ABSTRACT GPS trajectories are always embedded with errors, due to the weather or environmental variables. Existing trajectory repairing methods have employed Kalman filters or sequential data cleaning methods. Kalman filter or its variants change all observed measurements, while generally most measurements are originally accurate. Sequential data cleaning methods are mainly applied on one-dimensional data sequences, and when encountering multi-dimensional trajectories, their performance will be compromised due to that the features of multi-dimensional trajectories are not fully utilized. To address these issues, we propose to repair GPS trajectory with movement tendencies, speed change tendency, travel distance tendency and repair distance tendency. We formalize the tendency based trajectory repairing, and propose an exact solution to find the repair which minimize movement tendency score. Then we propose high quality candidate selection and dynamic error range estimation, to improve the efficiency and effectiveness of exact solution. Experiments on three data sets demonstrate the superiority of our proposal.

INDEX TERMS Trajectory repairing, movement tendency, error correction, data preprocessing.

I. INTRODUCTION

With the proliferation of mobile devices, massive trajectory data are generated to record GPS positions. With the GPS positions, trajectory data have been widely used by many applications, such as urban planning, intelligent transportation and mobility pattern analytics [21]. One fundamental task to enable these applications is the high quality of GPS positions. Though the accuracy of GPS positions has reached less than five meters in many areas, the GPS positions, if located around at high buildings and inside tunnels, suffer from high spatial errors [3]. When trajectory data contains errors, how to ensure the quality is the main focus of this paper.

In literature, the classic Kalman filter [1] and its variants such as EKF [7] and UKF [14] have been used to clean outliers. However, they could falsely modify the truly correct positions. The map-matching techniques [12], [17] project GPS points to road networks, but are unable to work with the trajectories of free-space moving objects. Recent works such as [15], [19], via meaningful speed constraint, first identify the violations to such constraints, next repair outlier points, and finally generate a cleaned sequence (which hence satisfies the constraints). In particular, the work SDC [19] aims to repair GPS trajectories by maximizing the likelihood of consecutive speed changes within the constraint of a certain total repair distance (a.k.a the budget). Unfortunately, such repair-based approaches suffer from two following issues. 1) Ineffectiveness: It is hard to tune the input parameter budget. Either a greater or smaller budget leads to rather different repair result. We will soon illustrate the issue by an example. 2) Inefficiency: These works simply set a fixed error radius to select repair candidates. Typically, a rather high error radius makes sure that no candidates are falsely missed. However, such a radius could overly select unnecessary candidates and lead to much high computation overhead.

We now give the following example to illustrate the ineffectiveness issue above.

Example 1: Give an observation trajectory \( T = \{p_1, p_2, \ldots, p_7\} \) in Figure 1, three dirty points \( p_2, p_4 \) and \( p_5 \) appear at time point 3, 11 and 14, respectively. We plot the trajectory \( T \) in black and the corresponding ground truth points in yellow. In addition, we give the proposed repair (namely ATR) of \( T \) in blue dashed line, and the
state-of-the-art approach SDC (in green and purple) with two repair budgets, 1 and 3, i.e., SDC-1 and SDC-3.

For the five trajectories above (i.e., the observation, ground truth, SDC-1, SDC-3 and ATR), we compute the overall speed change and travel distance by the following equations.

$$
SC = \sum_{i=2}^{n-1} \left| \frac{\text{Dist}(p_{i-1}, p_i)}{t_i - t_{i-1}} - \frac{\text{Dist}(p_i, p_{i+1})}{t_{i+1} - t_i} \right|
$$

$$
TD = \sum_{i=2}^{n} \text{Dist}(p_{i-1}, p_i)
$$

### TABLE 1. Features of Example 1

|          | Speed change | Travel distance |
|----------|--------------|----------------|
| Observation | 2.3448       | 12.7082        |
| Ground Truth | 0.7866       | 10.4344        |
| SDC-1     | 3.6038       | 12.3005        |
| SDC-3     | 2.8049       | 13.8704        |
| ATR       | 0.9331       | 10.4787        |

From Table 1, we have the following findings. 1) The speed change and travel distance of the ground truth trajectory are the smallest among all five trajectories. It makes sense because mobile users usually tend to move on short paths with trivial speed change. 2) For the observation trajectory, due to the noise and errors, the associated speed change and travel distance are much higher than those of the ground truth trajectory. 3) Given the various budgets 1 and 3, the two SDC approaches lead to different speed change and travel distance. For example, with a smaller budget 1, SDC-1 leads to a smaller travel distance, and vice versa. Thus, the SDC approach essentially involves the trade-off between the travel distance and speed change. How to tune a reasonable budget is non-trivial. 4) Finally, our approach ATR repairs the observation trajectory with the two values much close to the ground truth.

Besides the speed change and travel distance features, the repair of an observation trajectory \( T \) should not significantly modify the observation points, which is consistent with the minimum change principle (MCP) in the data repair community [2], [19]. The repair distance of a repaired trajectory \( T' \) of \( T \) is computed as \( \sum_{i=1}^{\min(n, m)} \text{Dist}(p'_i, p_i) \), \( p'_i \in T' \), \( p_i \in T \). Thus, the repair distance should be as small as possible.

### A. PROPOSAL

To address these issues, we propose an automatic trajectory repairing (ATR) which can be applied on free-space moving object. Unlike the previous works such as SDC requiring the non-trivial effort to tune input parameters such as budgets, we propose to repair observation trajectories in order to minimize speed change, travel distance and repair distance. Here, we define an overall score function to unify the three items, and repair the observation trajectories with the purpose to minimize the overall score.

Next, in terms of the inefficiency issue, we do not set a fixed error radius to select candidates, and instead exploit a well-trained regression model to adaptively tune an error radius for every GPS point. In this way, we can greatly reduce the amount of unnecessary candidates for much higher efficiency.

### B. CONTRIBUTION

To address challenges above, our major contributions in this paper are summarized as follows.

- We design an overall movement score to unify three tendencies including speed change, travel distance and repair distance. When given an observation trajectory, we study the problem of finding a repair trajectory to minimize the overall movement score. Our approach to solve this problem thus can effectively overcome the non-trivial effort of tuning parameters.

- We develop the efficient algorithms to solve the studied problem by a dynamic programming-based pseudo-polynomial algorithm and its improvement, and a regression model to tune an error radius for each GPS point to avoid overly selecting too many candidates, leading to higher computation efficiency.

