JOURNAL OF INFORMATION PROCESSING SYSTEMS **J**^{IP}**S**

Task Assignment Model for Crowdsourcing Software Development: TAM

Muhammad Zahid Tunio*, Haiyong Luo**, Cong Wang*, Fang Zhao*,
Abdul Rehman Gilal***, and Wenhua Shao*

Abstract

Selection of a suitable task from the extensively available large set of tasks is an intricate job for the developers in crowdsourcing software development (CSD). Besides, it is also a tiring and a time-consuming job for the platform to evaluate thousands of tasks submitted by developers. Previous studies stated that managerial and technical aspects have prime importance in bringing success for software development projects, however, these two aspects can be more effective and conducive if combined with human aspects. The main purpose of this paper is to present a conceptual framework for task assignment model for future research on the basis of personality types, that will provide a basic structure for CSD workers to find suitable tasks and also a platform to assign the task directly. This will also match their personality and task. Because personality is an internal force which whittles the behavior of developers. Consequently, this research presented a Task Assignment Model (TAM) from a developers point of view, moreover, it will also provide an opportunity to the platform to assign a task to CSD workers according to their personality types directly.

1. Introduction

Crowdsourcing has become an important factor for the quick development having the advantage of a short schedule due to the parallel and micro-tasking. It also offers low cost based on the knowledge of the crowd or "wisdom of the crowd". Crowdsource software development (CSD) is taken from the concept of crowdsourcing. It, usually, uses an open call online format to catch a large number of CSD workers to participate in different types of software project tasks; for instance, component development, designing, architecture, testing and debugging. Moreover, there are three types of models used in crowdsourcing software development: peer production, competition, and micro-tasking [1]. Many platforms are available to implement these models in different aspects of peer production open source projects like Linux, Apache, Firefox are used. Topcoder, Get a coder, Zhubajie, Utest, Guru, Freelance, Elance and Amazon Mechanical Tuck are few examples of competition and micro-tasking models, respectively.

CSD uses an open call format. This process involves three kinds of roles: requester (for the one the project is undertaken), platform (the service provider), and crowd-source developer (the person for coding and testing). However, this type of call format enables a large number of task accessibility and self-selection. On the platform, a number of developers can register and choose a task from available set. Once after the submission of the task from developers, the platform personnel has to evaluate the submission to decide on the best solution, from developers, to pay the rewards. Mao et al. [2,3] state that selection of a task from the extensive large set of tasks is a very hectic work for the developer. Besides, it is also a tiring and a time-consuming job for the platform to evaluate thousands of submitted tasks from developers. In the same view, Chilton et al. [4] and Aldhahri et al. [5] also mentioned that matching of their proper task to improper crowdsourcing software developer may not only decrease the quality of the software deliverables, but it causes overburden on both platform and developers. They further mentioned that most workers view a minimum amount of recent tasks which are posted on the crowdsourcing platform because the task is posted in hundreds. By considering the low level of skills and expertise level of the crowdsource software developers, unrealistic matching of CSD worker and the task may affect the software quality. LaToza and Van der Hoek [6] mentioned in his research that matching of workers with their expertise and knowledge and how we can benefit from the CSD worker is an issue. Geiger and Schader [7] described that by matching with extrinsic and intrinsic choice of the CSD worker, it is a fundamental self-identification principle for individual contributors to select those tasks which are the best match with their personal preferences. They further illustrated that for the CSD worker, it is a very important factor to comply with the choice and individuals' capabilities with the respective task requirements. Dang et al. [8] depicted that, for one task, there is a large number of CSD workers for participation and submission. Therefore, selection of a final submission is a hectic job for them. They further stated that every CSD worker is not qualified enough to give the best solution because of different hardware and software skills. In addition, CSD may have malicious workers in crowdsourcing platform for submitting tasks [9,10]. Therefore, nowadays finding a quality worker is a challenging task for the CSD model [11]. Machado et al. [12] stated that CSD model is very complex as it deals with technological, economic as well as personal issues. Therefore, it is very important to select appropriate worker among the large set of CSD workers. They further discussed that to assign a right task to the right CSD worker at a right time is indispensable and challenging for the crowdsourcing platform for their success. Moreover, Fu et al. [13] also stated that the selection of suitable CSD workers is extremely tiring and time-consuming job for the platform. An appropriate CSD task-worker will be beneficial for the platform to participate but also to perform the task with the best interest and better quality.

