average energy consumption rate p_i to depict the energy consumption of node s_i. Each sensor node may perform a different task in the network. Thus, they have different p_i. Let r_i denote the residual energy of s_i. Then r_i can be expressed as:

$$r_i = E_{sensor} - p_i * t^i_{lapse} \tag{1}$$

where t^i_{lapse} denotes the time lapse since the sensor s_i was charged. Once its residual energy falls below a threshold $\theta * E_{sensor}$, s_i will send a charging request REQ_i to the BS. If a sensor node runs out of energy, we consider it is a dead node and cannot work anymore. In addition, sensor nodes can only get charged by the MC.

The BS, as the control center of WRSN, receives the trajectory data and charging requests sent by sensors and predicts the future locations of each sensor (see Section IV.A). The BS also transfers the prediction results, the residual energy of sensors and charging requests to the MC.

The MC buffers the data transferred from the BS. Before beginning a charging task, the MC selects one node as the target node based on the buffered data (see Section IV.B) and then moves to the predicted location of this node to encounter and charge it. When the MC moves into the range R_{com} of the target node, they can communicate with each other directly by unicast transmission. The target node advises its current location at every time elapse of Δ, where $\Delta \ll \Delta T$. In this way, the MC can track the target node during the rest of charging task. After fully charging the target node, the MC can select another target node and begin a new charging task. As for energy transferring, we assume the point-to-point charging pattern is used, namely only the target sensor can be fully charged when the MC moves to its range of R. The MC does not charge any other nodes during its travel.

The efficiency of wireless charging is η. During the charging process, the energy got by the target sensor e_i will be:

$$e_i = E_{sensor} - r_i = \eta^* mc_i \tag{2}$$

where mc_i is the energy costed by the MC for charging s_i.

The MC moves around the area at max speed V_{mc}, while it can adjust its speed during the charging task. Let p_{mc} denote the energy consumption speed of the MC for every distance unit. the MC departs from the BS with full energy for its very first charging task. Obviously, the MC has to get recharging before its energy runs off. Therefore, before each charging task, the MC will check if its residual energy is enough to support the new charging task and return to the BS. If not, the MC has to abort current charging task and return to BS. We call the duration between the MC departs from and returns to the BS as one charging cycle. The energy consumption of the MC contains two parts, namely the energy consumed by travelling and the energy consumed by charging. The residual energy of the MC r is formulated as below:

$$r = E_{mc} - p^*_{mc} L - \sum_{s_i \in C} mc_i \tag{3}$$

where L is the total distance the MC has travelled and C is the set of sensors that have been charged in the current charging cycle.

B. PROBLEM DEFINITION

In a WRSN with static nodes or mobility deterministic nodes, the charging scheduling problem is essentially to find an optimized charging sequence so that the MC can access the locations of static nodes or the junctions on the future trajectories of mobile nodes based on this sequence. However, to charge nodes with mobility non-deterministic, the BS and the MC know neither the locations of nodes nor the future trajectories of nodes. The historical trajectory data and charging request messages are the only knowledge that the BS gets.

The charging problem for WRSN of nodes with non-deterministic mobility differs itself from other charging problems in there is no information on future locations of nodes. As a result, the MC does not know where to encounter mobile nodes and charge them. Therefore, to charge nodes, this problem has to be addressed first. In section I, we induce this problem as Node Discover Problem (NDP). Recent studies [24]–[26] on mobility pattern showed the non-deterministic movements have a high degree of temporal and spatial regularity and can be predicted. Therefore, NDP is essentially a problem to find a time series forecasting model to predict the future trajectories of nodes. The definition of NDP is given as follows:

NDP: Given the historical trajectory of s_i at time t, NDP is to find a time series forecasting model H that predicts the location of the sensor node s_i at time $t + \Delta T$. The formulation of NDP is depicted as below

$$\hat{l}_i(t + \Delta T) = H(l_i(0), l_i(\Delta T), \ldots l_i(t - \Delta T), l_i(t)) \tag{4}$$

where $\hat{l}_i(t + \Delta T)$ is the prediction result.

To ensure the MC can find the target node, the error of prediction should be less than R.

$$\left| l_i(t + \Delta T) - \hat{l}_i(t + \Delta T) \right| \leq R \tag{5}$$

After getting the future locations of nodes, the MC will select a proper node as the target node for charging. We represent this problem as Node Selection Problem (NSP). Similar to other scheduling problems in WRSN, the goal of NSP is to improve the charging performance. The charging performance is reflected in two aspects. The first is to minimize the number of nodes whose energy exhaust, which is the original goal to introduce charging schemes into the network. The other is to maximize the energy obtained by sensors from the MC with limited energy capacity. Since the energy consumption of the MC comprises two parts, namely the energy for travelling and the energy for charging, to maximize the energy obtained by sensors essentially is to minimize the travel distance of the MC. Clearly, to minimize the number of dead nodes has priority over to minimize the travel distance. Based on this consideration, we take the NSP as a hierarchical multi-objective optimization problem and give its definition as follows:

NSP: Given all the future locations and residual energy of mobile nodes, the objective of NSP is to find the sensor s_i, which minimizes the number of dead nodes firstly and then minimizes the travel distance of the MC, under the constraint of the residual energy of the MC, i.e.

$$\arg\min(N_{dn}(s_i)) \tag{6}$$

$$\arg\min(Tr(s_i)) \tag{7}$$

subject to

$$\begin{cases} Tr(s_i) \approx dis(l(t), \hat{l}_i(t + \Delta T)) \\ mc_i + p^*_{mc}(Tr(s_i) + dis(\hat{l}_i(t + \Delta T), l(BS))) \\ \qquad\qquad +\text{margin} \leq r \end{cases} \tag{8}$$

where $N_{dn}(s_i)$ and $Tr(s_i)$ are the number of dead nodes and the MC's travel distance, respectively, if the node s_i is selected as the target node, $l(t)$ is the current location of the MC, $dis(l(t), \hat{l}_i(t + \Delta T))$ is the distance between $l(t)$ and the predicted location of s_i. In practice, we can use $dis(l(t), \hat{l}_i(t + \Delta T))$ to approximate $Tr(s_i)$. $l(BS)$ is the location of the BS and $dis(\hat{l}_i(t + \Delta T), l(BS))$ is the distance between $\hat{l}_i(t + \Delta T)$ and $l(BS)$. The margin is a constant for safety since the MC may move away from $\hat{l}_i(t + \Delta T)$ during charging s_i. Thus, $p^*_{mc} dis(\hat{l}_i(t + \Delta T), l(BS)) + \text{margin}$ can ensure that the MC has enough energy to return to the BS after charging s_i.

When MC meets the target node s_i at $\hat{l}_i(t + \Delta T)$, the MC has to track the movement of s_i, since the mobility of s_i is non-deterministic and the energy transferring takes some time. Fortunately, the MC and s_i can communicate with each other directly. During the energy transferring, s_i will send its location periodically and the MC has to track s_i based on these locations advised. We name this problem as Node Escort Problem (NEP). In NEP, the MC still can't use the advised locations directly, because of the GPS error and the

TABLE 1. List of notations.

Notation	Meaning
N	the number of sensor nodes
S	the set of sensor nodes
E_{sensor}	the battery capacity of sensors
E_{mc}	The MC's battery capacity
p_i	the energy consumption rate of s_i
r_i	the residual energy of s_i
$l_i(t)$	the location of s_i at time t
ΔT	the time interval between node's advisements to the BS
θ	the node's energy level for charging request
REQ_i	the charging request of s_i
R_{com}	the direct communication range between the MC and nodes
Δ	the time interval between node's advisements to the MC
R	the charging range of MC
η	the efficiency of wireless charging
e_i	the energy obtained by s_i from MC
mc_i	the energy costed by MC for charging s_i
r	the residual energy of the MC
V_{mc}	the max speed of the MC
p_{mc}	the energy consumption speed of MC for every distance unit
L	the total distance the MC has travelled
$\hat{l}_i(t)$	The result of prediction at time t

very small energy transmission range, which is 1m in our case. Thus, we have to employ a target tracking model to trace the target node. The definition of NEP is given as follows:

NEP: Given a series of historical trajectory of s_i at time t, NEP is to find a target tracking model F to predict $l_i(t+\Delta)$. the MC moves to the prediction result. The formulation of NEP is depicted as below

$$\begin{aligned} l(t+\Delta) &= \hat{l}_i(t+\Delta) \\ &= F(l_i(t_0), l_i(t_0+\Delta), \ldots, l_i(t-\Delta), l_i(t)) \end{aligned} \quad (9)$$

where t_0 denotes the beginning time of this charging process.

