Oscillation Resolution for Mobile Phone Cellular Tower Data to Enable Mobility Modelling

Wei Wu[1], Yue Wang[2], Joao Bartolo Gomes[1], Dang The Anh[2], Spiros Antonatos[2], Mingqiang Xue[1], Peng Yang[1], Ghim Eng Yap[1], Xiaoli Li[1], Shonali Krishnaswamy[1], James Decraene[2], Amy Shi-Nash[2]
[1]Data Analytics Department, Institute for Infocomm Research, Singapore
{wwu, bartologjp, xuem, yangp, geyap, xlli, spkrishna}@i2r.a-star.edu.sg
[2]R&D Labs, Living Analytics, Group Digital Life, Singapore Telecommunication Limited, Singapore
{wangyue, anhkeen, antonatos, jdecraene, amyshinash}@singtel.com

***Abstract*—One major problem of using location data collected from mobile cellular networks for mobility modelling is the oscillation phenomenon. An oscillation occurs when a mobile phone intermittently switches between cell towers instead of connecting to the nearest cell tower. For the purpose of mobility modeling, the location data needs to be cleansed to approximate the mobile device's actual location. However, this constitutes a challenge because the mobile device's true location is not known.**

In this paper, we study the oscillation resolution problem. We propose an algorithm framework called DECRE (Detect, Expand, Check, REmove) to detect and remove oscillation logs. To make informed decisions DECRE includes four steps: Detect, to identify log sequences that may contain oscillation using a few heuristics based on the concepts of stable period and moving at impossible speed; Expand, to look before and after suspicious records to gain more information; Check, to check whether a cell tower is observed repeatedly (which is a strong indication of oscillation); and REmove, resolving oscillation by selecting a cell tower to approximate the mobile device's actual location.

Our experimental results on travel diaries show that our oscillation resolution approach is able to remove records that are far from mobile device's ground-truth locations, improve the quality of the location data, and performs better than an existing method. Our performance study on large scale cell tower data shows that the MapReduce implementation of our approach is able to process 1 Terabyte of cell tower data in five hours using a small cluster.

I. Introduction

There is much focus today on understanding the semantics of location logs. Many applications aim to derive insights from mobility data to understand human dynamics to support applications such as customer behaviour, location-based service delivery, urban planning and targeted marketing [9], [16], [17]. Such data are an ideal source of information to understand human dynamics and segmented customer behavior [1], [10]. Direct applications include urban planning [9], targeted marketing [16], [17] and human mobility profiling [3], [4], [11], [13], [14].

The three main sources of location data are typically: WiFi, GPS and cellular tower data. The first two types of data require users' engagement through connecting to a WiFi network or turning on specific applications with adequate settings. WiFi and GPS data can be as accurate as 5-10 meters. However they can only capture an incomplete picture of mobile device location because of limited WiFi coverage, GPS line of sight and battery drainage [15].

On the contrary, the cellular tower data is passive and does not require subscriber engagement [7]. Whenever a mobile phone subscriber triggers an activity like making a call, the mobile phone operator (i.e., company) automatically logs the identifier of the cellular tower the mobile phone is connected to. The locations of the cellular towers can be used to approximate the mobile device's locations, thus much richer and comprehensive location data can be obtained using cellular technologies when compared with WiFi and GPS [2].

However, due to dynamic changes in signal strength and various transmission conditions, significant noise can be observed in cellular tower data. One of the key challenges of analysing with cellular tower data is the problem of cellular tower *oscillation* [2], [3], [12]. An oscillation occurs when a communication transaction oscillates between multiple cellular towers even though the mobile device is not moving. Sequences of oscillation events may be observed and this clearly introduces undesirable noise, which may potentially reduce the accuracy of the data, and ultimately limit the quality of analytics based on such data.

Addressing the oscillation problem is very challenging because the mobile device's real location is not known. The oscillation resolution method has to make probabilistic inferences of where the mobile device roughly is, based on observed logs associated with the cellular towers, and then use that to detect and remove oscillation logs.

In contrast with existing methods that rely on semantic tags (that need to be contributed by external parties) of the cellular towers and merge the cellular towers into clusters to represent mobile device location [2], [12], we want to design oscillation resolution techniques and strategies that do not rely on other data sources and use cellular tower location (rather than cellular clusters) to represent the mobile device location. Our techniques are designed for logs data collected by cellular towers rather than data collected on mobile devices.

We design an algorithm called DECRE to detect and resolve oscillation. The algorithm consists of four steps called Detect, Expand, Check, and REmove.

In the Detect step we find sequences of logs that contain oscillations. We propose a few heuristics to detect such sequences of logs. The heuristics are based on the notions of *stable periods* and *moving at impossible speed*. A *stable period* is a duration when the mobile phone is consistently communicating through one cellular tower. It strongly indicates that the mobile device is close to that cellular tower during that period. *Moving at impossible speed* means that a mobile device

cannot travel very far in a very short period of time (e.g., travel 5 km in one minute).

Some of the detected sequences are quite short and do not contain enough information for making informed decision. We therefore introduce an Expand step where we look before and after the sequence until certain conditions are satisfied. Then in the Check step we test whether the expanded sequence contains logs that switches quickly between cellular towers which is a strong indication of oscillation. Finally the Remove step selects a cellular tower to represent the mobile device's location for the detected sequence and removes the oscillation logs.