- We validate our algorithms on three data sets to study the trade-off between repair accuracy and computation efficiency. The results demonstrate that our proposal outperforms both classic filtering methods and state-of-the-art repair approaches.

The rest of the paper is organized as follows. Section II first defines the movement tendencies and repair problem. Section III then gives the exact solution of repair problem and an improvement. Section IV next presents a regression model to tune error radius. After that, Section V reports evaluation result and Section VI reviews related work. Section VII finally concludes this paper.

### II. PROBLEM SETTING

#### A. PROBLEM STATEMENT

Consider a trajectory \( T = \{p_1, p_2, \ldots, p_n\} \) consisting of \( n \) GPS points. For brevity, we denote \( p_{i-j} \) the sub-trajectory \( p_i, p_{i+1}, \ldots, p_j \) with \( 1 \leq i \leq j \leq n \). Each point \( p_i = (x_i, y_i) \) contains the GPS longitude and latitude coordinate. In addition, the point \( p_i \) has a time stamp \( t_i \), and an error radius \( \theta_i \). The error radius means that the distance between \( p_i \) and its true position \( p^*_i \) is not greater than \( \theta_i \), i.e., \( \text{Dist}(p_i, p^*_i) \leq \theta_i \). Though each point \( p_i \in T \) has an
TABLE 2. Used symbols and associated meanings.

| Symbol | Meaning |
|--------|---------|
| $T, p_i$ | Observed trajectory, point $p_i \in T$ with coordinate $(x_i, y_i)$ |
| $t_i, \theta_i$ | Time stamp, error radius of $p_i$ |
| $T', p_i'$ | Repaired trajectory of $T$, repaired point $p_i' \in T''$ |
| $T', p_i''$ | Ground truth trajectory of $T$, ground truth point $p_i'' \in T'$ |
| $C_i$ | Candidate set of $p_i$ |
| $\epsilon$ | Granularity parameter |
| $V$ | Speed change tendency |
| $L$ | Travel distance tendency |
| $D$ | Repair distance tendency |
| $M$ | Movement tendency score |
| $\omega$ | Half window size of neighbors of $p_i$ considered |

individual radius $\theta_i$, the literature works [19] usually set a fixed radius for all points. For example, simply using the maximal radius $\theta_{\text{max}}$ among all points could avoid falsely missed candidates but at the cost of overly selecting too many candidates.

To repair a dirty point $p_i \in T$, at first we need to select the candidates used to repair $p_i$. Such candidates are those points whose distance to $p_i$ is within the error radius $\theta_i$. As shown in Figure 2, we intuitively consider that those candidates are within a circle with $p_i$ as the center and $\theta_i$ as the radius. To generate a finite candidate set $C_i$ for $p_i$, we use a granularity parameter $\epsilon$ and divide the circle into grid cells, where the cell width is $\epsilon$. In this way, we take the lattice points within the circle as the candidate set $C_i$. The formal definition of candidate set $C_i$ of $p_i$ is as follows.

$$C_i = \{(x_i + ae, y_i + be) | a, b \in \mathbb{Z}, \epsilon > 0, \sqrt{a^2 + b^2} \cdot \epsilon \leq \theta_i\}$$

Example 2: Let us consider the candidates of point $p_2$ in Figure 2. Given $\epsilon = 1$ and $\theta_2 = 2$, the candidate set $C_2$ contains the lattice points (blue circles) within the circle, i.e., $C_2 = \{p_2', p_2', \ldots, p_2', \ldots, p_2', 13\}$. Here, the point $p_2$, also denoted as $p_2'$, is within the set $C_2$.

From the previous work [9], we have the following lemma to give the size of a candidate set $C_i$ as follows.

**Lemma 1:** The size of a candidate set $C_i$ is $|C_i| = \pi \left(\frac{\theta_i}{\epsilon}\right)^2 + E\left(\frac{\theta_i}{\epsilon}\right)$.

In the lemma above, the term $E\left(\frac{\theta_i}{\epsilon}\right)$ is a relatively small absolute value, and we have $|E\left(\frac{\theta_i}{\epsilon}\right)| \leq 2\sqrt{\pi} \frac{\theta_i}{\epsilon}$. Thus, the size of a candidate set can be approximated by the area of the circle, i.e., $\pi \left(\frac{\theta_i}{\epsilon}\right)^2$.

Now, for each point $p_i \in T$ and one candidate $p_i' \in C_i$ with $1 \leq i \leq n$, we define a movement tendency score $\mathcal{M}_i$ to measure the goodness of this candidate $p_i'$ to repair $p_i$.

$$\mathcal{M}_i = \mathcal{M}(p_{i-1}', p_i', p_{i+1}') = D(p_i') + L(p_{i-1}', p_i') + V(p_{i-1}', p_i', p_{i+1}')$$

(1)

In the equation above, we introduce the three neighbouring candidates $p_{i-1}', p_i'$ and $p_{i+1}'$ to compute the score $\mathcal{M}_i$. In this way, we have chance to compute three tendency scores with respect to the repair distance (D), travel distance (L) and speed change (V). The detail of the three tendency scores will be soon given in Section II-B.

Until now, for each point $p_i$ within an observation trajectory $T$, we have an associated candidate set $C_i$. Given the candidate sets for all points within $T$, we define the following problem to repair $T$.

**Problem 1:** Given an observation trajectory $T$ consisting of $n$ points $p_i$ with $1 \leq i \leq n$, we need to find a repaired trajectory $T'$ with $n$ candidate points $p_i'$ corresponding to the points $p_i \in T$, such that the overall movement tendency score $\mathcal{M}(T')$ is minimized.

$$\min \sum_{i=1}^n \mathcal{M}_i = \sum_{i=1}^n \mathcal{M}(p_{i-1}', p_i', p_{i+1}')$$

(2)

s.t. $p_{i-1}' \in C_{i-1}, p_i' \in C_i, p_{i+1}' \in C_{i+1}$

B. COMPUTATION OF MOVEMENT TENDENCIES

In this subsection, we give the detail of three movement tendencies w.r.t repair distance (D), travel distance (L) and speed change (V). After that, in the next subsection, we normalize these tendencies.