To ensure the charging process not to be interrupted, the distance of the MC and s_i can't be less than the charging range R during the whole charging task.

$$dis(l(t), l_i(t)) < R \quad (10)$$

We notice that NEP is similar to NDP in the goal of either problem is to find an approach to predict the future locations of nodes. However, NEP differs from NDP in its real-time requirement. The goal of NEP is to track the recent movement of the target node. Thus, the distance between the MC and the target node can fulfil the charging requirements. If NEP uses the same approach as NDP, the significant transmission delay between the BS and the MC is unacceptable. We need a new node tracking algorithm for NEP.

The major notations used in this article are listed in Table 1.

In the next section, we present our algorithms to solve these three problems, respectively.

IV. PROPOSED CHARGING SCHEME

We have analyzed and formularized the three problems of charging nodes with non-deterministic mobility so far. In this section, we present our charging scheme to solve the three problems one by one. The proposed charging scheme comprises three algorithms, namely Trajectory Prediction Algorithm (TPA), MC Scheduling Algorithm (MSA), and Target Tracking Algorithm (TTA), to address these problems respectively. In the proposed scheme, at every time unit, the BS runs TPA to forecast the future locations of sensors and transfers the prediction results to the MC. The MC buffers the data from the BS. When departing from the BS or completing a charging task, the MC runs MSA based on the buffered data to determine the next charging target or to return to the BS. Once the target node is determined, the MC moves to the prediction location of the target node. When meeting the target node, the MC runs TTA to keep tracing the target node during the energy transferring. To make it easier to understand how the proposed scheme works, Fig. 1 presents the finite state machine (FSM) definitions for the BS, sensor nodes, and the MC, respectively. The FSM in Fig. 1a defines the operations of sensor nodes, while the FSM in Fig. 1b and the FSM in Fig. 1c define the operations of the BS and the MC, respectively. The arrows in the FSM descriptions show the transitions of the devices from one state to another. The event causing the transition is shown above the horizontal line labeling the transition, and the actions taken when the event occurs are shown below the horizontal line. When no action is taken on an event, we use the symbol Λ below the horizontal to explicitly denote the lack of an action.

In the rest of this section, we depict the details of three algorithms one by one.

A. TRAJECTORY PREDICTION ALGORITHM (TPA)

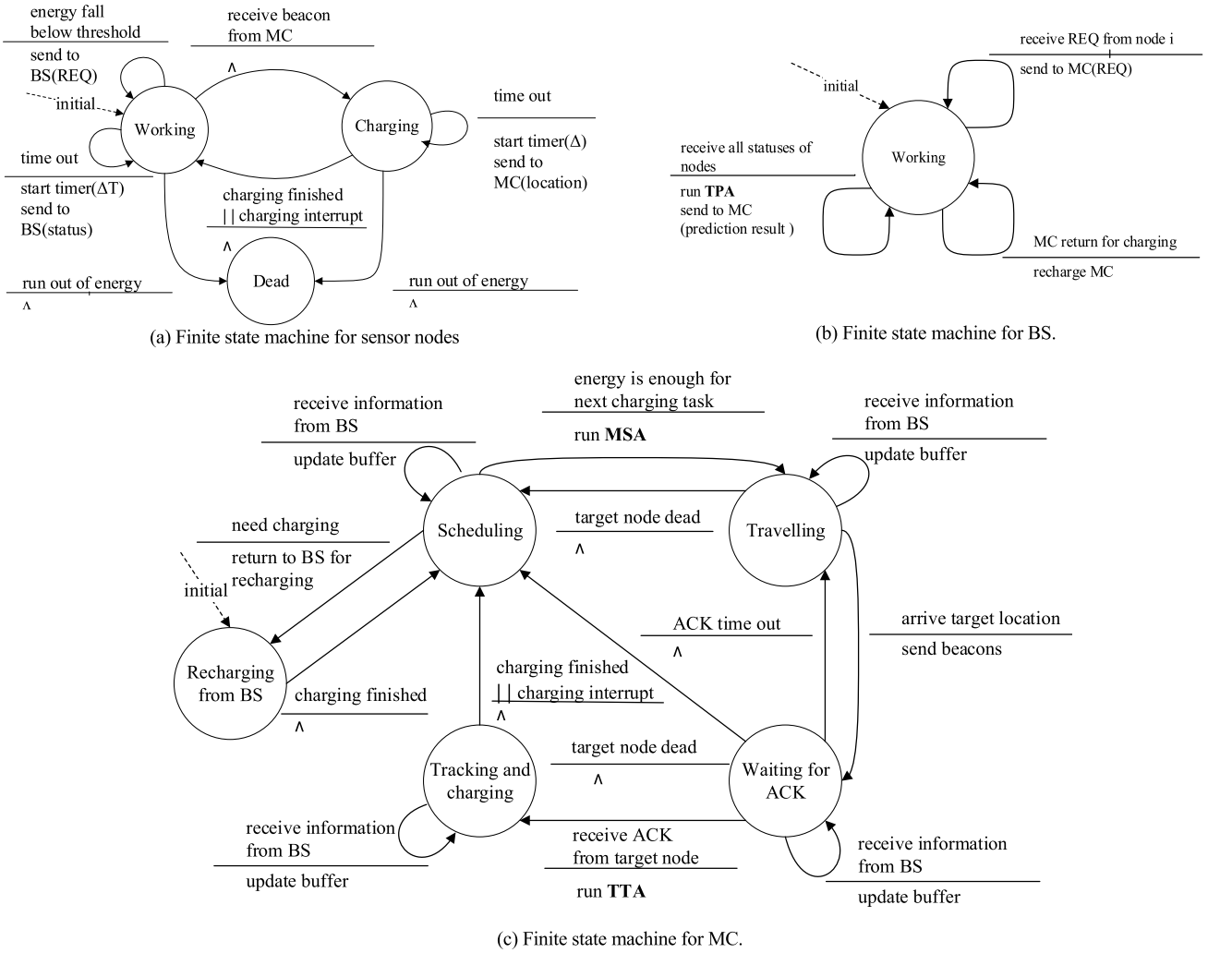
To address NDP, we have to find a time series forecasting model of nodes' trajectory.

Up to date, there are many studies for time series prediction, like Moving Average (MA), Exponential Smoothing (ES) and Auto-regressive Integrated Moving Average Model (ARIMA). Unfortunately, these approaches may not work well in our case because of the complexity of node mobility. Thus, we turn to neural network methods.

In the next, we present the data preprocessing first and then present the proposed neural network model to solve NDP.

1) DATA PREPROCESSING

The basic task of TPA is to address NDP problem in charging to nodes with non-deterministic mobility. This trajectory predicting algorithm has two characteristics that distinguish itself from the common time series forecasting approaches. First, there are inherent errors from GPS in the input historical trajectory data. Second, the goal of prediction is to let the MC find the target node and can be achieved when the MC moves into the communication range of the target node. Therefore, the prediction error is acceptable if it is less than R_{com}. Based



(a) Finite state machine for sensor nodes

(b) Finite state machine for BS.

(c) Finite state machine for MC.

FIGURE 1. Finite state machine.

on analysis above, we preprocess the trajectory by dividing the whole sensing area into non-overlapping cells sized in $L^*_{grid}L_{grid}$ and mapping the locations into corresponding cells, respectively. Without loss of generality, we have $L_{grid} < \frac{\sqrt{2}}{3}R_{com}$, which means the MC can communicate with the node directly if they are in the eight neighbor cells or in the same cell. In this way, after preprocessing, we ignore the GPS error and simplify the computation.

2) IMPROVED LSTM PREDICTING MODEL

In TPA, the LSTM model is introduced in light of its excellent performance in processing series data to achieve the prediction [26]. In more detail, owing to various nodes following different mobility patterns, it will be extremely huge calculation to figure out a unified mobility model for all sensor nodes. To this point, we try to find the mobility model for each sensor node individually. On the other hand, because all the sensor nodes move in the same monitoring area, there must be some common temporal and spatial regularity shared by them. If this kind of regularity is used, the convergence of the forecasting model will be accelerated.

Based on the above analysis, we design an end-to-end neural network to predict every sensor's trajectory based on its own previous trajectory and the most similar trajectory segment from the historical trajectories of all the other nodes. The model of the proposed network comprises two parts of input: the most similar segment and the most short-time segment. The short-time segment is the recent K locations of the predicted node. And the most similar segment is the same size segment from the previous trajectories of the other nodes estimated to have the most similarity to the short-time segment. Clearly, the most similar segment represents the common temporal and spatial regularity shared by the other nodes, while the short-time segment represents the mobile features of the predicted node.

