



Comparative Analysis of Common Software Cost Estimation Modelling Techniques

Syed Zaffar Iqbal¹, Muhammad Idrees², Ahmed Bin Sana¹, Nawab Khan¹

¹Department of Computer Science and Technology, Alhamd Islamic University, Quetta, Pakistan

²Department of Mathematics, Government Boys Degree College, Nushki, Pakistan

Email address:

zis.shah@gmail.com (S. Z. Iqbal), idrees.math@hotmail.com (M. Idrees)

To cite this article:

Syed Zaffar Iqbal, Muhammad Idrees, Ahmed Bin Sana, Nawab Khan. Comparative Analysis of Common Software Cost Estimation Modelling Techniques. *Mathematical Modelling and Applications*. Vol. 2, No. 3, 2017, pp. 33-39. doi: 10.11648/j.mma.20170203.12

Received: May 15, 2017; **Accepted:** June 12, 2017; **Published:** July 14, 2017

Abstract: As software cost estimation in software projects is a very difficult, confusing and challenging work for any software company and software cost estimation is the primary step to start any software project. It gives the overview of resources, efforts and time/schedule required for a software project in respect of cost to the software company. Software project success generally depends on software cost estimation as it provides us an initial idea of the track, challenges and risk involved in the software project development. The software cost estimation in software engineering is very challenging to match the actual cost of the software project with estimated cost. Effective software cost estimation can help software company make more consistent decisions in planning the software project risk. If the predicted estimates are wrong it may lead to negative results for a software company. Many software companies find, search and analyze software project performance by estimating software cost estimation accuracy. Unfortunately, regardless of the large body of experienced and skilled with estimation models, the accuracy of these models is not adequate. In this research paper observation on the performance of the software cost estimation methods and description of methodologies and technique used in to software project cost estimation included. This research paper give comparative comparison study of software cost estimation methods and reviews several classes of software cost estimation models and techniques. Also study the pros and cons of different software cost estimation modeling techniques.

Keywords: Comparative Analysis of Cost Estimation, Software Cost Estimation Models, Software Cost Estimations

1. Introduction

Software cost estimation is a vital part of most software development projects. Unfortunately, software development cost estimation has a difficult and inaccurate task. In spite of the availability of many estimation procedures, methods, techniques and guidelines, there is still a need for enhancement. One means of reducing software cost estimation error is through analysis of cost estimation error. Since the early 1950s, Software engineering cost models and estimation techniques are used for a number of will power.

These include software planning, software control, software improvement, software budgeting, risk analysis and investment analysis. In this modern era, software has become the most expensive element of computer system projects. Reliable and accurate software cost estimates are critical to both developers and clients/customers.

Research on software cost estimation started independently in a number of organizations and military organizations that develop large software systems [6]. They can be used for creating request for suggestions, scheduling, consultations, monitoring and control. The underestimating of budget may effect in management approving planned systems that then exceed their cost/budget, with underdeveloped functions, poor quality and failure to complete on estimated schedule. The determination and purpose of software cost estimation is to define:

- 1) Resources
- 2) Software quality assurance
- 3) Risks
- 4) Development tests
- 5) Development environment
- 6) Administration
- 7) Charges
- 8) Verification

If just have a look at the software development of

international market, easily notice the declination in hardware prices and an incredible increase in the computing volume, but still the cost of software development has increased significantly. Software development struggle and effort is an intensive procedure. For managing any software project effectively, it is very vital to judge the Software Cost and Effort correctly. Expecting a correct Software Cost Estimation and Software Effort Estimation is one of the most challenging, inspiring and demanding job in every software industry. For the projects which are of high budget, very big, important and have temporal constraints, having an accurate Software Cost and Effort Estimation becomes more and more important and at the same time analyze and predict an accurate Software Cost and Effort Estimation becomes more difficult.

Mostly classify the software projects into three categories:

- a) Small projects
- b) Medium projects and
- c) Large projects

The chances of having a precise software cost and effort estimation will be high, medium and low respectively. This is because, with the increase in the complexity of software, it becomes more difficult to have an accurate estimation of software cost and effort. The relationship between complexity of software projects and chances of having a precise software cost and effort estimation can be used by management for assessing those projects proposal and for effective management of the Software Development Process.