1) REPAIR DISTANCE TENDENCY

Given an observation point $p_i$ and its candidate $p_i'$, we compute the associated repair distance between $p_i$ and $p_i'$, i.e., $D(p_i, p_i')$. For an observation trajectory $T$ and a candidate repairing trajectory $T'$, we then compute the overall repair distance by $Dist(T, T') = \sum_{i=1}^n Dist(p_i, p_i')$. Following the minimum change principle (MCP) [2], we expect that the overall repair distance $Dist(T, T')$ should be small and otherwise the repaired trajectory $T'$ could distort the observation one $T$. The rationale behind MCP is as follows. Though observation data may contain noise, such observation more or less indicates the true data. Thus, the MCP is meaningful to minimize the overall repair distance $Dist(T, T')$.

2) TRAVEL DISTANCE TENDENCY

Differing from the repair distance above, we now consider the candidates of a point $p_i$ and its predecessor $p_{i-1}$, denoted by $p_i'$ and $p_{i-1}'$, and then compute the travel distance $Dist(p_{i-1}', p_i')$. After that, given the trajectories $T$ and $T'$, we compute the overall travel distance $Dist(T') = \sum_{i=2}^n Dist(p_i', p_{i+1}')$. We expect that the selected candidates $p_i'$ and $p_{i-1}'$ should lead to a small travel distance $Dist(p_{i-1}', p_i')$ and thus the overall travel distance $Dist(T')$. It makes sense because people usually would like to choose short travel paths for energy and
time saving. Note that we could compute the travel distance by point \( p'_i \) and its successor point \( p'_{i+1} \), instead of its predecessor \( p'_{i-1} \), but without degrading the proposed approach.

3) SPEED CHANGE TENDENCY

Let us consider the point \( p_i \), its predecessor \( p_{i-1} \) and successor \( p_{i+1} \), and the corresponding candidates \( p'_i \), \( p'_{i-1} \) and \( p'_{i+1} \). To compute the speed change of \( p_i \), first denote the speed of point \( p'_i \) by \( v(p'_{i-1}, p'_{i+1}) = \text{Dist}(p'_{i-1}, p'_i) \), and then define the speed change of point \( p'_i \) by \( u_i = u(p'_{i-1}, p'_i, p'_{i+1}) = |v(p'_{i-1}, p'_i) - v(p'_{i-1}, p'_{i+1})| \). Finally, the overall speed change of the candidate trajectory \( T' \) is computed by \( U(T') = \sum_{i=2}^{n-1} u(p'_{i-1}, p'_i, p'_{i+1}) \). It is not hard to find that the change of speeds in neighbouring points should not be significant in order to make sure safe driving of moving objects.

We give the following example to compute the three tendencies.

Example 3: In this example, we use the point \( p_2 \) in Figure 2 to show how the three movement tendencies above can help trajectory repair. For simplicity, we assume the error radii \( \theta_1 = \theta_0 = 0 \) and then no repair is needed for the points \( p_1 \) and \( p_3 \). Thus, we only need to consider the repair of \( p_2 \). Given the candidate set \( C_2 = \{p'_{2,1}, p'_{2,2}, \ldots, p'_{2,13}\} \) (see Example 2). Among the candidates in \( C_2 \), we want to find the best one to repair \( p_2 \). To this end, we take into account three movement tendencies as follows.

Firstly, depending upon the repair distance tendency, we have a sorted list of candidates by the ascending order of the repair distance: \( \{p_{2,7} \rightarrow p_{2,3} \rightarrow p_{2,6} \rightarrow p_{2,8} \rightarrow p'_{2,11} \rightarrow \ldots \rightarrow p'_{2,1}\} \). In terms of the repair distance, the candidate \( p'_{2,7} \) is the best option due to the repair distance \( \text{Dist}(p'_{2,7}, p_2) = 0 \). Secondly, if we compute the travel distance, we similarly have a sorted list of candidates by ascending order: \( p'_{2,11} \rightarrow p'_{2,12} \rightarrow p'_{2,8} \rightarrow p'_{2,13} \rightarrow \ldots \rightarrow p'_{2,1} \). Finally, the sorted list of candidates in ascending order of the speed change is \( p_{2,10} \rightarrow p_{2,13} \rightarrow p_{2,11} \rightarrow \ldots \rightarrow p_{2,9} \). If we take into account the three sorted lists together, we compute the overall movement tendency score \( M \) of each candidate and the candidate \( p'_{2,11} \) is selected as the best one to repair \( p_2 \).

C. NORMALIZATION OF MOVEMENT TENDENCIES

In this section, we give the detail to normalize the three tendencies mentioned above. Since the normalized travel and repair distance requires the subitems that appear within the normalized speed change, we thus first introduce the normalized speed change, then the normalized travel distance and finally the normalized repair distance.

Firstly, the speed change tendency of point \( p_i \) depends upon \( p_i \)'s predecessor \( p_{i-1} \) and successor \( p_{i+1} \). We thus leverage the candidates of \( p_{i-1} \) and \( p_{i+1} \) together to normalize the speed change tendency.

\[
V(p'_{i-1}, p'_i, p'_{i+1}) = \frac{\sum_{p'_{i-1} \in C_{i-1}} \sum_{p'_i \in C_i} \sum_{p'_{i+1} \in C_{i+1}} u(p'_{i-1}, p'_i, p'_{i+1})}{\sum_{p'_{i-1} \in C_{i-1}} \sum_{p'_i \in C_i} \sum_{p'_{i+1} \in C_{i+1}} \text{Dist}(p'_i, p'_{i+1})}, \quad 1 \leq i \leq n
\]  

(3)

Noted that \( p_1 \) has no predecessor and \( p_n \) has no successor. We thus add a head \( p_0 = p_1 \) and a tail \( p_{n+1} = p_n \) to trajectory \( T \), and meanwhile set their time stamps and error radii by \( t_0 = t_1 = \ldots = t_{n+1} = t_0 = 0 \). Thus, we have the candidate sets \( C_0 = \{p_0\} \) and \( C_{n+1} = \{p_{n+1}\} \).

Secondly, since the travel distance tendency only takes into account the point \( p_i \) and its predecessor \( p_{i-1} \), we have the normalized tendency as below.

\[
L(p'_{i-1}, p'_i) = \frac{1}{|C_{i+1}|} \sum_{p'_{i-1} \in C_{i-1}} \sum_{p'_i \in C_i} \sum_{p'_{i+1} \in C_{i+1}} \text{Dist}(p'_i, p'_{i+1})
\]  

(4)

In the equation above, the subitem \( \sum_{p'_i \in C_i} \text{Dist}(p'_i, p'_{i+1}) \), consistent with one subitem in Equation 3, makes sure that the tendencies in the two equations above are within the same scale.