The contributions of this paper are:

- We study the challenging problem of oscillation resolution in mobile phone cellular tower data.
- We propose oscillation resolution techniques based on the ideas of stable period and moving at impossible speed (defined in section IV), as well as an algorithm called DECRE that is able to detect and resolve various kinds of oscillation.
- We study the performance of our techniques and strategies on user travel diaries. Results show that our method successfully filtered out data points which were significantly erroneous, in the sense that those data points were 2 to 8 times more distant from the ground truth locations when compared with non-filtered data. To our knowledge this paper is the first work that uses travel diaries to study the performance of an oscillation resolution technique.
- We confirm the scalability of our techniques through applying a MapReduce implementation on 1 Terabyte of cellular tower data.

For this research project, a sample dataset of 3 months mobile network data is used. Mobile network data is the service log when a mobile phone is connected to the mobile network. It contains snonymised ID, latitude, longitude, TimeStamp and service type. The anonymised ID is a machine generated ID via a two-step non-reversible AES Encryption and Hashing process. Hence it is impossible to trace back to the original ID. There is no personal information about mobile subscribers in the dataset, nor any content of calls or SMSs. The location information in the dataset is mobile cell tower's location.

The rest of the paper is organized as follows: In Section II we survey the related work. We introduce the cellular tower data in Section III. We present our ideas and algorithms in Section IV. Section V shows the results of our performance study. We finally conclude this paper in Section VI.

II. Related Work

The problem of cell oscillation is to some extent related to the filtering of trajectory data from noisy GPS traces. For GPS trajectories approaches such as Mean and Median Filters, Kalman filter or particle filters [5], [8] are usually applied depending on the nature of the data and requirements of the output. A discussion of such approaches for GPS data can be found in [18]. However, cellular tower data are usually sparser both in time as well as geographically and thus more specific approaches are required.

There is a small amount of work that focus on the cell oscillation problem. In [2] Murat Ali Bayir et al. propose a framework for discovering mobile user profiles from mobile phone data. As part of this framework a cell clustering method is proposed to deal with cell oscillations in mobility paths. The method creates clusters by using majority voting over the location tags of its cellular towers. For untagged cellular towers, the frequency of these towers oscillating pairs is calculated. If a cellular tower pair (without order) appears at least three times in a mobility path, it is regarded as an oscillating pair and the two cells are put into a cluster. In Section V we compare our technique with this method.

The resolution of cell oscillation is also investigated in [12]. The method relies on semantic tag of locations that is used to identify the cellular tower locations which normally overlapped to the same semantic location. This can address the cell oscillation problem for cellular towers appearing at the same semantic location within a short period of time, where the pairs most likely represent a switching due to load balancing or handover effect. For cellular towers not semantically tagged, their method makes use of Location Area Code and cellular tower's radius information to cluster the cellular towers. If the distance between the pairs is less than the sum of their coverage radius and the stay time is less than a threshold time then the cells are clustered together. We do not compare with this method in experimental study because we do not have cellular tower's Location Area Code and radius information.

Both the method in [2] and the one in [12] are designed for cellular tower dataset collected using application on mobile phones that ask subjects to give semantic tags to cells. The method in [12] further relies on Location Area Code information of the cells. In fact, both [2] and [12] are based on the Reality Mining Dataset [6] that contains semantic tag and cellular tower Location Area Code.

Another work that is closely related to this work is [3]. In [3], clustering of consecutive records is used to identify minor oscillations. In particular, a sequence of records is regarded as a cluster if the maximum spatial distance between any two records in it is smaller than a threshold (e.g., 1 km). After the clustering, the centroid of the points in a cluster is used to represent the location of the cluster.

What distinguishes our work from them is that we limit our methods to the spatial and temporal information that is readily available such as the location of the cellular towers and the time stamps of the logs. We do not want to rely on data sources that are hard to collect and verify (such as semantic tags provided by users) or unreliable information (such as cellular tower coverage radius). Furthermore, since the output of our cleansing method needs to be fed into further analysis steps that assume a fixed set of cellular tower locations, our method cannot introduce new locations such as clusters of cellular towers. In other words, our method is designed for cellular tower logs collected passively and we need to meet practical constraints such as efficiency and output format.

III. Cellular Tower Data and Oscillation

The mobile phone cellular tower data include subscribers' mobile device activity logs. Each log in the sequence contains the time of the cellular event (e.g., making/receiving a call, sending/receiving a SMS) and the identifier of the cellular tower that the mobile phone is connected to when that event happens. Each cellular tower has a unique identifier and its location (i.e., latitude and longitude) is known. By joining

```
2013-11-01 06:34:31,833     C1 lat1 lon1
2013-11-01 07:56:54,257     C1 lat1 lon1
2013-11-01 07:56:54,546     C1 lat1 lon1
2013-11-01 07:58:00,477     C1 lat1 lon1
2013-11-01 07:59:21,457     C1 lat1 lon1
2013-11-01 07:59:21,728     C1 lat1 lon1
2013-11-01 09:18:11,187     C2 lat2 lon2
2013-11-01 09:30:28,951     C3 lat3 lon3
2013-11-01 09:32:07,999     C4 lat4 lon4
2013-11-01 09:32:21,911     C5 lat5 lon5
2013-11-01 09:32:28,177     C3 lat3 lon3
2013-11-01 09:32:53,416     C5 lat5 lon5
2013-11-01 11:32:54,779     C6 lat6 lon6
2013-11-01 11:51:37,701     C6 lat6 lon6
2013-11-01 12:38:04,584     C6 lat6 lon6
```

Fig. 1: Example of cellular tower data. Each log contains the timestamp of an event, the id and location of the cellular tower the mobile phone is connected to when the event happens. Values of cell ID and location are masked in this paper to protect business data.

the cellular tower data with cellular tower's location based on cellular tower identifier, we get a mobile device's cellular location data at given time. Figure 1 shows some logs of a mobile device. Each line in this figure is a log and it contains datetime, cellular tower identifier, latitude and longitude.