Hence, the correct calculation of cost of software development has a large economic influence: in fact, approximately 60% of large projects exceed their Software Cost Estimates and the projects which are never completed because of the false estimation of Software development Cost. Basically effort is the main driving factor for the Cost and because of this fact, estimation of Software Cost is ultimately a problem of Software Effort estimation. Number of scientists and researchers are trying their best for developing more and more accurate, new software cost estimation techniques. Most of the software cost estimation techniques are based on the algorithmic models, expert judgment and machine learning approaches [7, 8, 9]. Accurate cost estimation is important because:

- a) To measure the impact of variations and support planning and re-planning.
- b) To define what resources are require for the project and how well these resources will be utilized.
- c) To categorize and arrange development projects with respect to an overall business plan.
- d) Projects can be easier to control and manage when resources are better matched to real needs.

There are two factors which disturb the project costs:

- a) Efficiency and Proficiency of the implementation
- b) Cost of the studies done prior to software development

The above two factors are very vibrant and differ from project to project. There are many factors which are to be considered while software cost estimation.

2. Software Cost Estimation

Software cost estimation can be defined as the finding of the cost for a software project. Software cost estimation of software project emphasize on how to estimates efforts, time and cost with the project activities. Cost estimation contains answering the following demands of the software engineer:

- 1) How much effort is essential?
- 2) How much time is essential?
- 3) What is the total cost?

3. Software Cost Estimation

The idea of software project cost estimation come in 1960s and many cost estimation models have been developed by various researchers to calculate the software project estimation. The types into which these models are divided are three methods:

- 1) Algorithmic methods
- 2) Non-algorithmic methods
- 3) Machine Learning methods

In this research paper review all these methods and provide a comparative study based upon the effectiveness and usefulness of these software cost estimation models.

3.1. Algorithmic Methods

The algorithmic model is developed to deliver some mathematical equations to do software estimation. These mathematical equations are based on research and historical data and use inputs such as Source Lines of Code (SLOC), number of functions to perform, and other cost drivers such as language, design methodology, skill-levels, risk assessments, etc [14]. The algorithmic methods have been largely studied and many models have been developed.

3.1.1. Constructive Cost Model (COCOMO) Method

The Constructive Cost Estimation Model (COCOMO) is proposed by DR. Berry Boehm in 1981 and that's why it is also known as COCOMO'81. The COCOMO is the basic technique which is used in the industry, in all over the world. The COCOMO model is used for calculating the required number of persons in every month for the development of software. The COCOMO model is also able to give an estimate of the development schedule in months and with the support of this model can also estimate that how much efforts should be given to each and every major stages of software development.

The COCOMO model has: $MAN-MONTHS = K1 * (Thousands\ of\ Delivered\ Source\ Instructions)^{K2}$

Where K1 and K2 are two variable/constraints reliant on the development environment. Estimation of COCOMO model can be made more authentic by taking into account, other aspects like the required features of the software to be developed, the experience and qualification of the development team and the software development environment. The software complexity aspects are required as: reliability, efficiency, size of database, analyst and programmer

proficiency and also a good experience of development team. Many of these aspects affect the person months need by instruction of degree. COCOMO model observe that the software requirements and system have well defined and the requirements are unchangeable. COCOMO consider three development methods: Organic, Embedded and Semi-detached and provide three levels: Basic, Intermediate and Detailed [15].

Pros

- COCOMO model is realistic and easy to understand
- User can clearly know how it works
- Provide accounts for several features that affect cost of the software project
- Works on historical data/information and calculate precise results
- Helpful to understand the impact on different factors that affect the project costs

Cons

- COCOMO model ignores requirements and all documentation
- It ignores customer skills, cooperation, knowledge and other parameters
- It oversimplifies the impact of safety/security aspects and hardware issues
- It ignores personnel turnover levels
- It is dependent on the amount of time spent in each phase

3.1.2. Function Point Analysis (FPA) Method

The Function Point Analysis model is computing the size and complexity of a software project in terms of the functions that the project provide to the user. Function Point Analysis was designed by Albrecht [17] to count the functionality of software projects. This method measures the size of the software, it considers internal logical files [18], external interface file, external input-output, and external inquiries from functional viewpoint metric. ESTIMACS and SPQR/20 are the models based on Function point analysis estimation. There are two steps in counting function points:

Counting the user functions: The raw function counts are arrived at by considering a linear combination of five basic software components: external inputs, external outputs, external inquiries, logic internal files and external interfaces, each at one of three complexity levels: simple, average or complex. The sum of these numbers, weighted according to the complexity level, is the number of function counts (FC).

