Predicting relative topological stability of mobile users in a P2P mobile cloud

Pijush Kanti Dutta Pramanik¹ · Gautam Bandyopadhyay² · Prasenjit Choudhury¹

Received: 1 July 2020 / Accepted: 23 September 2020 / Published online: 15 October 2020
© Springer Nature Switzerland AG 2020

Abstract

In a P2P mobile cloud (PMC), an ad-hoc cloud is formed using the closely adjacent smart mobile devices (SMD). Here, a resource-deficient SMD user seeks its neighbouring resource-rich SMDs' help to carry out a resource-wanting job. To maintain reliability and QoS of the PMC, it is necessary to be assured of the availability of the SMDs before submitting a job request. If the job is distributed to multiple SMDs, instead of assessing the availability of a single SMD, the availability of a group of SMDs is required for a certain period of time. The objective of the paper is to find a stable group of SMD users to increase the QoS of PMC. We aim to propose a novel mobility prediction method to assess the probability of a group of SMD users to be relatively static with respect to each other. We propose an algorithm to estimate the relative stability of an SMD with respect to its neighbourhood over a period of time, irrespective of its geographical location. We assess the short-term relative stability between a group of SMD users as well as the long-term mobility pattern between them. For mobility calculation, we have used real user traces from the UCSD dataset. The estimated results are analyzed and validated using different metrics.

Keywords P2P cloud · Ad hoc mobile cloud · Mobility prediction · Relative mobility · Evaluation metrics · Logistic regression · Goodness of fit

1 Introduction

Along with the growing adoption of Internet of Things (IoT), Internet of Everything (IoE), smart homes, smart cities, smart healthcare, etc. the demand for pervasive computing services has increased significantly. Cloud is a good option to cater to these demands but not always. In the context of this paper, following are some of the issues with the cloud:

- It may happen that due to some reason (e.g., unavailability of internet connection or network infrastructure failures, mobile data used up and there is no Wi-Fi, etc.) cloud is not accessible.

- A real-time application may not tolerate the latency required to obtain a cloud service.
- A reluctant user may not want to use paid cloud services.
- With limited wireless bandwidth, sending compute and data-intensive jobs to the cloud causes network and performance bottleneck for mobile cloud services.

In these scenarios, peer-to-peer (P2P) mobile cloud computing, where a mobile device can lend resources from a peer mobile device, can be a worthwhile alternative.

✉ Pijush Kanti Dutta Pramanik, pijushjld@yahoo.co.in; Gautam Bandyopadhyay, gautam.bandyopadhyay@dms.nitdgp.ac.in; Prasenjit Choudhury, prasenjit0007@yahoo.co.in | ¹Department of Computer Science and Engineering, National Institute of Technology, Durgapur, India. ²Department of Management Studies, National Institute of Technology, Durgapur, India.

1.1 P2P mobile cloud computing

Smart mobile devices (SMDs¹) are becoming popular as the primary computing device among the users due to its all-round functionalities and increasing computing capacity, which is equivalent to the laptops and desktops [1]. Due to their usefulness, flexibility and capability to be an all-purpose device, SMDs have been integrated with our daily life. With the increased demand for diverse pervasive and ubiquitous computing and other online services, people are becoming greatly dependent on their SMDs for more effective and functional services to satisfy these needs. Accordingly, various utility and imperative applications are being developed. Many of these applications require real-time inputs such as sensor data and crowdsourced data. As a result, most of them comprise computing-intensive jobs which demand significant computing capacity. But all the SMDs are not equivalent in terms of their resources. The SMD market is crowded with a huge variety of devices incorporating a diverse range of resources. Some of them are very high-end devices and often outperform a standard computer, while many of the SMDs are low in resources, aimed only for basic purposes [2]. Therefore, an SMD with lesser configuration and capacity may not be able to run a resource-demanding application. If they want to, they need to offload their job to the cloud, which is known as mobile cloud computing [3]. But as we mentioned earlier, accessing cloud services are not always viable or desired. Alternatively, a resource-low SMD might take the required service from a neighbouring resource-rich SMD. For that, a P2P mobile cloud (PMC) need to be formed.

Using PMC, a resource requester can avail computing resources from one or multiple SMDs simultaneously. If a single SMD has sufficient resources and is willing to cater the service demand, then the job is directly sent to it. If multiple SMDs are required, the job is divided and distributed among the service providers. A typical PMC model is shown in Fig. 1.

As a collection of powerful SMDs can garner huge computing ability [4], PMC can deliver high-performance computing (HPC) by building a temporary virtual computing cluster, utilizing the computing resources of the available SMDs. This ad-hoc HPC facility can be set up anywhere by connecting the SMDs through Wi-Fi, or other short-range communication means such as Bluetooth and hotspots. Therefore, PMC can fulfil the purpose of a miniature cloud, providing satisfactory computing capacity pervasively. Not only cost and time-efficient but PMC is also much

Fig. 1 A typical P2P mobile cloud model

energy-efficient HPC option compared to energy-hungry cloud data centres [5].

1.2 Resource lending scenario in PMC

Let us portray a scenario depicting the usefulness of PMC. Akash is on a group tour with his friends. He is on the train. Suddenly he receives an urgent email from his office. He is asked to complete the financial calculations attached with the mail immediately. Obviously, Akash is not happy but also knows that he has to do it anyhow. As the job is computing-intensive, Akash faces the following two problems:

- He has not brought his laptop with him and neither his friends. All of them are carrying only smartphones. But none of these devices is capable enough to carry out the computing-intensive calculations individually.
- As he is in a moving train, there is no continuous internet connectivity. And so, he cannot use the cloud.

So, Akash decided to utilize PMC and requests his friends to lend their SMDs' resources. The PMC application on his smartphone divides the computation task into a number of small tasks and sends each small task to the SMDs of Akash's friends. Each SMD executes its part and sends the result back to Akash's smartphone. Once, all the results are accumulated, they are assembled to have the final result. Akash sends the result to his office and moves on to enjoy his holidays without any tension.

¹ SMDs include smartphones, phablets and tablets.

Fig. 2 Relative stability is maintained between a group of mobile users, although their topological positions are changed

1.3 Resource availability problem in PMC

Any ad-hoc network is prone to become fragile due to the nodes' random mobility. Similarly, the PMC is also fairly unstable because of the user's mobility. Availability of a service provider within an ad-hoc network cannot always be guaranteed. The resource availability is greatly affected by user mobility. It may happen that a service provider agreed to lend its resources but exits before the assigned job is finished. This issue is known as resource availability problem. Resource unavailability can be tackled by shifting the job to other SMDs, but if the SMD owners move in and out of the PMC very frequently, resource switching and the associated overhead, will increase accordingly and that will hamper the quality of service (QoS) of PMC in terms of throughput and reliability. Therefore, service provisioning in PMC must be supported with an accurate mobility prediction mechanism.

It should be ideal to assign the job to the SMD, which is supposed to be stable and would stay within the PMC for a longer time (till the job is finished). In order to select one or multiple SMDs as stable resource providers, predictions about the future mobility behaviour of each of the participating nodes are crucial. Particularly, the following information are needed:

- For how long the SMD is available in the PMC grid?
- What is the mean/average period of availability?
- What is the probability of the service requester and the provider being in contact until the request is fulfilled?

