Systematic Literature Review on Mining Software Repositories

Arvinder Kaur, Kamaldeep Kaur, Deepiti Chopra and Harguneet Kaur
University School of Information and Communication Technology (U.S.I.C.T), Guru Gobind Singh Indraprastha University (G.G.S.I.P.U), New Delhi, India
Email:arvinder@ipu.ac.in, kdkaur99@ipu.ac.in, dchopra27@gmail.com, harguneet.13316490018@ipu.ac.in

Abstract
Mining Software Repositories (MSR) explores the complex software systems to unfold valuable and interesting knowledge. The systematic literature review is performed on MSR studies published over a decade. The review is conducted on 300 selected papers. The study aims to identify – i) popular application areas, ii) popular tools, iii) datasets and software projects, along with their type and SDLC phase in which they are used, iv) popular techniques, v) top conferences and journals along with the types of studies and vi) current trends, of MSR research. The important results implied from the reviewed studies are: 1) Bug Prediction (25%) and Change Prediction (17%) are the most popular application areas of MSR, 2) Data Modelling and Statistical Tools, (19%) were employed in most of the reviewed studies, 3) About 71 % of reviewed studies use datasets from open source repositories, 4) There is a large number of mining techniques with Classification techniques (29%) dominating the field, 5) The most popular conference for MSR is the International Conference on Mining Software Repositories (MSR), and 6) Research gaps in evolving application areas of clone detection, code reuse and software evolution are identified.

Keywords: Systematic literature review, Mining software repositories, Software Repositories, Software Engineering

1. INTRODUCTION
Mining Software Repositories (MSR) is a dynamic research field that deals with extracting and analyzing the data available in software repositories. As the data is readily available with bug tracking repositories, source code repositories, control versioning systems (CVS) and mailing lists, this emerging field has extended its reputation since 2004, with the first MSR workshop and sustains to be a growing research area in software engineering. Most of the research work done in software engineering uses the data available in various software repositories. Some frequently examined areas include software defect prediction, determining software change patterns, the study of code clones, code reuse, software evolution and software performance analysis.

The purpose of this study is to explore the various techniques and tools that have been majorly used for MSR, determine popular approaches for MSR, identify the commonly explored repositories of the software projects and find out the current trends in the MSR research area. Our study not only investigates the literature in the context of software evolution but also reviews papers on all application areas of MSR. Apart from examining the techniques, repositories and methodology, our study also identifies the popular tools used for mining data repositories and determines the Software Development Lifecycle (SDLC) phase at which data is extracted.

Huzefa Kagdi et al. lead “A survey and taxonomy of approaches for mining software repositories in the context of software evolution”. Their paper examines the MSR field based on the four views: type of software repository, the aim, the approach used, and the assessment method. All application areas are explained in detail by the survey. They also described the usage of metrics in MSR.

K.K. Chaturvedi et al. carried out a review of “Tools in Mining Software Repositories”. Quality papers from the Mining Software Repositories (MSR) conferences/journals from 2007 to 2012 were reviewed and analyzed by K.K. Chaturvedi et al. [272]. In their work [K.K. Chaturvedi et al.], techniques, data sets and tools used are identified. Their paper categorizes different tools on the basis of the application areas and the datasets it works on. This review considers papers published in the proceedings of MSR conference and other related conferences/journals from the year 2003 to 2018.

Hadi Hemmati et al., gathered comments from MSR proceedings between 2005 and 2012 and analyzed and categorized them. After analysis, four themes were determined namely: “data acquisition and preparation, synthesis, analysis, and sharing/replication.” The trends in specific theme were identified and explored. The review in [Hadi Hemmati et al.]
provides a public forum consisting of the extracted patterns for enhancing the interaction among the MSR community for evolving best methodologies. However, none of the above papers [Huzefa Kagdi et al., K.K. Chaturvedi et al., Hadi Hemmati et al.], follow the systematic approach for conducting the Systematic Literature Review (SLR) according to the principles established by Kitchenham and Charters [Kitchenham and Charters], commonly recommended in software engineering systematic reviews. Our current paper follows the SLR principles [Kitchenham and Charters] and identifies which techniques and tools are prominently used in previous research, determining what all approaches for MSR have been studied, identify the most commonly explored repositories of the projects, explore the recent trends and identifies the research gaps in various application areas of MSR.

Some of the issues which have not been identified by the above mentioned papers [Huzefa Kagdi et al., K.K. Chaturvedi et al., Hadi Hemmati et al.] are answered in our paper including the quality assessment (QA) [Kitchenham and Charters] of the reviewed papers. QA is an essential step to discover valid and high quality papers for our analysis. The previous reviewed studies [Huzefa Kagdi et al., K.K. Chaturvedi et al., Hadi Hemmati et al.] did not identify the SDLC phase at which the repository is mined where as our study examines MSR at each SDLC phase. It is important to determine the correct SDLC phase for mining the required data. This study correlates the application areas and datasets of MSR with the SDLC phase from which data needs to be extracted, thereby providing a framework for new researchers who want to work in MSR research area.

This paper is composed as follows. Section 2 describes the methodology used, Section 3 discusses the review results, Section 4 provides the implications for research and practice, Section 5 discusses the threats to validity and Section 6 provides conclusion and future work.

2. METHOD

The Systematic Literature Review (SLR) process given by ‘Kitchenham and Charters’ is followed in our current study. In the earliest planning phase of the SLR, the systematic review protocol is formulated. The formulated protocol consists of six major steps:
- Research questions
- Search protocol
- Study selection
- Quality assessment
- Data extraction
- Data synthesis.

Figure 1 Steps of Review Model
Figure 1 depicts the six steps of the systematic review model. In the first step, research questions were developed. In the second step, the search strategy was planned, to find out the primary research papers related to the questions formulated in step 1. It involves identifying the search string and determining the databases to be searched. Then in the third step, the study selection criteria was planned to find the appropriate studies that contribute significantly to the research questions. Pilot study selection was also applied at this step to further filtering of the selected studies. In step 4, the selected papers were assessed for quality according to the defined quality checklist mentioned in section 2.3. Then at the data extraction step, data extraction card is designed and refined using pilot data extraction i.e. the form is tested on a representative sample of studies to be reviewed and is refined to collect desired information. The data extraction card is then planned to be used for extracting data from the selected studies. Finally, in the data synthesis step, the proper process for synthesizing the data to extract useful information based on the research questions is to be determined. A systematic review model is crucial for an SLR. The subsections 2.1, 2.2, 2.3, 2.4, 2.5 describe the review model in detail.