Adjusting for environmental processing complexity: The final function points is arrived at by multiplying FC by an adjustment factor that is determined by considering 14 aspects of processing complexity. This adjustment factor allows the FC to be modified by at most 35% or -35%.

Pros

- FPA model is not dependent on any tool, the language and approaches of software deployment
- Software project development costs can be measured in early phases of software development
- Results are improved than Source Lines of code (SLOC)

Cons

- It involves manual work which is more time consuming
- Difficult for a new developer to calculate the size of the software project
- The usage of FPA requires experience

3.1.3. Putnam Model

This model is designed by Lawrence H. Putnam in 1978. This model was design on the basis of manpower distribution and the research of many software projects. The Putnam model is a dynamic multivariate model for software cost estimation. In this model the association between effort and size is non-linear. The Putnam model is sensitive to deliver software project on time. According to Putnam model, fairly small additions in the project implementation schedule can result in extensive investments of effort. The main equation for Putnam's Model is [16]:

$$S = E * (\text{Effort})^{1/3} * t * d^{4/3}$$

Where, E is the environment factor that describes the environment ability, td is the delivery time, effort and S are expressed by person-year and line of code respectively.

Pros

- This model is initially work on two variables which are software project size and time
- Most of the software companies can easily collect size, efforts and time from past projects

Cons

- This model does not allow for other phases of software development life cycle.
- Time is very dominating issue in Putnam model.

More software cost estimation models based on algorithmic method.

Table 1. Algorithm methods.

Cost Estimation Model	Key Parameters
Walston Felix	Line of code size
Boeing	Line of code size
Bailey-Basili	Line of code size
Doty	Line of code size
GRC	Line of code size
Aron	Instructions
SEER SEM	Functional size
Mike II	Functional size
Gaffney	Functional size
Matson	Functional size
Kustanowitz	Man month
Farr & Zagorski	Man month

3.2. Non-algorithmic Methods

In non-algorithmic methods, the software project estimation can be completed through using the previous software projects and previous software experiences which is like to the under estimate software projects. In non-algorithm methods must have knowledge of a previously completed projects similar to our current software project. Estimation is completed on the basis of analysis of previous software projects. The methods based on Non-Algorithmic are as described below:

3.2.1. Analogy Method

The software cost estimation through analogy is that whenever receive a new software project for cost estimation, it is just compared to some other similar projects to reach at the adjacent similar software project through which can estimate our current software project cost. The data, information and its values from previously completed projects are assumed to calculate cost of our current project cost. This method can be used in both at system or component level [19, 20]. The details of the exact steps followed for estimation based on analogy are given below:

- 1) Select all attributes of current software project
- 2) Finding other similar software project
- 3) Compared current software project with already stored software in the database
- 4) Calculating the software cost of the current project from the previous similar software project

Pros

- a) Its depends on the data of previous similar software projects
- b) Experience of software engineer is required which helps in improved software cost estimate
- c) Recognize the minute distinction between the previous completed projects and current projects

Cons

- a) Software engineer required to find out the attributes using this method to through
- b) Project can be defined best also need to provide weightage to these to get a better analogy
- c) In this method cannot use this technique for every project

3.2.2. Top-Down Method

In this method calculate total cost from global properties selecting of non-algorithmic technique. Then this cost is divided to various components of the system. Top-down method is more beneficial in the early stages of software development because detailed information is not available during this stage [21, 22].

Pros

- a) Top-down method needs very less detail about the project
- b) It is faster to implement
- c) Unlike other techniques top-down estimation focuses on activities

Cons

- a) This technique does not take into consideration low level problems
- b) Low level problems are difficult and can increase the cost of the system

3.2.3. Bottom-up Method

Bottom-up estimation is opposite of Top-down estimation method. In this method cost of each software component and then the result is combined to achieve the overall cost of the software derived. Goal is to derive system estimate from the accumulated estimate of the small component [10].

Pros

- a) Bottom-up method is more stable

Cons

- a) It does not take into account the project level activities
- More software cost estimation models based on non-algorithmic method.

Table 2. Non-algorithm methods.