1.4 Relative topological stability

The fundamental approach of PMC is P2P resource sharing, where resources are being shared between the peer SMDs, when in need. Here, peer means the neighbouring SMDs, communicating locally and willing to lend their resources. As pointed in Sect. 1.3, these resource consuming and providing SMDs should be together until the resource is fully available. In a traditional mobility prediction approach, the mobility of each SMDs are assessed individually, and if there is a change in mobility, it is assumed that the device might be unavailable.

In this paper, instead of absolute mobility or stability, we stressed on assessing the relative stability between the resource provider and the consumer. It may happen that though a pair or a group of mobile users change their topological position, relatively they stay together, i.e. they are moving together. Figure 2 depicts such a scenario. The figure shows the mobility traces of a number of SMD users (A, B, C, D, E, F, G, H, I, J, K, L, M, N, and O) from access points (APs) AP₁ to AP₄ along with time t₁ to t₄. It can be seen from the figure that a group of users (C, G, I, J, and O) always move together. Which means, though their topological positions are changed they are relatively stable. This is the exact case in the scenario described in Sect. 1.2. Here, though Akash and his friends are in the mobile state, they are relatively static as they are moving together.

In these circumstances, i.e., when the resource seeker and provider move together, there is no need for hand-off or switching of the resource provider. Here, mobility

does not affect PMC topology and operation. Finding such a group of SMD users enables us to select the resource providers which minimizes the chance of incomplete job execution and job reassignment in PMC.

1.5 Contribution of this paper

To avoid the availability problem, a mobility prediction algorithm is required to predict the topological stability of a group of smartphone users. The solution should identify the users whose probability to be relatively static is higher, i.e. they stay together most often across different network segments and locations. The primary concept of the proposed approach is that even the SMDs are mobile, if they are relatively static, i.e., they move together, resource exchange can be done as opposed to the other mobility prediction approaches in which only absolute mobility is considered.

In this paper, we propose a novel lightweight algorithm that predicts the relative stability of the SMDs in a PMC with respect to their neighbourhood. The proposed algorithm is validated by applying on a real trace based dataset and performing a logistic regression analysis on the predicted results.

We attempt to find answers to the following questions:

- Which are the users stay together within an AP for a certain period of time? When and how long?
- Whether they remain together when they go to another AP.
- If b) is not true then is there any pattern that user (u_1) belongs to a group (g_1) in AP_1 but belongs to g_2 in AP_2 steadily?

Two studies were conducted using the algorithm. One involved predicting mobility on a short-term basis, taking time spans of 2-h duration, and the other one involved predicting mobility on a long-term basis, taking the tracing period of 78 days. The results were compared to draw interesting conclusions. We believe that the proposed algorithm gives a fair estimation of the SMDs' mobility behaviour which is required to make the decision about selecting the SMDs for job offloading.

1.6 Organization of the paper

The rest of the paper is organized as follows. Section 2 discusses the basics of mobility prediction and related work. Section 3 describes the dataset and the proposed algorithm for calculating relative stability. This section also presents the methodologies for short-term relative stability prediction and long-term mobility prediction. The proposed work is validated and analyzed in Sect. 4.

Fig. 3 Mobility prediction approaches

Section 5 concludes the paper by summarising the study and discussing further research.

2 Mobility prediction approaches and related work

The majority of the mobility prediction schemes collect users' movement history somehow and analyze them to assess and predict the mobility patterns of mobile users. The prediction accuracy depends not only on the method used for collecting the mobility patterns but also on the algorithm used for mobility prediction. Typically, the research on mobility prediction for the infrastructure-based wireless network is mainly aimed to improve the performance of wireless networks on different aspects such as admission control, QoS, handoff management, etc. The mobility prediction algorithm that is available in the literature can be classified into three major approaches, as shown in Fig. 3 and discussed in the following sub-sections.

2.1 Movement history based mobility prediction

This class of algorithms predicts the future location of the mobile nodes with the help of past and current states of the user and can be further classified into two categories, as shown in Fig. 3. The synthetic-based mobility prediction uses mathematical models (e.g., sets of equations), which try to capture the movement of the devices whereas, in trace-based, the traces are obtained by using the deployed

systems' measurements and typically consist of connectivity logs or location information [6].

2.1.1 The synthetic based mobility model

There is significant research on mobility prediction for cellular network available in the literature [7], where the prediction approach is based on the user's movement history. Synthetic models are based on the following two assumptions:

- (a) Current mobility states can be observed.
- (b) Nodes behaviour shows some pattern, i.e. generally people exhibit both random and regular movements (e.g. going to the department every day).

Several random mobility models have been proposed. The mobility prediction algorithm proposed Lium and Maguire [8] considers the notion that every human has some extent of consistency in his/her movement. The proposed approach consists of regularity detection algorithms (for predicting users' future regular movement) and a motion prediction algorithm (for predicting the next states of a user's movement). For predicting the subsequent movement trail of the mobile user, the motion prediction algorithm utilizes the database of regular movement patterns of the users and the random probability information with constitutional constraints. Therefore, this model is well suited for regular patterns like hourly, weekly, etc. The regular and random components of user movements can be corresponded with circle/track patterns, or they can be simulated by the Markov chain model also.

Liu et al. [9] proposed a hierarchic position prediction algorithm, in which, user's movements are mapped to previous mobility patterns by the proposed mobility prediction model.

For simulating the highway traffic that comprises mobile entities travelling in forward and backward directions with different constant speeds, a shadow cluster concept was proposed by Levine et al. [10]. In this distributed framework, mobile terminals inform the neighbouring base station about their requirements, position, and movement parameters, based on which, the base stations predict future demands and reserve resources accordingly. It helps the base stations to assess the mobile terminals of which the demands can be met sufficiently and based on that, mobile terminals are admitted to that particular bases station. To predict mobility, it uses the user's movement history traces.

Mobility prediction proposed in [11, 12] assumes that the nodes move according to Random Way Point Mobility model [13]. The mobility prediction algorithm proposed in

[14] uses different mobility models such as movement circle model, movement track model, and the Markov chain model to model the user mobility behaviour.

Synthetic mobility models use a mathematical model to study human mobility [15, 16], and these are used in simulations. These mobility models are generally simple to model and implement and easy to use, and have been reported as an effective means to solve problems like signal attenuation and also consume less power. However, the random nature of synthetic mobility model often fails to capture human movement patterns correctly and leads to unrealistic-scenarios and non-uniform distribution [17]. The observation made and the conclusion drawn from such models may be misleading, especially where the user movements are predictable to some extent (e.g., mobile users in an educational campus, office, etc.).

2.1.2 Trace based mobility prediction

A trace-based mobility model is developed based on datasets collected from real scenarios by tracing or monitoring the movements of the persons carrying mobile devices. The collected movement traces are then analyzed to find specific mobility patterns of those mobile users. The trace-based mobility models comprise of the following three steps [18]:

- (i) Collection of traces: The mobility traces are acquired as a time-bound sequence of locations that are visited by the mobile users at different times, in a network. Here, the location might represent the actual geographical location of the mobile node or the location of the AP to which the mobile node is connected.
- (ii) Processing of traces: Before the captured mobility traces can be used for further analysis, they need to be processed to eliminate noise from the collected data.
- (iii) Analysis of traces: The processed data is then analyzed to find specific patterns that may be present in the mobility behaviour of the users.

A significant amount of research work has been done to predict mobility using movement traces of real-life users [19]. The data collected for movements of real users are used to predict future locations of the users using various prediction methods.

