# Effective Mobility Identification in Mobile Fog Environment with the Internet of Things

https://doi.org/10.3991/ijim.v16i15.31501

Deepa D, Jothi K R(✉)
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India
jothi.kr@vit.ac.in

**Abstract—**Fog extends the cloud to be closer to the end devices so that it acts on IoT data within a millisecond. Almost 60% of data can be analyzed that is physically near to the IoT data. This proximity has various advantages, including reduced latency, which improves the user experience. However, because the distance to a fog service may vary as a user moves from one location to another, user mobility may restrict such benefits in practice. A fog service migration is based on a mitigation approach that allows the service to always be close enough to a user. Quality of Service is decreased because of the mobility of the user's location. Predicting the future location in advance improves the efficiency of service provisioning. In this work, a dynamic mobility model is proposed to find the user location in advance. This experiment was carried out by LuST mobility data set collected by Luxembourg Simulation of Urban Mobility (SUMO) Traffic (LuST). This result is give better accuracy of location prediction up to 98.87% when compared with existing methods.

**Keywords—**efficiency, fog computing, mobility, migration, prediction

## 1 Introduction

With the rapid development of the Internet of Things (IoT), modern mobile devices and their application have been rapidly increasing in the way of computation- demanding and delay-sensitive in many real-time applications [1]. Meanwhile, in cloud computing's high popularity of data from IoT devices, many time-sensitive applications, and their services cannot benefit from cloud computing technologies. (e.g., in a real-time pipeline application, there is an IoT application that will keep monitoring the application, suppose if any leakage in the pipeline immediate action will take place to prevent the damage but in cloud computing time take to process and get back to the application is too high by that time already we lost the opportunity to prevent the damage [2]. these issues can be resolved by the computing paradigm, i.e. fog computing. Fog computing extends all the capabilities of the cloud and does the computation near to the source device and gives immediate response. This improves reduced latency and real time-sensitive applications.

Due to their real-time computation, sensing communication, and storage capabilities, smart vehicles play an important role as the main data generator in a mobile fog computing system. The amount of data gathered by the different sensors is very high. Most mobility applications require real-time response, especially for applications for traffic control and safety enhancement. The traditional cloud computing architecture, however, is not planned to fulfill this requirement for low latency, as data obtained from mobility applications will be processed remotely instead of locally, because of the delay in transmission and any possible communication problems. Therefore, fog nodes located in proximity to mobile application fog computing systems will dramatically reduce the response time for vehicle applications [3]. Cloud will pre-push some essential resources to the fog to minimize network latency and release the traffic pressure over the links is the main function of fog in a mobile environment. The mobile customer is then able to conduct offline computation on the fog layer to produce and store only the important results in the cloud. The dense geographical deployment of fog servers, in addition, helps the device to be aware of the position of the end-user. The fog-aided cloud systems could therefore be well served by certain location-sensitive applications [4].

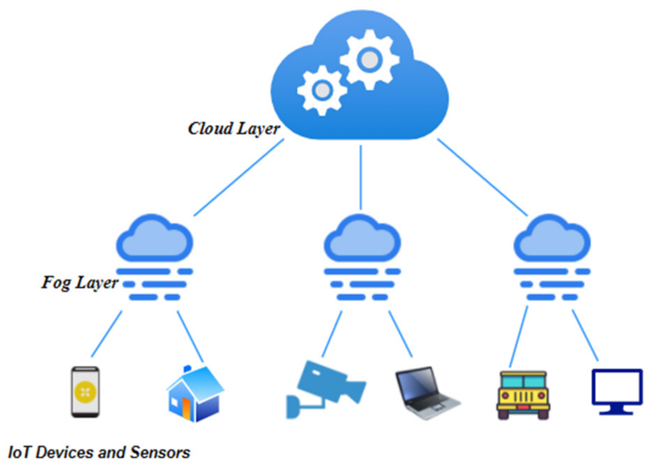


**Fig. 1.** Fog computing architecture

The basic fog computing architecture, consists of three layers in lower layer IoT devices and sensors are placed data is being generated in this layer Figure 1 In the middle layer, fog nodes are located that do their computation near to the source device so it gives response within milliseconds. In the top layer, the cloud server is placed only long store data and the high computation processing is only moved to the cloud processing.

Even though fog computing gives a solution to latency for real-time applications, mobility is one of the major problems which are unsolvable in fog computing. When the user migrates it finds the new fog node near the source device, and the continuation of the data processing is suspended due to the lack of mobility feature in fog computing. If the mobility of the user is identified in advance the continuation of the data processing improves the QoS.

Mobility prediction is different for humans and vehicles. Human mobility is identified is based on the location, time, contact, connectivity, temporal, spatial, and connectivity of the individual human models, and vehicular mobility is based on their geographical location, connectivity, RSU, and smart devices [5]. The general mobility prediction of humans is derived from the properties of the Generic mechanism like exploration and preferential method. The feasibility of migration between the fog nodes is considered a technique for mobility management. Once the mobility of the node is identified resource provisioning improves the performance of the fog node that reduces the latency of the fog computing. This work proposed a dynamic mobility prediction method, in this model mobile users' future location is calculated based on the coordinates X and Y corresponding to latitude and longitude [6].