Finally, in terms of repair distance tendency, we again add two subitems with respect to the amount of the candidates in \( p_{i-1} \) and \( p_{i+1} \) into the following equation.

\[
D(p'_i) = \frac{1}{|C_{i-1}||C_{i+1}|} \sum_{p'_{i-1} \in C_{i-1}} \sum_{p'_i \in C_i} \text{Dist}(p'_i, p'_{i+1})
\]  

(5)

III. EXACT SOLUTION

In this section, we first give an exact solution to solve Problem 1 by a pseudo-polynomial algorithm. This algorithm leverages the dynamic programming technique to repair the observation trajectory \( T \) by a candidate one \( T' \) with the purpose to minimize the overall movement tendency score. Next, to improve the repair efficiency, we propose to reduce the amount of used candidates.

A. DYNAMIC PROGRAMMING

We show the basic idea of this exact algorithm as follows. Since the speed change requires a triple of neighbouring candidates \( \{p'_{i-1}, p'_i, p'_{i+1}\} \), we consider all possible \( p'_{i-2} \), \( p'_{i-1} \), \( p'_{i} \) positions in each recurrence during the dynamic programming-based steps. Next, let us consider a repair of sub-trajectory \( p'_{i-1} \), we denote \( F(i, p'_{i-1}, p'_i) \) as the minimum overall movement tendency score \( M(p'_{i-1}, i) \). Since the successor candidate \( p'_{i+1} \) is still unavailable, we have to obtain \( M(p'_{i-1}, i) \) from \( p'_{i-1} \). Initially, for \( i = 1 \), we have \( F(1, p'_{0}, p'_1) = 0 \), \( p'_0 \in C_0, p'_1 \in C_1 \). Next, for each \( i \in \{2, \ldots, n+1\} \), we compute \( F(i, p'_{i-1}, p'_i) \) by the following recurrence.

\[
F(i, p'_{i-1}, p'_i) = \operatorname{min}_{p'_{i-2} \in C_{i-2}} F(i - 1, p'_{i-2}, p'_{i-1}) + M(p'_{i-2}, p'_{i-1}, p'_i)
\]  

(6)

Algorithm 1 gives the Pseudocode of the dynamic programming-based approach. By this algorithm, we could find \( F(n + 1, p'_0, p'_{n+1}) \) to be the minimum score, among all possible candidates \( p'_n \) and \( p'_{n+1} \). By retracting all
Algorithm 1 DP($T$, $\theta$, $\epsilon$)

Input: trajectory $T$, error radius $\theta$, granularity $\epsilon$
Output: the minimum score of the optimal repair

1. add head $p_0$ and tail $p_{n+1}$ to $T$;
2. construct $C_i$ ($0 \leq i \leq n + 1$) with each $p_0, p_1, \ldots, p_{n+1}$ with $\theta_i$ and $\epsilon$;
3. initialize $F(1, p'_0, p'_1) = 0$ for each $p'_0, p'_1$;
4. $\text{trac}e \leftarrow \emptyset$;
5. for $i \leftarrow 2$ to $n + 1$ do
6. for $p'_i \in C_i$ do
7. for $p'_{i-1} \in C_{i-1}$ do
8. $F(i, p'_{i-1}, p'_i) \leftarrow +\infty$;
9. for $p'_{i-2} \in C_{i-2}$ do
10. $\ell \leftarrow +\Delta(p'_{i-2}, p'_{i-1}, p'_i)$;
11. if $F(i - 1, p'_{i-2}, p'_{i-1}) + \ell < F(i, p'_{i-1}, p'_i)$ then
12. $F(i, p'_{i-1}, p'_i) \leftarrow F(i - 1, p'_{i-2}, p'_{i-1}) + \ell$;
13. $\text{trace}(i, p'_{i-1}, p'_i) \leftarrow p'_{i-2}$;
14. choose $p'_n \in C_n$, $p'_{n+1} \in C_{n+1}$ with minimum $F(n+1, p'_n, p'_{n+1})$;
15. $T' \leftarrow$ reverse candidates from $p'_{n+1}, p'_n$ back to $p'_1$ with $\text{trace}$;
16. return $T'$

$F(i, p'_{i-1}, p'_i)$ leading to this minimum score, we then find an exact solution $T'$.

Example 4 (Recurrence Computation): We consider again the trajectory in Example 1 with the granularity parameter $\epsilon = 1$. We perform the recurrence computation on $F(i, p'_{i-1}, p'_i)$ in Equation (6) from $i = 2$ to $i = n + 1 = 8$. The final state at $i = 8$ is $F(8, p'_{7,1}, p'_{6,1}) = 3.47$ which achieves the minimum movement tendency score. By retracing $F(i, p'_{i-1}, p'_i)$ leading to $F(8, p'_{7,1}, p'_{6,1})$, we obtain an exact solution $T' = \{p'_{1,1}, p'_{2,11}, p'_{3,11}, p'_{4,11}, p'_{5,11}, p'_{6,1}, p'_{7,1}\}$. It is exactly the solution $T'$ under $\epsilon = 1$ in Example 1, with the minimum tendency score $M(p'_{1,7}) = 3.47$.

Complexity: Algorithm 1 computes the exact solution in a pseudo-polynomial time $O(n|C_{\text{max}}|^3)$ with $O(2^n|C_{\text{max}}|^2)$ space (including states of $F$ and $\text{trace}$), where $|C_{\text{max}}|$ is the maximum size of candidate sets, i.e., $|C_{\text{max}}| = \max_{i=1}^{n}|C_i|$.

B. IMPROVEMENT

The time complexity of the exact solution (Algorithm 1) is determined by the size of candidate sets $|C_i|$. For example in Figure 2, with the granularity parameter $\epsilon = 0.2$ and error radius $\theta_2 = 2$, we have the amount $|C_2| = 317$ of candidates for point $p_2$. Moreover, the computation of movement tendency score depends upon the combinations of such candidates, which could degrade the repair efficiency. In this section, we propose a heuristic technique to select partial candidates from the candida set $C_i$ for more efficient repair.