Kim et al. [20] proposed a method for estimating the physical location of users from a sizeable trace data of mobile devices associated with APs in a wireless network. The mobility traces collected by Dartmouth College is used to extract users' locations and their movement trails

which were estimated using methods like triangle centroid, time-based centroid and Kalman filter.

The method proposed by Khalifa and Abbas [21] is supposed to predict future locations of mobile users and the duration for which they would remain at those locations. The algorithm is evaluated using UCSD dataset. The method is found to have better prediction accuracy than O(3) Markov predictor. A comparison is made among Markov-based predictors, compression-based predictors and LZ predictors after adding a simple fall-back feature to each of these predictors for the situation where the recent context has not been previously observed.

Burbey and Martin [22] used the UCSD dataset to evaluate the proposed Markov model based mobility prediction method. The method is based on the data compression algorithm called Prediction by Partial Match, which predicts the location of a user at a given time. The dataset was divided into a training set (for training the prediction model) and a test set which was used to test the performance of the prediction model by trying to predict the location of a user by giving a particular future time.

The mobility model proposed by Musolesi and Mascolo [23] uses the concept of social networking, in which the hosts are grouped based on their social relationships. This clustering is mapped to a topographical space, where the potency of dynamic social ties determines the movements. The model was validated with real traces indicating the potential of the synthetic mobility traces as a decent approximation of human movement patterns.

The Autoregressive Hello protocol (ARH) was proposed by Li et al. [24] for neighbourhood discovery. Here, each node and its neighbouring nodes predict their positions through an ever-updated, auto regression-based mobility model. Each node estimates its neighbour's position, regularly, using previous location traces. When the predicted location is very far from the actual location, which causes a topology distortion, the node transmits a 'hello' message to state and update its current location. This helps in auto-correction of the mobility model of the concerned node and its neighbours.

The different application scenario demands a different degree of granularity and prediction accuracy. In a cellular network, the location is determined from the cellular ID of the base stations. The prediction of a person's location is in terms of a base station, whose transmission ranges from several hundred meters (e.g. 500 m) to km (e.g. 30 km). However, in terms of a person's location with respect to Wi-Fi, demands precision in the order of Wi-Fi range. In the context of PMC, accuracy requirements are in the order of a few meters only. The mobility prediction algorithm in the cellular network runs in the base stations. However, in a PMC, it will run on each SMD, and therefore, not suitable in a resource-constrained environment.

The prediction of topological changes in trace-based mobility prediction requires constant prediction of node's location, which may be considered as an overhead in a resource-constrained environment. However, if the location is predicted from the change of neighbourhood, then it reflects the topological change in the network and helps to determine the node's stability, which is crucial for selecting stable SMDs in PMC.

2.2 Received signal strength based mobility prediction

In this approach, the mobility of a node is estimated by measuring how much the received signal strength is dependent on the distance from the source. The estimated values of the node's location and its mobility information are acquired using signal attenuation versus distance travelled. This type of mobility prediction method is very simple [25]. To figure out the location of a node, the following information are needed:

- (a) Received signal strength.
- (b) Transmitted signal strength and the radio propagation model, i.e. the reduction of the signal strength with increasing distance.

A novel mobility metric for mobile ad-hoc networks (MOBIC) was proposed by Basu et al. [26]. Here, the node with the lowest mobility in the neighbourhood is elected as the cluster-head. Each node assesses the signal strength received from its neighbours continuously and based on the variance, the movement rate of that node relative to the neighbouring nodes is estimated. This mobility measurement is used for the formation of mobile clusters to improve the scalability of different services. For selecting a cluster-head, MOBIC employs Aggregate Local Mobility (ALM), a novel mobility metric. However, in the cluster maintenance phase, nodes' mobility behaviours are not always considered, and hence, a cluster-head does not guarantee to stand a low mobility characteristic [27].

As an extension of MOBIC, MobDHop, proposed by Er and Seah [28], uses a distributed algorithm to form a stable cluster which can serve as underlying routing architecture. The variance in received signal strength is used to predict the neighbourhood mobility and, based on mobility metric, forms a d-Hop cluster, as suggested in [26]. Compared to MOBIC, it exercises more samples of the received signal to estimate the predicted mobility. The prediction model assumes that the mobility patterns of nodes will be exactly the same in the future as were in the recent past. In MAPLE [29], based on the signal strength received from the cluster-head, each host estimates its distance from the corresponding cluster-head.

In a wireless network, poor signal reception is frequently experienced due to the obstruction of different objects such as trees, buildings, etc. The different object has a significant effect on path loss and received signal strength. However, if the environmental condition is known, the power of the signal may include the actual power plus the signal loss. In such a case, the receiver must be aware of the information, which requires an extra field in the transmitted packet and results in an overhead per packet.

2.3 GPS based mobility prediction

The Global Positioning System (GPS) based mobility prediction approach uses geographical locational information to determine users' mobility patterns [30].

To elect a cluster-head, the algorithm proposed by Wang et al. [31] uses GPS to predict node mobility and location information. The mobility prediction algorithm proposed by Su et al. [11] is based on location and mobility information provided by the GPS system. The algorithm uses the Network Time Protocol [32] or GPS clock to synchronize the nodes to avoid inconsistent data traces. The duration of a node at a place is determined by using GPS system data like speed, direction, radio propagation range etc.

To predict the mobility of the node, a novel routing protocol called Zone-Based Stable Routing (ZBSR) is proposed in [33]. The area is divided into the non-overlapping square zone. This prediction algorithm uses the traces collected by GPS. Every zone has a zone-head that does the job of a router in the network and also keeps the information about the other nodes in that zone. The node ID is used to decide the path to every zone.

GPS may not work in particular environments (like indoor places, places lacking strong cellular network, etc.) and consumes considerable battery power; hence, dependence on GPS is undesirable in mobile cloud. We take a new approach to analyze the movements of the mobile nodes. The salient feature of the proposed algorithm is to predict whether a node is relatively stable or not instead of predicting the location of a node. Furthermore, our algorithm does not use GPS information to predict mobility. According to our proposal, mobility prediction is redefined in terms of relative stability.

We define relative stability as an attribute of a mobile node that determines its stability, which says whether it is stationary or in motion with respect to the nodes in its neighbourhood, not its absolute movement (that which is determined by GPS) with respect to its geographical

surroundings. With the best of our knowledge, we found that this work is not available in the literature and our research work provides an algorithm which can meet with all the three challenges and can cope up with the different dynamic situations. The existing mobility prediction algorithm proposed in the literature is good for different applications domain. Though, they fail to predict the relative stability of a node with respect to its clients, which is the key to select a service providing SMD in PMC.

2.4 Work related to mobile cloud computing

In his project Hyrax [34], Marinelli developed a mobile cloud computing system with a few Android-based smartphones as computing resources. Shi et al. [35] presented the design and implementation of an ad-hoc mobile system that allows mobile devices to offload computation to other mobile systems in contact. Mtibaa, Harras and Fahim [36] conducted a set of experiments to study the advantages of offloading tasks in a cloud of mobile devices, in terms of time and energy. They also tried to find out the type of services more suitable to be offloaded to the mobile cloud. Chang, Srirama, and Ling [37] proposed a mobile device-hosted service-oriented workflow-based mediation framework for Mobile Social Network in Proximity (MSNP). The proposed framework, named as adaptive mediation framework for service-oriented MSNP (AMSNP), is based on a public mobile P2P network in which, mobile users can interact with neighbouring mobile devices.