## 2 Related work

[1] Provide the task assignment in Mobile Edge Computing with mobility. This paper focuses on optimized assignment considering mobility prior to serving the task with minimal execution time. The average delay of the different networks and different type's users and the acceptance ratio of the user performance are improved.

[2] Developed Blockchain-based Mobility – aware Offloading (BMO) before predicting the mobility of the user they define the staying time between a mobile device in the service coverage of a fog server. Using the old location, new location, preferred location with the probability, and the complementary probability the mobility of the individual is identified.

[5] Provide the human mobility prediction in an opportunistic network based on the three aspects first one is mobility characteristics which include spatial, temporal, and connectivity and the second one is models which contain real traces and simulation models and the third one is prediction is based on location, time and contact.

[7] This paper uses linear mobility prediction for connected car systems that use the before and current location to predict its future location and it performs the task assignment optimized load balancing concept to avoid deadline misses count in the connected car environment in the fog environment.

[8] Provide the mobility prediction for both humans and vehicular in a fog environment. To predict the mobility of the user they classify the system into the low dynamic environment and highly dynamic environment. With this mobility prediction, the scheduling mechanism is implemented to reduce the latency and improve the QoE and QoS.

[9] Providing the feasibility of migration between cloudlets is considered a technique for mobility management devices and prediction algorithms are used to predict the specification of mobile applications in the future (e.g., location, bandwidth, processing speed requirements)

[10] Proposed Tessellation concepts that they divide the larger area into smaller groups and then apply the concepts to evaluate the mobility model. In the tessellation model, they follow four paths pure random, same direction, same sense, and skewed. Mobility pattern is identified based on all possibilities of six directions.

[11] Proposed Ubiquitous Resource Management for Interference and latency-Aware Service, in general, the mobility model is based on two types Probabilistic and deterministic in this they used a deterministic approach. In the deterministic approach, they follow an indoor experimental scenario with the user mobility in a small region.

In this paper mobility prediction method is proposed in the dynamic environment of the mobile users and the fog nodes. Dynamic mobility prediction improves the response when the user moves from one location to another location. Mobility prediction is more challenging in the mobile fog environment. By the nature of the fog computing minimum computation, the device is also able to predict the location based on availability. Existing predicting techniques increase the computation time that increases delay latency.

## 3 Mobility prediction model

In fog computing, the fog nodes can reserve some computation resources similar to the cloud computing services for mobile users and IoT devices. For example, if a mobile user is playing video content and moving from one node to another, the fog node needs to provide continuous video content without any deviation [8]. For this purpose, the mobility identification of the node is necessary to provide a better user experience. Once the mobility is identified in advance latency problem is easily reduced. The mobility prediction model is based on two categories human and vehicle. Both are having separate special characteristics to evaluate and identify mobility patterns.

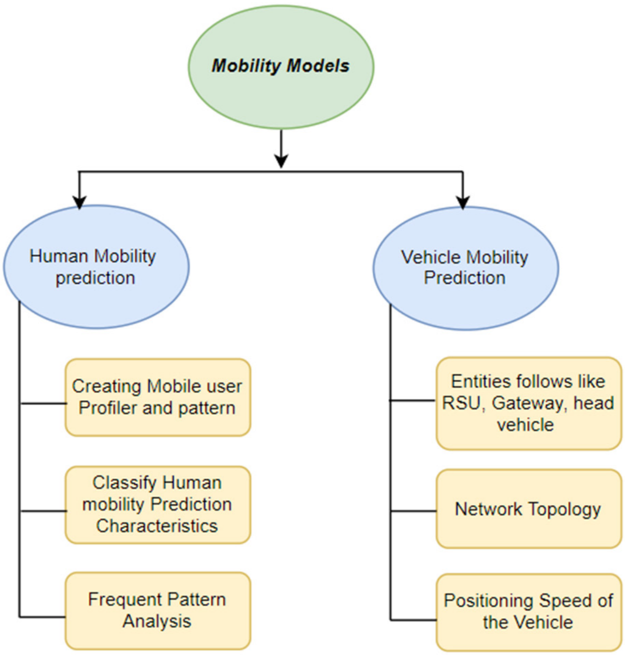


**Fig. 2.** Mobility model

In general, the mobility model problem is classified into two types one is mobile tracking and another one is mobile location positioning are represented in Figure 2. Mobile tracking is generally used for the simulating purpose and the location position is used to monitor and evaluate the management task [12]. Various mobility prediction methods with the different types of data set collected from the different locations are represented in Table 1. These datasets are pre-processed and results are compared to evaluate the simple mobility prediction. Almost mobility prediction is mainly two types one is human mobility characteristics and another one is vehicle mobility is collected from the different traffic locations [13].

