

Mobile Phone Maintainability Prediction using MCDM Methodology

Parita Jain, Puneet Kumar Aggarwal

Abstract: For software engineering it is essential to build a software that is qualitative from every perspective. As the interest of the clients in mobile applications getting higher and higher day by day, the quality turns out to be a major concern. Ample number of researches shows issues with the maintenance of the applications once they are public. Different studies showed various evaluation methods for estimating maintainability but only few were defined for mobile applications. Also, none of the approaches propounded gives a specified way for evaluating maintainability of these portable applications. The present research work, examines the key idea of maintenance of apps considering 9 vital factors for the estimation of maintainability utilizing an MCDM approach to evaluate the identified factors.

Keywords: Maintainability, Mobile Application, Multi Criteria Decision Making, Software Quality

I. INTRODUCTION

Mobile applications, by and large implied as portable applications that are developed with a defined process so that they can be easily runnable on different handheld devices [1]. Portable applications are a move away from the coordinated programming frameworks generally found on desktops, in such a way these applications direct restricted, disengaged and explicit functionalities, for example, games, office suites, mini-computers, document administrators, and so forth. In specific fields, applications may have maintained a strategic distance from the standards of performing various tasks on account of the constrained equipment assets of the underlying and crude handheld gadgets however in a contrast to this, their explicitness is the part of their appealing quality and rising interest since they permit buyers to choose and settle onto the attributes and highlights they need their gadgets to have.

Mobile applications should work with unexpected impulsions in comparison to the desktop apps, as clients perceive them differently with different demands. Cell phones can be used anywhere unlike to the desktops and laptops in easier manner. Due to these reasons, mobile applications are specified with some limitations in different extensions impacting the working qualities of an application created with a way to deal with work on various platforms [3][4].

Revised Manuscript Received on October 25, 2019.

Parita Jain*, Computer Science and Engineering Department, KIET Group of Institutions, Uttar Pradesh, India. Email: paritajain23@gmail.com

Puneet Kumar Aggarwal, Computer Science and Engineering Department, HMRITM, Delhi, India. Email: puneetagarwal7@gmail.com

Down to the perspective, mobile phones were built in a manner to have obliged limit, screen size, low preparing capacities, less intelligent and poor accessibility to the system. This decade has seen the variable development in the circles of versatile advancement, accessibility, registering and portable insight. Cell phones have now changed with more extensive scope of screen sizes, viewpoint proportions, resolution, expandable memory limits, extemporized processor capacities, intuition advances, smart touch functionalities, intelligent and easy to use interfaces, simple to utilize applications which further include the components of robotization, stimulation and creativity to our regular day to day existences [5][6]

The essential target of these applications is to convincingly run the exhibit, from utility, productivity, and different course of games, wellbeing, and practically. Social networking through web-based sites for instance, has turned out to be one of most prospering prerequisite which is powerfully mixing through everybody's life; which is perhaps the most grounded repercussion of this radical programming development. An application deal pattern exhibited the way that before the broad and wholehearted acknowledgment of cell phones, programming projects were intended to keep running on personal computers. Tablets and propelled cells telephones are in presently overpowering the usage of work area PCs and workstations. All things considered spotlight was hurled on the figure that 197 billion applications were check to be downloaded in 2018. In the same manner, various designers have swung to an "adaptable first" viewpoint, mirroring a relative example in website engineering [7].

Today, programming designers are anticipating making powerful applications as the ability to withstand sudden circumstances and conditions is the thing that different applications need to focus on. A progressively intricate methodology incorporates developing especially for the compact condition, misusing the two of its confinements and central focuses. Engineers are carefully embedding's different functionalities into their applications in order to outperform the set quality principles [8] [13].

Keeping up quality measures is constantly a test for programming designers. In programming building, it has been constantly an issue to improve quality in an adequate time with minimal effort. In past decades, ease of use of portable applications have expanded massively thus as its test towards quality.

