**SYSTEM ANALYSIS**

**EXISTING SYSTEM:**

* Recently, researchers have developed methods that concentrate on ranking source files for given bug reports automatically.
* Saha et al. syntactically parse the source code into four document fields: class, method, variable, and comment. The summary and the description of a bug report are considered as two query fields.
* Kim et al. propose both a one-phase and a two-phase prediction model to recommend files to fix. In the one-phase model, they create features from textual information and metadata (e.g., version, platform, priority, etc.) of bug reports, apply Na€ıve Bayes to train the model using previously fixed files as classification labels, and then use the trained model to assign multiple source files to a bug report.
* Rao and Kak apply various IR models to measure the textual similarity between the bug report and a fragment of a source file.

**DISADVANTAGES OF EXISTING SYSTEM:**

* Their one-phase model uses only previously fixed files as labels in the training process, and therefore cannot be used to recommend files that have not been fixed before when being presented with a new bug report.
* Existing methods require runtime executions.

**PROPOSED SYSTEM:**

The main contributions of this paper include: a ranking approach to the problem of mapping source files to bug reports that enables the seamless integration of a wide diversity of features; exploiting previously fixed bug reports as training examples for the proposed ranking model in conjunction with a learning-to-rank technique; using the file dependency graph to define features that capture a measure of code complexity; fine-grained benchmark datasets created by checking out a before-fix version of the source code package for each bug report; extensive evaluation and comparisons with existing state-of-the-art methods; and a thorough evaluation of the impact that features have on the ranking accuracy.

**ADVANTAGES OF PROPOSED SYSTEM:**

* Our approach can locate the relevant files within the top 10 recommendations for over 70 percent of the bug reports in Eclipse Platform and Tomcat.
* Furthermore, the proposed ranking model outperforms three recent state-of-the-art approaches.
* Feature evaluation experiments employing greedy backward feature elimination demonstrate that all features are useful.