According to Tomayko and Hazzan [14], the key complexities pertinent to the development of software are concerned with human aspects from their social and cognitive point of view. Thus, human aspects should be overemphasized to cope up with the challenges while developing projects under the umbrella of software engineering. Though both technological and managerial aspects also have prime importance to bring success for software development projects, yet these two aspects can be more effective and conducive if combined together with human aspects. The combination of all three aspects can resolve challenges related to software development.

The aim of this study is to present a conceptual framework for task assignment model for future research on the basis of personality types. This will give a right path for CSD workers to find suitable tasks and also a platform to assign the task directly to the CSD workers which match with the personality and tasks. Since, personality is an internal force which carves the behavior of developers.

Therefore, personality behavior can lead the model for suggesting or shortlisting the task set for developers. Therefore, this model can create a path to bring suitable developing personnel. It can also help the platform to select from limited tasks to evaluate. Based on the authors' knowledge, there is not a single research study available which has merged personality types with crowdsourcing software development.

2. Related Work

Mao et al. [2] employed the content-based technique to recommend developers for CSD tasks. This approach learns from historical task registration and winner records automatically that match tasks and developers. Snow et al. [15] proposed bias correction in crowd data in the form of modeling. They used a gold standard data set to estimate the CSD workers' model accuracy. However, this method is used in micro-tasking. Ambati et al. [16] used an implicit modeling based on skills and interest of CSD workers to recommend the classification based task. Yuen et al. [17] proposed an approach based on task matching to motivate CSD workers to perform task continually and in long run. This approach has focused on the recommendation of the tasks matching best with the workers. From this approach, we got inspired and motivated, therefore, will provide a new approach to the task matching problem for the best possible solution based on personality types that will be a novel approach. Sheng et al. [18] stated that, for a task matching, labeling is used as a technique but it also evident certain limitations. Liu et al. [19], Whitehill et al. [20], and Raykar et al. [21] used EM algorithm and Answer matrix to calculate the accuracy and mapping the quality of the CSD worker. Determination of single labeling is focused on these studies [22-25]. According to the author in the paper [13], by ignoring the task requirements and the relationship between CSD workers' skills these approaches may get undesired results. Therefore, a new approach is needed to relate the soft skills with the hard skills of the CSD workers.

In order to avoid the risks of giving the task to the improper personality types of CSD, Capretz and Ahmed [26] have proposed a suitable model. According to this model, they suggested that tasks assigned to developers must be based on their suitable personality types. For instance, the personality of a programmer should be Introvert (I), the personality of system analyst should be Extrovert (E) the tester has a sensing (S) and Thinking (T) personality. The software designer should be with intuitive (N) and think (T) personality. However, due to its non-empirical nature of the model, the effectiveness of this model is difficult to test. Hence, in CSD, we aim to propose an approach for assigning the task to the developers: testers, debugger, and coders according to their personality types. Because the personality types of the developer are one of the important human aspects to ensure quality for software tasks. It has also been confirmed that a technical sound individual cannot perform satisfactorily unless he/she is assigned development tasks based on their personality types. This raises new challenges for crowdsourcing tasks and it requires an in-depth understanding of matching of workers and their characteristics related to their work [27]. An individual performance in software development has co-relation and has direct interaction with the personality of CSD workers [28]. Capretz et al. [29] mentioned that assigning the task to the specific personality in software development is best suited for their traits to increase the successful outcome of the tasks.

Hence to integrate the CSD with the developer's personality and their interrelationship is required. This area of research is yet to be discovered and needs more work to be done.

2.1 Current CSD Model

The current model of CSD is working on open call format. As shown in Fig. 1, initially the requester has requested to the platform for solving their problem, secondly the platform has sorted out problems and made in micro-tasking then the tasks are posted on the platform. Here the role of requester has been completed and platform role has to begin now the CSD developers have to be registered for themselves to participate in the various tasks. They are required to submit their solutions to the platform and the platform has to review the submission. After reviewing the submissions, platform decides that the submission was according to their criteria or not. Moreover, developers also have the right of appeal against the rejection of their submission. In the end, the reward is given to the winner of submission.
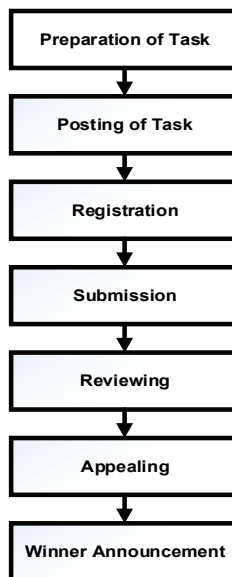

Fig. 1. Crowdsourced software development process.