We give the basic idea as follows. Among all candidates in the set $C_i$, some of them are much more qualified to correct a point $p_i$ than the others. We thus focus on the highly qualified candidates and then prune the remaining ones. To find such candidates $p'_i$, we expect that the overall tendency score of $p'_i$ should not be higher (a.k.a worse) than the overall score of the observation $p_i$. Thus, among the three tendency scores (w.r.t $D, L,$ and $V$) of $p'_i$, we consider those candidates $p'_i$ which should lead to at least one of three tendency scores smaller (a.k.a better) than the corresponding score of $p_i$. Based on the intuition, we estimate the quality scores of a candidate $p'_{i,j}$ of $p_i$ as below.

$$Q_D(p'_{i,j}) = \frac{\epsilon^{\text{Dist}(p_i, p'_{j})}}{\sum_{k \in C_i} \epsilon^{\text{Dist}(p_i, p'_k)}}$$

$$Q_L(p'_{i,j}) = \frac{\sum_{k \in C_i} \epsilon^{\text{Dist}(p_i, p'_k)}}{\sum_{k \in C_i} \epsilon^{\text{Dist}(p_i, p'_k)}}$$

$$Q_V(p'_{i,j}) = \frac{\epsilon^{\text{Dist}(p_i, p'_{i,j})+1}}{\sum_{k \in C_i} \epsilon^{\text{Dist}(p_i, p'_{i,j})+1}}}$$

Thus, by referring to the scores of $p_i$, we formally select the high quality candidates $H_i$ from $C_i$ as follows.

$$H_i = \bigcup_{K \in \{D, L, V\}} \{p'_{i,j} \in C_i | Q_k(p'_{i,j}) \leq Q_k(p_i)\} \quad (7)$$

Example 5 Still in Figure 1, for the point $p_2$, we compute the set $H_2 = \{p'_{2,7}\} \cup \{p'_{2,5}, p'_{2,6}, p'_{2,7}, p'_{2,10}, p'_{2,11}, p'_{2,12}, p'_{2,13}\} \cup \{p'_{2,5}, p'_{2,6}, p'_{2,7}, p'_{2,10}, p'_{2,11}, p'_{2,13}\} = \{p'_{2,5}, p'_{2,6}, p'_{2,7}, p'_{2,10}, p'_{2,11}, p'_{2,13}\}$. For the point $p_4$, we have
one is fixed to be $p_i$. As a result, given the half window size $\omega$, the total amount of triples is equal to $\omega^2$. In our experiment, we set $\omega = 3$ by default and thus typically select 9 triples for each point $p_i$.

In terms of a certain triple $\langle p_j, p_i, p_k \rangle$, we extract totally 6 movement features. The first three features include the distance, speed and turning angle. As shown in Figure 4(a), the point $p_i = (x_i, y_i)$ at timestamp $t_i$ has a predecessor $p_j$ and a successor $p_k$ ($j < i < k$). We compute the distance, speed and turning angle of $p_i$ by $f_L^{j,i} = \text{Dist}(p_j, p_i), f_V^{j,i} = \frac{\text{Dist}(p_i, p_k)}{t_k - t_j}$ and $f_A^{i,j,k} = \pi - \angle p_jp_ip_k$, respectively.

Secondly, we develop the next three movement features that are consistent with the aforementioned movement tendencies: speed change $f_L^{j,i,k}$, distance difference $f_D^{j,i,k}$ and repair distance $f_D^{j,i,k}$. As shown in Figure 4(b), we compute the speed change by $f_L^{j,i,k} = |f_V^{j,i} - f_V^{i,k}|$. In terms of distance difference, we note that moving objects are likely to move on short paths. Given the triple $\langle p_j, p_i, p_k \rangle$, we then compute the distance difference between the path $p_j \rightarrow p_i \rightarrow p_k$ and the one $p_j \rightarrow p_k$ by $f_D^{j,i,k} = |\text{Dist}(p_j, p_i) + \text{Dist}(p_i, p_k) - \text{Dist}(p_j, p_k)|$. As for repair distance $f_D^{j,i,k}$, we use $p_j$ and $p_k$ to infer the repair position $p'_i$ of $p_i$ by linear interpolation on $p_j \rightarrow p_k$. The inferred position $p'_i$ of $p_i$ is denoted as $p'_i = (x_j + \frac{(t_k - t_j)(x_k - x_j)}{t_k - t_j}, y_j + \frac{(t_k - t_j)(y_k - y_j)}{t_k - t_j})$. We heuristically use $p'_i$ as the repair of $p_i$ which locates on the short path with no speed change, which results in both minimum speed change tendency and repair path tendency. Thus the repair distance is denoted as $f_D^{j,i,k} = \text{Dist}(p_i, p'_i)$.

We note that the regression model could estimate an error radius that is smaller than the ground truth and the best repair candidate may be falsely missed. To tackle this issue, we follow the previous work [16] to estimate the variance of bagged predictors and random forests predictors based on the jackknife [4] and the infinitesimal jackknife (IJ) [5]. [16] proves that jackknife rule has an upward bias, while the IJ rule has a downward bias. By taking the arithmetic mean of the two variance estimations, it merges these two rules to achieve nearly unbiased variance estimation. In this way, we compute the error bars of the predicted radii. Then, in order to alleviate the candidate missing problem, we add the error bar on each predicted radius.

V. EXPERIMENT

A. COUNTERPART

In this section, we evaluate the performance of our proposed approaches including 1) DP: the dynamic
programming-based algorithm in Section III-A, 2) DP + H: the improved DP approach by selecting high quality candidates, and 3) DP + HRv: the enhancement of DP + H by a regression model to tune the error radius of each point with variance estimation. We compare these approaches with the state-of-the-art SDC [19] and classical Kalman filter (KF) [1] with its two variants, extended Kalman filter (EKF) [7] and unscented Kalman filter (UKF) [14]. Noted that SDC requires a repair budget parameter, we thus increment the repair budget until the likelihood of SDC remains stable. We do not repeat the evaluation of other approaches such as the smoothing-based EWMA, or the constraint-based method. It is mainly because such approaches suffer from worse performance than SDC (see [15], [19]).

B. DATASETS

We use three trajectory datasets.

- Ts: We collect a GPS trajectory by smartphones when mobile users are walking on the road network of a university campus, by the sampling rate of one GPS position per second. The ground truth positions of the trajectory are marked manually as follows. We first project the collected trajectory onto the digital map of the campus, visualize the projected trajectory, and carefully label each GPS point. In this way, we can manually identify 235 points with high spatial errors among the total 2710 collected GPS points.