Cellular tower data of a mobile device can be used to approximate the mobility trace of mobile phone subscribers. In the ideal case a mobile phone will connect to the nearest cellular tower whenever an event happens. In such cases the cellular tower data will be the best approximation of the subscriber's true trajectory we can get from the cellular event data. However, a mobile phone is not always connected to the cellular tower that is nearest to its actual location due to mobile phone network load balancing and other factors such as raining or proximity to water bodies (e.g., river, lake and sea). As a result, we often observe cell towers in a mobile device's log are very far from its actual location. Furthermore, even when the mobile device is stationary, his/her mobile phone can frequently switch between cell towers. These make the raw cellular tower data contain a lot of locations that do not reflect the mobile device's actual location.

The main objective of this research is to design an algorithm that detects logs that are far away from a mobile device's true location without knowing the its true location. We also want to detect the logs that switch between a few cell towers in a very short period of time. Such sequences of logs contain noise (in terms of modeling the location of the mobile device) and we want to remove such noisy logs.

Table I lists the symbols we will use and their meanings.

TABLE I: Symbols

Symbol	Meaning
C_j	cellular tower j where j is the identifier of the cellular tower
$C_j.lat$	latitute of the cellular tower C_j
$C_j.lon$	longitute of the cellular tower C_j
R_i	log i
$R_i.cid$	cell id of log i
$R_i.t$	date time of log i
$Distance(C_j, C_k)$	spatial distance between cell towers C_j and C_k
$TimeDiff(R_i, R_l)$	time difference between $R_i.t$ and $R_l.t$
$Distance(R_i, R_l)$	spatial distance between the cell towers in log R_i and log R_l
$Speed(R_i, R_l)$	$Distance(R_i, R_l)/TimeDiff(R_i, R_l)$
SP_i	a stable period
$SP_i.cid$	cell id of the logs in stable period SP_i
$SP_i.first$	the first log in a stable period
$SP_i.last$	the last log in a stable period

IV. DECRE Algorithm

We design an algorithm called DECRE (Detect, Expand, Check, and REmove) to find and remove oscillation logs. The DECRE algorithm has four steps. In the first "Detect" step we use four **heuristics** to find log sequences that contain oscillation logs. For some of the sequences we then use an "Expand" process to consider what are observed before and after the suspicious sequence. In the "Check" step we test whether the logs contain a cycle (defined in Section IV-C) that is a strong indication of oscillation. If the suspicious sequence is confirmed to contain oscillation logs, we delete the oscillation logs with a "Remove" step.

A. Detect

We use four heuristics to detect log sequences that may contain oscillation logs. Our heuristics are based on two concepts called stable period and moving at impossible speed.

1) Stable Period: A stable period is a time frame that is long enough and the mobile device continuously communicates with one cellular tower.

One of our fundamental understandings of the cellular tower data is that an individual log only tells us that the mobile device is **within the coverage** of that cellular tower at that particular time point. A single log is not enough for us to assert that the mobile device is **close** to the location of the cellular tower. However, if a mobile device continuously communicates with one cellular tower in a period that is long enough, it is very likely that the mobile device is **close** to that cellular tower.

Based on this, our first idea is to find a sequence of continuous logs that are associated with a same cellular tower.

Definition 1: given a sequence of log R_1 to R_n of a mobile device ordered by datetime, the **same-cell sequence**s are the continuous sequences of logs where the cellular tower is the same, i.e., $R_1.cid = R_2.cid, ..., = R_n.cid$. The **duration** of a same-cell sequence is the time duration from the time in the first log to the time in the last log of the same-cell sequence, i.e., $TimeDiff(R_1, R_n)$.

Figure 2 is an illustration of same-cell sequences in a sequence of logs. In this example there are 18 logs and four same-cell sequences are observed, namely SC1, SC2, SC3, and SC4. The first same-cell sequence SC1 contains three logs associated with cellular tower C1. The second same-cell sequence SC2 contains four logs associated with cellular tower C1. The third same-cell sequence SC3 contains two logs associated with cellular tower C4, and the fourth same-cell sequence SC4 contains four logs associated with cellular tower C7. Note that although both SC1 and SC2 contain logs with C1 they are separated by a log with C2.

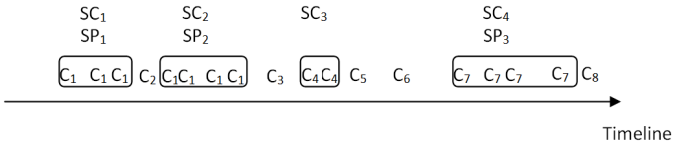


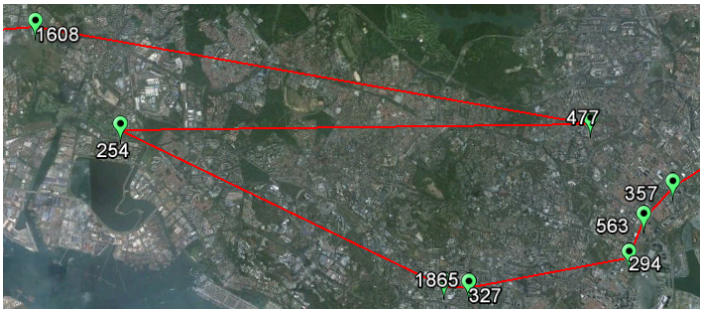
Fig. 2: Illustration of same-cell sequences and stable periods. SC1, SC2, SC3, and SC4 are the same-cell sequences. SP1, SP2, and SP3 are stable periods.