**Table 1.** Techniques in mobility prediction

| Ref. | Implementation Method | Location | Method | Data Set | Parameter Evaluated |
|---|---|---|---|---|---|
| [7] | Cloud and fog based environment | GPS data of taxis gathered in Rome with a radius of 500 meter | Linear mobility prediction model | CRAWDAD dataset Roma/taxi | Mobility prediction, deadline misses comparison |
| [8] | Cloud-based Environment | Real-time experiment | Mobility prediction using human movement | https://foursquare.com | Human mobility prediction |
| [1] | Cloud-based Simulation | Small cell base station users. | Lagranges interpolation and non-parametric approach | https://www.3gpp.org/ftp//Specs/archive/36_series/36.814/ | Average delay, acceptance rate mobility |
| [9] | C++ implementation | Mobility traces of taxi cabs in San Francisco | Online Assignment of Mobile Application | https://crawdad.org/epfl/mobility/20090224 | Migration rate |
| [2] | Docker blockchain network | A dataset in Melbourne central business district in the total area of 6.2km with 817 mobile users | Using probability and complementary probability with parameters | https://github.com/swinedge/eua-dataset | Human preferred speed of walking 1.4m/s |
| [5] | Cloud-based environment | GPS data-trace from the Lake Geneva region of Switzerland gathered over 18 months in the interval of 10seconds | History-based predictor using the expectation-maximization algorithm | CRAWDAD dataset Roma/taxi | Future mobility prediction |
| [14] | Java Environment | 350k hours of cell span data | Creating mobility Profiler Framework | Real-World Dataset | Finding frequent mobility pattern |
| [15] | MobFogSim simulator | 100 different buses in urban mobility patterns on average at 22.3kmph in a route of on average 26.44min | Migration Strategy and policy | Luxembourg SUMO traffic | Access point coverage, users speed, And the number of cloudlets |
| [16] | iFogSim | Real-time dataset | Delay Priority model | Real-time dataset with 2 cloudlets and 6 users | Number of users movement delay and total network usage |
| [17] | Real-time health monitoring IoT system | Square, hexagon, and random topologies | Focused on gateway Handover mechanism with different topology | Real-Time data | Handover latency, accuracy |

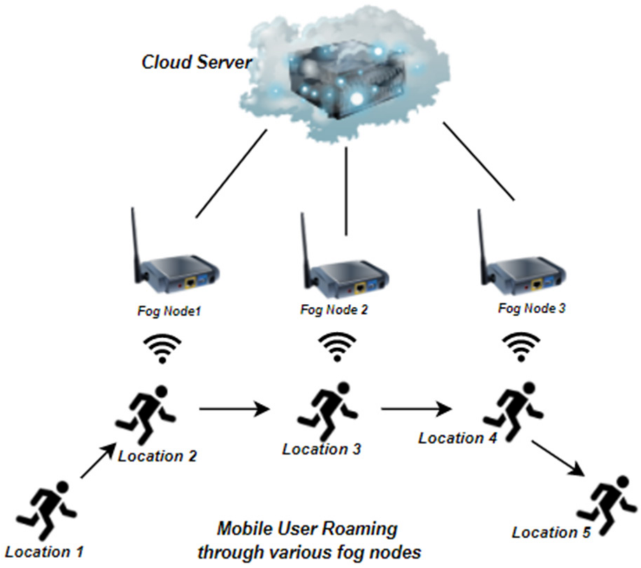


**Fig. 3.** Mobility in fog computing

User is moving from one location to another while moving migration takes place the will increase the latency in computation in Figure 3 Initially the user is accessing fog node1 whereas the user is moved the nearby fog node is selected to continue the execution without any deviation. For this, if we predict the mobility of the user in advance the concurrent action takes place without any delay [18].

## 4 Positing the coordinates (X and Y)

The coordinates pair (X, Y) is represented in two dimension position. The value x and y represents latitude and longitude. The latitude value lies on east-west and they are parallel to each other. If the node goes north, the latitude value is increased opposite direction decreasing the value. The range of the latitude lies between (−90 to +90) degrees are represented in Figure 4. The latitude value is represented by coordinate Y. The longitude values line north-south and horizontal to each other. The value of the longitude lines between (−180 to +180) degrees. The longitude value is represented by coordinate X. once the latitude and longitude values are located, we can find any location in the world [19].

To predict the future mobility the directions are divided into 8 regions and numbered as East = 1, …, Southeast = 8 (East, Northeast, North, West, Southwest, East, South, and Southeast), and each direction contains 45$^{o}$(0$^{o}$ to 360$^{o}$). Based on the user's existing location, the service-providing access point, migration zone, migration point, and region boundary is fixed, grounded on these values migration takes place while moving one location to another.