Cost Estimation Model	Key Parameters
Nelson Model, SDC	Man Month
Parkinson's Law	Available resources
Price to Win	Customer's budget
Expert Judgment	Consultant experience

3.3. Machine Learning Methods

All the methods/techniques covered till now, that well use statistical methods in software cost estimation and therefore flop in providing good strong results. Machine learning methods are utilize training rules for estimation ad repeat the run cycles. Therefore this can be the suitable approach as it can increase accuracy of the taking good results.

3.3.1. Neural Network Method

Neural network also called as artificial neural network method have an extensive representation and also have a very encouraging use in software cost estimation. The procedure of functioning of neural networks can be applied for software cost estimation easily. Every neuron can denote an activity for the software project and all of the neurons give outputs based on the input they receive. The overall output gives the software cost estimation of the whole project. A neural network is useful in the way that it becomes very easy to learn, as they modify the weights every time they are used. This leads to better and better estimation as the project tends to completion. Numerous methods exist inside neural networks and back propagation works the best in them.

Pros

- a) Neural Network method based estimation methods are consistent with unlike databases and they provide power of reasoning in estimation process

Cons

- a) Large amount of training data is required
- b) No guidelines or instructions are provided for designing

3.3.2. Fuzzy Logic

Fuzzy logic method based upon ideas of fuzzy logic. This method attempts to endorse the human behavior by searching for reasons. In cases where it's difficult to come up with a crisp decision, fuzzy logic can be effective in reaching the results. This technique always tries to support the facts which might be ignored usually. There are four stages in this approach [23]:

- 1) Produce trapezoidal numbers for the linguistic terms
- 2) Develop the complexity matrix by producing a new linguistic term
- 3) Determine the productivity rate and the attempt for the new linguistic terms
- 4) Determine the effort required to complete a task and to compare the existing method.

Additionally fuzzy logic method based estimation is also called soft computing technique. Soft computing techniques are emerging software estimation. Fuzzy logic has developed as an important tool to resolve such problems, for which mathematical models cannot be created. Design of software is many a times categorized by constraints that demonstration fuzziness. Application of fuzzy logic in cost estimation helps in overcoming many problems which exists in the already available cost estimation methods [24].

Pros

- In fuzzy logic method no training is necessary
- This method is more flexible
- Give dependable estimation

Cons

- This method is difficult to use
- Cost estimation of complex features is repetitive

4. Advantage and Disadvantage of Existing Cost Estimation Methods

Describe the advantages and disadvantages of existing cost estimation methods. This description could be useful for choosing an appropriate method in a particular project. Table 3 and 4 shows a comparison of mentioned methods for estimation. For doing comparison, the popular existing estimation methods have been selected.

Table 3. Advantages of Cost Estimation Methods.

Method	Type	Advantages
COCOMO	Algorithmic	Pure results, very mutual
FPA	Algorithmic	Language independent, results are improved than SLOC
Putnam	Algorithmic	Fast and easy. Fit for large scale project.
Walston & Felix	Algorithmic	Based on historical statistical data.
Analogy	Non-Algorithmic	Works based on actual experiences, having especial expert is not important
Top-down	Non-Algorithmic	Focuses on system or high level activities.
Bottom-up	Non-Algorithmic	Can estimate each components of the software to lead to a detailed estimation Results.
Price to win	Non-Algorithmic	Often gets the contract.
Neural Network	Machine Learning	Consistent with unlike databases, Power of reasoning
Fuzzy Logic	Machine Learning	Training is not required, Flexibility

Table 4. Disadvantages of Cost Estimation Methods.

Method	Type	Disadvantages
COCOMO	Algorithmic	Much data is required, It's not suitable for any project.
FPA	Algorithmic	Mechanization is hard to do, quality of output are not considered
Putnam	Algorithmic	Lack of some details required to estimate Parts of the software.
Walston & Felix	Algorithmic	Haven't been practiced a lot because of Statistical problems.
Analogy	Non-Algorithmic	A lots of information about past projects is required, In some situations there are no similar project.
Top-down	Non-Algorithmic	May not cover low level components and Activities.
Bottom-up	Non-Algorithmic	Needs a lot of time, resources to Implement it. Require understanding of the components.
Price to win	Non-Algorithmic	Generally produces large overruns.
Neural Network	Machine Learning	There is no guideline for designing, The performance depends on large training data.
Fuzzy Logic	Machine Learning	Hard to use, Maintaining the degree of meaningfulness is difficult.