Definition 2: if the time duration of a same-cell sequence is long enough (e.g., longer than a threshold L such as 10 minutes), we call such a same-cell sequence as **stable period**.

For example, in the four same-cell sequences of Figure 2 we may find three stable periods, shown as SP1, SP2, and SP3. Note that SC3 is not a stable period because its time duration is not long enough.

2) moving at impossible speed: Impossible movements are observed when the spatial distance between consecutive logs are too far for the mobile device to travel in the time duration between the logs.

Through visual analytics we notice that there are many instances where the mobile phone suddenly connects to a cellular tower that is very far from whether the mobile device is. Figure 3 shows an example of such event that happened when a mobile device moves from east to west by a train[1]. There was a sudden jump from cellular tower 254 (at time 18:50:16) to 477 (at time 18:54:20) and then jumps to cellular tower 1608 (at time 18:55:58). Clearly, an oscillation happened in this example. In fact, we observe many such cases when we visualize the data.



time	lat	lon	d(km)	t(minute)	v(km/h)
18:50:16	lat1	lon1	8.45	-	-
18:54:20	lat2	lon2	11.09	4.07	163.61
18:55:58	lat3	lon3	13.32	1.63	489.34

Fig. 3: An example where moving at impossible speed is observed. Cellular identifiers are removed to make room for distance, time, and speed information.

Another kind of moving at impossible speed is observed in log sequences where a mobile device has several logs from different cellular towers in just one minute or even a few seconds. For example, in Figure 1 we observe five logs in the one minute of 09:32 associated with four cellular towers (C4, C5, C3 and C6). A specific heuristic (heuristic 4) is designed to capture such sequences.

3) Heuristics: Based on the concepts of stable period and moving at impossible speed we design the following four heuristics to detect log sequences that exhibit moving at impossible speed and therefore contain oscillation logs.

Heuristic 1

If two consecutive stable periods' cell is the same and the time difference between them is short enough (e.g., shorter than a threshold L_1T = 2 minutes), the logs between the two stable periods are very likely due to oscillation. Let SP_i and SP_{i+1} be two two consecutive stable periods, the condition in this heuristic can be expressed as

$(SP_i.cid == SP_{i+1}.cid)$

AND

$(TimeDiff(SP_i.last, SP_{i+1}.first) < L_1T)$

For example as shown in Figure 4, stable periods SP1 and SP2 are with the same cellular tower C1, and the time difference from SP1 to SP2 is short, so we are sure the log between them (i.e., C2) is due to oscillation.

The intuition is that both the first stable period and the second stable period tell us that the mobile device is close to cellular tower C1, and the time between them is not long enough for the mobile device to move close to C2 and return to C1, so the log with C2 is very likely an oscillation log.

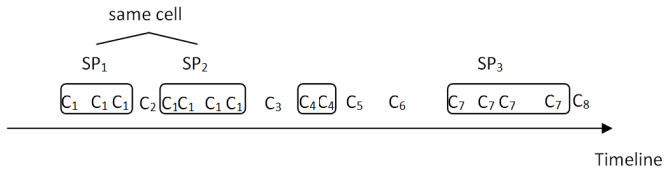


Fig. 4: Illustration of stable period based heuristic 1.

Heuristic 2

If shortly after a stable period there is a log whose cell is far away from the stable period's cell, that log is very likely due to oscillation. Let R_j be an immediate log after stable period SP_i, let L_2T and L_2D be two thresholds for time and distance, the condition in this heuristic can be expressed as

$(TimeDiff(SP_i.last, R_j) < L_2T)$

AND

$(Distance(SP_i.last, R_j) > L_2D)$

The intuition behind this heuristic is that the stable period tells us that the mobile device is close to the cellular tower in SP_i, and that the time between the stable period and the log is not enough for the mobile device to travel to a location that is close to the cellular tower of R_j.

Although it seems we can combine the conditions as $Distance(SP_i.last, R_j)/TimeDiff(SP_i.last, R_j) > L_2D/L_2T$ and use a threshold of speed to replace L_2D/L_2T, we do not do so for the following reason. Time difference can be very small, and therefore the derived speed can be very large even when distance is actually small. Such derived speed can be misleading. For example, if $Distance(SP_i.last, R_j)$ is 200 meters and $TimeDiff(SP_i.last, R_j)$ is 1 second, this will result in a speed of 720 km/h. If we use speed as threshold, such log will be regarded as oscillation. But in reality such log is possible when the mobile device enters the coverage of a new cellular tower. By using thresholds on time and distance we avoid removing many false positives.

Figure 5 shows an example of applying this heuristic. The time difference between stable period SP3 and the log of C8 is very short (e.g., 1 minute) but the distance between C7 and C8 is very far (e.g., 5 KM), according to the heuristic we are confident that the log of C8 is due to oscillation.

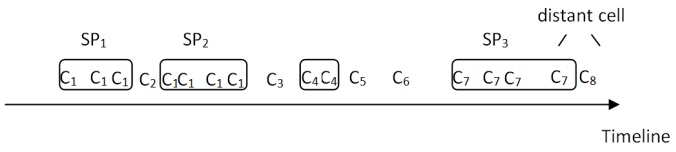


Fig. 5: Illustration of stable period based heuristic 2.

[1]We know the ground truth for this example because the data were contributed by one of the authors.

Heuristic 3

This heuristic is designed to capture the long jumps illustrated in Figure 3. Based on the knowledge of domain experts and also our insights gained from visual analytics of the data we know that such oscillation typically happens in a sequence of three logs. The second log suddenly jumps to a cellular tower that is far away from the cellular tower in the first log, and the third log jumps back to a log which is the same as (or close to) the first log.

