Chapter 4

Research Methodology
## 4.1 Tools and Techniques

To accomplish the research goal, the data is collected from open software repositories. Data in these repositories provide us with historic information, which is stored at the time of development of the software and testing of the software products. The large number of open software repositories, such as, source control repositories, bug repositories, achieved communication repositories, deployment logs and code repositories are publically available. The finest part of these repositories is that they are a treasure house of actual software development and approved for the international software communities for standard researches in software intelligence. Hassan et.al. [61] mentioned the following table which consists of information about software development.

<table>
<thead>
<tr>
<th>Repositories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Control Repositories</td>
<td>Source Control Repositories are the most commonly available and used repository in software projects. CVS, Subversion, Perforce, Clear-Case are examples of source control repositories that are used in practice. These repositories record the development history of a project. They track all the changes to the source code along with meta-data about each change.</td>
</tr>
<tr>
<td>Bug Repositories</td>
<td>These repositories track the resolution history of bug reports or feature requests that are reported by users and developers of large software projects. Bugzilla and Jira are examples of bug repositories.</td>
</tr>
<tr>
<td>Repositories</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Achieved Communication Repositories</td>
<td>These repositories track discussions about various aspects of a software project throughout its lifetime. Mailing lists, Emails, IRC chats, and instant messages are examples of archived communications about a project.</td>
</tr>
<tr>
<td>Deployment Logs</td>
<td>These repositories record information about the execution of a single deployment of a software application or various deployments of the same applications. For example, the deployment logs may record the error messages reported by an application at various deployment sites.</td>
</tr>
<tr>
<td>Code repositories</td>
<td>These repositories archive the source code for a large number of projects. Sourceforge.net and Google code are examples of large code repositories.</td>
</tr>
</tbody>
</table>

**The main objectives of the research are:**

1. To select and analyze effective software metrics for bug prediction.
2. To select statistical learning and data mining techniques to be applied on historical software data for prediction of software defect.
3. To address various measures to prevent defects in future software version releases.

To achieve these objectives, the Research Framework for Software Defect Prediction using historical databases is used as shown in Figure 4.1.
Figure 4.1: Framework for Software Defect prediction using Historical Databases
4.2 Data Collection

Data plays a very critical role in any organization and hence the selection of data (or dataset) [62] is an extremely important component for any software system [63]. The data has to be in a reliable, clean and accurate form to enable a developer to reach to some conclusion. The bug prediction dataset is a group of models and metrics of software system and their histories. The dataset is a dynamic component to accomplish bug prediction at the class level, as it uses a number of metrics, which can be used to create generalized linear regression models and the number of post-release defects. The performance of these models is evaluated by comparing the prediction results against the actual post-release defects provided as part of the dataset. The different datasets like NASA MDP repository [64,65,66,67], Promise Repository [68,69,70,71] [Marco D’Ambroset.al. [72]have created their own dataset by the following tools infusion(available at http://www.intooitus.com), Moose(available at http://www.moosetechnology.org), Churrasco (available at http://churrasco.inf.usi.ch).

The Open Science Promise Repository (available at http://openscience.us/repo/defect/ck/), a publicly available dataset is taken for the research. The product metrics, that is Chidamber and Kemerer metrics suite have been used and the various software modules were taken to discover the most significant or relevant software metrics. The software modules taken from CK & OO metrics [73] (Donor-Marian Jureckzo) were as follows:

Software modules used in the paper are named as Ant, Ivy, Tomcat, Berek, Camel, Lucene, POI, Synapse and Velocity as shown in Table 4.2.
### Table 4.2: Software Modules [74] [75]