A quality characteristic identifying with a plan of repressions compelled to the structure that characterizes a work in advancement, developing, for instance, how usable, engaging, brisk or reliable it is [12] [28] [31] [34]. Maintenance being the essential attribute for evaluating the quality of portable applications serves different issues [6] [10] [22]. It's being one of the difficult ways to enhance the quality and be at the top most ratings in the user's view point with maintaining all the functionalities and look and feel of the application till it exists. According to IEEE maintenance is defined as "Modification of a software product after delivery to correct faults, to improve the performance or other attributes, or to adapt to a modified environment". While developing of the applications it has been found that 60-70% of the whole development defines the maintenance of the software application [25]. That is, the effort required to develop a software application includes most of its effort in maintenance of the software application. This makes this phase as the most expensive phase of the software development life cycle model till the application exists in the market. One of the renowned author Marco [9] states that assessing the value of maintenance is the only way to control the cost of maintenance. Identification of the defined factors is a necessity for evaluating the value of maintenance. However, finding out the value for the same is a bit difficult process.

The present work centers' around how to enhance quality of mobile applications. For this, different basic factors and its sub-factors that influence maintainability are distinguished through various studies presented previously and experts assessment working in the field of portable application building. These components are additionally assessed utilizing ELECTRE-TRI strategy to prioritize them. This enables designers to concentrate more on maintenance of the applications to create quality applications.

II. RELATED WORK

A thorough review has been conducted related to the topics including mobile applications, maintainability, and quality defining different development approaches for mobile applications. Various techniques have been propounded for the estimation of maintenance in the past to decrease the endeavours required and cost of the product application upkeep dependent on particular measurements and criteria.

Najm [11] proposed an equation to calculate maintainability value contingent upon the figured formula. The maintainability value comprised of LOC, cyclomatic complexity, Halsted volume. The recommended equation came about because of count on the old equation, it relied only upon line of code, which produced results near the old maintainability value equation with less exertion and time to compute.

Merve Vildan üimúek [14] displayed that the Android applications ought to have its own measurements in order to be analyzed since android applications have extraordinary highlights that doesn't exist in object oriented metrics. Geoffrey Hecht et. al. [15] utilized a few measurements with

the object oriented measurements for assessing maintenance of the mobile applications. Tobias Ammann [16] introduced that there is no single metric that can be utilized as a fundamental measurement to foresee the application maintenance. A portion of the measurements defined in their research incorporates Source Lines of Code(SLOC), Cyclomatic Complexity(CC), Information Flow(IF), Depth of Inheritance Tree (DIT), and Coupling Between Object Classes (CBC).

Bandini et al. [19] anticipated the maintenance of product item situated frameworks by thinking about three free attributes, complexity of design, maintenance support, and software engineers' capacity. Four measurements are considered for estimation dependent on these three components and relationship was found between these variables to anticipate maintainability of the software application. A model for evaluating effort while doing the maintenance was introduced by Fioravanti and Nesi [18]. Different aspects were taken for evaluation including code parts, eradication of sureness's etc.

Kajko et al. [20] introduced 2 models for upkeep of software application, one for item and other for technique. The model handles the issues related to practicality by thinking about customary properties and characteristics of the application. Ahn et al. [21] propounded another model for evaluation of the effort for maintaining the application. The authors collected data by doing survey and utilized the productivity segment for assessment.

Ardimento et al. [23] propounded an appraisal with respect to extended support to the software application in the event that if the application is hard to be understood, at that point it is additionally hard to look after it. Additionally, the researchers recommended utilizing the product venture as on preliminary premise first before applying it.

Jamimi and Ahmed [24] propounded fuzzy approach for assessing maintenance of the software's. They considered the object oriented attributes for evaluation. One more model introduced by Pratap et al. [29] using the similar fuzzy approach. The researchers utilized different arrangements for anticipating maintenance of the programming applications.

Kumar and Rath [26] suggested an estimation model utilizing parallel computing and neuro-fuzzy approach. For verification the model gets implemented on 10 defined static codes revealing that the combination of both the approaches works well. Shiva Kumar et al. [27] introduced a technique to appraise programming improvement exertion utilizing neuro-fuzzy methodology. In any case, the confinement of the proposed technique was that it was not reasonable for enormous outcomes. Another technique was introduced by Sharma and Verma [17] to appraise the expense of programming frameworks by utilizing function points concept. They utilized Gaussian function for approval and to deliver excellent outcomes.