2.1. Research questions

The formulation of research questions (RQ) is done in such a way so that any researcher who wants to work in the field of MSR gain knowledge of popular application areas, the tools available for MSR, the datasets along with its types and software projects on which previous studies have been conducted. It also determines the SDLC phase in which dataset is used, popular MSR techniques implemented by researchers and conferences and journals in which different types of MSR papers are published. These RQ provides future directions and framework for new researcher. 6research questions (RQs) addressed in this study are:

i. **RQ1: Which application areas have been explored in MSR papers?**

   RQ1 aims at identifying the application areas in which MSR has been used, for example bug prediction, change prediction, clone detection, code reuse, software performance analysis, software evolution, etc. It helps in examining the current trends in MSR research along with analyzing popular application areas for further research in MSR field.

ii. **RQ2: Which are the tools used to mine repositories?**

   RQ2 tries to identify popular mining repository tools for the researchers and the software industry for extracting information from different repositories. Researchers in the field of MSR use different types of tools such as eclipse plugins, bug tracking tools, diff, parser, statistical tools, extraction tools and others. We also analyze the repositories on which these tools work, along with the application areas in which these tools are used.

iii. **RQ3: Which datasets and software projects are used for MSR studies? What type of datasets are used and in which SDLC phase dataset is available?**

   Datasets such as production data, archived data between project personnel, source code data and bug tracking systems are utilized to record the evolution of various software systems. Researchers and developers identify the perks of mining the repositories for controlling defects and changes in future software development, improving design and development issues so as to increase the efficiency of projects. So the knowledge of popular datasets is essential for an effective MSR research. It is important to analyze which datasets are primary i.e. directly collected and which are secondary i.e. indirectly collected. We also identify the type of datasets used i.e. Open Source, Industrial or Partial This helps in analyzing which datasets are available freely to researchers specially those belonging to academics.

   Popular software projects are also recognized in our study so that they can be used in other studies as a benchmark. It is easier to conduct comparative studies based on previously studied software projects. It is also significant to determine the software life cycle phase at which data sets are available. It will help researchers in analyzing at which phase of SDLC most datasets are available for extraction.

iv. **RQ4: What are the most investigated techniques for MSR?**

   Researchers want to develop and identify mining techniques that can be used for extracting hidden and valuable information from unstructured software datasets. Our study investigates the popular mining techniques used by researchers. It also identifies which techniques are used in which application areas, thereby providing the researchers working in those areas with a list of useful mining techniques.
v. RQ5: What are the different types of MSR studies? Which journals and conferences are popular in the field of MSR research?

MSR papers are categorized as theory, survey, experiment, review, development of tool, development of mining technique and case study. MSR community is growing at fast rate which has an impact on software research. It is important to distinguish different types of studies so as to comprehend the current challenges which will reflect the future work for all leading research in this area.

This SLR also determines the popular journals and conferences for MSR. This is the most beneficial research question for the researchers working in the field of MSR as it provides them with the details of relevant MSR conferences and journals. Researchers in MSR field follow a quantitative methodology for answering research questions. In last few years, MSR field is gaining its ground and has become one of the expanding area of software engineering.

vi. RQ6: What are the research gaps in various application areas of MSR?

Research gaps recognize the missing elements in the current research in a particular field of study. Finding gaps in the literature is a difficult task but it enlightens the areas that have scope for exploration and research in future. These research gaps should be filled by new research in the field of MSR.

2.2. Search protocol

The search protocol of SLR includes selection of search string and database resources which are explained in the following subsections.

2.2.1. Search string

In order to determine the search string, we have used the following procedure:

- We searched majorly used terms from the research questions.
- Derived alternative synonyms and spellings of these terms.
- Identified the keywords in related papers.
- The Boolean OR operator is used to combine the synonyms of the main terms.
- Boolean AND operator is used to join the main terms.

The keywords and their synonyms which are used for searching the research papers are mining, mine, software, systems, repository, repositories, evolution, analysis, change, bug, fault, defect and prediction. Thus search string finally formalized is given below:

Search string:

\[
[\text{(''Mine'' OR ''Mining'') AND (''Software'' OR ''Systems'') AND (''Repositories'' OR ''Repository'')}] \\
\text{OR} \\
[\text{(''Software'' OR ''Systems'' AND (''Evolution'' OR 'Analysis'' OR ``Refactoring''))}] \\
\text{OR} \\
[\text{(''Change'' OR ''Bug'' OR ''Fault'' OR ''Defector'' OR ''Clone'') AND (''Prediction'' OR ''Detection'')}] \\
\text{OR} \\
[\text{(''Code'' OR ''Software'' AND (''Reuse'' OR ''Reusability''))}
\]

2.2.2. Selection of resources

The six electronic databases which are searched are shown in Table 1. Studies are extracted from these databases using our search process. The search process comprises of the following phases:

Table 1 Selected Databases

<table>
<thead>
<tr>
<th>Source</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE Explore</td>
<td><a href="http://ieeexplore.ieee.org">http://ieeexplore.ieee.org</a></td>
</tr>
<tr>
<td>ACM Digital Library</td>
<td><a href="http://portal.acm.org">http://portal.acm.org</a></td>
</tr>
<tr>
<td>Springer Link</td>
<td><a href="http://www.springerlink.com">http://www.springerlink.com</a></td>
</tr>
<tr>
<td>MSR</td>
<td><a href="http://www.msrconf.org">http://www.msrconf.org</a></td>
</tr>
</tbody>
</table>
Search phase 1: The six electronic databases were searched using the above defined search string to obtain papers related to the field of MSR. The inclusion and exclusion criteria were applied on the extracted studies to select relevant studies.