5. Selection of Software Cost Estimation Methods

Based upon the above study, selecting the best method is a difficult task as it is clear that there is no best or worst method for software cost estimation rather each one of them has its pros and cons which are usually admiring. Non-algorithmic methods take decisions based upon past data and experience whereas Algorithmic methods use mathematical equations for the prediction. At one hand algorithmic methods require some efforts to understand but with enough data provide efficient results, on the other, non-algorithmic methods are easier to implement but sometimes provide inaccurate estimates. Thus, one must use a combination of different models to have all the benefits and get the best results. Moreover, the cost estimates

should be compared to actual cost and updated on regular basis throughout the project life cycle.

6. Usage of Software Cost Estimation Methods

It is very common that relate some software cost estimation methods to estimate the cost of software projects. But what have to analyze is that it is very important to repeatedly re-estimate software project cost and to relate marks against actual spending at each major milestone. This keeps the position of the project visible and helps to identify necessary corrections to budget and timeframe as soon as they happen.

At every software project estimation and re-estimation opinion, repetition is a vital tool to improve estimation quality. The system analyst who act as an estimator can apply

numerous estimation techniques and check whether their estimates unite. Different estimation methods may use different data. This results in better coverage of the knowledge base for the estimation process. It can help to identify cost mechanisms that cannot be allocated with ignored in one of the methods.

7. Recommendations

- 1) Do not stuck on any one cost estimation method
- 2) For cost and timeframe use different types of estimation methods
- 3) Use many cost estimating methods
- 4) Compare the results and analyze the explanations for variations.
- 5) Make documentation of the prediction made when making the cost estimates.
- 6) Do monitoring and evaluation of whole project accuracy in software cost estimate process.
- 7) In many ways of effective software process can increase accuracy in software cost estimation.
- 8) Maintain all previous/past data and make a database

8. Conclusion

When go through comparative analysis of various techniques for software cost estimation which have their own pros and cons. It is very hard to point out a failsafe method, as most of the cost estimation is scenario dependent. Some method are highly accurate but cumbersome to implement, while some only work best on certain known parameters. In order to pick a method for any given project, a proper analysis of the project is much needed. A wrong estimation technique can significantly delay a project, and a right technique can make a project breeze through its deadlines. A detailed analysis of global factors must be made, or else, it becomes very difficult to give correct deadlines. Also, an emphasis on smaller details is necessary, as they can cause delays when they add up together.