Formally the heuristic for capturing such oscillation logs can be expressed as follows where V is a threshold for speed and L_3 is a threshold for distance.

$(Speed(R_i, R_{i+1}) > V) \wedge (Speed(R_{i+1}, R_{i+2}) > V) \wedge (Distance(R_i, R_{i+1}) > L_3) \wedge (Distance(R_{i+1}, R_{i+2}) > L_3) \wedge (Distance(R_i, R_{i+2}) < L_3/2)$

Note that we need the conditions on distance to make sure a long distance jump happened. It is a very strong evidence of oscillation. As explained in heuristic 2, condition on speed itself is not enough because the computed speed can be misleading if the time difference between two logs is extremely short (e.g., a few seconds).

We also notice that sometimes the time difference between two logs can be reasonably long and it makes the computed speed looks normal although a long distance oscillation happened. For instance, in Figure 3 the speed from cell 254 to cell 477 is 163km/h which is not very odd. However the speed from cell 477 to cell 1608 is close to 500km/h. We capture such cases by changing the condition to the following expression.

$(Speed(R_i, R_{i+1}) * Speed(R_{i+1}, R_{i+2}) > V * V) \wedge (Distance(R_i, R_{i+1}) > L_3) \wedge (Distance(R_{i+1}, R_{i+2}) > L_3) \wedge (Distance(R_i, R_{i+2}) < L_3/2)$

Heuristic 4

Most of the oscillations happen in a short period of time and they are not adjacent to any stable period. They exhibit moving at impossible speed but do not satisfy heuristic 3 because the distance between cellular towers in consecutive logs is not long enough. Such oscillations typically happen among cellular towers that are close to each other.

We use the following criterion to identify such sequences: within a short period of time (e.g., a=1 minutes) there are at least a few (e.g., b=3) logs from at least a few (e.g., c=2) cellular tower. We call a continuous sequence of logs satisfying the above condition as **suspicious sequence** identified using parameters a, b, and c.

Note that not all suspicious sequences identified by this heuristic contain oscillation logs. We understand such sequences are possible and can be even reasonable when the mobile device is moving fast (e.g., driving on highway). We use the "Expand" and "Check" steps to find suspicious sequences that do contain oscillation logs. We ensure we only remove logs that we are confident that they are oscillations.

B. Expand

By running the heuristic 4 on cellular tower data we discover that most of the suspicious sequences contain 3 or 4 logs, and typically 2 or 3 cellular towers are involved. Since such sequences contain only one or two logs from each cellular tower and all happen in a short period of time, hence the logs in such suspicious sequences do not contain enough information to determine which of the cellular towers best represent the mobile device's location. So we decide to look before and after the suspicious sequence to gain more information (or evidence). We call this step "Expand".

Given a suspicious sequence detected using time window a (minute), we expand the sequence by looking at most a minute(s) before the suspicious sequence and at most a minute(s) after the suspicious sequence. The look-back (or look-after) process stops when it encounters a log whose cellular tower did not appear in the suspicious sequence.

The reason we limit the "Expand" process to a minute(s) before and after the suspicious sequence and limit it to the cellular tower that appeared in the suspicious sequence is that oscillation is typically a short term event (e.g., due to load balancing of the cellular network) and we want to focus on the cells that are involved in the suspicious sequence.

After the "Expand" step the suspicious sequence typically will have a few more logs, but still from the same set of cellular towers. As a result, we have more reliable information to decide on which cellular tower probably best approximates the mobile device's location.

C. Check

From the domain experts we also find out that an important characteristic (or evidence) of oscillation is that cycle of cellular towers is often observed in a short period of time. Here a cycle is defined as a continuous sequence of logs whose first log and last log have the same cellular tower and there is at least one log from other cellular tower between them. For example, a sequence of log C1C2C1 exhibits the cycle from C1 to C2 and then back to C1. On the contrary, C1C1C2 does not contain a cycle.

For each suspicious sequence identified by heuristic 4 and expanded by the "Expand" process, the "Check" step tests whether the sequence contains a cycle of events. If it has a cycle, we confirm that there is oscillation in the sequence. Otherwise we claim that the suspicious sequence is due to fast movement and will not remove the logs from it. For example, when a mobile device drives through an area with high density of cellular towers, it is possible to observe a few logs from a few cellular towers in a short period of time. Basically, this "Check" step ensures that we only remove oscillation logs for which we have enough evidence (i.e., a cycle is observed in a very short period of time).

D. Remove

For the oscillations detected by heuristics 1, 2, and 3, it is clear which logs are oscillation and they should be removed. However, for the suspicious sequences identified by heuristic 4 and further confirmed with the Expand and Check steps, we need to decide which logs in the sequence are oscillation logs and which cellular tower should be used to represent the location of the mobile device for this sequence.

We design a score based algorithm to select the cellular tower to approximate the location of the mobile device. Each cellular tower contained in the suspicious sequence gets a score based on its frequency in the sequence and its average distance to other cells appeared in the sequence. We want to favor the cellular tower that appears frequently in the sequence and is close to other cells.

After determining the cellular tower that will be used to represent the mobile device's location, we remove the logs from other cells in the suspicious sequence. Basically they are regarded as oscillation logs. Algorithm 1 lists the details of this remove process.