The proposed network mainly comprises the LSTM block mining the mobility pattern from the short-time segment and the fully connected block used to extract the features about the possible movement from the most similar segment. The

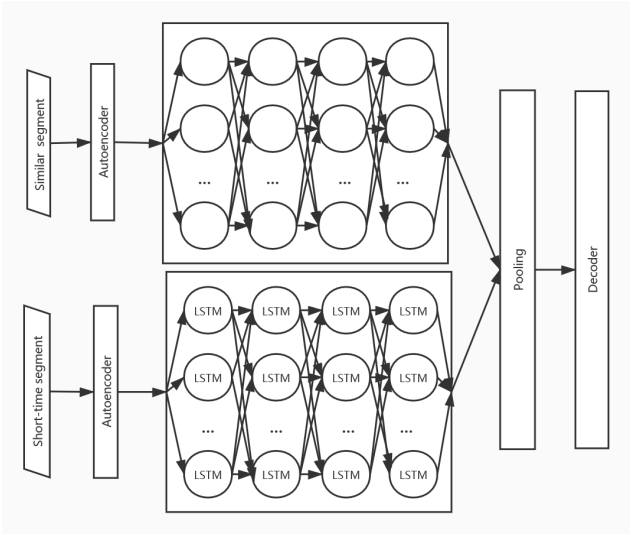


FIGURE 2. Model of the LSTM based predicting algorithm.

LSTM block consists of an autoencoder layer and four LSTM layers with 32 LSTM cells in each LSTM layer, and the fully connected block consists of an autoencoder layer and four fully connected layers with 32 neurons in each layer. A pooling layer is used to mix the output from the LSTM block and the fully connected block in proper proportions. The output of the pooling layer is passed on to a decoder layer for prediction. The architecture of the neural network is depicted in Fig. 2.

First, we extract the most similar segment. As is aforementioned that most similar segment from the previous trajectories of the other nodes can accelerate the convergence of the forecasting model, we first extract the most similar segments from the segments of previous trajectories. Let T_i denote the short-time segment of the predicted node s_i. At time t, we have:

$$T_i = [l_i(t-(K-1)\Delta T)), \ldots, l_i(t-\Delta T), l_i^t(t)] \quad (11)$$

The possible candidate of the similar segment from node s_j at time t' can be defined as:

$$T_j^{t'} = [l_j(t'-(K-1)\Delta T)), \ldots, l_j(t'-\Delta T), l_j(t')] \quad (12)$$

where $j \neq i$.

We select the Spearman's coefficient to be the metric of the similarity between T_i and the similar segment. Thus, the task of exacting the most similar segment becomes to find $T_j \in \{T_j^{t'}\}$ that has the max Spearman's coefficient with T_i. To simplify the calculation, we use a simple approach to find the candidates of the similar segment. We first find the smallest rectangle Rct_i which contains all the locations in T_i. For each sensor $s_j (j \neq i)$, we use all the historical locations $l_j(t_0)$ which are in Rct_i, as the $l_j(t'-(K-1)\Delta t)$ in equation (11) to generate $T_j^{t'}$. Then, for all the $T_j^{t'}$, we find the smallest rectangle $Rct_j^{t'}$ which contains all the location in $T_j^{t'}$. We calculate the Spearman's coefficient only if $Rct_i = Rct_j^{t'}$.

We can find the most similar segment of sensor s_j according to the calculated Spearman's coefficient. In this way, we can find the most similar segment and its calculated Spearman's coefficient θ_t from the previous trajectories of the other nodes after traversaling all the nodes other than s_i. The detailed algorithm is depicted in Algorithm 1.

Algorithm 1 Select the Most Similar Segment

```
T_j ← T_i and θ_t ← 0
Find the smallest rectangle Rct_i which contains all the
 locations in T_i
for each node s_j ∈ S − {s_i} do
    for (t_0 = 0; t_0 < t − K + 1; t_0 = t_0 + Δt ) do
      if l_j(t_0) in Rct_i then
        Find the smallest rectangle which contains
         all locations in T_j^{t_0} beginning with l_j(t_0) of
         length K
        if Rct_i = Rct_j^{t_0} then
          Calculate the Spearman's coefficient
           θ_Spearman between T_i and T_j^{t_0}
          if θ_Spearman > θ_t then
            T_j ← T_j^{t_0} and θ_t ← θ_Spearman
          end if
        end if
      end if
    end for
end for
Output: T_j and θ_t
```

In order to efficiently mine the mobility patterns of sensor nodes, we employ an autoencoder layer [27] which has been proved useful in mining process to encode both the segment T_i^t and the segment $T_i^{'t}$.

$$e_t = W_e T_i^t + b_e \quad (13)$$

$$e'_t = W_e T_i^{'t} + b_e \quad (14)$$

where W_e and b_e are the parameters in this fully connected layer. Segments are converted to $T^* N_{encode}$ matrix through the autoencoder layer, where N_{encode} denotes the number of the dimensions of the matrix.

In order to mine the long-term respective features of each sensor node's movement, we adopt four fully connected the LSTM layers to process the matrix e_t, which have been proved successful in sequence learning [28]. LSTM is a type of RNN with memory units, so it is widely used in long-term sequence modeling without suffering from "Vanishing gradients".

In the proposed LSTM, we take the first LSTM layer as an example to introduce the specific update process of the matrix e_t to the final prediction result:

At the beginning, we calculate the output of the input gate i_t:

$$i_t = g(W_{zi} z_t^n + W_{hi} h_{t-1}^n + W_{si} s_{t-1} + b_i) \quad (15)$$

where W_{zi}, W_{hi}, W_{si} are the parameters of the connection weight for the input from the upper layer, b_i is the bias of the input gate, and the activation function g is *Sigmoid*.

Then we update the value of the forget gate f_t :

$$f_t = g(W_{zf} z_t^n + W_{hf} h_{t-1}^n + W_{sf} s_{t-1} + b_f) \tag{16}$$

where W_{zi}, W_{hi}, W_{si} are the parameters of the connection weight for the input from the upper layer, b_f is the bias of the forget gate, and the activation function g is *Sigmoid*. The function of the forget gate is to control the memory of the previous moving pattern.

In the next step, the state of the memory units s_i is changed to:

$$s_t = i_t \odot \tanh(W_{zs} z_t^n + W_{hc} h_{t-1}^n + b_c) + f_t \odot s_{t-1} \tag{17}$$

where W_{zi}, W_{hi}, W_{si} are the parameters of the connection weight for the input from the upper layer, b_c is the bias of the memory units.

Last but not the least, we calculate the value of the output gate o_t:

$$o_t = g(W_{zo} z_t^n + W_{ho} h_{t-1}^n + W_{so} s_{t-1} + b_o) \tag{18}$$

where, W_{zi}, W_{hi}, W_{si} denote the parameters of the connection weight for the input from the upper layer, b_c is the bias of the output gate, and the activation function g is *Sigmoid*.

Finally, we get the output of this LSTM layer h_t^n:

$$h_t^n = o_t \tanh(c_t) \tag{19}$$

We acquire the information about the future locations of predicted node from its own trajectory through the four LSTM layers.

3) FULLY CONNECTED NETWORK

We adopt the most similar segment to gain more information to correct and improve the predicting of future locations. Different from the segment of the sensor node which need predicting, what we want to extract from the most similar segment is the information about the future locations just depending on the state in the same area, without considering the long-term effect. Hence, we adopt four fully connected layers to process the matrix e_t':

$$h_t' = We_t' + b \tag{20}$$

where W, b represent the connection weight between fully connected layers and the bias.

More information about the future locations is acquired from the most similar segment through fully connected layers.

4) DECODE THE INFORMATION TO FIGURE OUT THE FUTURE LOCATIONS

For the purpose to gain the future locations, we pool the output from the LSTM layers and the fully connected layers based on θ_t, while μ is set to limit the influence from the most similar segment:

$$d_t = \varphi(W_d \times (\mu\theta_t h_t + (1 - \mu\theta_t)h_t') + b_d) \tag{21}$$

where W_d is the parameter of the decoder layer and b_d is the bias of the decoder layer. Finally, we get the future cell where the target sensor will access from d_t. Then, the BS can use the center point of the cell as the future location of the target sensor.