- Tt: We use a real dataset provided by one of the largest telecommunication (Telco) in China. This dataset contains 5367 measurement report (MR) samples and corresponding GPS positions. Each MR sample measures the signal strength between mobile devices and nearby up-to-seven cellular towers. We apply a Telco localization algorithm [20] to recover the trajectories of outdoor positions of mobile devices from the MR samples, and the corresponding GPS positions of such mobile devices are then as the ground truth of the recovered trajectories. In terms the localization error, computed by the distance between a recovered position and ground truth, we can achieve the medium error 20.4 meters.

- Tr: We randomly generate synthetic noise (including various noise radius and noise rate) on each GPS position within a real GPS trajectory with 18610 GPS points, and the original trajectories can be treated as ground truth. We use two parameters, i.e., noise rate $N_r$ and noise radius $N_n$, to generate the noise points. Here, the noise rate $N_r$ indicates the percentage of GPS points that are selected to generate random noise. Given a selected point, we randomly generate a noise point such that the distance between the noise point and original one is smaller than the noise radius $N_n$. For example, by setting $N_r = 20\%$ and $N_n = 40$ meters, we first randomly select 20% points from the trajectory points, and then generate the noise points with the distance within 40 meters to the selected points. In this way, depending upon the parameters $N_r$ and $N_n$, we generate various synthetic trajectories.

C. METRICS

Besides the running time of repair algorithms, we also evaluate the repair effectiveness by RMS errors [10]. Let $T$ be an input observation trajectory, $T'$ be the ground truth of $T$, and $T''$ be the repaired trajectory. We then compute the RMS error by:

$$\Delta(T', T'') = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \text{Dist}(p'_i, p'_T)^2}, \quad p'_i \in T', \ p'_T \in T''$$

D. PERFORMANCE ON TWO REAL TRAJECTORIES

We evaluate all repair approaches on two real trajectories Ts and Tt. For Ts, we set $\theta_{max} = 16$ meters and $\epsilon = 4$ meters, and for Tt, we set $\theta_{max} = 80$ meters and $\epsilon = 20$ meters. In this way, we have 40 candidates for each point in both datasets of Ts and Tt.

| | Smartphone Trajectory Ts | | Teleco Trajectory Tt | |
|---|---|---|---|
| | RMS error | Time cost (s) | RMS error | Time cost (s) |
| Observe | 1.82 | 90.11 |
| KF | 4.89 | 1.32 | 115.83 | 3.2 |
| EKF | 3.29 | 1.40 | 107.46 | 3.5 |
| UKF | 2.80 | 1.51 | 103.25 | 4.1 |
| SDC | 1.79 | 1383.65 | 87.60 | 2513.3 |
| DP | 1.58 | 4079.12 | 86.81 | 7430.6 |
| DP+H | 1.41 | 33.38 | 85.57 | 80.6 |
| DP+HRv | 1.12 | 27.10 | 82.46 | 54.7 |  

From Table 4, we find that the RMS errors of two observation trajectories Ts and Tt (denoted as Observe) are 1.82 and 90.11, respectively. In terms of all listed repair approaches, we have the following findings.

First of all, the RMS errors of KF and its two variants are rather high. It is mainly because these approach could modify all points, no matter these points are errors or not. As online approaches, these KF-based methods are fast, leading to the least running time. Among the three approaches, KF simply adopts the linear model, and yet EKF and UKF non-linear models, leading to lower RMS errors than KF.

Secondly, the state-of-the-art approach SDC slightly repairs the input trajectories. Specifically, 1) SDC transforms the 2-dimensional GPS trajectory $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ into two 1-dimensional sequences $\{x_1, x_2, \ldots, x_n\}$ and $\{y_1, y_2, \ldots, y_n\}$. The two 1-dimensional sequences are then be repaired separately. 2) Among the three movement tendencies studied in this paper, SDC only adopts the speed change tendency. As a result, it is not hard to find that SDC suffers from higher RMS errors than all variants of our work. Moreover, the running time of SDC is much slower than the three variants of our work, except the baseline approach DP.

Thirdly, among the three variants, DP + HRv achieves the lowest RMS error, and DP + HRv leads to the fastest
running time. The high effectiveness and efficiency of DP + HRv are mainly because it adaptively tune the error radius for each point and greatly save the overhead to process a smaller amount of candidates. In addition, when compared with the baseline approach DP, the improvement DP + H significantly reduce the amount of candidates and leads to much faster running time.

Finally, for the two datasets Ts and Tt, all repair approaches lead to higher RMS errors and slower running time to process Tt than those to Ts. It is mainly because the Telco trajectory Tt suffers from much higher medium errors and more data samples.

As a summary, the filter-based approaches including KF and its two variants suffer from the highest RMS errors but with the fastest running time. The state-of-the-art approach SDC, though leads to much lower RMS errors than the filter-based approaches, but at the cost of significantly higher running time. Our approaches involve the tradeoff between RMS errors and running time, and lead to much lower RMS errors than the filter-based approaches and faster running time than SDC.

In the rest of this section, since our goal is to repair input trajectories with errors, we mainly focus on the comparison between our approaches and SDC.

E. PERFORMANCE ON THE SYNTHETIC TRAJECTORY

In this section, by varying the parameters to generate the synthetic trajectory Tr, we measure the performance of SDC and three of our approaches DP, DP + H and DP + HRv.

Figure 5 reports the performance of all methods on the dataset Tr. First, in Figure 5(a-b), the growth of the noise ratio \( N_r \) and noise radius \( N_\alpha \) leads to greater RMS errors for all approaches. In terms of a certain value of \( N_r \) and \( N_\alpha \), our approaches consistently outperform SDC.

In Figure 5(c-d), among the four approaches, the running time of DP and SDC remains relatively stable. It is mainly because \( N_r \) and \( N_\alpha \) have trivial effect on the time complexity in these methods. Instead, both DP + H and both DP + HRv are rather sensitive to \( N_r \) and \( N_\alpha \) because they need to prune more candidates when given greater \( N_r \) and \( N_\alpha \).

F. EFFECT OF MOVEMENT TENDENCIES

Since our approaches adopt three movement tendencies including the repair distance (D), travel distance (L) and speed change tendency (V), in Figure 6(a), we study the performance of various combinations of three tendencies. Though we use DP + HRv for illustration in this figure, the variants such as DP and DP + H are also applicable.