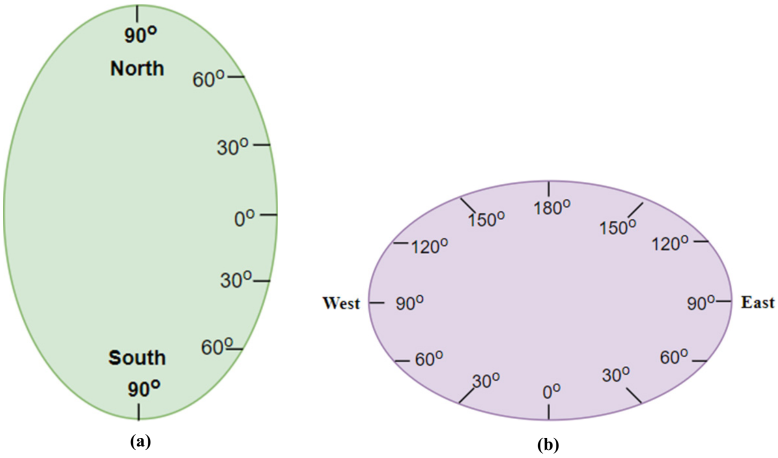


**Fig. 4.** Locating coordinates (a) coordinate x (b) coordinate Y

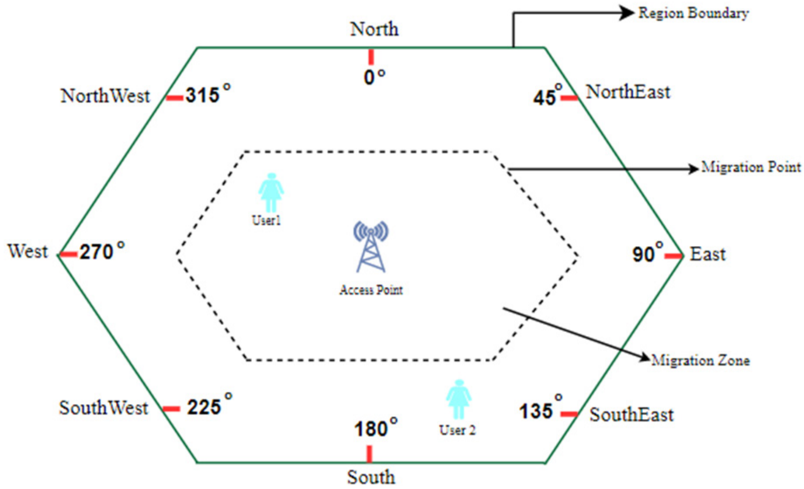


**Fig. 5.** Movement prediction model

As in Figure 5 the fog node monitors the user location continuously. Migration zone is a range where the migration decision is taking place, it returns TRUE or FALSE, once the user reaches the migration point it retunes TRUE. The migration point is placed in the region where the computed migration can be achieved. Once the user exits the migration point handoff takes place and performs the required action.

## 5 Dynamic mobility prediction method

To predict the upcoming location, need to define and evaluate basic information regarding the node status. The node status initialization is like coordinate values latitude (X), longitude (Y), movement speed(kmph), staying time, migration point, and migration zone. In Figure 6 the flow of location prediction is explained.

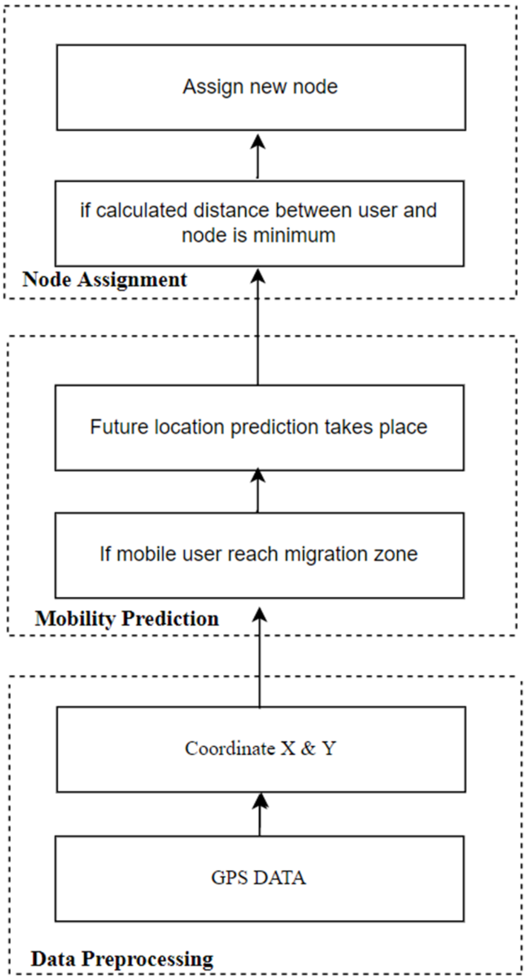


**Fig. 6.** Dynamic location prediction work flow

Staying time: Staying time is the difference between the end and start time of the mobile user entering into the new fog node and moving to another node. This time difference is used to know the how long user is staying in the node and monitor the network connectivity.

$$S_t = S_{end} - S_{start} \tag{1}$$

Here, St represents the starting time, Send represents the time the user moves to the next node, and Sstart denotes the time to enter the fog node.