Search phase 2: We applied snowballing analysis for finding new papers through the analysis of the references of the papers extracted using the search string. The inclusion and exclusion criteria were applied on these related papers to further identify the relevant studies. The inclusion criteria (IC) and exclusion criteria (EC) of the systematic literature review are as follows:

**Inclusion criteria:**
- The studies on the field of MSR.
- The studies that propose or compare MSR techniques.
- The studies that utilize MSR techniques to develop efficient methods for different MSR application areas.
- The studies that explain the development or implementation of a software tool that can be used for MSR.

**Exclusion criteria:**
- The studies that are not related to MSR.
- The studies that are not written in English.
- Summaries of tutorials, panels, editorials, posters and prefaces.

562 papers were initially found on MSR. The after applying IC and EC, 321 studies are selected for further reading. This search process is also known as replication package.

Table 2 provides statistics of obtained studies, studies included after removing duplicate studies, and the rate index. Rate Index is defined as proportion of included studies to the total number of studies considered for the systematic review.

**Table 2 Obtained and Included Papers**

<table>
<thead>
<tr>
<th>Database</th>
<th>Obtained</th>
<th>Included</th>
<th>Rate Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE xplore</td>
<td>195</td>
<td>137</td>
<td>42.68</td>
</tr>
<tr>
<td>ACM digital</td>
<td>155</td>
<td>67</td>
<td>20.87</td>
</tr>
<tr>
<td>Springer link</td>
<td>60</td>
<td>38</td>
<td>11.84</td>
</tr>
<tr>
<td>MSR</td>
<td>78</td>
<td>49</td>
<td>15.26</td>
</tr>
<tr>
<td>Elsevier</td>
<td>53</td>
<td>24</td>
<td>7.48</td>
</tr>
<tr>
<td>Google Scholar</td>
<td>30</td>
<td>6</td>
<td>1.87</td>
</tr>
</tbody>
</table>

2.3. Study quality assessment

The Quality Assessment (QA) criteria were enforced to the research papers so as to select the papers with acceptable quality, which were eventually used for data extraction. A number of questions were formulated to assess the quality and relevance of the selected papers. QA questions are defined on the basis of different parameters: research questions, application areas, tool, dataset, techniques and research methodology.

These QA questions are presented in Table 3. Each question has one of the three answers: “Yes”, “Partly/Not Relevant”, or “No” which have been given score as: “Yes” = 1, “Partly/Not Relevant” = 0.5, and “No” = 0. For a given paper, its quality score was calculated by summing up the scores of the answers to the QA questions. The studies which have the quality score greater than 8 are considered reliable with acceptable quality and further used for the subsequent data extraction and data synthesis. The summary of quality assessment is presented in Table 11 in Appendix A.
### Table 3 Quality Assessment Questions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>No.</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research Questions</td>
<td>QA1</td>
<td>Are the research questions clearly stated?</td>
</tr>
<tr>
<td></td>
<td>QA2</td>
<td>Are the research questions justified in the context?</td>
</tr>
<tr>
<td></td>
<td>QA3</td>
<td>Are all the research questions answered?</td>
</tr>
<tr>
<td>Application Area</td>
<td>QA4</td>
<td>Does the paper contribute significantly to the application area?</td>
</tr>
<tr>
<td>Tool</td>
<td>QA5</td>
<td>Is the usage of tools adequately described?</td>
</tr>
<tr>
<td></td>
<td>QA6</td>
<td>If the study involves tool development, then is the development of a tool explained clearly?</td>
</tr>
<tr>
<td></td>
<td>QA7</td>
<td>If the study involves tool development, then is its performance assessed?</td>
</tr>
<tr>
<td></td>
<td>QA8</td>
<td>Is the tool benchmarked against existing tools?</td>
</tr>
<tr>
<td>Dataset</td>
<td>QA9</td>
<td>If the study involves data collection, are the data extraction methods defined?</td>
</tr>
<tr>
<td></td>
<td>QA10</td>
<td>If the study involves analysis, are the data sets adequately described?</td>
</tr>
<tr>
<td></td>
<td>QA11</td>
<td>Is the data available or can be collected by the method defined?</td>
</tr>
<tr>
<td>Techniques</td>
<td>QA12</td>
<td>Is the purpose of the technique defined clearly?</td>
</tr>
<tr>
<td></td>
<td>QA13</td>
<td>Is the result of the used technique clearly stated?</td>
</tr>
<tr>
<td></td>
<td>QA14</td>
<td>Is the result of the technique compared with other techniques?</td>
</tr>
<tr>
<td>Research Methodology</td>
<td>QA15</td>
<td>Is research methodology clearly described?</td>
</tr>
<tr>
<td></td>
<td>QA16</td>
<td>Are the threats to validity of the study clearly presented?</td>
</tr>
<tr>
<td></td>
<td>QA17</td>
<td>Does the results of the study are further added to the literature?</td>
</tr>
<tr>
<td></td>
<td>QA18</td>
<td>Does the study discuss the issues related to validity/reliability of their measures?</td>
</tr>
</tbody>
</table>

### 2.4. Data extraction

After the selection of papers, data extraction card is formed as shown in Table 4. It is used to gather relevant information from the selected papers. This table is formed through pilot data extraction, i.e., the data extraction card is tested on representative samples from the studies to be reviewed and refined to collect desired information. Items in Table 4 are written according to the associated research questions which help in data synthesis step. During the data extraction, some data could not be extracted directly from the selected studies. Nevertheless, we could
obtain them indirectly by processing the available data appropriately. Not every selected study provides answers to all the 6 research questions. For ease of tracing the extracted data, we explicitly labeled each study with the IDs of the research questions to which the study can provide the corresponding answers. Some of the research questions may have multiple answers, for example one paper may apply two techniques for mining the software repositories, and so to resolve this issue, reviewed paper has been counted in both the techniques.