There are various procedures for estimation of different factors subject to their distinctive sub-factors. Various creators have quite recently given powerful

results with these heuristic strategies yet simultaneously, there is an augmentation for improvement as the sub-factors reliably changes. Similarly, it has been found that for portable applications improvement there is no accurate algorithmic reason to survey upkeep. Likewise, extremely less research has been found to evaluate elements of mobile applications in a comprehensive manner. Consequently, there is a necessity for an estimation model for different parts of it.

III. PROPOSED APPROACH

Based on the literature review and opinion from the experts in the field of mobile application development, 9 critical factors for assessing maintainability have been determined and evaluated using a MCDM approach named ELECTRE-TRI method. The method is applied on the identified factors and the relationship among these factors get identified based on four criteria and the critical factors get outranked accordingly with respect to their weight coefficients. The factors that are identified are described as:

A. Lucidness

Lucidness is the capability of programming code which makes it intelligible and understandable notwithstanding for a non-specialized person. Generally, intelligibility is estimated by the proportion between number of lines of code and remarks which are accommodated the understandability of developers, and the system try not to get them. That is if without looking for the definitions or usage of the language if one can comprehend the working of the code, the program is said to be comprehensible. It is without a doubt, gives the idea that meaningfulness is a trademark identified with reusability, practicality, modifiability, and power.

B. Detectability

Detectability is the capacity to follow something. It is pertinent in various fields, including programming advancement and human services. Crosswise over businesses, detectability is utilized to consent to guidelines and limit hazard. By and large, detectability is used to give the historical backdrop of a thing and to enables monitoring in a situation.

C. Coupling

Coupling is the proportion of the level of association between the modules. In general, good programming will have low coupling. Maintaining a software application is very difficult if the coupling level is very high. That is, if coupling will be high then it's very difficult to go for any change during maintenance as that may lead for some defects in other functionalities. So, its important factor from the maintenance perspective.

D. Attachment

Attachment is an extent of how much the segments of the module are for all intents and purposes related. It is how a lot of all parts composed towards playing out a single task are

contained in the section. In a general sense, the association is the internal stick that keeps the module together. A better than average programming plan will have a high association that is the application must have high connection towards one another.

E. Size

Estimating programming size necessitates that the entire source code is expertly assembled, including database structure contents, information control source code, part headers, arrangement records, and so on. There are two sorts of programming sizes to be estimated, the specialized size (impression) and the useful size. There exists few programming specialized estimating techniques that have been generally depicted. The most widely recognized specialized measuring methods include number of Lines of Code per innovation, number of records, capacities, classes, tables, and so forth., from which exploded backwards Function Points can be figured.

F. Control Structure

The control structure determined by the contingent proclamations (rotation and cycle) of a module. It defines how actually the control is moving in between the modules. And the factor is very much important to get the maintenance of the mobile application as all the applications have to be very instant in terms of control transferring.

G. Independence

Independence is the degree to which the source code can be executed on various PC design and different working frameworks. Independent testing is a gathering of tests performed by at least one expert programming analyzer, who is not identified with the product item development group. They are classified "free" or "independent" since they are neither related with the maker nor the purchaser.

H. Consistency

Consistency is one of the quality traits of a software application regardless of whether it isn't expressed unmistakably in the necessities archives, and analyzers must check for it. It can likewise be named as how much the source code contains uniform documentation, expressing, and symbology inside itself.

I. Documentation Accuracy

It is defined as the degree to which the literature depicts the usefulness actualized in the system. Documentation is the most fundamental bit of a product item's accomplishment and productivity. Poor documentation influences the nature of programming or application. And this ultimately leads towards worse maintenance of the mobile application.

The relationship between these 9 identified maintainability factor and four criteria gets determined with respect to the rational analysis by the experts and the literature that is reviewed. The criteria are defined as:

A. Understandability

This refers to the degree to which the maintenance team

understands the software application functionality and how it runs in order to maintain them efficiently.

B. Immovability

The maintenance must be done in such a way that the implicit and explicit structure of the software application doesn't not affected. The functionality must remain the same.

C. Testability

This refers to the testing process performed while maintenance is going on. Testing is one of the important perspective while maintenance. While doing maintenance the errors or the defects must be resolved in an efficient way so that they will not occur again.