3. Proposed Model for CSD

The proposed approach will work the same as the open call format. But, the proposed task assignment CSD model includes the personality based categorization and selection of task (see Fig. 2). Once after the requester prepared a request for the task, the platform shortlists the posts of tasks for competition based on personality types. In this case, the registration of developer requires the personality measurement test to know the type of personality of developers. Meanwhile, this proposal also suggests that the task should also be included with a specific explanation which can be used to define the required traits of personality for the task. For example, the social networking based tasks may require a developer with the extrovert trait. Since the extrovert developer can understand and work on the task with interest as they involve themselves in social activities. Hence, the task will be directly available for the developers if the developers are already registered with the platform along with their personality types (Myer Briggs Type Indicator [MBTI]). The classification of individual personality types are classified on the MBTI test which follows with the combination of four-dimensional pairs, as shown in Table 1, and from that four combinations, there are 16 possible personality combinations

which are shown in Table 2. To evaluate a personality of the CSD worker, this study, will use MBTI personality type as an instrument as this instrument is widely used in the research of software engineering [30-35].

Table 1. Four dimensions of MBTI personality

(E) Extroversion	(I) Introversion
(S) Sensing	(N) Intuition
(F) Feeling	(T) Thinking
(P) Perceiving	(J) Judging

Table 2. MBTI 16 personality types with pair of 4 each

ISTJ	ISFJ	INFJ	INTJ
ISTP	ISFP	INFP	INTP
ESTP	ESFP	ENFP	ENTP
ESTJ	ESFJ	ENFJ	ENTJ

3.1 The Detailed Process of Proposed Model

(1) When requester plans to post a task, the categories of the task should be predefined on the CSD platform. When any task is submitted by a requester, the task will be placed in the set of categorized task (for example, developing, debugging, and testing). When the requester wants to publish the task on a CSD platform, the requester also has the description of tasks. For instance, the category of the task removes a bug from the code in Python is categorized as shown in Table 3.

Table 3. Task categories and personality type

Task description	To remove a bug from a code for a designing program
Category of task	Designing/debugging
Personality type	ENFJ (Python), ISTJ (Java)

(2) The CSD platform has a database of a CSD worker who has already registered and participated. The CSD worker participation record has been kept in the database, based on choosing a preference and task submission along with personality types (Here we assume that the platform has already registered participants along with personality type). However, when platforms post tasks according to the categories of the tasks and a new CSD worker, wants to participate, they also have the option to choose the task according to their preferred personality types. If they are not registered before, they have the option to register themselves and submit the solutions. The task selection choice is calculated on basis of the information of the workers and the task which is selected previously. The information consists of the category of task, rewards and personality type.

(3) When logging on to the CSD platform, the list of tasks is available for the CSD worker that match best with his personality and task matching. From that list, a CSD worker will select his preferred task to work on. As we have discussed above that most of the workers only browsed the few tasks when searching tasks.

(4) The list of tasks available for the CSD worker on the platform will be of great importance to attract a large number of CSD workers to match their task with their preferred personality type and interesting category of tasks easily and quickly.

Mathematically we can define the proposed model as:

Let T = {t1, t2, …} is a set of tasks and W = {w1, w2, …} is a set of CSD workers, and P = {p1, p2, …} is a set of personality types. We can represent the personality type which is required for a task t by a node Personality(t) ∈ P, and the set of personalities of the CSD worker w by Personality(w) ∈ P. by assigning a set of tasks and CSD workers, a task assignment set U is a mapping from T to w which will mapped task t ∈ T to U(t) = w ∈ W. The best situation is to map a task with required personality P to a worker with his exact personality type.
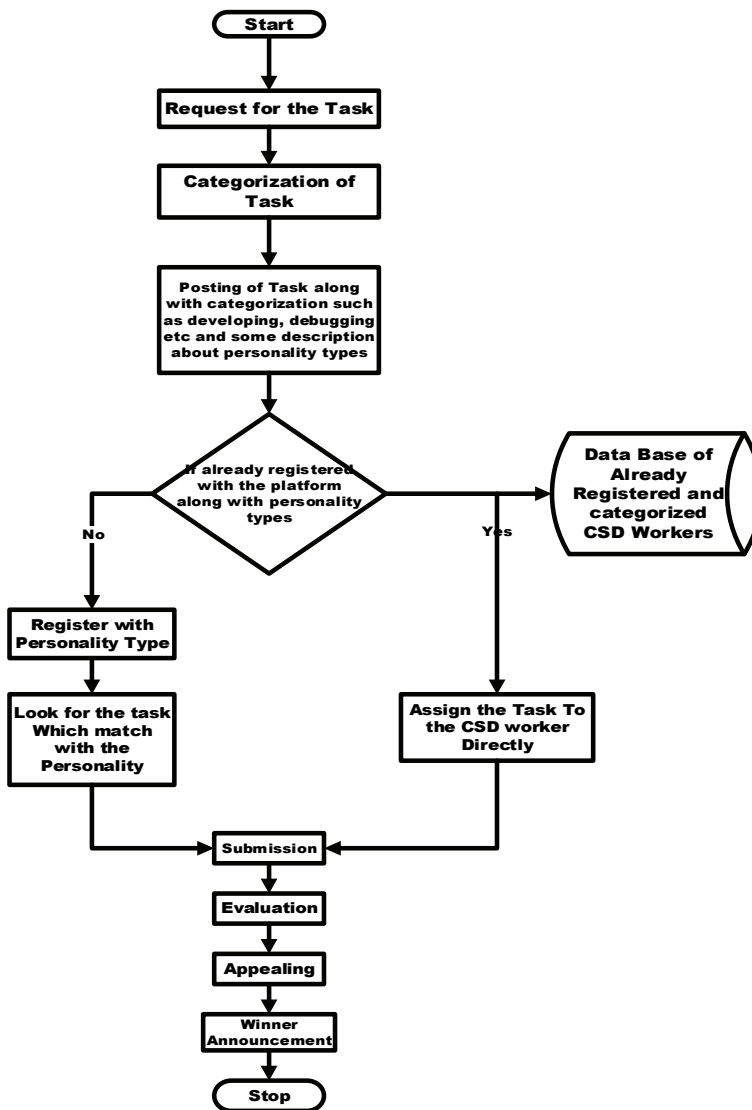
Fig. 2 displays the proposed model based on personality.

Fig. 2. Proposed model of crowdsourcing software development.

4. Contribution

The contribution of this study will be taken into account from many ways. Firstly, this study would benefit and suggest CSD platform to classify the tasks, for both platform and developers, on personality based types. Secondly, the results emanating from this study will not only help the CSD but it will also ensure the satisfaction of CSD workers for choosing the task according to their personality types. This plays a key role for workers in the selection of a suitable task and it has a correlation with the quality of the results. Thirdly, the results of this research will produce a Task Assignment Model (TAM) for CSD not only from CSD workers' point of view but it will also provide an opportunity for the platform to assign or shortlist the tasks to CSD worker according to their personality types directly. This will be very helpful in saving the time spending on searching on the platform for their preferred task. Moreover, it will also give enormous relief to the platform to reduce their burden for analyzing the thousands of daily submissions by CSD workers.

5. Conclusions

The focus of this paper is to propose a model in the CSD domain for task assignment on the basis of developers' personality. This model will help developers for finding suitable tasks and also the platform to assign the task directly to the CSD workers that match with the personality and task. This will help the developers to choose or register for those tasks that are suitable for them according to their personality type. It also facilitates the platform to assign a task to the suitable developer based on the developers' personality type. This approach will provide a path to the quality development with less effort. In future, we want to design an algorithm based on this model and some experiments on platform data to validate our model for its functionality and generalization.

Acknowledgement

References

[1] T. D. LaToza and A. Van der Hoek, "Crowdsourcing in software engineering: models, motivations, and challenges," *IEEE Software*, vol. 33, no. 1, pp. 74-80, 2016.

[2] K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," *Journal of Systems and Software*, vol. 126, pp. 57-84, 2017.

[3] K. Mao, Y. Yang, Q. Wang, Y. Jia, and M. Harman, "Developer recommendation for crowdsourced software development tasks," in *Proceedings of 2015 IEEE Symposium on Service-Oriented System Engineering*, San Francisco Bay, CA, 2015, pp. 347-356.

[4] L. B. Chilton, J. J. Horton, R. C. Miller, and S. Azenkot, "Task search in a human computation market," in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, Washington, DC, 2010, pp. 1-9.