References

- [1] San Juan, M. N. D. (2011). Practical software project estimation: a toolkit for estimating software development effort & duration.
- [2] Al Khatib, S. M., & Noppen, J. (2017). Benchmarking and Comparison of Software Project Human Resource Allocation Optimization Approaches. *ACM SIGSOFT Software Engineering Notes*, 41 (6), 1-6.
- [3] Sehra, S. K., Brar, Y. S., & Kaur, N. (2016). Predominant Factors Influencing Software Effort Estimation. *International Journal of Computer Science and Information Security*, 14 (7), 107.
- [4] Iqbal, S. Z. (2017). Z-SDLC Model. *International Journal of Engineering and Advanced Research Technology (IJEART)*, 3 (2), 8.
- [5] Grimstad, S., & Jørgensen, M. (2006, September). A framework for the analysis of software cost estimation accuracy. In *Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering* (pp. 58-65). ACM.
- [6] Jones, C. (2005). *Software Quality in 2002: A Survey of the State of the Art*. Software Productivity Research, Marlborough, Massachusetts.
- [7] Boraso, M., Montangero, C., & Sedehi, H. (1996). Software cost estimation: An experimental study of model performances.
- [8] Jensen, R. W. (2004). Extreme software cost estimating. *CrossTalk: The Journal of Defense Software Engineering*.
- [9] Stamelos, I., Angelis, L., Morisio, M., Sakellaris, E., & Bleris, G. L. (2003). Estimating the development cost of custom software. *Information & Management*, 40 (8), 729-741.
- [10] Boehm, B. W. (2001). Software engineering economics. In *Pioneers and Their Contributions to Software Engineering* (pp. 99-150). Springer Berlin Heidelberg.
- [11] Hughes, R. T. (1996). Expert judgement as an estimating method. *Information and Software Technology*, 38 (2), 67-75.
- [12] Wittig, G., & Finnie, G. (1997). Estimating software development effort with connectionist models. *Information and Software Technology*, 39 (7), 469-476.
- [13] Kashyap, D., & Misra, A. K. (2013, July). Software development cost estimation using similarity difference between software attributes. In *Proceedings of the 2013 International Conference on Information Systems and Design of Communication* (pp. 1-6). ACM.
- [14] Marbán, O., Menasalvas, E., & Fernández-Baizán, C. (2008). A cost model to estimate the effort of data mining projects (DMCoMo). *Information Systems*, 33 (1), 133-150.
- [15] S. Malathi and S. Sridhar, "Optimization of Fuzzy Analogy in Software cost estimation using linguistic Variables," *Procedia Engineering*, vol. 38, pp. 177-190.
- [16] Software Cost Estimation Methods: A Review" - Vahid Khatibi, Dayang N. A. Jawawi. Volume 2 No. 1 ISSN 2079-8407 *Journal of Emerging Trends in Computing and Information Sciences* ©2010-11 CIS Journal.
- [17] Yinhan, Z., W. Beizhan, et al. "Estimation of software projects efforts based on function point", *Computer Science & Education. ICCSE, 4th International Conference*, 2009.
- [18] Mustafa, K. K. Gowthaman, and R. A. Khan, "Measuring the Function Points for Migration Project: A Case Study", *American Journal of Applied Sciences*, 2005.
- [19] Gupta, Syona, Geeta Sikka, and Harsh Verma, "Recent methods for software effort estimation by analogy", *ACM SIGSOFT Software Engineering Notes*, 2011.
- [20] Keung, J. W., B. A. Kitchenham, et al. "Analogy-X: Providing Statistical Inference to Analogy based Software Cost Estimation". *Software Engineering, IEEE Transaction on* 34 (40) 471-484, 2008.
- [21] Kusuma Kumari B. M, "Software Cost Estimation Techniques", *International Journal of Engineering Research in Management and Technology*, Volume -3, Issue- 4, 2014.

- [22] Hareton Leung, Zhang Fan, “Software Cost Estimation”, Article 2001.
- [23] Narendra Sharma, Aman Bajpai, Mr. Ratnesh Litoriya, The International Journal of Computer Science & Applications (TIJCSA) ISSN – 2278-1080, Vol. 1 No. 3 May 2012.
- [24] Attarzadeh, I. Siew Hock Ow, “Improving the accuracy of software cost estimation model based on a new fuzzy logic model”, World Applied Science Journal 8 (2): 117-184, 2010.
- [25] Bhatia, P., Mishra, K. K., & Misra, A. K. (2016). An Approach to Software Cost Estimation by Improved-Time Variant Acceleration Coefficient Based PSO. Journal of Multiple-Valued Logic & Soft Computing, 27 (1).
- [26] Mishra, K. K., Tripathi, A., Tiwari, S., & Saxena, N. (2017). Evolution based memetic algorithm and its application in software cost estimation. Journal of Intelligent & Fuzzy Systems, 32 (3), 2485-2498.
- [27] Phillips, J. J., & Phillips, P. P. (2016). Handbook of training evaluation and measurement methods. Routledge.
- [28] Briand, L. C., El Emam, K., Surmann, D., Wiecezorek, I., & Maxwell, K. D. (1999, May). An assessment and comparison of common software cost estimation modeling techniques. In Software Engineering, 1999. Proceedings of the 1999 International Conference on (pp. 313-323). IEEE.

Biography



Syed Zaffar Iqbal works as a Lecturer at Alhmad Islamic University, Quetta-Pakistan. He received his MCS degree from University of Balochistan in 2010. His research interests include are software engineering, database management system, programming and development. He is a supervisor of final year projects at his university and ICT R&D Funded projects. He is the author of computer science books. He is also running a software company named as Logical Creations as Managing Director.



Muhammad Idrees works as a Lecturer in the Department of Mathematics at Government Boys Degree College, Nushki Balochistan. He received his MSc Mathematics from University of the Punjab, MCS from Virtual University of Pakistan and M. Phil Scholar in Mathematics at University of Balochistan, Quetta. His research interests include are Fuzzy Logic, Fuzzy Algebras and Theoretical Computer Science. He is also an advisor in Logical Creations.