D. Portability

Portability of the software application defines the ability of the application to run on different platform without any extra effort. All the applications running today are almost cross-platform applications running on multiple platforms by number of users. It's important to focus on this aspect while doing maintenance so that the platform in-dependency must not change.

The relationship is defined as direct or indirect relationship using a symbol (+) and (-) respectively. With this determined relationship the outranking of the critical factors get done by utilizing a Multi-criteria decision making approach named ELECTRE-TRI method. The relationship is shown in Table I.

Table I: Relationship between the critical factors and criterion

Criterion	Understandability	Immovability	Testability	Portability
Factors				
Lucidness	+	+	+	-
Detectability	+	+		
Coupling	+	+		+
Attachment	+	+		+
Size		+		-
Control Structure	-	+	-	+
Independence		+	+	
Consistency	-		+	+
Documentation Accuracy	+	-	+	+

IV. EVALUATION OF THE PROPOSED APPROACH

An ELECTRE-TRI is one of the multi-criteria decision making strategies that was first proposed by Bernard Roy and his associates at SEMA consultancy organization. ELECTRE represents: ELimination Et Choix Traduisant la REalite (ELimination and Choice Expressing REality) [30]. The strategy can be applied to three fundamental issues: choosing, sorting and ranking of the variables [32] [33]-[35]. The technique incorporates 2 stages: first is the development of one or a few outranking relations, which targets looking at in an exhaustive way each pair of activities; second is the exploitation method that explains on the proposals acquired in the first stage. For the research work, the approach is applied to outrank the critical factors for maintaining a mobile application.

After determining the relationship among the factors and the criteria, the weight expectation function for each critical factor gets evaluated based on which outranking is done and the most vital factor gets determined for the maintenance of mobile applications. For evaluation, each parameter gets assumed with respect to the expression defined for weight calculation. The weight expectation function can be defined as follows:

$$We(ai) = Wb(ai) * Rb + TRi / Rb + \sum TRj (aj \in A)$$

Where, A= {ai|i=1,2,3,...,k} be a set of factors.

Wb is a base weight vector function for all A.

Rb is the difference of the direct and indirect relationships such that (Rb ≥ 0).

TRi is the constant value, reflecting the total number of criteria's.

TRj is the summation of the relationships defined for each critical factor with the sub-factors.

After substituting the values in the given expression determined from the Table 1, the weight values for each maintenance critical factor gets evaluated as shown in Table II. The more the evaluated value closer to zero the more its having the higher priority and must be focused at most while doing maintenance.

Table II: Weight Expectation Function Values

Factors	Weight Expectation Function Values (W _a)
Lucidness	0.70
Detectability	1.05
Coupling	0.72
Attachment	0.72
Size	2.0
Control Structure	1.0
Independence	1.05
Consistency	1.02
Documentation Accuracy	0.70

From the calculation, it is found that lucidness and documentation accuracy defined as the critical factors that lead to the success of maintenance of the mobile applications. The more the application is understood properly the more it will be maintained easily. According to the scale in increasing order coupling and cohesion found to be next critical factors in the order. Similarly, all the factors are determined with their weight values and defined with their importance towards maintenance of the mobile application.

V. CONCLUSION

Quality of mobile application is being an essential concern nowadays. It has been seen that designers are just not stress over the quality rather they are stressed over the expedient dispatch of the application. With the ample number of portable applications available on the play store, it's basic to evaluate the quality parameters and find the portable application having incredible quality. Quality involves diverse different parameters. One of the



major parameter is maintenance of a portable applications while its being utilized by the clients. The present research work centers around the assessment of the maintenance of a mobile application. For this an ELECTRE-TRI method has been used to identify the critical factors that leads towards the assessment of maintainability of mobile application after its gets deployed. In total nine factors have been determined and then a multi-criteria decision making approach applied to outrank these identified factors based on the demonstrated weight esteems for the chosen factors. Further, number of criteria can be expanded in order to assess the maintenance more precisely for mobile application.