Mobility path: A mobility path θ = [θ1, θ2, θ3, θ4, …, θn] represents user movement from one place to another place, user update their location in terms of θ for every second, using this θ value mobility of the user is identified while moving from one location to another location.

Migration Point: A migration point is a location on the map where it is appropriate to perform the computed migration. Before the handoff process happens, the Migration Stage should be initiated. Depending on infrastructure characteristics and the wireless link, the migration point can be set.

Migration Zone: The migration zone is an environment in which migration decisions are continuously computed. The zone of migration is the region inside the point of migration that is limited. Once a migration decision returns TRUE, it is only carried out when the user reaches the point of migration, which is any point at the location.

To compute the future location of the user assuming that each user updates their location for every θ when the user updates its location at time t in the form of (X, Y) the agreeing fog node uses the user's previous location with corresponding speed and direction. Using this information future location is predicted. After the location prediction then the user will find the closest fog node if the user predicted location is outside of the existing fog node.

$$\text{Users Existing Location: } l = t \tag{2}$$

$$\text{User Previous Location: } l = t - \theta \tag{3}$$

$$\text{Users Upcoming Location: } l = t + \theta \tag{4}$$

Based on the user's existing location, the future location is identified that is based on the θ value. The θ value is calculated using the user X and Y values. Using this X and Y value there are different possibilities to predict the future location

| ***Algorithm:* Before migration process** |
|---|
| **Input:** Location (*x,y*)<br>**Output:** Migration Decision<br>**If** mobile user moving<br>PredictFutureLocation()<br>**if**(*x!=0*)<br>Verify if the point is on Y-axis and must not do the y/0<br>(or)<br>θ is negative, but it is the first quadrant and it needs to be in the second quadrant<br>**else-if**(*x<0&&y>=0*)<br>θ is positive and it needs to be in the third quadrant<br>**else-if**(*x<0&&y<0*)<br>θ is negative and it needs to be in the fourth quadrant<br>**else-if**(*x>0&&y<0*)<br>θ is zero and it needs to be on Y (positive)<br>**else-if**(*x==0&y>0*)<br>θ is zero and it needs to be on y (negative)<br>**If** the distance between user and migration point is minimum<br>Make migration decision *True*<br>**else** *False*<br>**else** no migration |

If we find these coordinate values then the future location will be identified. After identifying the quadrant value, the distance between the two points is identified to locate the position most appropriately.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5}$$

By observing the position the basic values are mentioned in the form of radiant so it is converted radiant into a degree, given by

$$d = directions * (180 / \pi) \tag{6}$$

Using the CoordX and CoordY and the updated θ value, the current location of the user is identified, using these values the previous and the future values are calculated using the proposed formula. Once the mobility of the user is identified the future process of the migration of the data will take place to do the further work progress.

## 6 Experimentation description

This experiment was evaluated in the iFogSim simulator, with predefined input parameters listed in Table 2. The fog environment is created in the form of one cloud server, and each fog node has mobility users connected with them. Due to the wide range of users in the fog computing environment, each of them requires its identification like a requirement, characteristics, and architectural nature, and some user devices like smartwatches or IoT devices embedded in vehicles so require different mobility patterns like speed and their direction. To evaluate mobility prediction, the dataset is taken from the Luxembourg SUMO Traffic (LuST). In a more realistic form, the data sets are selected from 100 different buses mobility in the LuST. The average speed of moving buses at 22.3kmph, on average of 26.44min.In the mobility dataset the mobility parameters are in the form of time (in seconds), direction (in rad), position X and position Y, and speed (m/s). Example: 3.1 −1.68755 10369.2 2234.57 2.34286

**Table 2.** Simulation parameters

| Parameter | Value |
|---|---|
| Access point Coverage | 1000m |
| Fog Node Coverage | 1000m |
| Cloudlet Coverage | 1000m |
| Users Speed | 20kmph |
| Number of cloudlets per access point | 1:1 |
| Max_Handoff_Time | The 1200s |
| Min_Handoff_Time | 700s |
| Tota Prediction Count | 160 |
| Correct Prediction Count | 152 |
| Accuracy | 97.87% |

The mobility prediction of the user was evaluated using the GPS data of mobile users gathered in the LuST traffic. The fog environment is created with seven fog nodes with a radius of 500 meters. The proposed mobility prediction keeps monitoring, which taxi is entering or leaving the region in the given fog nodes. The future direction is predicted using the current location of the taxi.

## 7 Result

The migration of the user node is shown in Figure 7 with the seven fog nodes. Each fog node has an access point to cover the region 500m. Each fog node monitors the user while entering and leaving within the coverage point, once it exceeds the coverage the future location is predicted, and handover another access point to carry out the task without any delay in processing. In Figure 7 where the blue dot represents the current location of the user. The green line indicates the correction prediction and the red line indicate the incorrect prediction.