Among the three combinations using the individual tendency, we find that the travel distance tendency (L) leads to the least RMS error and the repair distance tendency (D) the greatest error. For the three combinations of two tendencies, we again find that the travel distance tendency (L) consistently leads to the major contribution. Finally, we see that three tendencies DLV leads to the best performance.

G. EFFECT OF REGRESSION MODEL

To evaluate the performance of the regression model to tune error radii, we implement various regression models on the smartphone trajectory Ts. In Figure 6(b), we vary the parameter \( \omega \) (a.k.a the half window size) from 1 to 8. Among the five regression models, the Random Forest (RF) leads to the least RMS errors on \( \omega = 3 \). Instead, comparatively smaller \( \omega \) could miss the context information of neighbouring points, and
comparatively greater $\omega$ adds more irrelevant features into the regression model. Thus, either those greater or smaller values of $\omega$ degrades the performance. In addition, we noted that the RMS error of SVM grows with a greater $\omega$. This is mainly because a greater $\omega$ leads to more features and it is harder for SVM to separate the entire feature space by a hyperplane.

In addition, as shown in Figure 6(c), we report the importance of the features used by the Random Forest (RF)-based regression model. Recall that each triple of three points $(p_{i-j}, p_i, p_{i+k})$ with $1 \leq j, k \leq \omega$ leads to 6 features which are then used by the regression model. In this figure, the $x$-axis represents the pairs $(j, k)$. For example, the pair $(2, 3)$ represents the triple $(p_{i-2}, p_i, p_{i+3})$ illustrated in Section IV.

From Figure 6(c), we find that 1) the features with respect to the pairs $(1, 1), (2, 1)$ and $(3, 1)$ are more important than others, 2) among the 6 features in each pair, the three features $f_{AL}, f_{AV}, f_{AP}$ are more important than others (i.e., the three features $f_L, f_V$ and $f_A$), and 3) the points far away from $p_i$, e.g., the pair $(3, 3)$, have trivial affect on the error radius estimation of $p_i$.

**H. EFFECT OF TWO PARAMETERS $\varepsilon$ AND $\theta_{\text{max}}$**

Finally, we study the effects of two parameters $\varepsilon$ and $\theta_{\text{max}}$ on the smartphone trajectory $T_s$. As shown in Figure 7(a), when the granularity parameter $\varepsilon$ becomes smaller, we have more candidates (See Figure 2) and smaller RMS errors for all three variants of our approach. Again in Figure 7(b), a greater $\theta_{\text{max}}$ indicates more selected candidates and also leads to smaller RMS errors. As shown in Figure 7(c-d), more candidates, caused by a smaller $\varepsilon$ and greater $\theta_{\text{max}}$, lead to higher running time.

**VI. RELATED WORK**

**A. FILTER-BASED REPAIR**

Kalman filter (KF) [11] has been widely used in navigation and control of vehicles. It repairs the sequence of measurement data with noise and other errors. KF works in a two-stage process: prediction and update. In the prediction stage, KF generates the estimations of the current state variables, along with previous updated state estimate and the uncertainty of current state. These estimations are updated by using a weighted average when the next measurement is observed. These two stages are recursively computed along with the sequentially generated observation in real time. In this way, KF gives the optimal linear estimate for linear system models.

Unfortunately, most systems are nonlinear and two variants of KF are proposed to address the non-linearity issue, extended Kalman filter EKF [7] and unscented Kalman filter UKF [14]. The difference between EKF and UKF is how they handle non-linear equations. EKF takes the Jacobian matrix [8] to linearly approximate non-linear functions. UKF takes a bunch of representative points from a Gaussian distribution to construct a better approximation of non-linear functions.

Comparing with our proposed approaches, KF based methods modify every observed measurements and the estimations are computed by only previews estimation and uncertainty. Our methods have the chance to remain the accurate observed measurement. Moreover, our solution DP globally considers the entire trajectory to repair the observed measurements. Our proposal thus can obtain more accurate repair results than KF-based methods.

**B. DATA REPAIR**

The two previous works [6], [15] repaired the sequential data by solving a constraint-based optimization problem. Sequential dependencies [6] concerns the range of value changes between two consecutive data points. Taking advantage of the (widely existing) associated timestamps, [15] states that the repaired sequence should satisfy the constraint on the speed of data changes. For the distances on timestamps, sequential dependencies can be also interpreted as a special case of speed constraints with fixed time intervals.

Both of the above methods can only detect and repair violations to the specified constraints and do not consider any likelihood aspects among data points. As reported in SDC [19], the statistical-based repair could outperform constraint-based methods. It is particularly meaningful for the case of some small scale errors, which is common in GPS trajectories studied in the paper.

**C. STATISTICAL-BASED REPAIR**

The statistical-based repair techniques have been applied on relational data [11], [13], [18] and sequential data [19]. [18] identifies the attributes with dirty values as flexible attributes and the attributes with correct values as reliable attributes. The relations between reliable and flexible attributes are then modeled and it can be applied to repair the value with flexible attributes. After the repair, the data replacement with the most likelihood w.r.t. the data distribution in the relation is applied.
VII. CONCLUSION

In this paper, we propose to repair GPS trajectory with the objective to minimize the overall movement tendency score which involves the three specific movement tendencies (i.e., speed change, travel distance and repair distance). Based on the candidates of each GPS point, we propose a dynamic programming (DP)-based algorithm to obtain an exact solution with pseudo-polynomial cubic-time complexity. Next, by selecting a small amount of high quality candidates, we improve the efficiency of the DP algorithm. After that, with help of a regression model, we adaptively estimate the error radius of each point to further reduce the amount of candidates per node. Experimental studies on three trajectory data sets validate the effectiveness and efficiency of our algorithms.