[5] E. Aldhahri, V. Shandilya, and S. Shiva, "Towards an effective crowdsourcing recommendation system: a survey of the state-of-the-art," in *Proceedings of 2015 IEEE Symposium on Service-Oriented System Engineering*, San Francisco Bay, CA, 2015, pp. 372-377.

[6] T. D. LaToza and A. Van der Hoek, "A vision of crowd development," in *Proceedings of 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Florence, Italy, 2015, pp. 563-566.

[7] D. Geiger and M. Schader, "Personalized task recommendation in crowdsourcing information systems: current state of the art," *Decision Support Systems*, vol. 65, pp. 3-16, 2014.

[8] D. Dang, Y. Liu, X. Zhang, and S. Huang, "A crowdsourcing worker quality evaluation algorithm on MapReduce for big data applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 7, pp. 1879-1888, 2016.

[9] B. Carpenter, "Multilevel Bayesian models of categorical data annotation," 2008 [Online]. Available: https://lingpipe.files.wordpress.com/2008/11/carp-bayesian-multilevel-annotation.pdf.

[10] A. Brew, D. Greene, and P. Cunningham, "Using crowdsourcing and active learning to track sentiment in online media," in *Proceedings of the 19th European Conference on Artificial Intelligence*, Lisbon, Portugal, 2010, pp. 145-150.

[11] J. Howe, "The rise of crowdsourcing," *Wired Magazine*, vol. 14, no. 6, pp. 1-4, 2006.

[12] L. Machado, R. Prikladnicki, F. Meneguzzi, C. R. de Souza, and E. Carmel, "Task allocation for crowdsourcing using AI planning," in *Proceedings of the 3rd International Workshop on CrowdSourcing in Software Engineering*, Austin, TX, 2016, pp. 36-40.

[13] Y. Fu, H. Chen, and F. Song, "STWM: a solution to self-adaptive task-worker matching in software crowdsourcing," in *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, Zhangjiajie, China, 2015, pp. 383-398.

[14] J. E. Tomayko and O. Hazzan, *Human Aspects of Software Engineering*. Hingham, MA : Charles River Media, 2004.

[15] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast: but is it good? Evaluating non-expert annotations for natural language tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Honolulu, HI, 2008, pp. 254-263.

[16] V. Ambati, S. Vogel, and J. G. Carbonell, "Towards task recommendation in micro-task markets," *Human Computation*, vol. 11, pp. 1-4, 2011.

[17] M. C. Yuen, I. King, and K. S. Leung, "Task matching in crowdsourcing," in *Proceedings of 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, Dalian, China, 2011, pp. 409-412.

[18] V. S. Sheng, F. Provost, P. G. Ipeirotis, "Get another label? Improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, NV, 2008, pp. 614-622.

[19] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang, "CDAS: a crowdsourcing data analytics system," *Proceedings of the VLDB Endowment*, vol. 5, no. 10, pp. 1040-1051, 2012.

[20] J. Whitehill, T. F. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: optimal integration of labels from labelers of unknown expertise," *Advances in Neural Information Processing Systems*, vol. 22, pp. 2035-2043, 2009.

[21] V. C. Raykar, S. Yu, L. H. Zhao, A. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy, "Supervised learning from multiple experts: whom to trust when everyone lies a bit," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 889-896.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society Series B (Methodological)*, vol. 39, no. 1, pp. 1-38, 1977.

[23] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 20-28, 1979.

[24] M. R. Gupta and Y. Chen, "Theory and use of the EM algorithm," *Foundations and Trends in Signal Processing*, vol. 4, no. 3, pp. 223-296, 2011.

[25] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, 2nd ed. Hoboken, NJ: John Wiley & Sons, 2007.

[26] L. F. Capretz and F. Ahmed, "Making sense of software development and personality types," *IT Professional*, vol. 12, no. 1, pp. 6-13, 2010.

[27] G. Kazai, J. Kamps, and N. Milic-Frayling, "Worker types and personality traits in crowdsourcing relevance labels," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, Glasgow, Scotland, 2011, pp. 1941-1944.

[28] C. M. Karapicak and O. Demirors, "A case study on the need to consider personality types for software team formation," in *Proceedings of the International Conference on Software Process Improvement and Capability Determination*, Bremen, Germany, 2013, pp. 120-129.

[29] L. F. Capretz, D. Varona, and A. Raza, "Influence of personality types in software tasks choices," *Computers in Human Behavior*, vol. 52, pp. 373-378, 2015.