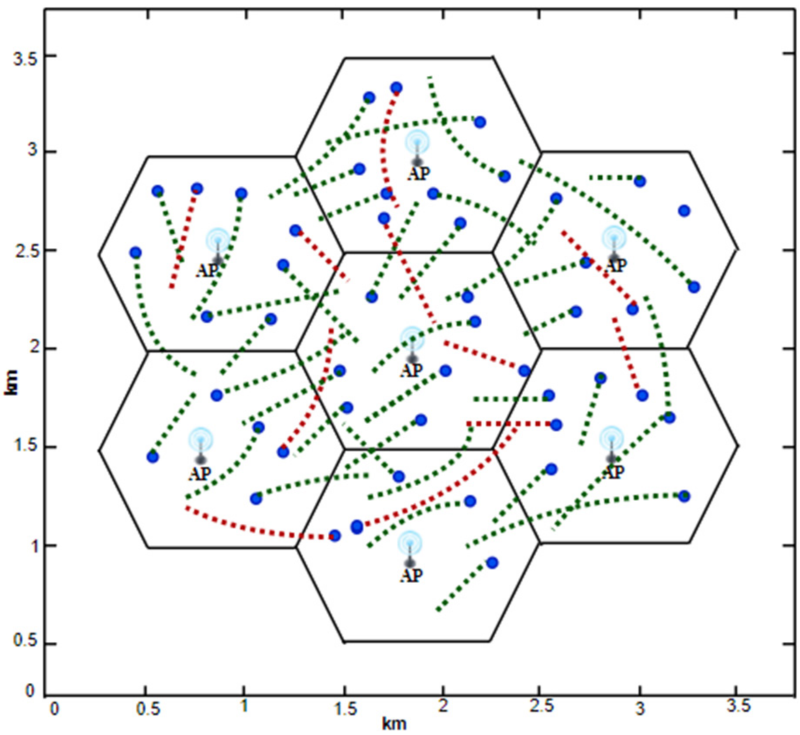


**Fig. 7.** Location prediction

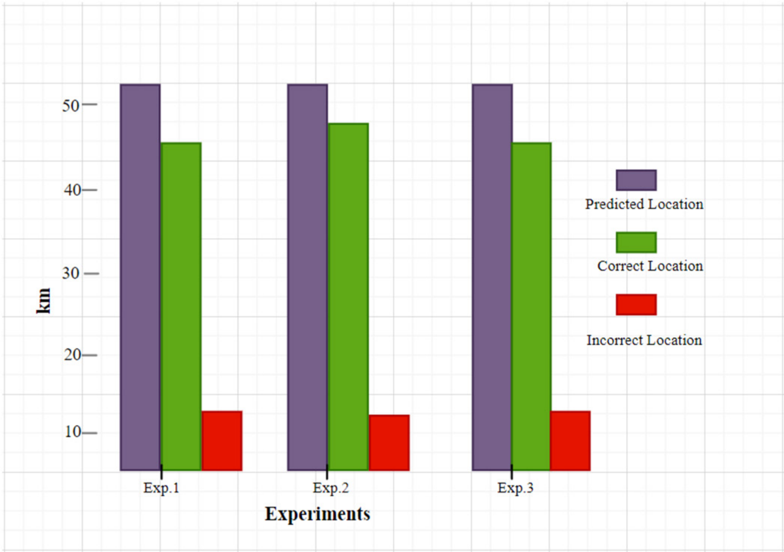


**Fig. 8.** Experiment comparison with different speed

Due to the wide range of users in the fog computing environment, aiming to increase the covered scenario of the mobility prediction of the mobile users. Initially, the experiment was tested by selecting 100 buses mobility patterns from LuST. In Figure 8 Experiment 1 evaluated on the average at 22kmph, the mobility prediction is evaluated. By keeping the same route was evaluated earlier the user speed of the vehicle is increased as 44kmph was done in experiment 2. In experiment 3 again it's doubled to increase the speed of the vehicle to 66kmph to find the migration of nodes among the different fog nodes. By observing all three experiments future prediction accuracy got up to 98.87% for different speeds of the mobile users. Once the user's future location is identified, the user experience is improved in the fog computing environment.

**Table 3.** Comparison of results

| S.No. | Mobility Model | Total Prediction Count | Prediction Count Accuracy |
|---|---|---|---|
| 1 | Linear Mobility Prediction | 500 | 96.54% |
| 2 | Dynamic Pattern Tree | 250 | 93.24% |
| 3 | Tessellation Model | 300 | 94.56% |
| 4 | Random Walk Model | 450 | 92.34% |
| 5 | Blockchain Mobility model | 200 | 96.89% |
| 6 | **Dynamic Mobility Prediction** | **Random** | **98.87%** |

In Table 3 dynamic mobility prediction model accuracy is compared with existing models Linear mobility prediction [20], Dynamic pattern tree [21], Tessellation model [22], Random walk model [23], and blockchain mobility prediction [24], by this comparison proposed method provide 98.87% accuracy. Mobility prediction improves the efficiency when the user moves from one location to another without any delay the computation is carried by a nearby fog node. dynamic mobility prediction can be implemented in the real time applications like mobile phone use in class room [25] mobile learning [26], micro teaching video resources [27].

