CHAPTER 11

CONCLUSION

This thesis has presented some metrics to estimate size, effort, maintainability and understandability of software. These metrics can be useful for software project managers to understand, control and improve the software process and product.

The thesis presented an introduction to software estimation and the difficulties in estimation. It presented an overview of the work done by various researches in the field of software cost estimation since the early 60’s till date. It presented the literature survey that was carried out where the thrust has been on size/effort/cost estimation techniques and models as well as measurement of maintainability and understandability. Further, literature survey was also carried out in the field of the soft computing approach (neural network models, fuzzy models and probabilistic reasoning ) being used in software engineering application particularly in estimating size, effort, cost, maintainability and understandability.
It reviewed various cost estimation models and classified them. It further, compared the various models, discussing the strengths and weaknesses of each model.

The soft computing approach was discussed and the possibility of using neural network model and fuzzy model for the measurement of software metrics was explored. It further proposed hybrid models, which combine two or more paradigms of computing to arrive at better solutions. There is also a caution made that it is not necessary that hybrid models may be intrinsically better at times they could provide more opportunities for misuse then similar techniques. Although, motivated by combined strengths of the system, the hybrid model in the worst case may contain none of the strengths and all weaknesses of the components systems.

The estimation of size is very critical and difficult area and the challenge lies in estimating the size of the software in the beginning. The function point analysis method, which is independent of the language tools or methodologies, was used for estimating size. The thesis proposed a fuzzy model to capture the inherent fuzziness that is present in all model measurement parameters i.e. all inputs and complexity factors. The uncertainty that prevails in the inputs is captured by fuzzification and the imprecision of complexities also managed by fuzzy logic. Instead of using crisp weights to define simple, average and complex
measurement parameters, the fuzzy model proposed, exploits overlapping amongst them and provides better solutions.

A neural network was proposed for estimating the Line of Code once the function point count, the FP standard used, the language used and team size are known. This model was validated by using data of 88 projects available from ISBSG (release 9). The feed forward network with the back propagation algorithms was used. Different training algorithms were evaluated and it was concluded that train br (Bayesian regularization) was found to be more suitable.

The possibility of estimating Lines of Code, when only Function Point count is known, is also dealt with in this work. Models are proposed that use simple linear regression techniques, neural networks and a combination of the two. This thesis also proposed an expert committee model, which is a combination of robust regression neural network model. All the models proposed including the expert committee model are validated by project data (release 9) from ISBSG. It was verified that more number of training and test cases further improved the performance of the model.

In order to use the software for years together, every software needs to be changed to meet the changing requirements of users & customers during its
lifetime. It is generally impractical and uneconomical to produce software which does not need to be changed. Introduction of new hardware also demands change in software. The process of changing software after it has been delivered and is in use is called software maintenance. Maintenance is reported to be the costliest phase. Cost of maintenance can be up to 65-75% of total cost of software. A good measure of maintainability can help manage the software maintenance phase better. It cannot be measured adequately by only source code or documents. The thesis provides a new unified measure of software maintainability. The four parameters of which maintainability is an integrated measure are average live variables in a program, the average life span of variables, the comment ratio which is a measure of documentation quality and average cyclomatic complexity.

The model proposed here is a fuzzy model which is validated by project data for all parameters, as well as corrective maintenance time. Empirical results prove that integrated measure of maintenance proposed by this fuzzy model showed strong co-relation to the average the corrective maintenance time.

Software is a collection of various documents and source code. Understandability of the software is one aspect, which has a lot of impact on maintainability, and requires a good comprehension of logic of the source code. It is a measure of how well the programmer understands the software code as well as other related documents. If the software developer cannot correlate the documentation and
source code, maintenance is going to be very difficult. The fundamental reality that "you can't control what you can't measure", highlights the importance of a good measurement of software understandability. Two measures of spatial complexity, which are based on the important aspects of procedure, oriented program - code as well as data are combined to form the total spatial complexity. Code spatial complexity is a measure of the understanding of the source code. It depends on the psychological complexity of the source code and data spatial complexity is a measure of the cognitive abilities to comprehend various inputs, outputs and other intermediate results. The quality of the documentation associated with the code is as important as the code itself. Gunning's Fog index is a measure of readability of a passage of text.

An important factor contributing to the understandability of software is cohesiveness of source code and documents. An integrated measure of software understandability, which depends on total spatial complexity, documentation quality and cohesiveness source code and documents, is provided in this thesis. A neural network model that can handle the subjectivity of their individual contribution towards understandability is used.

Fuzzy as well as neural models have been proposed for measuring various software metrics in this thesis. To find out which model is better, a sensitivity analysis of the two models was carried out. It was carried out by finding out its
condition number where condition number is defined to be the maximum value of the relative error in the solution to the relative error in data over the problem domain. The lower the condition number, the better is the model or the more stable is the model.

Sensitivity Analysis was carried out with the help of a case study where the two models were used to measure software maintainability. To be able to do this the Fuzzy model proposed earlier to measure maintainability was used and further a neural network was modeled to measure maintainability. These two models were then evaluated for the condition number.

During working in this area of research a lot of scope for future work has been observed. The work done here is limited to procedure-oriented software, but this could further extended to object oriented software. Models proposed for software effort need empirical investigation. The concept of metrics has not been applied to web base software and client server software. Many metrics such as size, complexity, effort, maintainability and understandability can be proposed for these kinds of software and their relevance / validity can be investigated.

The hybrid approach in soft computing could definitely by used for example a neuro fuzzy system could be used where a neural network architecture can simulate a fuzzy system or alternatively a neural network could be used to adapt some parameters of the fuzzy system. In brief, this thesis has proposed a soft
computing approach to some metrics for procedure-oriented software to measure various software engineering practices. Various metrics for size, effort, maintainability and understandability have been proposed and their respective validation / usefulness has been presented in the respective chapters.