<table>
<thead>
<tr>
<th>Article Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year of publication(2003-2018)</td>
</tr>
<tr>
<td>Name of authors</td>
</tr>
<tr>
<td>Source</td>
</tr>
<tr>
<td>Application areas explored in MSR papers (RQ1) Bug Prediction, Change Prediction, Software Evolution, Software Performance Analysis, Clone Detection and Code Reuse</td>
</tr>
<tr>
<td>Tools used to mine repositories (RQ2) (Bug Tracking tools, Data Modeling and Statistical tool, Extraction tool)</td>
</tr>
<tr>
<td>Which data repositories are mined (NASA, PROMISE, GitHub, Jira, FOSS, GCC, Sourceforge, StackOverflow, Bugzilla)</td>
</tr>
<tr>
<td>Which software project are mostly used (Eclipse, Firefox, GNOME desktop suite, Net Beans, Android, Apache, FreeBSD, PostgreSQL, Linux, ArgoUML and others)</td>
</tr>
<tr>
<td>Distribution of Type of datasets used in MSR papers (Open Source, Industrial, Partial and None)</td>
</tr>
<tr>
<td>Identify software life cycle phase at which data sets is gathered (Requirements, Design, Development, Testing and Maintenance) (RQ3)</td>
</tr>
<tr>
<td>Investigated techniques for MSR (RQ4) (Association Rule, Regression, Text Mining, Clustering, Frequent Pattern Matching, Classification, Machine Learning)</td>
</tr>
<tr>
<td>Type of MSR study (theory, survey, experiment, Implementation, review, development of tool, development of mining technique and case study)</td>
</tr>
<tr>
<td>The research gaps in various application areas (RQ6) (Clone Detection, Code reuse, Software evolution)</td>
</tr>
</tbody>
</table>

2.5. Data Synthesis

The purpose of data synthesis phase is aggregation of information from the selected papers to provide response to the specified research questions. One piece of information alone might not lead to a conclusive answer, so we need to aggregate the extracted information. The data synthesis strategy used for this review is the narrative synthesis method [275], i.e., the studies are organized into homogenous groups and the characteristics, quality, context and results of the studies are reported in a standardized format. In narrative synthesis, the extracted data is tabulated in accordance with the research questions. Visualization tools such as pie chart, line graph, bar chart and box plot, are used in our study to report and present the results.
3. Results and Discussion

An overview of the studies selected for the review is given in this section. Then the findings of the review are presented and discussed. Each research question is answered in a separate subsection. The results of the review are interpreted not only in the context of the respective research questions, but also in the wider context linked to all research questions.

3.1. Overview of selected studies

A total of 300 studies are reviewed in our SLR. They consist of papers spanning over entire decade (2003-2018). Figure 2 shows a graph indicating the number of publication in each year. The first workshop on MSR (held in 2004) helped in naming a field that was arising in software engineering research. MSR field continues to be a rapidly growing area in software research thus we have included majority of MSR papers after 2004.

![Number of Papers per year](image)

Figure 2 Number of Research Papers per year in review

After applying IC and EC, 321 studies were selected on which QA was performed. The studies with quality score greater than 8 were reviewed. 21 papers were rejected because their quality score was found to be less than 50%. Therefore majority of the studies selected have high quality. As mentioned in Table 5, about 61.05% (196 of 300) of the studies selected for review have a very high or high quality level.

<table>
<thead>
<tr>
<th>Quality level</th>
<th>Number of studies</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very high (15≤score≤17)</td>
<td>66</td>
<td>20.56</td>
</tr>
<tr>
<td>High (12≤score≤14.5)</td>
<td>130</td>
<td>40.49</td>
</tr>
<tr>
<td>Medium (8≤score≤11.5)</td>
<td>104</td>
<td>32.40</td>
</tr>
<tr>
<td>Low (5.5≤score≤8)</td>
<td>12(rejected)</td>
<td>3.74</td>
</tr>
<tr>
<td>Very low (0≤score≤5)</td>
<td>9(rejected)</td>
<td>2.81</td>
</tr>
<tr>
<td>Total</td>
<td>321(300 included 21 rejected)</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 5 Quality Levels of relevant studies

3.2. Application areas explored in MSR papers (RQ1)

The observations made after reviewing 300 papers in the field of MSR suggest that it has many applications. Below is a brief description of popular applications areas of MSR:

- **Bug Prediction**: A bug indicates an error or fault in the software system that may cause the system to behave unexpectedly. Bug Prediction uses the bug history of software system to predict bugs that may be introduced in future. Most of the work in bug prediction predicts bugs during design and coding phase.

- **Change Prediction**: Change prediction helps developers by predicting program entities that are likely to be
changed in next version or next phase of software development.

- **Clone detection**: Code clones are pieces of code that are identical to each other. Researchers employ mining techniques for detection of similar fragments in project. Clone detection is usually performed in design and development phase of a SDLC.

- **Code reuse**: As a large amount of code is freely accessible through the Internet, there is a chance that a given problem has been solved already, and its implementation is open source. It is essential to mine the software code repositories of previous versions during the design and development of the next reuse.

- **Software Evolution**: Software Evolution refers to the tasks and activities involved in addition of new functionality to existing software. Major organizations spend more on evolving existing software than on developing new one. In order to maintain the existing software, there is a need to mine software repositories during the maintenance phase.

- **Software Performance Analysis**: The growing complexity of the software and its profound use in day to day life has promoted interest in the field of software performance analysis. Researchers tend to analyze the attributes of the software by mining repositories in order to suggest improvements in the software development and maintenance phase.

- **Others**: All the application areas of MSR which have not been covered above are refactoring, software maintenance etc.

Figure 3 presents the distribution of reviewed papers focusing on different application areas. If a paper focuses on multiple application areas then the same paper is included in all the application areas it covers. This figure 3 suggests that:

![Figure 3 Application areas explored in MSR papers](image)

The dominant application found in research papers are Bug prediction and change prediction. 25% of the papers are based on bug prediction, with the next-highest proportion being the change prediction area (17%). This shows that researchers are more concerned on improving the mining techniques of bug and change prediction.

- Researchers are also working on Software Performance Analysis (16%) and Software Evolution (16%) application areas. These are important research areas as they analyze the software repositories and suggest improvements in the existing software development practices which helps in increasing the productivity of the software.

- Code Reuse (nine percent) has been relatively less explored in research work. An important aspect of code reuse is that if used efficiently, it not only reduces the cost of development but also increases the productivity of software.

- Six percent of papers included in our study are on clone detection. Clones are useful for reengineering and can be advantageous in many ways. But it is difficult and costly to manage clones. Researchers are trying to develop effective clone detection approaches.
Table 6 provides a replication package of the reviewed studies for the different application areas of MSR.