B. MC SCHEDULING ALGORITHM (MSA)

Since NSP is a hierarchical multi-objective optimization problem and minimizing the number of the dead nodes is the first goal, MSA aims at minimizing the number of the dead nodes first and then minimizing the charging travel distance. Therefore, the node selection algorithm works the following way:

First, MSA checks the request messages received, because REQ_i means s_i is in the state of energy emergency. The MC chooses the sensor s_i with the lowest energy in the buffered REQ_i. Then, the MC verifies the constraint (Eq. 7). If the constraint is satisfied, the MC starts the charging task for the sensor s_i. Otherwise, the MC chooses the other node whose energy is the second lowest in the buffered request messages. The selection repeats until all the request messages are checked or a node is selected.

If there is no qualified node selected based on the request messages, MSA tries to select the target node with minimizing the charging travel distance. The MC broadcasts beacons to discover the sensor node nearby and selects the sensor node with the lowest residual energy from nodes responding to the MC. To void charging the sensors nearby repeatedly, a queue is used to log the sensors charged by the MC recently. As a result, the MC can ignore the responses from these sensor nodes in the queue.

With no sensors need charging nearby, MSA guides the MC to charge the sensors in the sub-area of the largest energy requirement. For this purpose, we define the set S_i' as the neighbor sensor nodes of s_i. Each sensor node $s_j \in S_i'$, $(j \neq i)$ means $\hat{l}_j(t+\Delta t)$ belongs to the same cell or the eight neighbor cells of $\hat{l}_i(t + \Delta t)$. Then, we can use the metric $E_{gain}(i)$ as below to indicate the energy requirements of sub-area of node s_i :

$$E_{gain}(i) = (E_{sensor} - r_i) + \lambda \sum_{s_j \in S_i'} (E_{sensor} - r_j) \tag{22}$$

where $0 \leq \lambda \leq 1$ is a coefficient.

The sensor node with the max energy requirement in its proximity will be selected as the target of the next charging task. Once the target node is selected, the MC checks the constraint in Eq. 7. If its residual energy can't meet the constraint, the MC has to abandon this charging task and return to the BS for recharging.

The details of the proposed MC scheduling algorithm are described in Algorithm 2.

Algorithm 2 Select the Target Sensor for Charging

while the MC's buffer for REQ is not empty **do**
  Find the sensor s_m's REQ in the MC's buffer with the lowest energy level and delete s_m's REQ, $s_k \leftarrow s_m$
  if s_k satisfies Eq. 7 **then**
    Return s_k
  end if
end while
The MC sends beacons
if sensor s_j responds to the MC and not in the queue for recent charged sensors **then**
  $s_k \leftarrow s_j$
else:
  Find the sensor node s_i with max energy requirement $E_{gain}(i)$ in its proximity
  $s_k \leftarrow s_i$
end if
if s_k can't satisfy Eq. 7 then
  $s_k \leftarrow BS$
Output: s_k

If the target node is a node nearby, the MC tracks and charges it by the target tracking algorithm (see Section IV.C). Otherwise, the MC leaves for the prediction location of the target sensor node at the max speed V. The target location may be updated if the new prediction location of the target node is received before the MC arrives. After arriving, the MC starts to send out beacons right away to get connected with the target sensor. Once the direct connection is made, the MC can track and charge the target node via the target tracking algorithm. If failing to meet the target node, the MC has to abandon this charging task and start a new one.

C. TARGET TRACKING ALGORITHM

Once connected to the target node, the MC can communicate with the target node for the recent movement information. To achieve the energy transferring, the MC has to escort the target node during the whole transferring process. The target tracking algorithm is proposed to achieve this goal.

We normally regard the mobility of the target node as a linear trace, because the prediction step Δ is small. The GPS error of locations can be treated as the Gaussian noise approximately. Therefore, the Kalman filter is introduced to predict the location of the target node, which has excellent performance in short-time prediction. The details of the target tracking algorithm are as follows.

According to the Kalman filter, the location $\hat{l}_i(t+\Delta)$ of the target sensor s_i at time $t+\Delta$ relates to the location $l_i^+(t)$ estimated by the Kalman filter at previous time t, which can be denoted as

$$\hat{l}_i(t+\Delta) = F_t l_i^+(t) + \varepsilon_{t+\Delta} \tag{23}$$

$$\varepsilon_{t+\Delta} \sim N(0, Q_{t+\Delta}) \tag{24}$$

where F is the matrix of the mobility model of the target node, $\varepsilon_{t+\Delta}$ is the noise of our mobility model and $Q_{t+\Delta}$ is the covariance matrix of the mobility model. Also, the observed location $l_i(t+\Delta)$ of the target node s_i is related to the observation location, which can be denoted as:

$$l_i(t+\Delta) = G\hat{l}_i(t+\Delta) + \omega_{t+\Delta} \tag{25}$$

$$\omega_{t+\Delta} \sim N(0, R_{t+\Delta}) \tag{26}$$

where G is the matrix of the observation model, ω_t is the noise of the observation model and $R_{t+\Delta}$ is the covariance matrix of the observation model. In this work, we treat the error caused by observation as the Gaussian noise because of the use of GPS application.

The Kalman filter mainly consists of two steps. The first step in this process is to predict the current location on the previous location:

$$\hat{l}_i(t+\Delta) = Fl_i^+(t) \tag{27}$$

Then the predicted location can be denoted as:

$$P_{t+\Delta}^- = FP^+F^T + Q_{t+\Delta} \tag{28}$$

where $P_{t+\Delta}^-$ is the predicted covariance matrix of the location vector and $P_{t+\Delta}^+$ is the updated covariance matrix of the location vector.

For the second step, we update the location based on the observation model:

$$l_i^+(t+\Delta) = \hat{l}_i(t+\Delta) + K_{t+\Delta}(l_i(t+\Delta) - G_{t+\Delta}\hat{l}_i(t+\Delta)) \tag{29}$$

and update the covariance of the updated location:

$$P_{t+\Delta}^+ = (I - K_{t+\Delta}G_{t+\Delta})^T P_{t+\Delta}^-(I - K_{t+\Delta}G_{t+\Delta}) + K_{t+\Delta}^T R_{t+\Delta}K_{t+\Delta} \tag{30}$$

where K_t is the Kalman gain matrix, which is estimated as:

$$K_{t+\Delta} = P_{t+\Delta}^- G_{t+\Delta}^T (G_{t+\Delta}P_{t+\Delta}^- G_{t+\Delta}^T + R_{t+\Delta})^{-1} \tag{31}$$

During the whole process of the wireless charging, we use the location $\hat{l}_i(t+\Delta)$ as the prediction result of the target nodes at time $t+\Delta$.

Then the MC adjusts its speed of travelling to track the target node:

$$V_t = d_t/\Delta \tag{32}$$

where d_t is the distance between the MC and the location $\hat{l}_i(t+\Delta)$, if $V_t > V$, we modify the speed V_t to V.

After completing the energy transferring, the MC runs MSA to start a new charging task.

V. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of the proposed recharging scheme. We carried out three groups of simulations. The first group of simulations is used to validate the LPA algorithm, which is the fundamental of the proposed scheme. The second group of simulations aims at verifying

the effectiveness of the three proposed algorithms. Finally, we compare the performance of the proposed scheme with the RL-based charging algorithm (RLC) [13]. To the best of our knowledge, RLC and our proposed scheme are the only two charging schemes for nodes with non-deterministic mobility.

In the next part, we first introduce the simulation setup, and then depict the three groups of simulations respectively.

A. SIMULATION SETUP

To evaluate the performance of the proposed scheme under the real environment, we have carried out the simulation by replaying mobility from the CRAWDAD data set [29]. The data set used in our simulation records the human mobility trace of 36 students in NCSU. Each trajectory is represented by a sequence of timestamped points and each point gives the information of the latitude and longitude. In our simulation, we choose 17 students as the mobile nodes, whose trajectory is over 30000 seconds.

In our simulation, to simplify the calculation, we convert the latitudes and longitudes in trajectories into 2D coordinates. According to the maximum and the minimum values of each dimension for all the locations in 2D coordinates, we can figure out a 2D rectangle as the network area. Actually, the network area is set as $2021.00m \times 1990.93m$. The BS is placed in the center of the rectangle. The time interval between two location advisements to the BS is 30 seconds, while the time elapse between two location advisements to the MC directly is 1 second.

The communication range between the node and the MC is $45m$, and the L_{grid} is $20m$, which ensures $L_{grid} < \frac{\sqrt{2}}{3}R_{com}$ Each node is equipped with a $3.6V575mAH$ rechargeable battery, namely the capacity is $7500J$. The static energy consumption rate for each node is uniformly set in the range of $[40 \times 19.5/1000W, 120 \times 19.5/1000W]$. Each node will send the request message to the BS when its energy level is below 0.2×7500J. As for the MC, we assume the MC's capacity of batteries is $250kJ$, which is used for both travelling and recharging. The maximum speed of the MC is set to $4m/s$ and the travelling costs at the speed of $25J/m$. According to [10], the charging range of the MC is $1m$ and the efficiency of wireless charging is 0.9.