## 8 Conclusion

Due to the rapid advancement in IoT devices, the amount of data generation is very high. Computing this large amount of data in cloud computing increase the latency problem. Mobility in fog computing also increases the latency when roaming from one location to another location. The proposed dynamic mobility prediction method generates the future location of the mobile nodes in advance. This prediction improves the QoS and user experience. This prediction is compared with existing models, it gives a 97.87% accurate value of the user location. In the future, mobility-based resource provisioning like load balancing, scheduling, and offloading can be performed with the secure authentication model.

## 9 References

[1] Chen, Yu-An, John Paul Walters, and Stephen P. Crago. "Load balancing for minimizing deadline misses and total runtime for connected car systems in fog computing." In 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), pp. 683–690. IEEE, 2017. https://doi.org/10.1109/ISPA/IUCC.2017.00107

[2] Pirozmand, Poria, Guowei Wu, Behrouz Jedari, and Feng Xia. "Human mobility in opportunistic networks: Characteristics, models and prediction methods." Journal of Network and Computer Applications 42 (2014): 45–58. https://doi.org/10.1016/j.jnca.2014.03.007

[3] Waqas, Muhammad, Yong Niu, Manzoor Ahmed, Yong Li, Depeng Jin, and Zhu Han. "Mobility-aware fog computing in dynamic environments: Understandings and implementation." IEEE Access 7 (2018): 38867–38879. https://doi.org/10.1109/ACCESS.2018.2883662

[4] Wang, Zi, Zhiwei Zhao, Geyong Min, Xinyuan Huang, Qiang Ni, and Rong Wang. "User mobility aware task assignment for mobile edge computing." Future Generation Computer Systems 85 (2018): 1–8. https://doi.org/10.1016/j.future.2018.02.014

[5] Maleki, Erfan Farhangi, and Lena Mashayekhy. "Mobility-aware computation offloading in edge computing using prediction." In 2020 IEEE 4th International Conference on Fog and Edge Computing (ICFEC), pp. 69–74. IEEE, 2020. https://doi.org/10.1109/ICFEC50348.2020.00015

[6] Wu, Xianjing, Shengjie Zhao, Rongqing Zhang, and Liuqing Yang. "Mobility prediction-based joint task assignment and resource allocation in vehicular fog computing." In 2020 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6. IEEE, 2020. https://doi.org/10.1109/WCNC45663.2020.9120524

[7] Fan, Jianhua, Jianwei Liu, Jin Chen, and Jing Yang. "LPDC: mobility-and deadline-aware task scheduling in tiered IoT." In 2018 IEEE 4th International Conference on Computer and Communications (ICCC), pp. 857–863. IEEE, 2018. https://doi.org/10.1109/CompComm.2018.8780904

[8] Dou, Wanchun, Wenda Tang, Bowen Liu, Xiaolong Xu, and Qiang Ni. "Blockchain-based mobility-aware offloading mechanism for fog computing services." Computer Communications 164 (2020): 261–273. https://doi.org/10.1016/j.comcom.2020.10.007

