CHAPTER 5

5.1 PROPOSED MODEL

In the existing software reliability models the failure check is performed after the coding and the implementation phase. Sometimes due to faulty designs requires reimplementation of the project. It leads to wastage of the resources and the time. The proposed software reliability model performs its working in two phases. First phase of the model is completed before the coding after the design phase. In this phase the design is checked against the requirements. This phase uses the error back propagation of the neural network. The second phase of the model is placed after the implementation phase. This model uses the mean time failure and intensity to increase the reliability. The detail working of the model can be understood by following model:

Figure 5.1: Block Diagram of Software Reliability Model.
5.2 METHODOLOGY

The methodology complete in two phases one is training phase and the other is testing phase.

Training Phase:
1. Input the requirement analysis.
2. Input the design issues corresponding to the requirement analysis.
3. Train the network (calculate weight matrix) by using threshold activation function.

Testing Phase:
4. Input the requirement
5. Analyze the requirement
6. The requirement analysis is given as input to the neural network
7. Neural network process the requirement analysis and provides corresponding design issues.
8. The design issues are checked in the design.
9. If any error occur then design is updated and go to step 8
10. Perform Coding
11. Then get mean time failure

\[ m(t) = \left( f v = \frac{[1 - \exp(-\phi t)]}{[1 + \exp(-\phi t)]} \right) > 0? f v: 0 \]

12. Calculate intensity of failure

\[ \lambda(t) = \frac{dm(t)}{d(t)} \]

13. Remove failures.