Since the Δt is set as 30 seconds, there are 1000 locations for each trajectory. Since TPA needs the first 750 locations to learn the mobility patterns of sensors, we start the simulation from $750^*30s = 22500s$. The simulation lasts for $250 \times 30s = 7500s$. Moreover, we set the residual power of each sensor is uniformly range from $2600J$ to $5200J$ at the beginning of simulation, which reflects the energy of the sensor closer to actual. It is noticed that the minimum energy consumption rate of sensors is $40 \times 19.5/1000W$ and the time simulation is $7500s$. At end of the simulation, each sensor consumes at least $40 \times 19.5/1000W \times 7500s = 5850J$. At the beginning of simulation, the maximum residual power of sensors is $5200J$. Therefore, each sensor runs out of energy if it does not get charged during the simulation.

TABLE 2. Simulation parameters.

Parameter	Value
L_{grid}	$20m$
N	17
E_{sensor}	$7500J$
E_{mc}	$250kJ$
p_i	$[40\times19.5/1000W, 120\times19.5/1000W]$
Δt	$30s$
θ	0.2
R_{com}	$45m$
Δ	$1s$
R	$1m$
η	0.9
V_{mc}	$4m/s$
p_{mc}	$25J/m$
λ	0.5

Following similar settings in [13], [30], the main parameters are listed in Table 2.

B. VALIDATION OF TPA

In the experiment, the proposed predicting model is constructed in TensorFlow and evaluated on the platform of an NVIDIA GeForce GTX 1080Ti.

We cut the first 750 locations as the training set and the rest 250 locations as the testing set for TPA.

The mean square error (MSE) is used as the loss function and the optimization method is RMSprop [31] with a learning rate of 0.001 and the decay of 0.9. For initialization, we use the random values as the initial parameters of the network. For each sensor node, the network is trained for more than 200 epochs.

Since the goal of TPA is to guide the MC to find the target sensor and to make direct communication with it, we can accept the prediction results which are close to the actual locations of the predicted sensor. Thus, the performance of the prediction results can be observed through the deviations which are the distance between the prediction results and their actual locations, and the accuracy rate, which is the ratio of acceptable prediction results and all prediction results. The deviation d is defined as:

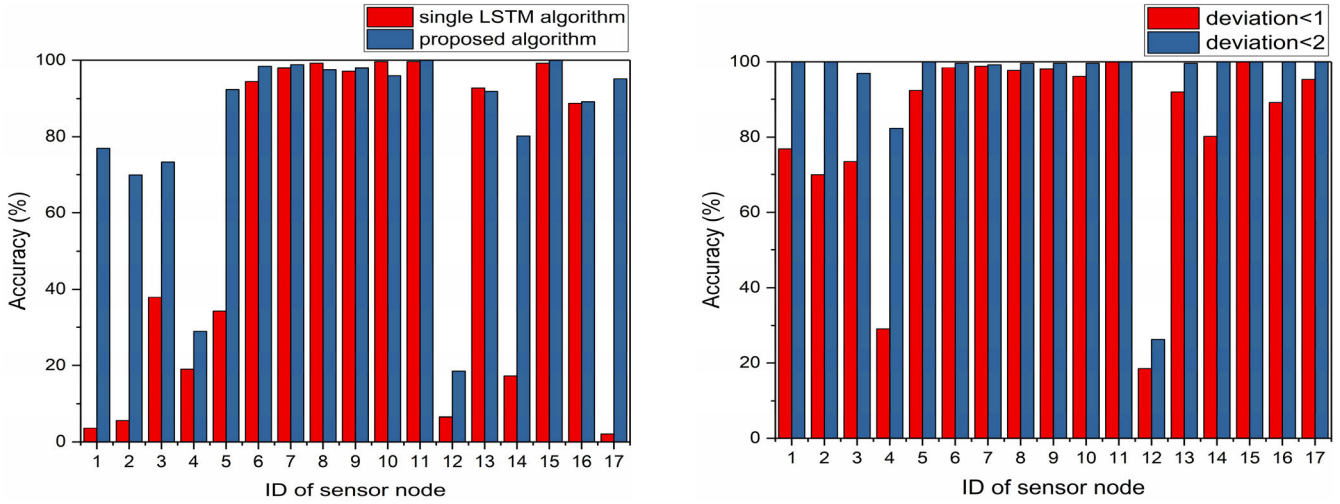
$$d = \max(\left|\hat{x}_t - x_t\right|, \left|\hat{y}_t - y_t\right|) \tag{33}$$

where (x_t, y_t) is the cell the target sensor accessed at time t and $(\hat{x}_t, \hat{y}_t)$ is the prediction result.

The accuracy rate A is defined in below:

$$A = \frac{P(d < \varepsilon)}{P} \tag{34}$$

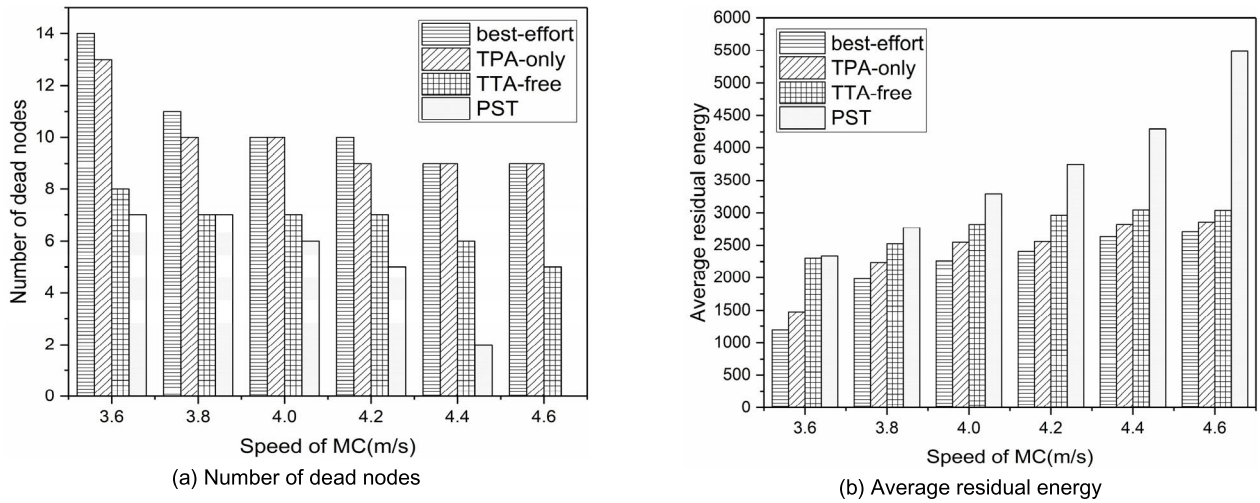
where P is the number of prediction results and $P(d < \varepsilon)$ is the number of prediction results whose deviation less than the constant ε. Notice that the deviation is the distance of cells, since we mapped the locations into cells in data preprocessing. Therefore, according to Eq. 33, the $d < 1$ means the prediction is correct only if the prediction cell is exactly the



(a) Compare with single LSTM model.

(b) Different deviation.

FIGURE 3. Accuracy of trajectory prediction.



(a) Number of dead nodes

(b) Average residual energy

FIGURE 4. Impact of the speed of MC.

real cell, and the $d < 2$ means the prediction is correct if the prediction cell is the real cell or one of the eight neighbor cells of the real cell.