[9] Roig, Pedro Juan, Salvador Alcaraz, Katja Gilly, and Carlos Juiz. "Study on mobility and migration in a fog computing environment." In 2018 22nd International Conference Electronics, pp. 1–6. IEEE, 2018. https://doi.org/10.1109/ELECTRONICS.2018.8443636
[10] Shekhar, Shashank, Ajay Chhokra, Hongyang Sun, Aniruddha Gokhale, Abhishek Dubey, and Xenofon Koutsoukos. "Urmila: A performance and mobility-aware fog/edge resource management middleware." In 2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC), pp. 118–125. IEEE, 2019. https://doi.org/10.1109/ISORC.2019.00033
[11] Salvador, Paulo, and António Nogueira. "Markov modulated bi-variate Gaussian processes for mobility modeling and location prediction." In International Conference on Research in Networking, pp. 227–240. Springer, Berlin, Heidelberg, 2011. https://doi.org/10.1007/978-3-642-20757-0_18
[12] Puliafito, Carlo, Diogo M. Gonçalves, Márcio M. Lopes, Leonardo L. Martins, Edmundo Madeira, Enzo Mingozzi, Omer Rana, and Luiz F. Bittencourt. "MobFogSim: Simulation of mobility and migration for fog computing." Simulation Modelling Practice and Theory 101 (2020): 102062. https://doi.org/10.1016/j.simpat.2019.102062
[13] Wang, Dongyu, Zhaolin Liu, Xiaoxiang Wang, and Yanwen Lan. "Mobility-aware task offloading and migration schemes in fog computing networks." IEEE Access 7 (2019): 43356–43368. https://doi.org/10.1109/ACCESS.2019.2908263
[14] Gia, Tuan Nguyen, Amir M. Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. "Fog computing approach for mobility support in internet-of-things systems." IEEE Access 6 (2018): 36064–36082. https://doi.org/10.1109/ACCESS.2018.2848119
[15] Buccafurri, Franscesco, Gianluca Lax, and Antonia Russo. "Exploiting digital identity for mobility in fog computing." In 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), pp. 155–160. IEEE, 2019. https://doi.org/10.1109/FMEC.2019.8795342
[16] Rahmani, A. M., Gia, T. N., Negash, B., Anzanpour, A., Azimi, I., Jiang, M., and Liljeberg, P. "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach." Future Generation Computer Systems, 78 (2018): 641–658. https://doi.org/10.1016/j.future.2017.02.014
[17] Song, Chaoming, Tal Koren, Pu Wang, and Albert-László Barabási. "Modelling the scaling properties of human mobility." Nature Physics 6, no. 10 (2010): 818–823. https://doi.org/10.1038/nphys1760
[18] Djemai, Tanissia, Patricia Stolf, Thierry Monteil, and Jean-Marc Pierson. "Mobility support for energy and QoS aware IoT services placement in the fog." In 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–7. IEEE, 2020. https://doi.org/10.23919/SoftCOM50211.2020.9238236
[19] Laurila, Juha K., Daniel Gatica-Perez, Imad Aad, Olivier Bornet, Trinh-Minh-Tri Do, Olivier Dousse, Julien Eberle, and Markus Miettinen. The mobile data challenge: Big data for mobile computing research. No. CONF. 2012. https://doi.org/10.1016/j.pmcj.2013.07.014
[20] Lera, Isaac, Carlos Guerrero, and Carlos Juiz. "YAFS: A simulator for IoT scenarios in fog computing." IEEE Access 7 (2019): 91745–91758. https://doi.org/10.1109/ACCESS.2019.2927895
[21] Huang, Cheng, Rongxing Lu, and Kim-Kwang Raymond Choo. "Vehicular fog computing: architecture, use case, and security and forensic challenges." IEEE Communications Magazine 55, no. 11 (2017): 105–111. https://doi.org/10.1109/MCOM.2017.1700322
[22] Wang, Tian, Lei Qiu, Arun Kumar Sangaiah, Guangquan Xu, and Anfeng Liu. "Energy-efficient and trustworthy data collection protocol based on mobile fog computing in Internet of Things." IEEE Transactions on Industrial Informatics 16, no. 5 (2019): 3531–3539. https://doi.org/10.1109/TII.2019.2920277

[23] Lin, Fuhong, Lei Yang, Ke Xiong, and Xiaowen Gong. "Recent advances in cloud-aware mobile fog computing." Wireless Communications and Mobile Computing 2019 (2019). https://doi.org/10.1155/2019/8204394
[24] Yadav, Rahul, Weizhe Zhang, Omprakash Kaiwartya, Houbing Song, and Shui Yu. "Energy-latency tradeoff for dynamic computation offloading in vehicular fog computing." IEEE Transactions on Vehicular Technology 69, no. 12 (2020): 14198–14211. https://doi.org/10.1109/TVT.2020.3040596
[25] Diego Calderón-Garrido and Francisco Javier Ramos-Pardo. "The use of mobile phones in classrooms: A systematic review." International Journal of Emerging Technologies in Learning 17, no.06 (2022): 194–210. https://doi.org/10.3991/ijet.v17i06.29181
[26] Ling Han and Jing Zhao. "Optimal allocation of mobile learning resources based on a complex network." International Journal of Emerging Technologies in Learning 17, no. 06 (2022): 211–225. https://doi.org/10.3991/ijet.v17i06.30017
[27] Jie Liu and Haiping Lv. "Recommendation of micro teaching video resources based on topic mining and sentiment analysis." International Journal of Emerging Technologies in Learning 17, no. 06 (2022): 243–256. https://doi.org/10.3991/ijet.v17i06.30011

## 10 Authors

**Deepa D** is pursuing her doctoral studies at the school of computer science and engineering of vellore institute of technology, vellore, India. The area of research is cloud and fog computing. (email: deepadevendiran@outlook.com).

**Jothi K R** is currently working as Associate Professor in the School of Computer Science and Engineering, Vellore Institute of Technology, Vellore. He has more than 20 years of teaching experience in various engineering institutions and 10 years of research experience. His research area includes Blockchain, Computational Intelligence, Machine Learning, Deep Learning, Soft Computing, Energy Aware Computing, Vehicular networks, Cloud Computing, Educational Technology and Internet of Things. He is an active Senior Member of IEEE, ACM and other professional societies. (email: jothi.kr@vit.ac.in; orchid id: 0000-0003-0106-2804).

Article submitted 2022-04-07. Resubmitted 2022-05-16. Final acceptance 2022-05-16. Final version published as submitted by the authors.