CHAPTER 6

6.1 RESULTS AND DISCUSSION

The proposed methodology is analyzed in two manners. In the first way, library software is built from the initial phase and complete methodology is applied on the software for high reliability. The software is built for the Vaish College of Engineering Rohtak Haryana. In the second way, the proposed neural network based methodology is analyzed on the datasets downloaded from [5]. The dataset predicts the defects in the five modules of the NASA products. The NASA products under analysis are JM1, PC1, KM1, KC1, and KC2. The variables in this dataset are evaluated by using static measures i.e. prediction variables. The subsets in the dataset are prepared by classifying the set on the basis of size of module. This results in the high prediction performance.

6.1.1 Simulation Environment

MATLAB supports the entire data analysis process, from acquiring data from external devices and databases, through preprocessing, visualization, and numerical analysis, to producing presentation-quality output.

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and +. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.
MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

**Key Features**

1. High-level language for technical computing
2. Development environment for managing code, files, and data
3. Interactive tools for iterative exploration, design, and problem solving
4. Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
5. 2-D and 3-D graphics functions for visualizing data
6. Tools for building custom graphical user interfaces
7. Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java, COM, and Microsoft Excel

The MATLAB development environment lets you develop algorithms, interactively analyze data, view data files, and manage projects. MATLAB provides a high-level language and development tools that let you quickly develop and analyze your algorithms and applications.

**Develop algorithms using the high-level language and development tools in MATLAB:**

The MATLAB Language

The MATLAB language supports the vector and matrix operations that are fundamental to engineering and scientific problems. It enables fast development.
With the MATLAB language, you can program and develop algorithms faster than with traditional languages because you do not need to perform low-level administrative tasks, such as declaring variables, specifying data types, and allocating memory. In many cases, MATLAB eliminates the need for ‘for’ loops. As a result, one line of MATLAB code can often replace several lines of C or C++.

At the same time, MATLAB provides all the features of a traditional programming language, including arithmetic operators, flow control, data structures, data types, object-oriented programming (OOP), and debugging features.

A communications modulation algorithm that generates 1,024 random bits, performs modulation, adds complex Gaussian noise, and plots the result--all in just 9 lines of MATLAB code.

MATLAB lets you execute commands or groups of commands one at a time, without compiling and linking, enabling you to quickly iterate to the optimal solution.

**Optimize the performance and maintainability of MATLAB code:**

For fast execution of heavy matrix and vector computations, MATLAB uses processor-optimized libraries. For general-purpose scalar computations, MATLAB generates machine-code instructions using its JIT (Just-In-Time) compilation technology.

This technology, which is available on most platforms, provides execution speeds that rival those of traditional programming languages.
Development Tools:

MATLAB includes development tools that help you implement your algorithm efficiently. These include the following:

**MATLAB Editor** - Provides standard editing and debugging features, such as setting breakpoints and single stepping.

**M-Lint Code Checker** - Analyzes your code and recommends changes to improve its performance and maintainability.

**MATLAB Profiler** - Records the time spent executing each line of code.

**Directory Reports** - Scan all the files in a directory and report on code efficiency, file differences, file dependencies, and code coverage.

### 6.1.2 Analysis

![Fault Criticality in Software Project](image)

*Figure 6.1 : Fault Occurrence in Software Project*
Here figure 6.1 is showing the results of fault occurrence in a software project. Here the fault criticality is between 0 and 1. Higher the bar more critical the fault is.

Figure 6.2 : Fault Criticality in Software Project (Less Critical)

Here figure 6.2 is showing the results of low fault criticality in a software project.
Figure 6.3: Fault Criticality in Software Project (Medium Critical)

Here figure 6.3 is showing the results of low fault criticality in a software project.
Figure 6.4 : Fault Criticality in Software Project (High Critical)

Here figure 6.4 is showing the results of low fault criticality in a software project.
The quality is better in the class level data prediction as compared to method level data prediction. The defect prediction is more accurate in the large modules as compared to small modules. The present work uses the 60% of the dataset for the training purpose and rest for the testing purpose. The target of this work on this dataset is to find the software reliability by finding the defects accurately.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Data</td>
</tr>
<tr>
<td>JM1</td>
<td>0.65</td>
</tr>
<tr>
<td>PC1</td>
<td>0.63</td>
</tr>
<tr>
<td>KM1</td>
<td>0.64</td>
</tr>
<tr>
<td>KC1</td>
<td>0.63</td>
</tr>
<tr>
<td>KC2</td>
<td>0.61</td>
</tr>
<tr>
<td>Library Software</td>
<td>0.48</td>
</tr>
</tbody>
</table>
The graphical representation of the above values is shown below:

![Graph](image)

The reduce in the error confirms the better performance of the model.