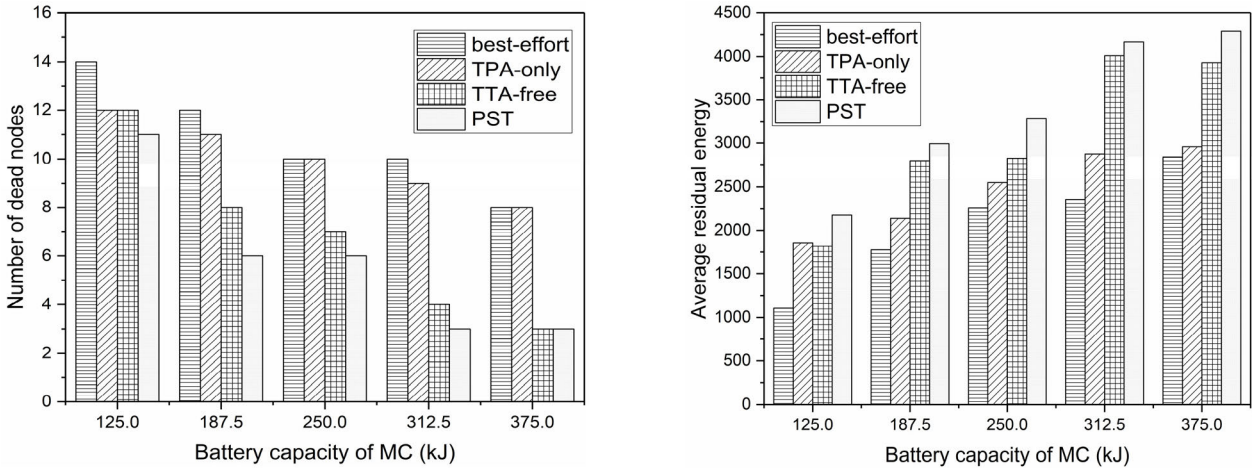
To evaluate the performance of the proposed algorithm, we compare it with a single LSTM model which use short-time segment as the network input and the LSTM block consisting of an autoencoder layer, four LSTM layers and a decoder layer as the same size as the proposed improved model. Fig. 3 shows our simulation results.

As is shown in Fig. 3a with deviation $d = 1$, compared with the single LSTM model, the proposed predicting model shows significant improvement in the prediction for most nodes like 1,2 and 14, meaning the similar trajectory segment of the other sensors is helpful. Contrary to expectations, there is some little performance degradation in some nodes like the node 8 and 10 on which the single LSTM predicting algorithm performs extremely well, an explanation for this might be that some temporal and spatial regularity from the similar trajectories of different nodes brings unexpected error to the prediction.

As shown in Fig. 3b, when the acceptable deviation is expanded from 1 to 2, the performance is much better, the prediction accuracy rate of 16 nodes exceeds 80% with the proposed algorithm.

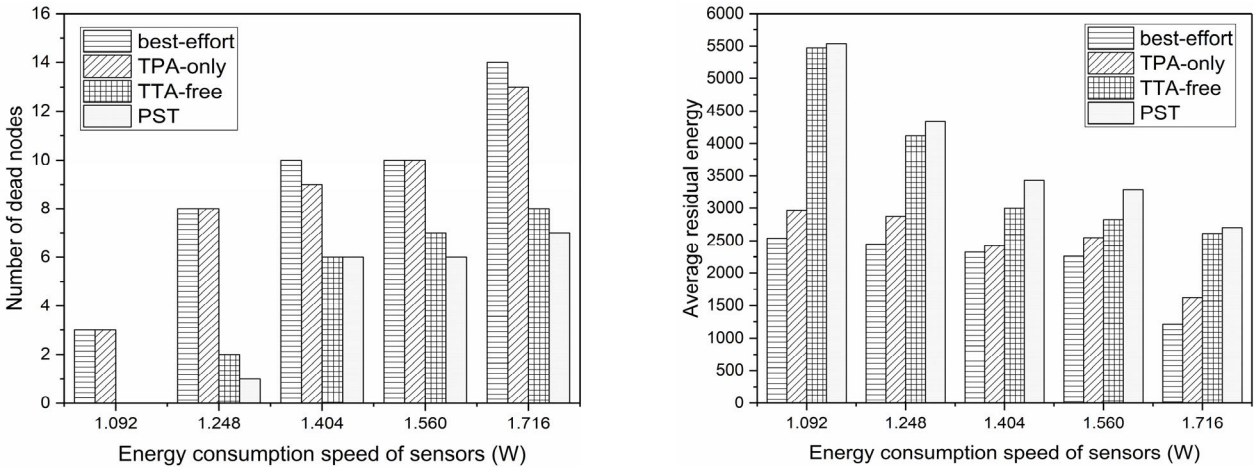
C. EFFECTIVENESS OF THE THREE PROPOSED ALGORITHMS

We further carry out some extensive simulations to verify the effectiveness of the three proposed algorithms. To illuminate the effectiveness of our proposed algorithms, we compare our PST scheme with the three simplified versions in conditions with various speeds of the MC. These three simplified charging schemes are described as below;



(a) Number of dead nodes. (b) Average residual energy.

FIGURE 5. Impact of the battery capacity of MC.



(a) Number of dead nodes. (b) Average residual energy.

FIGURE 6. Impact of the energy consumption speed of sensors.

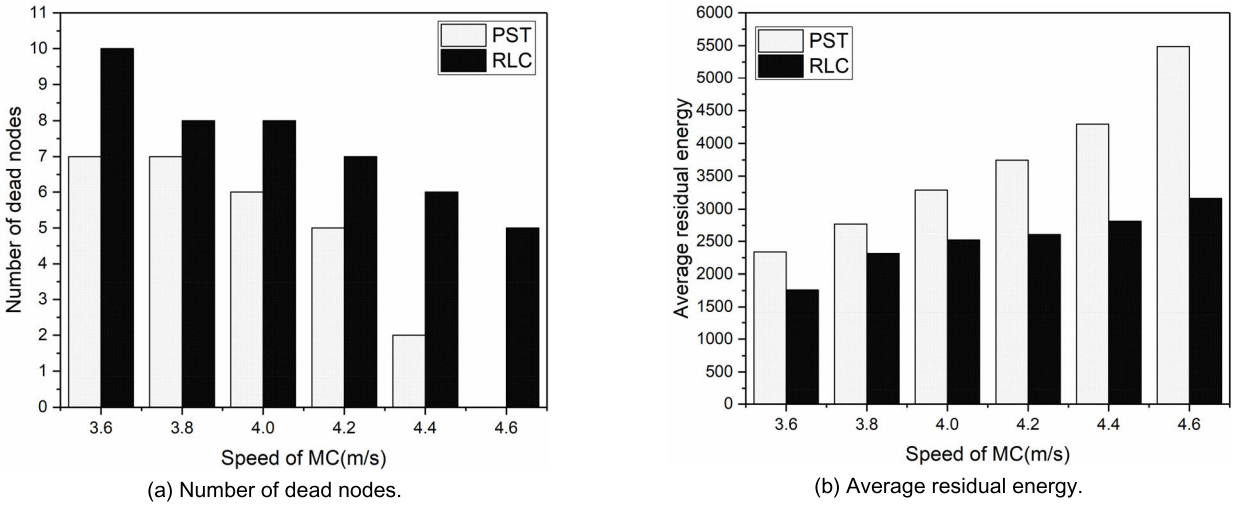
The first simplified version named best-effort charging, in which the MC always selects the sensor with the lowest energy as the target and then moves to the location reported by the target node for the energy transferring. During the energy transferring, the MC always moves to the location advised by the target node. Clearly, none of three proposed algorithm is used in this approach.

The second simplified charging scheme is called TPA-only charging, in which the MC still tries to charge the node with the lowest energy. Differing from the best-effort charging approach, the TPA-only charging approach introduces the TPA to find the target node. During the energy transferring, the MC acts in the same way as the best-effort charging.

The last simplified charging scheme, called TTA-free charging, employs the TPA and the MSA. The difference between the TTA-free charging and the proposed scheme is TTA-free charging does not use the TTA instead of acting as the best-effort charging or TPA-only charging during the energy transferring.

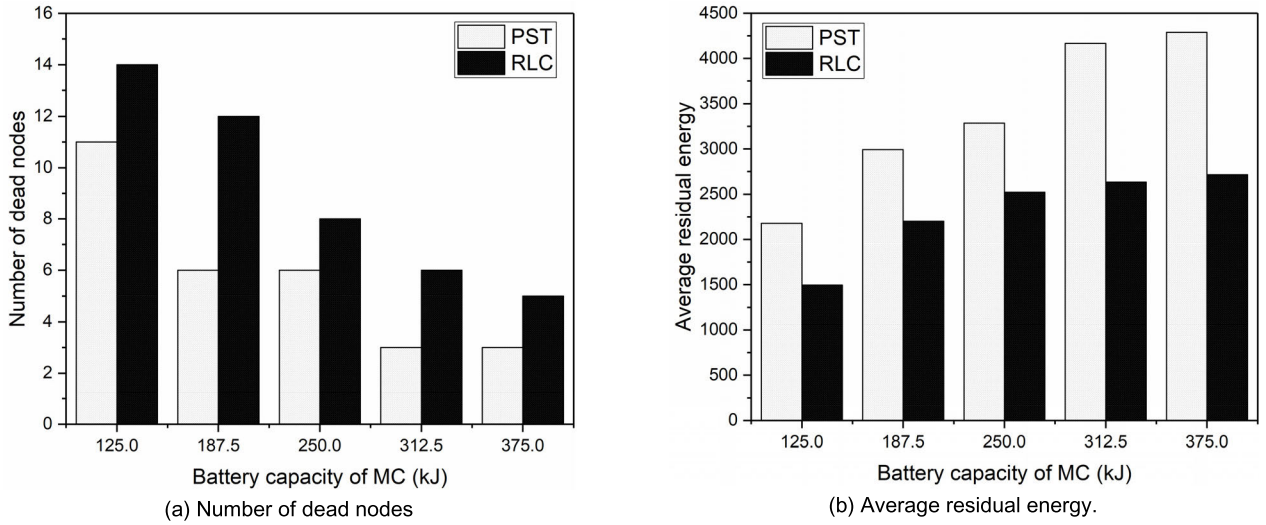
Therefore, the comparison between the best-effort charging and the TPA-only charging illuminates the effectiveness of the TPA, while the comparison between the TPA-only charging and the TTA-free charging demonstrates the effectiveness of the MSA. The effectiveness of the TTA can be depicted by comparing the TTA-free charging and our proposed scheme.