REFERENCES

- 1 H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, "Diversity in smartphone usage," *In Proceedings of the 8th International Conference on Mobile systems, applications, and services (Mobisys '10)*, pp. 179-194, 2010.
- 2 A. Ahmad, K. Li, C. Feng, S.M. Asim, A. Yousif, and S. Ge, "An Empirical Study of Investigating Mobile Applications Development Challenges", *IEEE Access*, Vol. 6, pp. 17711-17728, 2018.
- 3 A. Nitze, and A. Schmietendorf, "A survey on mobile users' software quality perceptions and expectations," *In Proceedings of the Eighth International Conference on in Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 1-2, 2015.
- 4 Z. Jiang, R. Kuang, J. Gong, H. Yin, Y. Lyu, and X. Zhang, "What Makes a Great Mobile App? A Quantitative Study Using a New Mobile Crawler," *IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pp. 222-227, 2018.
- 5 P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Exploring Quality Aspects of Smart Mobile Phones Applications", *Journal of Advanced Research in Dynamical and Control Systems (JARDCS)*, pp. 292-297, 2018.
- 6 P. Jain, A. Sharma, and L. Ahuja, "The Impact of Agile Software Development Process on the quality of Software Product", *In Proceedings of the 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2018.
- 7 "Number of apps available in leading app stores 2018 | Statistic." [Online]. Available: <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>.
- 8 P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Assessing Quality of Mobile Applications Based On a Hybrid MCDM Approach", *International Journal of Open Source Software and Processes (IJOSSP)*, Vol. 10, Issue 3, Article 4, 2019.
- 9 T. D. Marco, "Controlling Software Projects: Management Measurement and Estimation", Prentice-Hall, Yourdon Press, 1986.
- 10 P. Jain, A. Sharma, and L. Ahuja, "Software Maintainability Estimation in Agile Software Development" *International Journal of Open Source Software and Processes (IJOSSP)*, Vol. 9, Issue 4, Article 4, pp. 65-78, 2018.
- 11 Najm, N. MAM. "Measuring Maintainability Index of a Software Depending on Line of Code", *Journal of Computer Engineering (IOSR-JCE)*, Volume 16, Issue 2, Ver. VII (Mar-Apr. 2014), PP 64-69
- 12 P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Evaluating Self-Management Features for Mobile Applications", *International Journal of E-Services and Mobile Applications (IJESMA)*, Vol.11, Issue 2, pp.43-45, 2019.
- 13 P. Jain, A. Sharma, and L. Ahuja, "ISM Based Identification of Quality Attributes for Agile Development", *In Proceedings of the 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 615-619, 2016.
- 14 M. V. Simsek, "a software quality model for android applications, a thesis submitted to the graduate school of informatics of the middle east technical university", February 2016
- 15 G. Hecht1, O. Benomar, R. Rouvoy1, N. Moha2 and L. Duchien1 in "Tracking the Software Quality of Android Applications along their Evolution", 17 Sep 2015.
- 16 T.Ammann, "A metrics-based comparison of secondary user quality between iOS and Android", Master Thesis Software Engineering. August 26, 2014.
- 17 V. Sharma, and H. K. Verma, "Optimized Fuzzy Logic Based Framework for Effort Estimation In Software Development", *International Journal of Computer Science*, Vol. 7, Issue 2, pp. 30-38, 2010.
- 18 F. Fioravanti, and P. Nesi, "Estimation and Prediction Metrics for Adaptive Maintenance Effort of Object -Oriented Systems", *IEEE Transactions on Software Engineering*, Vol. 27, Issue 12, pp. 1062-1084, 2001.
- 19 S. Bandini, F. D. Paoli, S. Manzoni, and P. Mereghetti, "A support system to COTS based software development for business services. *In Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering*, Vol. 27, pp. 307-314, 2002.
- 20 M. Kajko-Mattsson, M., Canfora, G., Chorean, D., Van Deursen, A., Ihme, T., Lehman, R. Reiger, T. Engel, and J. Wernke, "A Model of Maintainability - Suggestion for Future Research. *In Proceedings of International Multi-Conference in Computer Science & Computer Engineering (SERP'06)*, pp. 436-441, 2006.
