<div align="right">

Chapter 6

</div>

TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is done in order to identify any gaps, errors or missing requirements. Testing is done at various levels and by various methods.

6.1 Testing Methods

There are many approaches available in software testing. A few of them are discussed in this section.

6.1.1 Static Testing

Static testing is a form of software testing where the software/program is not used. It involves reviewing of the code to find errors. It is not detailed but checks for the sanity of the code or algorithm. It includes reviews, walkthroughs and inspections.

6.1.2 Dynamic Testing

Dynamic testing is a form of software testing where the execution of programmed code is done with a given set of test cases. In dynamic testing, the program must be compiled and run. Unit tests, integration tests, system tests and acceptance test utilize dynamic testing.

6.1.3 White Box Testing

White box testing is a method of testing the internal structures of an application. In white box testing, an internal perspective of the system as well as programming skills are used to design test cases. It can be applied to unit, integration and system levels of the software testing process. It is usually done at the unit level.

6.1.4 Black Box Testing

Black box testing is a method of testing that examines the functionality of an application without peering into its internal structures. It can be applied to every level of testing like unit, integration and system although it is mostly used for higher level testing.

6.2 Testing Levels

A software testing strategy provides a road map for the software developer. Testing begins at the module level and works outward towards the integration of the entire computer based system. Different testing techniques are used. The commonly used testing strategies are Unit Testing, Integration Testing and System Testing.

6.2.1 Unit Testing

It is a testing method where individual units of code, set of one or more modules together with associated control data are tested to determine if they are fit to use. Test cases are defined to perform unit testing. The following are a few test cases considered-

Test Case 1: When window size (N) is lesser than the given threshold

This test case is checked by varying the value of N from its threshold value to a value significantly low. The following table summarizes the result of the test case.

Threshold Value (value of N)	Input (value of N)	Expected Output (in terms of accuracy)	Observed Output (in terms of accuracy)
100	40	<100%	62.5%

Table 6.1 Test Case 1

Result:The observed output is low in a practical perspective but numerically consistent.

Test Case 2: When window size (N) is greater than the given threshold

This test case is checked by varying the value of N from its threshold value to a value significantly high. The following table summarizes the result of the test case.

Threshold Value (value of N)	Input (value of N)	Expected Output (in terms of accuracy)	Observed Output (in terms of accuracy)
100	225	100%	100%

Table 6.2 Test Case 2

Result: The observed output matches with the expected output.

Test Case 3: When number of bands (K) is lesser than the given threshold

This test case is checked by varying the value of K from its threshold value to a value significantly low. The following table summarizes the result of the test case.

Threshold Value (value of K)	Input (value of K)	Expected Output (in terms of accuracy)	Observed Output (in terms of accuracy)
40	5	<100%	53.29%

Table 6.3 Test Case 3

Result: The observed output is low in a practical perspective but numerically consistent.

Test Case 4: When number of bands (K) is lesser than the given threshold

This test case is checked by varying the value of K from its threshold value to a value significantly high. The following table summarizes the result of the test case.

Threshold Value (value of K)	Input (value of K)	Expected Output (in terms of accuracy)	Observed Output (in terms of accuracy)
40	80	100%	100%

Table 6.4 Test Case 4

Result: The observed output matches with the expected output.

Test Case 5: When error proportion rate (t) is lesser than the given threshold

This test case is checked by varying the value of t from its threshold value to a value significantly low. The following table summarizes the result of the test case.

Threshold Value (value of t)	Input (value of t)	Expected Output (in terms of False Alarm Rate)	Observed Output (in terms of False Alarm Rate)
0.1	0.088	0%	1.61%

Table 6.5 Test Case 5

Result: The observed output matches closely with the expected output and acceptable.

Test Case 6: When error proportion rate (t) is greater than the given threshold

This test case is checked by varying the value of t from its threshold value to a value significantly high. The following table summarizes the result of the test case.

Threshold Value (value of t)	Input (value of t)	Expected Output (in terms of False Alarm Rate)	Observed Output (in terms of False Alarm Rate)
0.1	0.13	0%	0.92%

Table 6.6 Test Case 6

Result: The observed output matches closely with the expected output and acceptable.