We vary some important parameters to observe their impacts on the number of the dead nodes and the average residual energy. Fig. 4 compares the performance of these four schemes on the impact of various maximum moving speeds of the MC. In Fig. 4a, as expected, the proposed scheme always has a smaller number of the dead nodes than other approaches. Especially when the maximal speed of the



(a) Number of dead nodes. (b) Average residual energy.

FIGURE 7. Impact of the speed of MC.



(a) Number of dead nodes (b) Average residual energy.

FIGURE 8. Impact of the battery capacity of MC.

MC is higher than 4.6m/s, there is no dead node for our proposed scheme. We also notice that the TPA-only scheme only slightly outperforms the best-effort scheme, while the TTA-free scheme has a much better performance. One reason is that the MSA algorithm guides the MC to the nodes with the least travel distance while there is no request message. Meanwhile, the best-effort scheme or the TPA-only scheme always drives the MC to charge the node with the lowest energy. Therefore, in the two schemes with MSA, the MC has more opportunities to charge sensors than in the two schemes without MSA. Although with TPA the TPA-only scheme has more chances to find the target node than the best-effort scheme, it may not perform the energy transferring successfully because of the lack of MSA. Thus, the TPA-only scheme only has a minor improvement on the best-effort scheme.

We can conclude the similar conclusion from Fig. 4b, where the average residual energy ratio is used as the comparing metric. As presented in Fig. 4b, the proposed scheme always achieves the higher average residual energy than the other schemes. We also notice that the average residual energy of the PST scheme increases sharply with the maximum moving speed, while the same metrics of the other schemes have a limited increase. This is because, although TTA is used in PST, the MC can't chase the target node successfully and fail the energy transferring when having a low moving velocity. With a high moving velocity, TTA can ensure to track the charged node successfully during the energy transferring.

We also vary the MC's battery capacity in our simulations, with results presented in Fig. 5. As the battery capacity increases, the numbers of dead nodes of all schemes decrease,

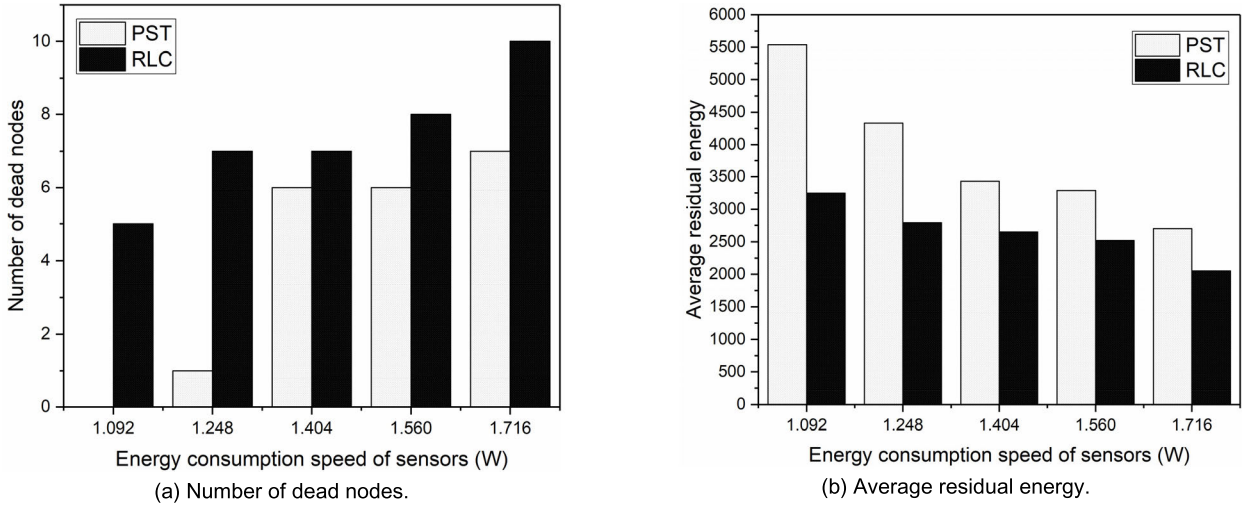


FIGURE 9. Impact of the energy consumption speed of sensors.

while the nodal average residual energy of all schemes rises. This is because the MC with a larger battery capacity needs fewer times to return to the BS for recharging and can charge more nodes during one charge cycle. As expected, the scheme that uses more algorithms we proposed achieves the better performance.

Fig. 6 shows the impact of the sensors' energy consumption speed. We carry out simulations where the static average energy consumption rate of each sensor is multiplied by 0.7, 0.8, 0.9, 1.0, 1.1 and set the expectations as the vertical coordinates of Fig. 6a and Fig. 6b. With the energy consumption speed increase, the number of dead nodes of all schemes rise, while the sensors' average residual energy of all schemes decreases. When the sensors' energy consumption speed increase, only one MC cannot serve all the sensors with critical energy state. As a result, the number of dead nodes rises and the sensors' residual energy decreases. It is also noticed in Fig. 6a the best-effort scheme has a close performance to the TPA-only scheme, while the TTA-free scheme has a close performance to the PST scheme. The reason, which is the same as in Fig. 4a, TPA only ensure the MC can find more target nodes than the best-effort scheme, but MSA optimizes the charging performance. On the other hand, TTA is not sensitive to the sensor's status.

As a result, the TTA-free scheme has more similar performance to the PST scheme in Fig. 6 than in Fig. 4.

D. PERFORMANCE COMPARISON

Finally, we evaluate the performance of the proposed charging scheme by comparing it with RL-based Charging Algorithm (RLC) [13]. As mentioned in section I, RLC is a reinforcement learning based charging scheduling algorithm. After mining some fixed locations, which are frequently visited by sensors, as the charging places in the application area, RLC uses reinforcement learning method to construct a charging scheme, which directs the MC to visit these hotspots at an off-line sequence and stay at each charging place for different time units to provide charging service.

Fig. 7 compares the performance on the impact of the maximum speed of the MC. As the maximum speed of the MC rises, the numbers of the dead nodes of both schemes decrease, while the average residual energy of both schemes rises. In particular, there is no sensor running out of its energy during the simulation when the maximum speed of the MC is 4.6m/s, while there are over 25% sensors (5 dead nodes in 17 nodes) exhaust. We also notice that PST has a better performance than RLC. The simulation results can be explained as follows. Essentially, the main idea of our algorithm is to chase the target node for providing charging service. Therefore, for PST, the increase of the MC's maximum speed shortens the cost of time for chasing the target node in each task, and then PST can achieve more charging tasks in the simulation. On the other hand, RLC chooses to stay at each hotspot for a certain time to wait for the target node's access. As a result, faster speed only shortens the time cost on the travel of the MC and visits more hotspots. Therefore, our algorithm benefits more than RLC when the MC's speed rises.

Fig. 8 shows the performance on the impact of the MC's battery capacity. With the MC's battery capacity increasing, the number of dead nodes decreases and the average residual energy rise in both schemes. This result can be explained as for both PST and RLC, the increase on the MC's capacity makes the MC can provide more charging service during one charging cycle. Still, PST achieves a better performance than RLC with the varying of MC's battery capacity. One reason is that PST can start a new charging task right after completing the current task while RLC has to wait for a certain time at hotspots no matter the charging is completed.