Data: oscillation sequence
Result: a sequence of logs where oscillation logs are removed
C = the set of cells in the sequence ;
for *each cell* c *in* C **do**
  F_c = the number of times c appears in the oscillation sequence;
  D_c = the average distance from c to cells in $C - \{c\}$;
  $Score_c = F_c / D_c$
end
C_h = the cell with the highest score;
remove logs whose cells are not C_h;
return logs

Algorithm 1: Remove oscillation logs

Figure 6 shows an example where a suspicious sequence C9C10C9C11C9C11 is identified and it contains cycles (i.e., C9C10C9, C9C11C9, and C11C9C11). Figure 7 shows the relative locations of the cells involved in this sequence. The distance between C9 and C10 is 0.9 KM. The distance between C9 and C11 is also 0.9 KM. The distance between C10 and C11 is 0.1 KM. They are in a busy commercial area where density of cellular towers is high.

The algorithm counts the number of times each cell appears and get F_{c9}=3, F_{c10}=1, F_{c11}=2. Then it calculates the average distance from each cell to other cells. Since distance(C_9,C_{10})=0.9, distance(C_9,C_{11})=0.9, distance(C_{10},C_{11})=0.1, we get D_{c9}=(9+9)/2=0.9, D_{c10}=(9+1)/2=0.5, D_{c11}=(9+1)/2=0.5. Then the score for each cell is calculated as $Score_{c9}$=3/0.9, $Score_{c10}$=1/0.5, and $Score_{c11}$=2/0.5. As a result, cell C_{11} is selected as the mobile device location for this oscillation sequence.

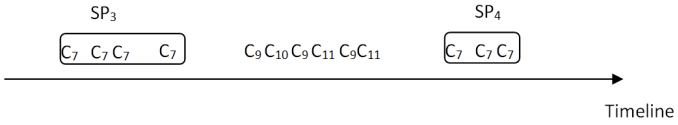


Fig. 6: An example of suspicious sequence discovered by the "Detect", "Expand", and "Check" steps.

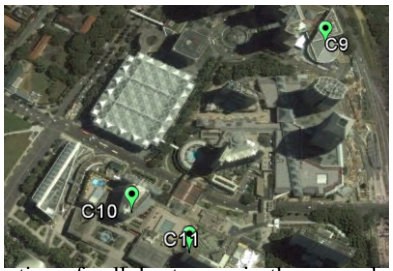


Fig. 7: The location of cellular towers in the example shown in Figure 6. We do not have cellular tower radius (or coverage) information, so we do not draw the circles around the towers.

E. Discussion

The DECRE algorithm is designed in a modular manner so that each step of the algorithm can be replaced with new algorithm in the future. For example, the heuristics in the "Detect" step can be replaced with more sophisticated detection criterion. As another example, if the radius of the cells are known, we can design a new REmove algorithm by taking that into account and then plug the new REmove algorithm to the DECRE algorithm.

V. Performance Evaluation

In this section we describe the experiments we performed to study the effectiveness and efficiency of our oscillation resolution techniques. For effectiveness, we want to see whether the records after oscillation resolution approximate the mobile device's true trajectory better than the original data. For efficiency we want to find out whether the methods can handle Big data. We report results on both travel diaries and large scale cellular data. We also compare our technique to the oscillation resolution method used in [2]. The comparison study is presented in Section V-C2.

Due to space limit we do not report the results of experiments that study the effect of the heuristic parameters. The results reported below are based on parameters values we used in the examples in Section IV-A3.

A. Performance Metric

The performance metric we use to measure effectiveness of the methods is the distance between location in cellular data and the mobile device's true location at corresponding time.

Recall that each cellular tower log LOG_i contains the time information $LOG_i.t$. Suppose we have the true location of the mobile device at time $LOG_i.t$ and denote it as LOG_i^l, we can compare the location in log LOG_i to LOG_i^l and calculate the distance between them as $distance(LOG_i, LOG_i^l)$.

Given a set of cellular tower records of a mobile device and the ground-truth locations at corresponding times, we define the average distance between locations in cellular records and corresponding true locations as the measure of **how the records approximate the mobile device's real mobility trace**. Formally, suppose N is the number of records, the performance metric is

$$\frac{\sum_i^N distance(LOG_i, LOG_i^l)}{N}$$

Recall that our methods remove some of the cellular tower records that are regarded as oscillation records. Let us use $LOG^{original}$ and $LOG^{cleansed}$ to denote the records before and after oscillation resolution respectively. We use $LOG^{removed}$ to denote the records that are removed. Thus we have $LOG^{cleansed}$ = $LOG^{original}$ - $LOG^{removed}$.

We compute the performance metric for $LOG^{original}$, $LOG^{cleansed}$, and $LOG^{removed}$. We can conclude that our methods are effective if

- $LOG^{cleansed}$ is closer to the ground-truth than $LOG^{original}$ is; and
- $LOG^{removed}$ is much farther away from ground-truth than both $LOG^{original}$ and $LOG^{cleansed}$ are.

B. Data Collection and Correction

Since we compare the locations in the cellular records to the actual locations of the subjects at the corresponding

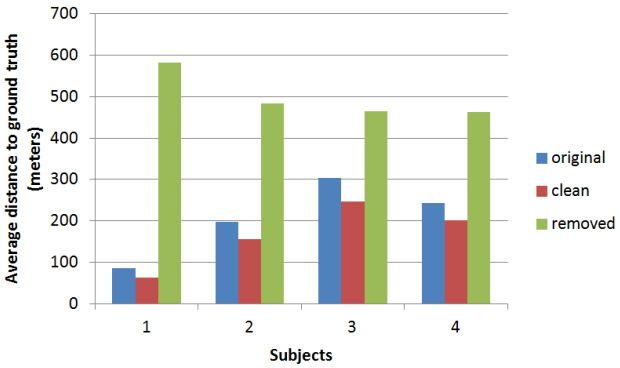


Fig. 8: Average distance (in meters) from records to subject's true locations.