- 21 Y. Ahn, J. Suh, S. Kim, and H. Kim, "The Software Maintenance Project Effort Estimation Model Based on Function Points", *Journal of Software Maintenance: Research and Practice*, Vol. 15, Issue 2, 71-85, 2003.
- 22 P. Jain, A. Sharma, L. Ahuja, "A Customized Quality Model for Software Quality Assurance in Agile Environment", *International Journal of Information Technology and Web Engineering (IJITWE)*, Vol. 14, Issue 3, Article 4, pp. 64-77, 2019.
- 23 P. Ardimento, A. Bianchi, and G. Visaggio, "Maintenance-oriented Selection of Software Components. *In Proceedings of 8th European Conference on Software Maintenance and Reengineering*, pp. 115 -124, 2004.
- 24 H. A. Ai-Jamimi, and M. Ahmed, "Prediction of Software Maintainability Using Fuzzy Logic", *In Proceedings of 3rd International Conference on Software Engineering and Service Science (ICSESS)*, pp. 702-705, 2012.
- 25 P. Jain, A. Sharma, L. Ahuja, "The Model for Determining Weight Coefficients of Maintainability Criteria in Agile Software Development Process", *In Proceedings of the International Conference on Internet of Things: Smart Innovation and Usages (IOT-SIU)*, 2019.
- 26 L. Kumar, and S. K. Rath, "Software maintainability prediction using hybrid neural network and fuzzy logic approach with parallel computing concept", *International Journal of System Assurance Engineering and Management*, Vol. 8, Issue 2, pp. 1487-1502, 2017.
- 27 N. Shivakumar, N. Balaji, and K. Anathakumar, "A neuro Fuzzy Algorithm to Compute Software Effort Estimation. *Global journal of Computer Science and Technology*, Vol. 16, Issue 1, pp. 1-6, 2016.
- 28 P. K. Aggarwal, P.S. Grover, L. Ahuja, "A Performance Evaluation Model for Mobile Applications", *In Proceedings of the International Conference on Internet of Things: Smart Innovation and Usages (IOT-SIU)*, 2019.
- 29 A. Pratap, R. Chaudhary, and K. Yadav, "Estimation of Software Maintainability using Fuzzy Logic Technique. *In Proceedings of International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pp. 486-492, 2014.
- 30 G. Lu, H. Wang, and X. Mao, "Using ELECTRE TRI Outranking Method to Evaluate Trustworthy Software," *In Proceedings of International Conference on Autonomic and Trusted Computing (ATC)*, pp. 219-227, 2010.
- 31 P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Security Aspect in Instant Mobile Messaging Applications," *In Proceedings of IEEE International Conference on Recent Advances on Engineering, Technology and Computational Sciences (RAETCS)*, pp.1-5, 2018.
- 32 R. Mousseau, R. Slowinski, and P. Zielniewicz, "A User-oriented Implementation of the ELECTRE TRI Method Integrating Preference Elicitation Support," *Journal of Computers and Operations Research*, Vol. 27, Issue 7, pp.757-777, 2000.
- 33 J. Figueira, S. Greco, and M. Ehrgott, "Multiple Criteria Decision Analysis: State of the art surveys," Springer, ISBN: 0-387-23067-X, 2005.
- 34 P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Incorporating Autonomic Capability as Quality Attribute for Software Systems", *In Proceedings of IEEE International Conference on Reliability, Infocom Technologies and Optimization*, 2018.
- 35 P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Locating Usability Critical Factors for Mobile Applications Using ELECTRE-TRI Method", *In Proceedings of IEEE International Conference on Confluence*, pp. 597-600, 2019.

AUTHORS PROFILE



Ms. Parita Jain completed her M. tech (IT) from CDAC Noida. Currently, she is working as Assistant Professor, Department of Computer Science and Engineering at KIET Group of Institutions, UP. She is pursuing Ph.D. from AIIT, Amity University. Her research interest includes Software Engineering, Software Testing, and Data Mining. She has published more than 15 papers in international journals and conferences.



Mr. Puneet Kumar Aggarwal completed his M. tech (IT) from CDAC Noida. Currently, he is working as Assistant Professor, Department of Computer Science and Engineering at HMRITM, Delhi. He is pursuing Ph.D. from AIIT, Amity University. His research interest includes Software Engineering, Software Testing, and Steganography. He has published more than 15 papers in international journals and conferences.