Table 6 Replication Package for Application Area of MSR

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Application Area</th>
<th>No. of papers</th>
<th>Citations</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Change Prediction</td>
<td>47</td>
<td>1, 2, 5, 10-12, 19, 21, 22, 30, 32, 36-39, 43, 44, 53, 56, 59, 62, 66, 68, 69, 75, 85, 92, 100, 101, 103, 104, 107, 113, 119, 121, 124, 131, 156, 170, 176, 195, 251, 260, 262-265, 269</td>
</tr>
<tr>
<td>3</td>
<td>Clone detection</td>
<td>17</td>
<td>29, 132, 135, 169, 178, 203, 215-221, 246, 249, 268, 270</td>
</tr>
<tr>
<td>4</td>
<td>Code reuse</td>
<td>25</td>
<td>51, 54, 55, 76, 80, 102, 103, 105, 106, 109, 125, 140, 146, 175, 184, 187, 233-239, 241, 255</td>
</tr>
<tr>
<td>5</td>
<td>Software Evolution</td>
<td>42</td>
<td>8, 14, 16, 20, 23-26, 28, 29, 38, 46, 63, 86-88, 95, 97, 119, 138, 143-145, 150, 177, 205, 211, 212, 221-225, 227, 228, 232, 242-244, 250, 253, 268</td>
</tr>
<tr>
<td>6</td>
<td>Software Performance Analysis</td>
<td>42</td>
<td>8, 10, 11, 13, 14, 16, 33, 41, 50, 52, 61, 70, 73, 78, 79, 81, 93, 105, 117, 120, 129, 149, 155, 158, 160, 163-165, 167, 182, 183, 185, 188, 191, 192, 200-207, 208, 254, 257, 258</td>
</tr>
</tbody>
</table>

3.3. Tools used to mine repositories (RQ2)

Researchers have used various types of tools to mine the datasets. Tools have been categorized into three categories: Bug Tracking tools, Data Modeling and Statistical Tools, Extraction Tools. There are some papers which do not use or develop any tool, so those papers are excluded for this research question. The distribution of tools is given in Fig. 4 indicates:

- Data Modeling and Statistical Tools (41 %) are the most widely used tools to evaluate the performance of MSR. The popular statistical tools used by researchers include WEKA, SPSS, Rapid Miner, DTreg, DB Miner, R Tool, checkstyle etc. These tools are used in majority of studies for generating and analyzing the results.
- Extraction Tools (29%) such as ckjm (Chidamber and Kemerer java metrics) extractor, ParseHub, Visual Scraper are used to extract desired information from software repositories and other online resources.
- Bug Tracking Tools (17%) such as Buginfo, trac, FOSSIL, Redmine etc. are used to identify patterns of bugs and changes from the data repositories. 13% of the papers used other tools. The tools included in this category are used less than three times in different research papers. Some of these tools include MALLET, Doc Miner, etc.

Different tools work on different datasets and different applications. It is important that we choose the correct tool according to our dataset and application area. Table 7 lists the datasets and application areas for which the MSR tools can be used.
### Table 7 Datasets and application areas of MSR tools

#### 3.4. Frequently used datasets and software projects along with its type and SDLC phase in which it is used (RQ3)

Data is usually stored in one or more data centers or repositories from which it can be extracted for research work. Some datasets are directly used for example NASA, PROMISE, GitHub, Jira, Sourceforge and some are mined like Google Code, FreeBSD, StackOverflow, FOSS, Apache Software Foundation are some of the prominent repositories which are mined. Figure 5 shows the distribution of datasets mined. This figure indicates that:

<table>
<thead>
<tr>
<th>Tools</th>
<th>Dataset</th>
<th>Application Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Modelling and Statistical Tools</td>
<td>GitHub, Sourceforge, NASA, Bugzilla</td>
<td>Bug Prediction, Change Prediction, Software Performance Analysis</td>
</tr>
<tr>
<td>Extraction Tools</td>
<td>PROMISE, StackOverflow, GitHub, FOSS, Bugzilla</td>
<td>Clone Detection, Bug Prediction, Change Prediction, Code Reuse</td>
</tr>
<tr>
<td>Bug Tracking Tool</td>
<td>Jira, Bugzilla</td>
<td>Bug Prediction</td>
</tr>
</tbody>
</table>

![Figure 5 Distribution of Datasets mined](image)

- 20% of the papers used SourceForge [3] repository, which is a central online hub for software developers to manage and control open source software projects. It was the first service available for open source projects where developers are provided with bug tracking system, source code repository, free access for hosting, news bulletin board as a blog, user and developer comment lists etc. This repository is widely used by the researchers for mining data.
- 18% of papers collect data from GitHub repository [226] which is a popular open source repository consisting of code, commits and issues of various software projects. GitHub is a web based software repository which provides all versions of the software along with the source code management functionality. It offers access to bug tracking systems, task management and wikis for software projects etc. It also act as a storage area for private repositories as well as open source projects.
- StackOverflow (10%) [193, 209] is a popular socio technical network. It acts as a podium for end users to inquire and respond the questions on a wide range of topics related to software engineering. It also assign flag to each question based on its priority which is a useful feature for researchers in mining data.
Bugzilla (11%) is a web-based general purpose bug tracking software repository. It was originally developed and used by Mozilla Project but now is used by many software projects as a bug tracking system.

Promise repository (eight percent) is one of the research largest dataset repository used mainly for software engineering area. It is formed to promote verifiable, repeatable refutable and improvable prediction models. Datasets and tools are available in this repository to serve researchers for mining useful information from it.

Jira (11%) is a proprietary issue tracking and bug tracking product which provides project management functions. It is written in Java language.

Google code (four percent) is a source code repository where codes of various software or web pages are stored globally or confidentially and support version control, bug tracking, and mailing lists for researchers to mine data.

FOSS repository (five percent) contains data such as reports, presentations, conference proceedings, videos, photos, and digital collections of software which is produced by open source software society which is a great help for researchers

NASA (six percent) is a collection of datasets that have been contributed by various universities, agencies and companies. This repository aims to contain datasets which are used for development of prognostic algorithms by researchers.