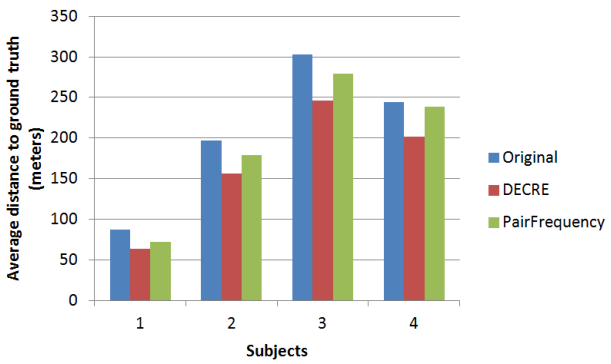


Fig. 9: Average distance (in meters) from cleansed records to subject's true locations. The shorter, the better.

timestamps, we need to have the actual locations of the subjects as the ground-truth for comparison.

Initially, we asked a number of subscribers to collect GPS data as ground-truth. However, GPS data can only be collected when the mobile phone is outdoor. GPS data are also very unreliable when the mobile device is in a bus or a train. As a result, very limited amount of GPS data (in terms of time) could be collected.

Consequently, we decided to ask the subjects to manually correct their cellular tower data as their travel diaries rather than trying to collect GPS data. For each point in the cellular data (that corresponds to a cellular tower location at a time point) the subject checks whether that point is close to his/her real location at that time point based on his memory and journal. If they do not match well, the subject simply changes (by dragging on a map) the location in the cellular log to the actual location. Therefore, after this process we obtained a trace with ground-truth locations and timestamps matching the original cellular tower trace.

A number of subjects help in correcting their cellular records collected for a period of 2 weeks. The traces include a wide range of activities such as staying at home, working at office, commuting, visiting shopping malls, transportation hubs, airport, theme parks, etc.

C. Results on Travel Diaries

1) Performance of DECRE algorithm: Figure 8 shows the distance (in meters) between different sets of records and the ground-truth for four randomly selected subjects. Each bar represents the distance between the records previously defined and the ground-truth. Note that different subject's records exhibit different characteristics. For example, subject U1's records generally approximate the real locations quite well. The records removed from his/her records are very far from the real locations. Such characteristics are subject dependent because they depend on where the subject lives and works and where he/she went during the two weeks.

We observe that the records in the cleansed set is closer to the ground-truth than the original records. Overall we gain about 10% improvement in terms of average distance. The reason it is not so significant is that most original records are not removed as oscillation records. This is as expected because most of the time our mobile phones are connected to the nearest cellular towers. Only about 5% of records are removed, therefore the average distance to ground-truth won't be affected significantly.

Comparing the removed records to the cleansed records, we observe that the removed records are much further away from true locations. For subjects U2, U3 and U4, the removed records are about 2-3 times further away than cleansed records. For subject U1, the removed records are even 10 times further away than cleansed records. Removing these noisy points (although the percentage of them is not very high) is very important for studying subject's mobility trace.

Using annotated data we confirm that our methods can effectively identify oscillation records and remove them. As a result, the records after cleansing can approximate the subject's true trajectory better. In particular, the misleading records that are far away from true locations are removed.

2) Comparison to Existing Method: We also compare our technique to the oscillation resolution method used in [2]. It first clusters the cellular towers based on subject's semantic tags, then detects oscillation cellular tower pairs without tags based on the number of times each cellular tower pair appears together. Since our dataset does not have semantic tags, we implement the second step of that method which handles towers without semantic tags. We use "PairFrequency" to refer to this method. Figure 9 and Figure 10 show the results.

Figure 9 shows the average distances from the datasets to the ground-truth. For each experimental subject we show distances from three datasets: the original, after DECRE resolves oscillation, and after PairFrequency resolves oscillation. The shorter the distances, the better. We can see the DECRE performs much better than PairFrequency. Its improvement is about two times of that of PairFrequency.

Figure 10 shows the average distances from the records removed by Human, DECRE and PairFrequency to the ground-truth. The larger the distances, the better. We can see that Human performs the best because they remember where they were and therefore they can easily identify the noisy points. DECRE performs better than PairFrequency. The points removed by DECRE are farther away from the ground-truth than the ones removed by PairFrequency.

Figure 9 and Figure 10 show that comparing to PairFrequency our algorithm DECRE is able to retain the points closer to the ground-truth and remove the points farther away from the ground-truth, without knowing the ground-truth.

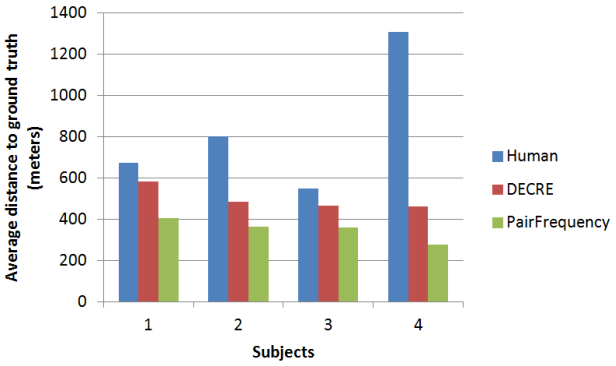


Fig. 10: Average distance (in meters) from removed records to subject's true locations. The higher, the better.

D. Results on Large Scale Data

We run our methods on a large scale dataset consisting of cellular tower data of collected in three months. The size of the dataset is about 1 Terabyte. A MapReduce implementation of our method is able to complete the cleansing process in a few hours using a small four-machine cluster.