<table>
<thead>
<tr>
<th>Software Module</th>
<th>Version</th>
<th>Observation</th>
<th>Variables</th>
<th>Description</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ant</td>
<td>1.7</td>
<td>745</td>
<td>21</td>
<td>Ant is a Java library and command-line tool, to build Java and non-Java applications.</td>
<td><a href="http://ant.apache.org">http://ant.apache.org</a></td>
</tr>
<tr>
<td>Ivy</td>
<td>2</td>
<td>352</td>
<td>21</td>
<td>Ivy is a tool for managing project dependencies. It is flexible and tightly integrated with Apache ant</td>
<td><a href="http://opencourse.u.s/repo/defect/ck/ivy.html">http://opencourse.u.s/repo/defect/ck/ivy.html</a></td>
</tr>
<tr>
<td>Tomcat</td>
<td>6.0</td>
<td>858</td>
<td>21</td>
<td>Tomcat is an open source implementation of the java servlet, JSP, Java web-socket specifications.</td>
<td><a href="http://tomcat.apache.org">http://tomcat.apache.org</a></td>
</tr>
<tr>
<td>Berek</td>
<td>1</td>
<td>44</td>
<td>21</td>
<td>Open source academic projects.</td>
<td><a href="http://opencourse.u.s/repo/defect/ck/ber">http://opencourse.u.s/repo/defect/ck/ber</a> ek.html</td>
</tr>
<tr>
<td>Camel</td>
<td>1.6</td>
<td>965</td>
<td>21</td>
<td>Camel is a library in Java with minimal dependencies for Embedding java application and support bean binding.</td>
<td><a href="http://came.l.apache.org">http://came.l.apache.org</a></td>
</tr>
<tr>
<td>Software Module</td>
<td>Version</td>
<td>Observation</td>
<td>Variables</td>
<td>Description</td>
<td>URL</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
<td>-----</td>
</tr>
<tr>
<td>Lucene</td>
<td>2.4</td>
<td>340</td>
<td>21</td>
<td>Lucene is an open source high performance text search engine library written in java.</td>
<td><a href="https://lucene.apache.org">https://lucene.apache.org</a></td>
</tr>
<tr>
<td>POI</td>
<td>3</td>
<td>442</td>
<td>21</td>
<td>POI is employed to create and maintain Java APIs for manipulating various file formats like to read and write MS Excel and MS Power Point file using Java.</td>
<td><a href="https://poi.apache.org">https://poi.apache.org</a></td>
</tr>
<tr>
<td>Synapse</td>
<td>1.2</td>
<td>256</td>
<td>21</td>
<td>Synapse is a light-weight, high performance, free and open source software.</td>
<td><a href="http://synapse.apache.org">http://synapse.apache.org</a></td>
</tr>
<tr>
<td>Velocity</td>
<td>1.6</td>
<td>229</td>
<td>21</td>
<td>Velocity is a Java-based template engine and powerful template language to reference objects defined in Java code.</td>
<td><a href="http://velocity.apache.org">http://velocity.apache.org</a></td>
</tr>
</tbody>
</table>

### 4.3 Data Cleaning

Data Cleaning is supportive in improving the quality of the data as it deals in detecting inconsistencies, removing errors, missing values. Rahm et.al [76] in their paper addresses the issue of data cleaning, which is a foremost part of
extraction, transformation, loading (ETL) process in a data warehouse. There are a variety of tools available to clean the data, but at times, a major portion of the data needs to be cleaned manually which are tough to write and maintain. Though various types of tools are available, yet there is a complexity in a cleaning problem.

4.4 Feature Selection

In a larger dataset, all the variables are not so important to consider, the more the number of variables, the complexity will be on the increase. Therefore, it is always desirable to reduce the variables and should include important variables in a dataset. Through a Feature Selection technique, we can reduce the variable and locate the importance of the variable in a dataset. There are various Feature Selection techniques that are available in this paper. The Feature Selection techniques, like Boruta [77], regsubsets [78], and FSelector[79] were employed to discover the vital software metrics in a software system.

**BORUTA:** Boruta[80] is one of the most significant Feature Selection packages to explore the relevant element from a large dataset. It uses a Wrapper algorithm, which is better than the filter method as in the Wrapper Method classifier is used as a black box returning a feature ranking. The R package Boruta is available at [http://CRAN.R-project.org/package=Boruta](http://CRAN.R-project.org/package=Boruta). When the R package Bouruta is in operation, it displays the entire confirmed variable and the rejected variable in a dataset. When the box plot is drawn Green, Blue and Red represent a Z- Scores of confirmed, minimal or average, rejected attribute respectively.

**REGSUBSETS:** regsubsets[81][82] are used for regression subsets selection to come across the model that ideally suits the data by calculating its R2, AIC and BIC values. The model can be ranked according to adjusted R2 criteria and BIC, when the graph is being plotted as shown in Figure 4.2, there are two indicators, black and white. Black indicates that the variable is included in the model and white signifies that it is excluded in the model.
**FSELECTOR:** FSelector[83] [84] is also considered to be one of the important selection functions for selecting the attributes from a dataset. This function is used to find the irrelevant and redundant attribute as much as possible from a given dataset. FSelector consists of Feature Selection algorithm, like wrapper and filter. The Wrapper method uses a predictive model and trains a new model for each subset. The Wrapper Algorithm used in FSelector is best first search, backward search, forward search, hillclimbingsearch. The Filter Method uses a proxy measure to score a feature sub-set. Filters are usually less computationally intensive than wrappers. It is generally used as feature ranking. Filter methods have also been applied as a pre-processing step for wrapper methods. Filter method used in FSelector are CFS, chisquared, information gain, gain ratio, symmetrical uncertainty, linear correlation, rankcorrelation, oneR, relief, consistency, random forest importance. [85] Other algorithm used are cutoff. k, cutoff. k. percent, cutoff. biggest. Diff as. simple. formula. Filter method Random Forest, Information Gain, Linear Correlation and Rank Correlation were used.