The reviewed papers were categorized based on not only the datasets but also on the basis of software projects used. Projects can be software or operating systems. Categorization is as follows: Eclipse, Firefox, GNOME desktop suite, Net Beans, Android, Apache, PostgreSQL, Linux, ArgoUML and others. Figure 6 shows the distribution of software projects used. The figure indicates that:

- The Android operating system [154, 181, 231] (17%) is the mostly used project by researchers. It is the mobile operating system developed by Google which is used in touch screen mobile devices such as smart phones and tablets. Its source code is released under open source license which is generally used by many researchers in their specific field.
- 13% of the papers used repositories are based on Eclipse software [35]. Eclipse is a java based Integrated development environment (IDE) used in software engineering. It uses plug-ins to provide different functionalities within and on top of the runtime system. It contains a base workspace where researchers can mine for the changes and bugs in the software.
• Datasets of Linux and GNOME [102, 109] desktop suite both contribute same proportion (nine percent) of papers reviewed. Linux is an open source Operating System. The main advantage of Linux is it can be used, modified and distributed by any researcher under the GNU General Public License. GNOME (GNU Network Object Model Environment) is a desktop environment that is based on Linux OS and composed of free and open source software. GNOME is an international project which aims to develop software frameworks and to coordinate researchers for accessing the details of software projects.

• Data from Apache software projects [74] is analyzed in six percent of the papers. Apache is a decentralized open source area of developers where project is managed by team of experts to provide legal protection and access software code by researchers of apache projects.

• Firefox (five percent) and ArgoUML (five percent) software projects are also used in the papers reviewed in our study. ArgoUML is UML diagram software based on java which is developed under open source Eclipse public license. It is basically used by researchers to mine the software in design phase of SDLC. Firefox is an open source web browser invented by Mozilla Foundation based on Windows, MacOS and Linux operating system. Firefox provides an environment for web developers or researchers to use built in tools for extracting useful information for example bugs of a web page.

• Netbeans (five percent) and PostgreSQL (four percent) are open source software projects. While FreeBSD (five percent) [48] is a free UNIX like operating system which is also relatively less evaluated software project.

It is further important to find that whether research in the field is verifiable and repeatable. For studies to be verifiable, data should be available openly to other researchers. A reproducible research promotes new researchers to conduct research in this field. But most industrial companies do not make their data repository public pertaining to confidentiality. In this case, alternatives to share confidential data should be devised by researchers and companies to promote research without compromising with the confidentiality.

The selected papers were classified on the basis of type of data repositories used: open source, industrial, partial and none. Data repositories which are freely accessible by everyone are referred to as open source data repositories. The industrial repositories are privately owned by a software company or researcher. Some papers use both industrial and open source repositories, the repositories in such of papers are considered under partial data repositories. If no information about the repository is mentioned in the paper, then that is considered as none. Figure 7 depicts the distribution of type of data repositories indicating that:

- The percentage of papers that make use of open source repositories [139, 204] to assess the performance of the proposed mining technique is 71%. The proportion of research papers that evaluated industrial data repositories is less (13%)—this trend shows that more researchers prefer open source repositories to carry out their research as they are easily available to them. The correctness and appropriateness of papers that report results on the basis of evaluation of data from industrial data repositories is worthy of retrospection as the results of these papers may not be repeatable.
Only a few papers (seven percent) used open source as well as industrial data repositories. Partial repositories provide assurance to people planning to apply the results, as using both of open source repositories and industrial data repositories together in a paper makes the evaluation results more dependable and credible.

Nine percent of the papers did not use any data repository. These papers may include survey papers, review papers, theory papers that do not rely on data repositories for validation.

The datasets are also categorized as primary and secondary depending on whether they are directly collected (primary) or indirectly collected (secondary). Some of the datasets discussed in this study are primary while some are secondary irrespective of whether it is open source, industrial and partial. Table 8 gives the classification of datasets on basis of whether they are primary or secondary, and type of dataset, i.e., open source, industrial or partial.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Type (Open Source / Partial / Industrial)</th>
<th>Primary/Secondary Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sourceforge</td>
<td>Open Source</td>
<td>Primary</td>
</tr>
<tr>
<td>GitHub</td>
<td>Open Source</td>
<td>Primary</td>
</tr>
<tr>
<td>StackOverflow</td>
<td>Open Source</td>
<td>Secondary</td>
</tr>
<tr>
<td>Promise</td>
<td>Open Source</td>
<td>Primary</td>
</tr>
<tr>
<td>Jira</td>
<td>Partial</td>
<td>Primary</td>
</tr>
<tr>
<td>Google Code</td>
<td>Open Source</td>
<td>Secondary</td>
</tr>
<tr>
<td>FOSS</td>
<td>Open Source</td>
<td>Secondary</td>
</tr>
<tr>
<td>NASA</td>
<td>Industrial</td>
<td>Primary</td>
</tr>
<tr>
<td>Bugzilla</td>
<td>Partial</td>
<td>Primary</td>
</tr>
</tbody>
</table>

Table 8 Classification of Datasets

The ‘Software Development Life Cycle’ (SDLC) property is used to discover the relationship between the SDLC phase and data set. It investigated the effect of SDLC phase on MSR. The studies were categorized according to SDLC at which data is collected into: Requirements, Design, Development, Testing and Maintenance. The SDLC lays down a foundation outlining the tasks to be carried out during the development of software. SDLC defines a formal organization structure to be pursued by the software development team.

SDLC comprises of following activities:

- **Requirements:** In this phase requirements for the software product are finalized and analyzed. It is a vital step in the process of software development and requires the expertise of proficient and knowledgeable software engineers. It is important to mine data from requirements gathering phase in order to understand the implications of decisions made in initial stage of the software development.

- **Design:** It is a critical phase in the SDLC where approved requirements are taken as an input and logical system is converted into physical system design. Hence making it necessary to analyze the design phase of the software development to comprehend fully the evolution of software.

- **Implementation/Development:** In this phase the software code is actually written on the basis of the requirements gathered in the Requirements phase. Code analysis, change prediction and bug prediction all make use of data from this phase of the software development.

- **Testing:** This is the process of discovering and fixing bugs in the software code. It involves systematically finding bugs in the software and debugging them. The information from testing phase is usually mined for bug prediction and software performance analysis.