In total our technique removes about 6 percent of the records as oscillation. Figure 11 shows the breakdown of the percentage of total records removed by the heuristics. We see that heuristic 4 removes most of the oscillation records identified. It is because most of the oscillations happen in a very short period of time and they are not adjacent to stable periods. They also do not oscillate between cells that are far away, and therefore they are not captured by heuristic 3 which has conditions on both movement speed and distance. Note that heuristic 4 detects most of the oscillation does not mean that heuristics 1, 2 and 3 are not important. In fact, they are able to remove points that deviate a lot from ground-truth.

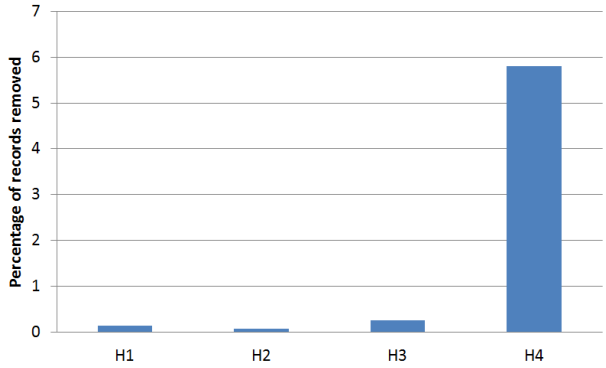


Fig. 11: Percentage of total records that are detected by the four heuristics.

VI. Conclusion

In this paper we study the problem of detecting and removing the oscillation records from cellular location data. We propose an algorithm called DECRE that detects and removes the oscillations to improve the quality of the data for mobility modelling. We studied the effectiveness of our methods using a travel diaries dataset. The results show that our methods are able to detect and remove the records that are far away from mobile devices' real location without knowing the real locations. We test the efficiency of our methods on a large scale cellular location dataset and show that our program is able to clean 1 Terabyte data in a few hours.

In the future, we plan to use the cleaned data in mobility modeling. In particular, the data will be used to identify important locations, to detect transport mode, and to build movement prediction models.

References

[1] Miriam Baglioni, José Antônio Fernandes de Macêdo, Chiara Renso, Roberto Trasarti, and Monica Wachowicz. Towards semantic interpretation of movement behavior. In *Advances in GIScience*, pages 271–288. Springer, 2009.

[2] Murat Ali Bayir, Murat Demirbas, and Nathan Eagle. Mobility profiler: A framework for discovering mobility profiles of cell phone users. *Pervasive and Mobile Computing*, 6(4):435–454, 2010.

[3] Francesco Calabrese, Giusy Di Lorenzo, Liang Liu, and Carlo Ratti. Estimating origin-destination flows using mobile phone location data. *IEEE Pervasive Computing*, 10(4), 2011.

[4] Meng-Fen Chiang, Wen-Yuan Zhu, Wen-Chih Peng, and Philip S. Yu. Distant-time location prediction in low-sampling-rate trajectories. MDM, 2013.

[5] Arnaud Doucet, Nando De Freitas, Kevin Murphy, and Stuart Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, 2000.

[6] Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[7] Nathan Eagle, John A Quinn, and Aaron Clauset. Methodologies for continuous cellular tower data analysis. In *Pervasive computing*, pages 342–353. Springer, 2009.

[8] Dieter Fox. Adapting the sample size in particle filters through kld-sampling. *The international Journal of robotics research*, 22(12):985–1003, 2003.

[9] Shan Jiang, Gaston A Fiore, Yingxiang Yang, Joseph Ferreira Jr, Emilio Frazzoli, and Marta C González. A review of urban computing for mobile phone traces: current methods, challenges and opportunities. In *2nd International Workshop on Urban Computing*, 2013.

[10] Eric WT Ngai and Angappa Gunasekaran. A review for mobile commerce research and applications. *Decision Support Systems*, 43(1):3–15, 2007.

[11] Christian M Schneider, Vitaly Belik, Thomas Couronné, Zbigniew Smoreda, and Marta C González. Unravelling daily human mobility motifs. *Journal of The Royal Society Interface*, 10(84), 2013.

[12] Shafqat Ali Shad and Enhong Chen. Cell oscillation resolution in mobility profile building. *International Journal of Computer Science Issues*, 9(3), 2012.

[13] Wanita Sherchan, Prem P Jayaraman, Shonali Krishnaswamy, Arkady Zaslavsky, Seng Loke, and Abhijat Sinha. Using on-the-move mining for mobile crowdsensing. In *Mobile Data Management (MDM)*, 2012.

[14] Zbigniew Smoreda, Ana-Maria Olteanu-Raimond, and Thomas Couronné. Spatiotemporal data from mobile phones for personal mobility assessment. *Transport survey methods: best practice for decision making. Emerald Group Publishing, London*, 2013.

[15] Alex Varshavsky, Mike Y Chen, Eyal de Lara, Jon Froehlich, Dirk Haehnel, Jeffrey Hightower, Anthony LaMarca, Fred Potter, Timothy Sohn, Karen Tang, et al. Are gsm phones the solution for localization? In *Workshop on Mobile Computing Systems and Applications, 2006*.

[16] David Jingjun Xu, Stephen Shaoyi Liao, and Qiudan Li. Combining empirical experimentation and modeling techniques: A design research approach for personalized mobile advertising applications. *Decision Support Systems*, 44(3):710–724, 2008.

[17] Soe-Tsyr Yuan and You Wen Tsao. A recommendation mechanism for contextualized mobile advertising. *Expert Systems with Applications*, 24(4):399–414, 2003.

[18] Yu Zheng and Xiaofang Zhou. *Computing with spatial trajectories*. Springer, 2011.