2.5 Mobility and stability prediction in mobile cloud

Shah et al. [38] proposed to use the history of user's mobility patterns, in an ad-hoc mobile grid, to select a resource that would probably remain connected for a long period of time. Haryanti and Sari [39] predicted the mobility of a group of mobile nodes to identify a cluster of nodes, among a number of available resources providing nodes. Each node of that cluster has corresponding mobility with respect to the resource requesting node. The idea was to identify a stable cluster of nodes, which would ensure the completion of the assigned tasks. Zhou et al. [40] proposed a device selection method based on the stability resource status of the devices in a mobile device cloud. In the proposed model, the cloud controller maintains the historical information of each participating device. Based on these information, the stability of each device is assessed, which helps in selecting suitable device as a resource provider.

3 Calculating relative stability

In this section, we propose a strategy to estimate an SMD's relative stability. The relative stability of an SMD is measured by its stability (or dynamism) with respect to its neighbourhood.

Each SMD monitors its neighbourhood at different time intervals. Let, x denotes an SMD and NH be its set of neighbourhood SMDs. The mobility of an SMD at t_1 is calculated using Eq. 1, where $NH_{t_1}^x$ = list of neighbours of x at time t_1 , $NH_{t_1-1}^x$ = list of neighbours of x at time $(t_1 - 1)$ and $M_{t_1}^x$ is the mobility of x at time t_1 .

$$M_{t_1}^x = \frac{|NH_{t_1-1}^x \cap NH_{t_1}^x|}{|NH_{t_1}^x|} \quad (1)$$

The relative stability, over a time span T , is calculated using Eq. 2, where, $RS_{t_1}^x$ = relative stability of x at time t_1 which is stored for the last S_T time steps.

$$RS_{t_1}^x = \frac{1}{S_T} \sum_{i=t_1-x} M_{t_1}^x \quad (2)$$

The algorithm for relative stability calculation is given in Algorithm 1.

Algorithm 1: Relative stability calculation

Input: Node neighbour list database of x .

Output: Relative stability $RS_{t_1}^x$ of x .

1. Find current neighbour list $NH_{t_1}^x$.
2. Calculate the change in the neighbourhood using Eq. 1.
3. Store the obtained values in a linear list.
4. Calculate relative stability using Eq. 2.
5. Return relative stability $RS_{t_1}^x$.

The mobility at a particular time t_1 is considered as a temporal process. The spatiotemporal prediction tool is not considered because the node mobility at a particular time instance t_1 depends on the neighbour list at $t_1 - 1$ and t_1 , for $i = 1$ to n , where n is the number of intervals.

Fig. 4 Fields of `ap_locations.csv` and their descriptions

Fig. 5 Fields of `wtd.csv` and their descriptions

3.1 The UCSD dataset of real user traces

The correctness of the proposed mobility prediction algorithm was tested with real user traces obtained from the UCSD dataset [41]. The UCSD dataset contains the mobility information of 275 across more than 400 APs, recorded at University of California, San Diego over an 11-week period in 2002. The WTD software was used to collect the information in every 20 s for all APs that were detected across all frequencies. The dataset contains two trace files: `ap_locations.csv` and `wtd.csv`. `ap_locations.csv` contains four fields, as given in Fig. 4 and `wtd.csv` contains seven fields, as given in Fig. 5. The trace file `wtd.csv` was processed to generate the set of files required for the mobility study.

3.2 Short-term relative stability assessment

This section describes our study of relative stability made on a short-term basis. The descriptions of the variables

Table 1 Notations used in the relative stability analysis

Variable	Dependency	Description
U	Independent	USER_ID of users from trace dataset.
T	Independent	Time interval of 15-minute duration, where T = 1 denotes 00:01 am–00:15 am, T = 2 denotes 00:16 am–00:30 am, and so on. T lies in [1, 96].
D	Independent	Day of observation selected from the trace period, where D = 1 denotes 22nd September 2002, D = 2 denotes 23rd September 2002, and so on. D lies in [1, 78].
$A_{i,k}^j$	Partially dependent on U, T and D	AP_ID of the AP to which user with U = j is associated with during time interval T = i on day D = k; its value is “NC” if the user is not connected to any AP.
$N_{i,k}^j$	Partially dependent on U, T and D	Neighbourhood set of users with U = j during time interval T = i on day D = k; its value is “NC” if the user is not connected to any AP.
S_{i_1,k_1,i_2,k_2}^j	Dependent on N	Relative stability of user with U = j calculated over the time span: time interval T = i_1 on day D = k_1 to time interval T = i_2 on day D = k_2 . R lies in [0, 1]. It is to be noted that if $k_1 = k_2$, S_{i_1,k_1,i_2,k_2}^j can be written as $S_{i_1,i_2}^{j,k}$.

Table 2 Sample user's information

Time interval (T)	Lazy/busy hour	USER_ID (U)	Weekend neighbourhood (D = 8)	Weekday neighbourhood (D = 25)
00:01–00:15 (1)	Lazy hour	16	$N_{1,8}^{16} = \{\}$ (empty neighbourhood)	$N_{1,25}^{16} = \{3, 24, 127, 225\}$
12:45–13:00 (52)	Busy hour	16	$N_{52,8}^{16} = \{50, 64, 191, 235, 264\}$	$N_{52,25}^{16} = \text{NC}$
00:01–00:15 (1)	Lazy hour	103	$N_{1,8}^{103} = \text{NC}$	$N_{1,25}^{103} = \{165, 207, 238\}$
12:45–13:00 (52)	Busy hour	103	$N_{52,8}^{103} = \{171, 185, 193, 238\}$	$N_{52,25}^{103} = \text{NC}$
01:00–01:15 (5)	Lazy hour	39	$N_{5,8}^{39} = \{32, 128, 164, 166, 220\}$	$N_{5,25}^{39} = \text{NC}$
09:45–10:00 (40)	Lazy hour	39	$N_{40,8}^{39} = \{32, 164, 166, 208, 244\}$	$N_{40,25}^{39} = \text{NC}$
01:00–01:15 (5)	Lazy hour	65	$N_{5,8}^{65} = \{235, 264\}$	$N_{5,25}^{65} = \{16, 47, 64, 212, 243\}$
09:45–10:00 (40)	Busy hour	65	$N_{40,8}^{65} = \text{NC}$	$N_{40,25}^{65} = \{\}$ (empty neighbourhood)

which were used in our analysis are given in Table 1. Here, A and N are partially dependent variables, depending on independent variables U, T and D, whereas S is a completely dependent variable, depending on N.

For computing the short-term mobility, the value of relative stability was calculated from 20 users selected randomly from the dataset, and the relationship between their mobility and the obtained relative stability was analyzed. The selected users had the following USER_ID's: 10, 12, 22, 30, 35, 18, 24, 2, 7, 70, 77, 98, 135, 142, 208, 255, 263, 270, 240, and 210.

Since the relative stability prediction algorithm requires information about the neighbourhood of an SMD for the calculation of its relative stability, the original trace file was processed to obtain the neighbourhood information of all users. Each day was divided into 96 intervals, each of 15 min duration. The connection information of each user during these intervals was recorded. The connection information answered the following two questions:

- To which AP a particular user was connected during a particular 15-min interval on a particular day?
- What is the set of neighbours of a particular SMD (user) during a particular 15-min interval on a particular day?

For each user, a particular session of 2-h duration on a particular day during which he/she was connected to some AP was selected. The set of neighbours of a user U_i for a particular time interval was defined as the set of users, other than U_i itself, who were connected to the same AP as the user U_i during the particular time interval.