- **Maintaining:** Software maintenance is required for any software that needs to continuously improve its performance after initial release. It involves fixing existing bugs, improvement and addition of new features. Hence it is generally a longer phase than the development phase. This phase provides a lot of data that needs to be mined not only to understand the software evolution but also for bug prediction and change prediction. Figure 8 shows the phases of SDLC at which data sets are gathered by the researchers.
It suggests:
- Majority of the data sets are collected at the maintenance phase [40]. 45% of the research papers have collected the data at maintenance phase.
- The other dominant phase, constituting 31% of the papers reviewed, is the development phase followed by testing (10%).
- It is noticed that only eight percent of papers extract the data from requirement (five percent) and design (three percent) phase. It implies that it is difficult to extract data at requirement and design phase.

Table 9 gives the classification of application areas according to the SDLC phase.

Table 9 Classification of Application Areas

<table>
<thead>
<tr>
<th>Application Area</th>
<th>SDLC Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bug Prediction</td>
<td>Design , Implementation, Testing, Maintenance</td>
</tr>
<tr>
<td>Change Prediction</td>
<td>Implementation, Maintenance</td>
</tr>
<tr>
<td>Code Reuse</td>
<td>Design, Implementation</td>
</tr>
<tr>
<td>Software Evolution</td>
<td>Design, Maintenance</td>
</tr>
<tr>
<td>Software Performance Analysis</td>
<td>Implementation, Testing, Maintenance</td>
</tr>
<tr>
<td>Clone Detection</td>
<td>Design, Implementation</td>
</tr>
</tbody>
</table>

3.5. The most investigated techniques for MSR (RQ4)
The objective of this research question was to categorize and document the existing mining techniques [9]. Figure 9 depicts the distribution of different mining techniques. This distribution indicates that there exist a wide range of techniques used for MSR:

- Classification [26, 92, 98, 131, 146, 172, 203] based techniques form the highest proportion (29%) of the techniques
used in the studied papers. Classification algorithms like Naïve Bayes classifier, Decision Trees etc. are quite popular for classifying bugs, changes, code clones etc.

- Clustering technique [8, 191] (14%) is also commonly used in studies reviewed in this SLR. Clustering algorithms like K-means, DBSCAN etc. are used for recognizing patterns and similarities from the vast amount of data available in repositories.

- Machine Learning [180] (12%) is also used in some of the studies reviewed in this SLR. Machine Learning provides a range of popular predictive modeling techniques used for finding out the association between dependent and independent variables in a study. Regression techniques such as Linear Regression, Logistic Regression, Polynomial Regression etc. have been widely used for prediction in MSR research.

- Frequent pattern matching [49, 53, 56, 190, 202] (11%), Association Rule mining [19, 101, 121] (10%) and Text Mining [6, 34, 52, 60, 82, 104, 106, 136, 137, 147, 148, 152, 160, 171, 186] (10%) are evolving techniques for MSR. Frequent Pattern matching and Association rule mining has been used by researchers to uncover useful patterns from repositories. For instance it might be interesting to find out relation between different kinds of changes in the software and the frequency of these changes. Text Mining is also being widely used for extracting information from rich but unstructured text available from various repositories and artifacts produced during software development.

It is important to understand which techniques are useful for which applications. Table 10 provides a list of application areas for which the MSR techniques can be employed.

3.6. Different types of MSR studies and popular Mining Software Repositories journal and conferences (RQ5)

The research studies are segregated based on the research method employed, i.e. theory, survey, experiment, implementation, review, development of tool, development of mining technique and case study. Figure 10 depicts the distribution of research methods. This figure suggests the following:

Table 10 Application Areas for MSR Techniques

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Application Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>Bug Prediction, Change Prediction, Code Reuse, Clone Detection</td>
</tr>
<tr>
<td>Frequent Pattern Matching</td>
<td>Code reuse, Change Prediction</td>
</tr>
<tr>
<td>Association Rule Mining</td>
<td>Software Performance Analysis, Bug Prediction, Change Prediction, Software Evolution</td>
</tr>
<tr>
<td>Text Mining</td>
<td>Code Reuse, Bug Prediction, Change Prediction, Software Evolution</td>
</tr>
<tr>
<td>Machine Learning</td>
<td>Bug Prediction, Change Prediction, Software Performance Analysis</td>
</tr>
<tr>
<td>Clustering</td>
<td>Clone Detection, Software Evolution</td>
</tr>
</tbody>
</table>

- A majority of the papers (28%) reported results of an experiment [210, 213], with the next-highest proportion being the development of mining technique (15%) and case study (15%). This shows that researchers are keen on exploring new results by conducting experiments.

- Theory papers [1, 58, 64, 197] form 12% each of the total papers reviewed. Also 14% of the papers describe the development of mining tool [7, 57, 168]. Only 10% of the papers are found to be survey [91, 196] papers in our study.

- Only six percent of the papers are review papers [42, 65, 141, 142, 206] so there is a need to conduct more reviews in the field of MSR.

International Conference on Mining Software Repositories (MSR) [157, 189] is the dominant conference in the MSR field. International Conference on MSR provides a platform for researchers and practitioners in the field to discuss and collaborate on new research ideas. Research in the field of MSR has attained recognition since the year 2004 when the first International Conference on MSR took place. It is from then that MSR carries on to being a rapidly expanding research field. Figure 11 represents the distribution of popular journals and conferences.
3.7. Research gaps in various application areas of MSR (RQ6)

Among the various application areas as discussed in section 3.2, it was found that the proportion of the selected papers for code reuse, software evolution and clone detection areas were to be less in comparison to other application areas of MSR. This section tries to identify the research gaps in these areas.

3.7.1 Clone Detection
Clone detection aims at detecting the reused code fragments with or without minor revisions. It is important to detect code clones not only during software development, but also during the maintenance phase to speed up the process. For example if a bug is found inside a code fragment then all fragments similar to it should also be checked to detect the same bug. From the papers reviewed related to clone detection it is observed that detecting software clones have made a significant contribution, but it is still not identified whether clones are good or bad? It is required to carry out more studies which determine the impact of clones on software quality and maintainability. It is still a challenge for researchers to determine the behavior and impact of code clones. It cannot be said for sure that non-cloned codes contain less defects than cloned code. Thus more empirical studies need to be conducted using automatic tools for clone detection to arrive at such general conclusions. It is also observed that there is a need to record the history of clone detection to analyze the reason for occurrence of code clones. Clone management studies should be encouraged more to understand economic tradeoff software clones in software development lifecycle. There is an urgent need to have empirical and analytical studies that compare and evaluate various clone detection techniques and tools.