We also vary the energy consumption speed of sensors in our simulation. We carry out simulations where the static average energy consumption rate of each sensor is multiplied by 0.7, 0.8, 0.9, 1.0, 1.1 and set the expectations as the vertical

coordinates of Fig. 9a and Fig. 9b. As the energy consumption speeds of sensors rise, the number of dead nodes rises and the average residual energy decreases in both schemes. These results can be attributed to the shortening of sensors' lifetime. As a result, only one MC has no enough time to serve all the sensors. More detailed, when the energy consumption speeds of sensors are high, PST shows better performance than RLC because PST chooses to chase the target node for charging in the nearest time, as for RLC, waiting at hotspots may cause the target node energy depletion before it visits the hotspots. As the simulations in Fig. 7 and Fig. 8, PST outperforms RLC in all the conditions.

In a word, PST uses the idea of chasing the target node rather than the idea of waiting at hotspots for a certain time to charge the target node to achieve more flexibility and better performance.

VI. CONCLUSION

In this article, we have studied how to charge nodes with non-deterministic mobility with an energy-limited MC. We deduced three major problems for providing charging service in this case, namely NDP, NSP and NEP. We proposed the charging scheme named Predicting-Scheduling-Tracking (PST), which comprises the trajectory prediction algorithm, the MC scheduling algorithm, and the target tracking algorithm, to address these three problems, respectively. Finally, we evaluated the performance of the proposed PST charging scheme against the RLC and some simplified charging schemes through simulations. The proposed PST charging scheme outperforms the others and maintains the WRSN system in a good state. However, due to the trajectory prediction algorithm is based on a neural network approach, the training process for learning the mobility patterns of each sensor node is a heavy load and needs a very long period of time. As a result, the proposed approach may not well work for a network with a large number of nodes. Therefore, we will further improve the trajectory prediction algorithm and extend this work to multiple MCs in the future.

REFERENCES

[1] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, "Network-lifetime maximization of wireless sensor networks," *IEEE Access*, vol. 3, pp. 2191–2226, 2015.

[2] W. Xu, W. Liang, X. Lin, and G. Mao, "Efficient scheduling of multiple mobile chargers for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7670–7683, Sep. 2016.

[3] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *Proc. 6th Int. Conf. Mobile Syst., Appl., Services*, 2008, pp. 29–39.

[4] A. Lavric, V. Popa, and S. Sfichi, "Street lighting control system based on large-scale WSN: A step towards a smart city," in *Proc. Int. Conf. Expo. Electr. Power Eng. (EPE)*, Oct. 2014, pp. 673–676.

[5] A. Naureen, N. Zhang, S. Furber, and Q. Shi, "A GPS-less localization and mobility modelling (LMM) system for wildlife tracking," *IEEE Access*, vol. 8, pp. 102709–102732, 2020.

[6] M. T. R. Khan, S. H. Ahmed, and D. Kim, "AUV-aided energy-efficient clustering in the Internet of underwater things," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 4, pp. 1132–1141, Dec. 2019.

[7] A. H. Dehwah, S. Elmetennani, and C. Claudel, "UD-WCMA: An energy estimation and forecast scheme for solar powered wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 90, pp. 17–25, Jul. 2017.

[8] Z. Fan, Z. Jie, and Q. Yujie, "A survey on wireless power transfer based charging scheduling schemes in wireless rechargeable sensor networks," in *Proc. IEEE 4th Int. Conf. Control Sci. Syst. Eng. (ICCSSE)*, Aug. 2018, pp. 194–198.

[9] C. Wang, J. Li, Y. Yang, and F. Ye, "Combining solar energy harvesting with wireless charging for hybrid wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 560–576, Mar. 2018.

[10] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Soljacic, "Wireless power transfer via strongly coupled magnetic resonances," *Science*, vol. 317, no. 5834, pp. 83–86, Jul. 2007.

[11] L. He, P. Cheng, Y. Gu, J. Pan, T. Zhu, and C. Liu, "Mobile-to-mobile energy replenishment in mission-critical robotic sensor networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr./May 2014, pp. 1195–1203.

[12] C.-F. Cheng and C.-C. Wang, "The energy replenishment problem in mobile WRSNs," in *Proc. IEEE 15th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2018, pp. 143–144.

[13] T. Liu, B. Wu, W. Xu, X. Cao, J. Peng, and H. Wu, "Learning an effective charging scheme for mobile devices," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2020, pp. 202–211.

[14] V. Balasubramanian, F. Zaman, M. Aloqaily, I. A. Ridhawi, Y. Jararweh, and H. B. Salameh, "A mobility management architecture for seamless delivery of 5G-IoT services," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[15] W. Yao, M. Li, and M. Y. Wu, "Inductive charging with multiple charger nodes in wireless sensor networks," in *Proc. Adv. Web Netw. Technol., Appl., Apweb Int. Workshops: Xra, Iwsn, Mega, & Icse*, Harbin, China, Jan. 2006, pp. 262–270.

[16] L. Xie, Y. Shi, Y. T. Hou, W. Lou, and H. D. Sherali, "On traveling path and related problems for a mobile station in a rechargeable sensor network," in *Proc. 14th ACM Int. Symp. Mobile ad hoc Netw. Comput. (MobiHoc)*, 2013, pp. 109–118.

[17] W. Liang, W. Xu, X. Ren, X. Jia, and X. Lin, "Maintaining large-scale rechargeable sensor networks perpetually via multiple mobile charging vehicles," *ACM Trans. Sensor Netw.*, vol. 12, no. 2, pp. 1–26, May 2016.

[18] T. Liu, B. Wu, H. Wu, and J. Peng, "Low-cost collaborative mobile charging for large-scale wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2213–2227, Aug. 2017.

[19] Y. Ma, W. Liang, and W. Xu, "Charging utility maximization in wireless rechargeable sensor networks by charging multiple sensors simultaneously," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1591–1604, Aug. 2018.

[20] P. Guo, X. Liu, M. Chen, and K. Zhang, "Should interference be avoided? Charging WSNs with efficient placement of wireless chargers," *IEEE Access*, vol. 6, pp. 54876–54883, 2018.

[21] A. Tomar, L. Muduli, and P. K. Jana, "An efficient scheduling scheme for on-demand mobile charging in wireless rechargeable sensor networks," *Pervasive Mobile Comput.*, vol. 59, Oct. 2019, Art. no. 101074.

[22] A. Tomar, L. Muduli, and P. K. Jana, "A fuzzy logic-based on-demand charging algorithm for wireless rechargeable sensor networks with multiple chargers," *IEEE Trans. Mobile Comput.*, early access, Apr. 27, 2020, doi: 10.1109/TMC.2020.2990419.

[23] A. Madhja, S. Nikoletseas, and A. A. Voudouris, "Adaptive wireless power transfer in mobile ad hoc networks," *Comput. Netw.*, vol. 152, pp. 87–97, Apr. 2019.

[24] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, Feb. 2010.

[25] L. K. Gallos, C. Song, and H. A. Makse, "A review of fractality and self-similarity in complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 386, no. 2, pp. 686–691, Dec. 2007.

[26] C. Wang, L. Ma, R. Li, T. S. Durrani, and H. Zhang, "Exploring trajectory prediction through machine learning methods," *IEEE Access*, vol. 7, pp. 101441–101452, 2019.

[27] Y. Chen, L. Jiao, Y. Li, and J. Zhao, "Multilayer projective dictionary pair learning and sparse autoencoder for PolSAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 6683–6694, Dec. 2017.

[28] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[29] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong. (2009). *CRAWDAD trace ncsu/mobilitymodels/GPS/NCSU*. [Online]. Available: http://crawdad.cs.dartmouth.edu/ncsu/mobilitymodels/GPS/NCSU

[30] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," *Proc. 9th Int. Conf. ASPLOS*, vol. 28, 2000, pp. 1–12.
[31] M. C. Mukkamala and M. Hein, "Variants of RMSProp and Adagrad with logarithmic regret bounds," presented at the 34th Int. Conf. Mach. Learn., vol. 70. Sydney, NSW, Australia, 2017.

YUANHAO LI received the B.S. degree from Sichuan University, China, in 2018, where he is currently pursuing the M.S. degree in computer science. His current research interests include wireless sensor networks and machine learning.

LINGYUN ZHONG received the B.S. degree in Internet of Things engineering from Southwest Petroleum University, China, in 2019. She is currently pursuing the M.S. degree in computer technology with Sichuan University. Her research interests include wireless sensor networks.

FENG LIN received the B.S., M.S., and Ph.D. degrees in computer science from Sichuan University, China. He is currently an Associate Professor with the College of Computer Science, Sichuan University. His research interests include wireless sensor networks, vehicle networks, information center networks, and distributed systems.