Table 2 shows the neighbourhood information of four sample users during busy hours (10 am–5 pm) and lazy hours (5 pm–10 am), on a weekday and a weekend, using the notation defined in Table 1. It is expected to have more SMDs in the busy hours while very few or nil in lazy hours.

The detailed information about the chosen session for a set of representative users and relative stability calculated over the session is given in Table 3 which also shows the relative stability of a set of representative users, calculated on a short-term basis.

Table 3 Selected session and relative stability information of a set of representative sample users

USER_ID	Day	Session (T_1 – T_g)	Relative stability	Mobility/stability behaviour
10	18	1–8	$S_{1,8}^{10,18}=0.8333$	Static; high relative stability
12	13	2–9	$S_{2,9}^{12,13}=0.95$	Static; high relative stability
22	31	36–43	$S_{36,43}^{22,31}=0.25$	Dynamic; low relative stability
30	2	7–14	$S_{7,14}^{30,2}=0.875$	Static; high relative stability
35	32	61–68	$S_{61,68}^{35,32}=0$	Dynamic; low relative stability

Fig. 6 Relative stability of 20 sample users

It is observed that the users who have a higher value of relative stability are relatively more static compared to the users who have a lower value of relative stability. By relatively static, we mean that a user is static with respect to its neighbours, irrespective of its actual geographical location. Detailed analysis reveals that the users who have relative stability value close to '1' have unchanging AP connection, whereas the users who have relative stability value close to '0' have frequently changing AP connection. From Fig. 6, we intuitively choose 0.5 as the threshold value for determining relative stability of the users, separating them into two groups, namely static users and dynamic users. A user whose relative stability is above 0.5 is regarded as static, and one whose relative stability is below this threshold value is regarded as dynamic.

Furthermore, a study of the relative stability of two sample users (USER_ID's 39 and 156) on a weekday and a weekend in terms of the mean and standard deviation of the neighbourhood change (according to Eq. 1) was performed; the related graphs are shown in Fig. 7. The mean and standard deviation values are calculated by taking

spans of 2 h (eight 15-min intervals) throughout the day. The mean and standard deviation values corresponding to an interval t are the mean and standard deviation of the neighbourhood change values over a time span t' to t , where $t' = t - 7$. Observing the weekday and weekend graphs of the two users, we come across different scenarios, as mentioned in Fig. 8.

It is to be noted that a user getting disconnected (going out of range of all APs/turning off his or her SMD) is regarded as a neighbourhood change, and the neighbourhood change value at the disconnection interval is assumed to be zero.

From the above observations, it is concluded that the short-term mobility study is not sufficient to determine a user's inherent behaviour, the reason being that when the standard deviation value becomes greater than the mean value of the neighbourhood change over a given timespan, the user's behaviour is unpredictable, even if it exhibits a high value of relative stability. To determine the inherent mobility behaviour of users, a logistic regression analysis of their relative stability for the entire trace period is needed, as described in the next section.

Fig. 7 Relative stability/mobility analysis for two users (USER_ID's 39 and 156) on weekday and weekend

3.3 Long-term mobility prediction using logistic regression analysis

A long-term study of mobility behaviour of all users is necessary to determine the effectiveness of our proposed prediction algorithm when users' neighbourhood records are taken over a long period of time. It indicates the degree of correctness of our method when a large database of users' movements is analyzed. If both short term and long-term prediction results are consistent for a particular user, the user can be predicted to be static or dynamic with a high chance of the prediction being correct.

The proposed algorithm was applied on the UCSD dataset and the average value of relative stability, termed as average stability ratio (ASR), was calculated for all the users in the dataset over the entire trace period of 78 days (taking the average of 78 daily values of relative stability for each user), after which a logistic regression model was derived to predict whether a user is static or dynamic. Out of the 275 users in the trace, 240 users

were used for the analysis (the other users were filtered out as they remained active in zero or very few sessions during the entire trace period, leading to an insignificant value of the standard deviation of neighbourhood change). This set of users was divided into two sets for training and testing. Records of the first 160 users were used as the training set to derive the logistic regression model, and of the remaining 80 users were used as the test set to validate the model.

Before performing logistic regression, we needed a method to classify a user as static or dynamic. We took the dependent variable as static or dynamic and one independent variable of ASR. The descriptions of variables are summarised in Tables 4 and 5. There is no definite method for defining the status of a node as static or dynamic. We used a method that is simple and objective, i.e. if the coefficient of variation (CV), defined by Eq. 3, of the relative stability of a user is less than 1, it is classified as *Static*, and otherwise, it is classified as *Dynamic*.

Case 1	<i>Stability Ratio < Standard Deviation and Stability Ratio < Threshold Value</i>
<ul style="list-style-type: none"> • The user's behaviour is unpredictable. • There is no fixed group of neighbours with which the user stays connected. • For example, such behaviour is exhibited by user 39 during intervals 66 to 73 on day 6. 	
Case 2	<i>Stability Ratio < Standard Deviation and Stability Ratio ≥ Threshold Value</i>
<ul style="list-style-type: none"> • The user behaviour is erratic. • The user stays connected to a fixed group of neighbours for some intervals but it changes its neighbourhood at other intervals. • For example, such behaviour is exhibited by user 156 during intervals 53 to 60 on day 6. 	
Case 3	<i>Stability Ratio > Standard Deviation and Stability Ratio < Threshold Value</i>
<ul style="list-style-type: none"> • The user's behaviour is predictable and it is dynamic. • The user changes its neighbourhood in a similar pattern leading to an almost constant low value of neighbourhood change during the timespan. • For example, such behaviour is exhibited by user 156 during intervals 69 to 76 on day 8. 	
Case 4	<i>Stability Ratio > Standard Deviation and Stability Ratio ≥ Threshold Value</i>
<ul style="list-style-type: none"> • The user's behaviour is predictable and it is static. • The user stays connected with almost the same fixed group of neighbours throughout the timespan leading to an almost constant high value of neighbourhood change. Such a user can be regarded as static. • For example, such behaviour is exhibited by user 156 during intervals 56 to 63 on day 8. 	

Fig. 8 Different possible scenarios based on Fig. 7

Table 4 Dependent variable based on the coefficient of variation

Degree of relative stability	Type of user based on ASR	Criterion
High relative stability	Static	CV < 1
Low relative stability	Dynamic	CV > 1

$$\text{Coefficient of variation} = \frac{\text{Standard deviation}}{\text{Average stability ratio}} \quad (3)$$

Table 5 Independent variable

Name of the variable	Description of the variable
Avg_SR	ASR of a user, calculated for the full trace period of 78 days

Table 6 Dependent variable encoding

Original value	Internal value
Static	0
Dynamic	1

As the dependent variable or outcome is a dichotomous one, we have taken *Static* = 0 and *Dynamic* = 1 to classify the nodes, as shown in Table 6.

4 Validation and analysis

4.1 Logistic regression analysis

Logistic regression [42, 43] has popularly been used for predicting the outcome of an event, or the probability of the presence or absence of a characteristic in a considered context is based on the values of a set of predictor variables. Precisely, it is the most popular modelling approach for binary outcomes. Since, in our prediction requirement, there are only two possible outcomes, either static or dynamic, the logistic regression is the automatic choice for prediction modelling. The basic characteristics of logistic regression are shown in Fig. 9.