REFERENCES

[1] G. Bishop et al., “An introduction to the Kalman filter,” in Proc SIGGRAPH, vol. 8, 2001, p. 41.
[2] P. Bohannon, W. Fan, M. Flaster, and R. Rastogi, “A cost-based model and effective heuristic for repairing constraints by value modification,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2005, pp. 143–154.
[3] J. Dai, F. He, W.-C. Lee, G. Chen, and B. C. Ooi, “DTNC: A new server-side data cleansing framework for cellular trajectory services,” 2017, arXiv:1703.00123. [Online]. Available: https://arxiv.org/abs/1703.00123
[4] B. Efron, “Jackknife-after-bootstrap standard errors and influence functions,” J. Roy. Stat. Soc., B (Methodol.), vol. 54, no. 1, pp. 83–111, 1992.
[5] Model Selection Estimation Bootstrap Smoothing, Division Biostat, Stanford Univ., Stanford, CA, USA, 2012.
[6] L. Golub, H. J. Karlloff, F. Korn, A. Saha, and D. Srivastava, “Sequential dependencies,” VLDB, vol. 2, no. 1, pp. 574–585, 2009. [Online]. Available: http://vldb.org/pvldb/vol2/vldb09-149.pdf
[7] N. Gordon and M. Pitt, “A comparison of sample based filters and the extended kalman filter for the bearings-only tracking problem,” in 9th Eur. Signal Process. Conf., (EUSIPCO), Island of Rhodes, Greece, Sep. 1998, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/document/7089645/
[8] J. S. Gradshteyn and I. M. Ryzhik, Table of Integrals, Series, and Products. New York, NY, USA: Academic, 2014.
[9] G. Hardy and S. Ramanujan, Twelve lectures subjects suggested by his life work. Cambridge, U.K.: Cambridge Univ. Press, 1940.
[10] P. Zhao et al., “ATR: Automatic Trajectory Repairing With Movement Tendencies,” IEEE Access, vol. 8, 2020.
[11] C. Mayfield, J. Neville, and S. Prabhakar, “ERACER: A database approach for statistical inference and data cleaning,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, (SIGMOD), Indianapolis, Indiana, USA, Jun. 2010, pp. 75–86. [Online]. Available: https://doi.org/10.1145/1807167.1807178
[12] P. Newson and J. Krumm, “Hidden Markov map matching through noise and sparseness,” in Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geographic, Inf. Syst., 2009, pp. 336–343.
[13] T. Rekatsinas, X. Chu, I. F. Ilyas, and C. Ré, “Holoclean: Holistic data repairs with probabilistic inference,” Proc. VLDB Endowment, vol. 10, no. 11, pp. 1190–1201, 2017.
[14] D. Simon, Optim. state estimation: Kalman, H infinity, nonlinear approaches, Hoboken, NJ, USA: Wiley, 2006.
[15] S. Song, A. Zhang, J. Wang, and P. S. Yu, “SCREEN: Stream data cleaning under speed constraints,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, Melbourne, VIC, Australia, May/Jun. 2015, pp. 827–841, doi: 10.1145/2723372.2723370.
[16] W. Wagner, H. Raack, and B. Efron, “Confidence intervals for random forests: The jackknife and the infinitesimal jackknife,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1625–1651, May 2014.
[17] H. Wu, W. Sun, B. Zheng, L. Yang, and W. Zhou, “CLSTERS: A general system for reducing errors of trajectories under challenging localization situations,” Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol., vol. 1, no. 3, p. 115, 2017, doi: 10.1145/3130081.
[18] M. Yakout and L. Berti-Equille, and A. K. Elmagarmid, “Don’t be scared: Use scalable automatic repairing with maximal likelihood and bounded changes,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, New York, NY, USA, 2013, pp. 553–564, doi: 10.1145/2426367.2426706.
[19] A. Zhang, S. Song, and J. Wang, “Sequential data cleaning: A statistical approach,” in Proc. Int. Conf. Manage. Data. (SIGMOD), San Francisco, CA, USA, Jun/Jul. 2016, pp. 909–924, doi: 10.1145/2882903.2915233.
[20] Y. Zhang, W. Rao, M. Yuan, J. Zeng, and H. Yang, “Confidence model-based data repair for telco localization,” in Proc. 18th IEEE Int. Conf. Mobile Data Manage., (MMD), Daedeon, South Korea, May/Jun. 2017, pp. 186–195, doi: 10.1109/MMD.2017.33.
[21] K. Zhao, M. Musolesi, P. Hui, W. Rao, and S. Tarkoma, “Explaining the power-law distribution of human mobility through transportation modality decomposition,” Sci. Rep., vol. 5, p. 9136, Jun. 2015.

PENG ZHAO received the B.E. degree from Tongji University, Shanghai, China, in 2016, where he is currently pursuing the Ph.D. degree with the School of Software Engineering. His research interests include data mining, spatial-temporal data management, and mobile computing.

AOQIAN ZHANG received the B.E. degree in computer software from the School of Software, Tsinghua University, in 2013, and the Ph.D. degree in software engineering from the School of Software, Tsinghua University, under the supervision of Prof. J. Wang and Prof. S. Song. He is currently a Postdoctoral Fellow with the Data System Group, Cheriton School of Computer Science, University of Waterloo, under the supervision of Prof. I. F. Ilyas.

CHENXI ZHANG received the degree from the National University of Defense Technology, in 1988. He is currently a Professor of computer science with the School of Software Engineering, Tongji University. He has written or edited two books, including New Generation Computing: Recent Research (North-Holland Press, 1990) and Research on Frontiers in Computing (Tsinghua University Press, 1989). His research interests include methods and theory in distributed systems, peer-to-peer networks, and next generation networks. He received the National Science Fund for Distinguished Young Scholars, in 1996.
JIANGFENG LI received the B.Sc. degree from Wuhan University, Wuhan, China, in 2003, the M.Sc. degree from the Huazhong University of Science and Technology, Wuhan, in 2006, and the Ph.D. degree from Tongji University, Shanghai, China, in 2010. He is currently with the School of Software Engineering, Tongji University. His current researches are focused on distributed computing, information-centric networks, and mobile networking.

QINPEI ZHAO received the B.Sc. degree in automation technology from Xi’an Jiaotong University, Xi’an, China, in 2004, the M.Sc. degree in pattern recognition and image processing from Shanghai Jiaotong University, Shanghai, China, in 2007, and the Ph.D. degree in computer science with the University of Eastern Finland, in 2012. She is currently with the School of Software Engineering, Tongji University, China. Her current researches are focused on clustering algorithm and multimedia processing.

WEIXIONG RAO received the B.Sc. degree from Beijing Jiaotong University (formerly Northern Jiaotong University), the M.Sc. degree from Shanghai Jiaotong University, and the Ph.D. degree from the Chinese University of Hong Kong, in 2009. He was with the Hong Kong University of Science and Technology, in 2010, the University of Helsinki, from 2011 to 2012, and the University of Cambridge Computer Laboratory Systems Research Group, in 2013. He is currently with the School of Software Engineering, Tongji University, China. His research interests include networked data systems and energy-aware mobile computing.