### 4.5 Modeling Technique

Machine learning is a branch of Artificial Intelligence, which will build a system that learns from and makes predictions on the data. Machine learning has turned out to be one of the intense topics as every person wants to build an intelligent application [86] or make the prediction accurately. Machine learning can also be defined as learning from the past experience so that better or correct techniques can be applied in the future. Machine learning can be classified as unsupervised learning and supervised learning as shown in Figure 4.2.
Unsupervised learning is a technique to discover the hidden patterns in input data. Clustering is however, an unsupervised learning technique. Supervised learning [87] is used when there is a requirement to train the model to make a prediction. Supervised learning can be categorized into two types: Regression and Classification.

The machine learning technique is an important branch of computers for software bug prediction. This research applied regression technique on machine learning models to predict the best model for the software bug prediction. Six machine learning models Linear Regression, Random Forest, Decision Tree, Support Vector Machine, Neural Network and Decision Stump were used to find the most optimal machine model.

**Linear Regression**

Linear Regression is generally used for the predictive analysis. This model finds the relationship between Response Variable (dependent variable) and a single or more Explanatory Variable (independent variable).

**Random Forest**
Random Forest [88] is one of the ensemble learning user-friendly methods [89] used in the prediction for better performance and can be applied for the software bug prediction. It is also employed to grade the prominence of the variable. This paper has used Random Forest as one of the Feature Selection techniques to locate the important software metrics. In Random forest, each node is split and randomly chosen to explore the best predictor. It uses two parameters, the number of variables in the random subset at individual node and the number of trees in the forest. The parameters used in the random forest are the quantity of variables in the random subset at every node and the number of trees in the forest. [90].

**Decision Tree**

`Decision tree is one of the most supervised learning methods used in the classification and regression for predictive modeling approach [91]. It is a dominant device for variable analysis. Decision tree has the capability to handle datasets which have errors and missing values [92]. However, one of the major disadvantages of the decision tree is oversensitivity to the teaching set, which is not relevant or noisy data.

**Support Vector Machine**

The Support Vector Machine is supervised machine learning used in the classification, regression [93] and outlier’s detection. The SVM works well when data sets are diminutive since the requisite training time is reduced. If the data sets are less noisy, it provides a suitable model [94]. A SVM is applied in several applications, like Face Recognition field [95], Optical Character Recognition [96], Spam Categorization [97], Financial Time Series Forecasting [98] etc.

**Neural Network**

Neural Network can be used to find the correlation between input and output, to predict software defects and find the pattern.

**Decision Stump**

Decision Stump is a machine based learning model with a single level decision tree [99]. As compared to the decision tree, the decision stumps are easily
constructed. A Decision stump generally gives the best result or continues to improve when feature selected has useful values.

**Research Methodology**

Research methodology is also by far one of the important components to achieve the goal of any system. A major purpose of the paper is to find the best machine based learning model for software bug prediction.

**Mathematical Model Used**

One of the machine-learning techniques is a regression [100] which can be made use of to formulate the prediction model [101]. In this regression method were applied such as Linear Regression, Random Forest, Decision-Tree, Support Vector Machine, Neural network, Decision Stump to explore the most optimum machine model. The objective of the research paper is achieved by applying different performance parameters [102] R squared, correlation, mean square error and accuracy of different machine learning models on the dataset.

The Performance parameters were calculated as follows:

- **Correlation (corr) [103]**

Correlation can be well-defined as the association between the actual and predicted values. It lies between 0 and 1 and the values that tend towards 1 are considered as really the best. The Mathematical representation is provided as follows:

\[
corr = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}
\]  

Where, \(x =\) Actual Value; \(y =\) Predicted Value; \(\bar{x} =\)mean of all the actual values; \(\bar{y} =\)mean of all the Predicted values; \(n =\) total number of instances
RSquared (R^2)

RSquared is recognized as the coefficient of determination. The coefficient of determination R-Squared is applied to investigate how the variances in one variable can be explained by an alteration in a second inconstant and determine as percent. The higher the coefficient, the higher percentage of points, the line passes through when the data points and lines are plotted. The values of 1 or 0 would indicate the regression line represents all or none of the data respectively.

\[ R^2 = \text{corr} \times \text{corr} \]  

Mean Square Error (MSE)

Mean Square Error is applied to compute the error-rate of a regression model.

\[ MSE = \frac{\sum_{i=1}^{n} (p_i - a_i)^2}{n} \]  

Where, a = actual target; p = predicted target; n = total number of instances

Accuracy

The accuracy is calculated as the percentage deviation of predicted target with actual target with acceptable error.

\[ \frac{100}{n} \sum_{i=1}^{n} q_i \]

\[ q_i = \begin{cases} 1 & \text{if } |abs(p_i - a_i)| \leq \text{err} \\ 0 & \text{otherwise} \end{cases} \]  

Where, a = actual target; p = predicted target; n = total number of instances; err=acceptable error