Compared to the linear regression model, it is more appropriate to model the situation where the dependent variable is dichotomous (a 0/1 variable). In a linear regression model, the input of the dependent variables might be greater than 1 or less than 0. Therefore, it is impractical to model probabilities with linear regression since the boundary of the probability of an event should be within 0–1. To overcome this, the logistic regression model adopts a generalized form of the linear model and extends it by mapping the real numbers to the 0–1 range [44].

The relationship between the regression coefficient (z) and the probability (p) of an event of interest (i), in a logistic regression model, is described by the link function, given in Eq. 4. Here, the dependent variable (z_i) is a logit, which is a log of odds, in other words, it is the value of the unobserved continuous variable for the i th case, and p_i is the probability that the i th case experiences the event of interest and z_i and is defined by Eq. 5.

$$\text{logit}(p_i) = z_i = \ln \frac{p_i}{1 - p_i} \quad (4)$$

Fig. 9 Basic characteristics of logistic regression **Fig. 10** Link function p_i

$$p_i = \frac{e^{\text{logit}(p_i)}}{1 + e^{\text{logit}(p_i)}} = \frac{e^{z_i}}{1 + e^{z_i}} = \frac{1}{1 + e^{-z_i}} \quad (5)$$

In Fig. 10, the predicted variable p_i and the explanatory variable z_i are represented through the vertical and horizontal axes, respectively. The odds ratio (z value) of each independent variable in the logistic regression model can be estimated using the regression coefficients, as given in Eq. 6.

$$\text{logit}(p_i) = z_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} \quad (6)$$

where x_{ij} is the j th predictor for the i th case, β_j is the j th coefficient, and k is the number of predictors.

Here, β s are the regression coefficients that are evaluated by means of an iterative maximum likelihood method. Logistic regression analysis does not consider the prior probabilities of failure as well as the restrictive assumptions regarding the normal distribution of independent variables or the equal dispersion matrices [45, 46]. Though, due to the subjectivity of the choice of these misclassification costs in practice, most researchers aim

Table 7 Estimated results of the logistic regression model of the ASR of the mobile users

	Coefficient of the predictor (β)	Standard error (SE)	Wald $((\beta/SE)^2)$	DF	Significance
ASR	21.293	3.453	38.019	1	.000
Constant	-6.455	.981	43.251	1	.000

Table 8 Classification table of the proposed model

Observed performance		Predicted performance	
Static	Positive observation	Static Positive prediction	Dynamic Negative prediction
		TN	FP
Dynamic	Negative observation	FN	TP

to minimize the total error rate, implicitly assuming equal costs of type I and type II errors [46, 47].

4.2 Empirical testing

To test the proposed model, we opted for the Wald statistic that is used to assess the significance or contribution of the coefficients in the model [48]. It is defined by Eq. 7.

$$\text{Significance} = \frac{(\text{Regression coefficient})^2}{(\text{Standard error of the coefficient})^2} \quad (7)$$

Wald statistic is based on Chi square distribution with DF (degrees of freedom) is 1; i.e., the number of independent parameters is one. The parameter for which the Wald statistic is maximum is the best one that contributes to predicting the outcome properly. Maximization of Wald statistic means the minimization of the standard error of the corresponding parameter.

Table 9 Classification table for training dataset

Observed performance	Predicted performance		Correct prediction percentage
	Static	Dynamic	
Static	40	11	78.4
Dynamic	5	104	95.4
Overall correct prediction percentage			90.0

Table 10 Classification table for evaluation dataset

Observed performance	Predicted performance		Correct prediction percentage
	Static	Dynamic	
Static	31	4	88.6
Dynamic	2	43	95.6
Overall correct prediction percentage			92.5

The estimated results of the logistic regression model for the ASR of the users in the total sample are shown in Table 7. The maximum likelihood estimation for user classification is used to estimate the final logistic regression, as expressed in Eq. 8, where, z is defined as in Eq. 6. Here, the event (i) is *Static*; hence, p_i denotes the probability of the outcome *Static*.

$$z = -6.455 + 21.293 \times \text{ASR} \quad (8)$$

4.3 Classification accuracy

The accuracy of the proposed model has been diagnosed using classification table (also known as confusion matrix), as shown in Table 8, which gives a statistical estimation of the number of observations that have been predicted correctly. Classification tables help to evaluate the correctness of the model by cross tabulating the actually observed successes with the predicted number of successes and actually observed failures with a predicted number of failures. For each case, if the predicted probability of a category is greater than the user-specified cut-off, the predicted response of that category is considered as 0. The cut-off value was considered as 0.5.

4.3.1 Classification of training dataset

Table 9 shows the comparison of the observed and the predicted mobility behaviour of the users and to the extent that it can be correctly predicted on the training

dataset. Dynamic users can be classified 95.4% correctly, while 78.4% of Static users can be properly classified. The overall prediction is 90.0% correct. The cut-off value can be changed to notice the change/improvement in the predicted correct response.

4.3.2 Classification of evaluation dataset

Table 10 shows the comparison of the observed and the predicted mobility behaviour of the users and to the extent that it can be correctly predicted on the evaluation dataset. *Dynamic* users were classified 95.6% correctly, while 88.6% of *Static* users were predicted accurately. The overall prediction is 92.5% correct.

4.4 Validation of the model

In model validation, the prediction performance of the model is observed by checking against independent data. Typically, the validation process includes collecting an independent dataset and validation of the results on it. To validate our model, as shown in Table 10, we took 80 test users, and the results were validated using various evaluation metrics, as listed in Table 11. The performance measures of the prediction model are compared for the training and evaluation datasets and shown in Fig. 11.

4.5 Tests for goodness of fit

The goodness of fit asserts how well a statistical model and the sample data fit to a set of observations. It helps to identify if there is any discrepancy between observed values and the values predicted from the considered model in a normal distribution case. Although several methods are there for testing the goodness of fit of the logistic regression, in this work, we used the popular Hosmer and Lemeshow test (HL test) [49] that provides useful information about the calibration of the model. If the p value produced by the HL test for goodness of fit is small, the model is considered as a poor fit. Typically, if it is less than 0.05, then the model is rejected; otherwise, it is passed.

The HL test is fundamentally a Chi square goodness of fit test for grouped data, where the samples are divided into g groups according to their predicted probabilities. Typically, the value of g (number of groups) is chosen as 10. The observations with the lowest 10% predicted probabilities are placed in the first group, the next lowest 10% are placed in the second group, and so on. The grouping is done on the estimated parameter values (z_i), for the probability that stability is predicted correctly for each observation in the sample, based on each observation's covariate values, as defined by Eq. 4.