[30] S. Cruz, F. Q. da Silva, and L. F. Capretz, "Forty years of research on personality in software engineering: a mapping study," *Computers in Human Behavior*, vol. 46, pp. 94-113, 2015.

[31] R. Valencia-Garcıa, F. Garcia-Sanchez, D. Castellanos-Nieves, J. T. Fernandez-Breis, and A. Toval, "Exploitation of social semantic technology for software development team configuration," *IET Software*, vol. 4, no. 6, pp. 373-385, 2010.

[32] N. R. Mead, "Software engineering education: how far we've come and how far we have to go," *Journal of Systems and Software*, vol. 82, no. 4, pp. 571-575, 2009.

[33] A. R. Gilal, J. Jaafar, M. Omar, S. Basri, and A. Waqas, "A rule-based model for software development team composition: team leader role with personality types and gender classification," *Information and Software Technology*, vol. 74, pp. 105-113, 2016.

[34] M. Z. Tunio, H. Luo, W. Cong, Z. Fang, A. R. Gilal, A. Abro, and S. Wenhua, "Impact of personality on task selection in crowdsourcing software development: a sorting approach," *IEEE Access*, vol. 5, pp. 18287-18294, 2017.

[35] M. Z. Tunio, H. Luo, C. Wang and F. Zhao, "Crowdsourcing software development: task assignment using PDDL artificial intelligence planning," *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 129-139, 2018.

Muhammad Zahid Tunio https://orcid.org/0000-0002-9406-0586

He received his B.E. in Computer Systems Engineering and Masters of Engineering in Information Technology from Mehran University of Engineering and Technology, Jamshoro, Pakistan, in 2000 and 2010, respectively. He has 15 years experience in teaching at graduate and Post graduate level. currently, he is a Ph.D. scholar at School of Software Engineering, Beijing University of Post and Telecommunication, China. He is also Assistant Profesor in Department of Computer Systems Engineering at Dawood University of Engineering and Technology, Karachi, Pakistan. His research interests are crowdsource software development, pervasive devices, and mobile computing.

Haiyong Luo https://orcid.org/0000-0001-6827-4225

He received the B.S. degree in the Department of Electronics and Information Engineering from Huazhong University of Science and Technology, Wuhan, China in 1989, M.S. degree in School of Information and Communication Engineering from the Beijing University of Posts and Telecommunication, China in 2002, and Ph.D. degree in Computer Science from the University of Chinese Academy of Sciences, Beijing, China in 2008. Currently, he is Associate Professor at the Institute of Computer Technology, Chinese Academy of Science (ICT-CAS) China. His main research interests are location-based services, pervasive computing, mobile computing, and the Internet of Things.

Cong Wang https://orcid.org/0000-0002-2504-4056

She received the Ph.D. degree in Control Theory and Engineering from the University of Science Technology, Beijing, China in 2002. She was the Vice Director in charge of pattern recognition and intelligent systems between 2002 and 2011 with the Intelligent Science and Technology Research Center, Beijing University of Posts and Telecommunication. Currently, she is a Professor with the School of Software Engineering, Beijing University of Post and Telecommunications. She is also the vice director of the Key Laboratory of Trustworthy Distributed Computing and Services, Ministry of Education. Her research interests include control theory, Internet of Things, and industrial systems engineering.

Fang Zhao https://orcid.org/0000-0002-4784-5778

She received the B.S. degree in the School of Computer Science and Technology from Huazhong University of Science and Technology, Wuhan, China in 1990, M.S. and Ph.D. degrees in Computer Science and Technology from Beijing University of Posts and Telecommunication, Beijing, China in 2004 and 2009, respectively. She is currently Professor in School of Software Engineering, Beijing University of Posts and Telecommunication. Her research interests include mobile computing, location-based services, and computer networks.

Abdul Rehman Gilal https://orcid.org/0000-0002-1904-1588

He is a faculty member of Computer Science department at Sukkur IBA University, Pakistan. He has earned a Ph.D. in Information Technology from University Teknologi Petronas (UTP), Malaysia. He has been researching in the field of software project management for finding the effective methods of composing software development teams. Based on his research publication track record, he has contributed in the areas of the human factor in software development, complex networks, databases and data mining, programming and cloud computing.

Wenhua Shao https://orcid.org/0000-0002-2440-9981

He is currently pursuing the Ph.D. degree with the School of Software Engineering, Beijing University of Posts and Telecommunications, China. His, current main interests include location-based services, pervasive computing, and convolution neural networks and machine learning.