Table 11 Performance measure details for prediction model evaluation

Evaluation metrics	Significance	Ideal value	Calculation
True positive rate (TPR) or sensitivity or recall	Indicates how many static SMDs, out of all the static SMDs, have been predicted correctly	Close to 1	TP/(TP + FN)
True negative rate (TNR) or specificity	Indicates how many dynamic SMDs, out of all the dynamic SMDs, have been predicted correctly	Close to 1	TN/(TN + FP)
False positive rate (FPR) or fall-out (Type-I error)	Indicates how many dynamic SMDs, out of all the dynamic SMDs, have been predicted incorrectly	Close to 0	FP/(TN + FP)
False negative rate (FNR) or miss rate (Type-II error)	Indicates how many static SMDs, out of all the static SMDs, have been predicted incorrectly.	Close to 0	FN/(TP + FN)
Positive predictive precision (PPP) or positive predictive value (PPV)	Indicates how many SMDs, out of all the predicted static SMDs, are actually static	Close to 1	TP/(TP + FP)
Negative predictive precision (NPP) or negative predictive value (NPV)	Indicates how many SMDs, out of all the predicted dynamic SMDs, are actually dynamic	Close to 1	TN/(TN + FN)
False discovery rate (FDR)	Indicates how many SMDs, out of all the predicted static SMDs, are actually dynamic. In other words, it is the probability of making any Type-I error at all	Close to 0	FP/(FP + TP)
False omission rate (FOR)	Indicates how many SMDs, out of all the predicted dynamic SMDs, are actually static. In other words, it is the probability of making any Type-II error at all	Close to 0	FN/(FN + TN)
Accuracy (ACC)	Determines the overall predicted accuracy of the model. In other words, it measures the fitment of the model	Close to 1	(TN + TP)/(TN + FN + TP + FP)
Balanced accuracy (BA)	Gives a balanced accuracy calculation for the imbalanced test set, like ours. It is calculated as the average of TPR and TNR for each class (here, static and dynamic)	Close to 1	(TP/(TP + FN) + TN/(TN + FP))/2
F1 score	Provides a better understanding of the performance of the model than ACC, especially in case of an uneven class distribution, like ours. It is calculated as the harmonic mean of TPR and PPP	Close to 1	2TP/(2TP + FP + FN)

Chi square statistic calculated by HL test is given by Eq. 9.

$$X_{HL}^2 = \sum_{j=1}^{10} \frac{(O_j - P_j)^2}{P_j(1 - P_j/n_j)} \sim \chi_8^2 \quad (9)$$

where n_j =number of observations in the j th group; O_j =number of observed cases in the j th group; P_j =number of predicted cases in the j th group; χ^2 =standard Chi square.

The degree of freedom and the p-value are calculated using Eqs. 10 and 11 respectively.

$$DF = k - 2 \quad (10)$$

where k =number of regrouping (i.e., 10 in HL test) of the observed and predicted cases of two dependent variables.

$$p - \text{value} = \int_0^{11.053} f(X^2) d(X^2) \quad (11)$$

The observed significance level for Chi square value is found to be 0.199, as shown in Table 12, which indicates that the null hypothesis of the model is acceptable, i.e. the observed values are close to the predicted values. The acceptance of the null hypothesis ensures that the hypothetical model fits the observations reasonably satisfactory. The calculated Chi square value of this model at the level of significance below 0.05 indicates that logistic regression is very meaningful and our null hypothesis for goodness of fit, i.e., observed mobility are equal to the predicted mobility tested by the HL test.

Fig. 11 Performance measures for training and evaluation datasets **Table 12** Hosmer and Lemeshow test results

Chi square	DF	p value
11.053	8	.199

4.6 Comparison of the two studies of relative stability

Previously, the instantaneous relative stability of 20 sample users from the trace dataset was calculated and analyzed. The relative stability of two additional users throughout a particular weekday and weekend was analyzed as well. Later, the overall ASR of all users (240 users out of 275 users of the entire trace dataset) was calculated, and a logistic regression model was derived to predict the mobility of the users' in terms of *Static* or *Dynamic*. The significance of the logistic regression analysis is that it shows us the

inherent nature of the users. The short-term relative stability and its analysis may show a result different from the overall nature of a particular user. For an exemplar demonstration, Table 13 summarises the analysis information of two users.

It is observed from the table that the user with USER_ID 24 has a relative stability value of 0.6875 on the 61st day of the trace period during intervals 42–49, giving us the impression that it is nearly static. Logistic regression analysis classifies it into the category of *Dynamic*. Due to this difference in the two results, the exact tendency (to be static or dynamic) of the user in the 50th interval cannot be determined convincingly. On the other hand, the user with USER_ID 255 has a relative stability value of 0.9583 on the 45th day of the trace period during intervals 29–36. Logistic regression analysis shows that it is indeed static on a long-term basis, classifying it as static analyzing its ASR. This user

Table 13 Analysis of two sample users

USER_ID	Instantaneous relative stability	Behaviour (by intuition)	Classification according to logistic regression model	Remarks
24	0.6875	Static	Dynamic	Mobility cannot be predicted convincingly; behaviour is misleading
255	0.9583	Static	Static	Mobility can be convincingly predicted because of consistent behaviour

can be convincingly predicted as static in the 37th interval. For users that satisfy a consistent behaviour with both the instantaneous and regression analysis, i.e. if their results in both the analysis match, prediction about their stability can be made convincingly.

5 Conclusions

A solution to the problem of finding a stable peer service provider, faced in P2P mobile cloud (PMC), is presented in this paper. To avoid frequent job loss and job handover, it is desired that the SMDs which are relatively static to the service requesting SMD for a longer period should be chosen as the service providers. The proposed work finds the mobility pattern of a group of mobile users, connected to different Wi-Fi access points (APs), from the UCSD dataset, created using real user traces. Two studies have been conducted on the dataset using the proposed algorithm. The first study applied the algorithm to calculate relative stability for 20 sample users from the trace, choosing time spans of 2-h duration. These values gave the instantaneous behaviour of the nodes. In the second study, the average relative stability of a user was calculated taking its neighbourhood information during the entire trace period of 78 days. A logistic regression analysis of the average stability of all users from the trace was performed. The regression model was derived using a training set of 160 users and was evaluated using a test set of 80 users.

The general observation is that the short-term grouping does not necessarily reflect the true behaviour in the long term. It is observed that the result of the instantaneous analysis and that of the regression analysis, which takes into account the overall mobility of users for a long time period, do not match in case of some users. These users can be considered to be erratic in their movements on the campus in consideration, and therefore their behaviour is unpredictable. Based on their relative stability, the users were classified into two categories: static and dynamic. If a user's instantaneous relative stability and its classification derived by the regression model indicate similar behaviour, its mobility, i.e. whether it is static or dynamic, can be predicted with a high probability of accuracy.

The results show that the proposed algorithm can be used to efficiently predict the behaviour of the nodes in a PMC. However, the proposed algorithm does not make any prediction about how long a node in the network is going to remain static once it has been predicted to be static. If a node is found to be static at a particular location, its duration of stay at that location is a criterion that should be considered while making the decision about

selecting a stable SMD in PMC. Further research will be carried out to predict the duration of stay of a static node to bring more improvement in the mechanisms of service management in PMC.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Pramanik PKD, Choudhury P, Saha A (2017) Economical supercomputing thru smartphone crowd computing: an assessment of opportunities, benefits, deterrents, and applications from india's perspective. In: 4th international conference on advanced computing and communication systems (ICACCS—2017), Coimbatore, India
2. Pramanik PKD, Choudhury P (2020) Mobility-aware service provisioning for delay tolerant applications in a mobile crowd computing environment. *SN Appl Sci* 2(3):1–17
3. Fernando N, Loke SW, Rahayu W (2013) Mobile cloud computing: a survey. *Future Gener Comput Syst* 29:84–106
4. Pramanik PKD, Pal S, Pareek G, Dutta S, Choudhury P (2018) Crowd computing: the computing revolution. In: *Crowdsourcing and knowledge management in contemporary business environments*. R. Lenart-Gansiniec, Ed., IGI Global, pp 166–198
5. Pramanik PKD, Pal S, Choudhury P (2019) Green and sustainable high-performance computing with smartphone crowd computing: benefits, enablers, and challenges. *Scalable Comput Pract Exp* 20(2):259–283
6. Musolesi M, Mascolo C (2008) Mobility models for systems evaluation—a survey
7. Chan J, Seneviratne A (1999) A practical user mobility prediction algorithm for supporting adaptive QoS in wireless networks. In: *Seventh IEEE international conference on networks (ICON'99)*
8. Lium G, Maguire G Jr (1996) A class of mobile motion prediction algorithms for wireless mobile computing and communications. *Mob Netw Appl* 1(2):113–121
9. Liu T, Bahl P, Chlamtac I (1998) Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks. *IEEE J Sel Areas Commun* 16(6):922–936
10. Levine D, Akyildiz I, Naghshineh M (1997) A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept. *IEEE/ACM Trans Netw* 5(1):1–12
11. Su W, Lee SJ, Gerla M (2001) Mobility prediction and routing in Ad Hoc wireless networks. *Int J Netw Manag* 11(1):3–30
12. Agarwal A, Das SR (2003) Dead reckoning in mobile Ad Hoc networks. In: *IEEE Wireless Communications and Networking (WCNC)*, New Orleans, USA
13. Camp T, Boleng J, Davies V (2002) A survey of mobility models for Ad Hoc network research. *Wirel Commun Mob Comput* 2(5):483–502
14. Erbas F, Steuer J, Kyamakya K, Eggesieker D, Jobmann K (2001) A regular path recognition method and prediction of user movements in wireless networks. In: *IEEE vehicular technology conference (VTC)*
15. Gonzalez M, Hidalgo C, Barabási AL (2008) Understanding individual human mobility patterns. *Nature* 453:479–482

16. Navidi W, Camp T (2004) Stationary distributions for the random waypoint mobility model. *IEEE Trans Mob Comput* 3(1):99–108
17. Wang W, Guan X, Wang B, Wang Y (2010) A novel mobility model based on semi-random circular movement in mobile Ad hoc network. *J Inf Sci* 180(3):399–413
18. Munjal A, Camp T, Aschenbruck N (2012) Changing trends in modeling mobility. *J Electr Comput Eng* 2012:2012
19. Gross T, Truduce C (2005) A mobility model based on wlan traces and its validation. In: 24th annual joint conference of the IEEE computer and communications societies, Miami, USA
20. Kim M, Kotz D, Kim S (2006) Extracting a mobility model from real user traces. In: 25th IEEE international conference on computer communication (INFOCOM)
21. Khalifa I, Abbas H (2012) Mobility prediction in dynamic grids. *J Comput Inf Sci* 5(3):90
22. Burbey I, Martin TL (2008) Predicting future locations using prediction-by-partial-match. In: First ACM international workshop on Mobile entity localization and tracking in GPS-less environments (MELT'08), San Francisco; USA
23. Musolesi M, Mascolo C (2007) Designing mobility models based on social network theory. *ACM SIGMOBILE Mob Comput Commun Rev* 11(3):59–70
24. Li X, Mitton N, Simplot-Ryl D (2011) Mobility prediction based neighborhood discovery in mobile Ad Hoc networks. In: NETWORKING 2011. Lecture notes in computer science, vol 6640, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont and C. Scoglio, Eds., Berlin, Heidelberg, Springer, 2011, pp 241–253.
25. Velayos H, Karlsson G (2004) Limitations of range estimation in wireless LAN. In: 1st workshop on positioning, navigation and communication (WPNC'04), Hannover, Germany
26. Basu P, Khan N, Little T (2001) A mobility based metric for clustering in mobile ad hoc networks. In: 21st international conference on distributed computing systems workshops, Mesa, AZ, USA
27. Yu J, Chong P (2005) A survey of clustering schemes for mobile ad hoc networks. *IEEE Commun Surv Tutor* 7(1):32–48
28. Er I, Seah W (2004) Mobility-based d-hop clustering algorithm for mobile ad hoc networks. In: IEEE wireless communications and networking conference, Atlanta, GA, USA
29. Palit R, Hossain E, Thulasiraman P (2006) MAPLE: a framework for mobility-aware pro-active low energy clustering in ad hoc mobile wireless networks. *Wirel Commun Mob Comput* 6(6):773–789
30. Tsui JBY (2004) Fundamental of global positioning system receivers: a software approach, 2nd edn. Wiley, Hoboken
31. Wang G, Zhang L, Cao J A virtual circle-based clustering algorithm with mobility prediction in large-scale MANETs. In: Lu X, Zha W (eds) Networking and mobile computing (ICCNMC 2005). Lecture notes in computer science, vol 3619. Springer, Berlin, Heidelberg, pp 364–374
32. Mills D (1991) Internet time synchronization: the network time protocol. *IEEE Trans Commun* 39(10):1482–1493
33. Hassanein HS, Du H, Yeh C (2004) Robust route establishment in high mobility MANETS. In: International computer engineering conference
34. Marinelli EE (2009) Hyrax: cloud computing on mobile devices using mapreduce, Master's thesis, Carnegie Mellon University
35. Shi C, Lakafosis V, Ammar MH, Zegura EW (2012) Serendipity: enabling remote computing among intermittently connected mobile devices. In: Thirteenth ACM international symposium on mobile Ad Hoc networking and computing (MobiHoc'12), South Carolina, USA
36. Mtibaa A, Harras KA, Fahim A (2013) Towards computational off-loading in mobile device clouds. In IEEE 5th international conference on cloud computing technology and science, Bristol, UK
37. Chang C, Srirama SN, Ling S (2012) An adaptive mediation framework for mobile P2P social content sharing. In: Liu C, Ludwig H, Toumani F, Yu Q (eds) Service-oriented computing (ICSOC 2012). Lecture notes in computer science, vol 7636. Springer, Berlin, Heidelberg, pp 374–388
38. Shah SC, Nizamani Q-U-A, Chauhdary SH, Park M-S (2012) An effective and robust two-phase resource allocation scheme for interdependent tasks in mobile ad hoc computational Grids. *J Parallel Distrib Comput* 72(12):1664–1679
39. Haryanti SC, Sari RF (2014) Improving resource allocation performance in mobile Ad Hoc grid with mobility prediction. In: International conference on intelligent green building and smart grid (IGBSG), Taipei, Taiwan
40. Zhou A, Wang S, Li J, Sun Q, Yang F (2016) Optimal mobile device selection for mobile cloud service providing. *J Supercomput* 72(8):3222–3235
41. McNett M, Voelker GM (2005) Access and mobility of wireless PDA users. *Mob Comput Commun Rev* 9(2):40–55
42. Balakrishnan N (2019) Handbook of the logistic distribution. CRC Press, Boca Raton
43. Hilbe JM (2017) Logistic regression models. Chapman and Hall/CRC, Boca Raton
44. Agresti A (2012) Categorical data analysis, 3rd edn. Wiley, Hoboken
45. Hosmer DW, Lemeshow S (2000) Applied logistic regression, 2nd edn. Wiley, Hoboken
46. Zavgren CV (1985) Assessing the vulnerability of American industrial firms: a logistic analysis. *J Bus Finan Acc* 12(1):19–45
47. Ohlson JA (1980) Financial ratios and the probabilistic prediction of bankruptcy. *J Acc Res* 18(1):109–131
48. Menard SW (2002) Quantitative applications in the social sciences: applied logistic regression analysis, 2nd edn. SAGE Publications Inc., Thousand Oaks
49. Hosmer DW Jr, Lemeshow S, Sturdivant RX (2013) Applied logistic regression, 3rd edn. Wiley, Hoboken

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.