3.7.2 Code Reuse
Code Reuse involves the usage of already existing software for developing new software with the help of reusability principles. It aims to save resources and time by using already created software in development phase of software life cycle. For example software library, subroutines are the basic practices used in code reuse area. The research in the code reuse field is at a growing stage. Therefore, there is a requirement of automated tools that help in enhancing the reusability of code. In the software development area, there is a necessity to understand the role of reusability. Thus new and improved methods need to be developed for organizing, classifying and retrieving software code for reuse. Code reuse should not only consider the software routines or libraries rather take the whole system in context of maintaining or building new systems.
3.7.3 Software Evolution

Software evolution is the process of developing software initially using software engineering principles and then updating till it met the user's requirements. Once the desired software is developed, advancing technology and requirements force to change the software accordingly. Thus recreating software from the scratch is not feasible, so it’s economical to update the existing software to meet the new requirements. The laws of software evolution faced many challenges since their introduction, and now it has been shown that they are not universal. Similar to the changes in software, these laws also need to be changed. These laws need to be adapted not only according to the changes in software, but should also account for the changes in software development and maintenance processes.

With time, it is necessary to carry out empirical studies that assist in the formulation of new software evolution laws which can be shown valid according to the current trends.

4. Implications for research and practice

The results of the review suggest that there are a wide range of techniques used for MSR with no single technique dominating the research. Classification techniques were used in maximum of the studies but still it constitutes only 29% of the reviewed research. The proportion of papers employing machine learning are increasing, but at a slow rate. Researchers should explore this technique. Association Rule Mining, Frequent Pattern Mining, Text Mining, Clustering are other mining techniques that are applied by the researchers. Hence, researchers should be encouraged to conduct more studies and experiment on these lesser used techniques to strengthen evidence regarding their performance. Additionally, researchers must continue exploring the possibilities of developing new techniques and frameworks for MSR.

Partial repositories are known to provide assurance to people planning to employ the results of a study, as using both open source repositories and industrial data repositories makes the evaluation of results more credible. But only seven percent of the papers used partial data repositories for their research and majority of studies (71%) relied only on Open source data repositories. This trend is noticed because open source repositories are easily accessible unlike industrial repositories. It is encouraged to validate the findings of an experiment both with open source and industrial data repositories.

5. Threats to validity

The major threats to validity of the review model are examined on the basis of the following aspects: publication bias, study selection bias and probable imprecision in data extraction.

Due to publication bias, positive findings in the MSR field are published more than the negative findings. Also researchers have a tendency to claim that their technique or methodology perform better than others’. This may result in overestimation of performance of MSR techniques. Luckily, one inclusion criterion (i.e., the third inclusion criterion) may help alleviate this threat. This criterion includes the studies that perform comparison of MSR techniques. These studies which perform a comparison of MSR techniques are not biased towards any technique and report the results of the comparison in an impartial way. Hence, these studies report the negative results in addition to the positive results. However, this review does not consider gray literature (i.e., white papers, technical reports, thesis and work in progress) as it is difficult to obtain, publication bias is unavoidable.

Papers have not been searched using issue-by-issue, manual reading of research titles and abstracts of all papers published in various journals and conferences. The reason for not preferring this approach has been related to practical concerns such as workload and limited query string. Therefore, some MSR papers might have been excluded from some less cited journals or conference proceedings.

Research in the field of MSR has gathered momentum since the year 2004 with the occurrence of the first International Conference on MSR. That is why we have included majority of papers after 2004 and excluded before 2003. The results and trends observed are based on all the papers reviewed during the entire time frame from 2003 to 2018. The review findings do not represent any increasing or decreasing trends over the time series.

Another limitation of this study is that even though it discusses the various application areas of MSR, it does not dwell upon the different metrics used by these application areas.
In order to decrease the threat of imprecision in data extraction, specialized data extraction cards have been elaborated. Any disagreement between researchers is resolved through discussion. Nevertheless, perfection in the extraction process cannot be guaranteed, as human errors are unavoidable.

6. Conclusion and future work

The present paper investigated 300 MSR papers published in journals and proceedings of conferences. The purpose of this systematic literature review is to help researchers to find out popular application areas, tools, repositories and techniques in the field of MSR. The research gaps and popular conferences and journals for publishing MSR studies are also described. Thus our review provides a roadmap for upcoming research in MSR field.

The following important observations are made from this systematic literature review:

- Bug Prediction (25%) is found to be the most popular application area of MSR followed by Change Prediction (17%), Software Performance Analysis (16%) and Software Evolution (16%).

- Researchers have developed and used many tools for mining and analysis of data available in various software repositories. Data Modeling and Statistical tools (41%) are widely employed for analyzing the performance of MSR techniques.

- The proportion of papers that use open source repositories is 71%, while the proportion of studies that employ partial repositories is only seven percent. This trend does not appear to be satisfactory, and more papers need to employ partial datasets. Partial repositories gives assurance to people who intend to apply the results, as using both open source repositories and industrial data repositories makes the results more credible and dependable.

- Classification (29%) is the most popular technique for mining software repositories. Text Mining (10%) and Association Rule Mining (10%) are emerging techniques and should be explored more in future research.

- International Conference on Mining Software Repositories (MSR) is the most popular conference on MSR. It provides ideas and opportunities for researchers working in this field. Moreover most of the papers report the studies of an experiment (28%) and very few are review (six percent) or survey papers (10%).

- Research gaps of relatively less explored and evolving application areas like clone detection (six percent), code reuse (nine percent) and software evolution (16%) are identified and researchers should focus on filling these research gaps.

The future work includes comparing accuracy of various techniques for MSR, so that performance of MSR techniques can also be assessed. It is also recommended to discuss with the experts and existing literature in the field for summarizing the strengths and weaknesses of MSR techniques.

Appendix A. Summary of Quality Assessment Table 11 provides a summary of quality assessment of selected assessment
Table 11 Summary of Quality Assessment

<table>
<thead>
<tr>
<th>Parameter</th>
<th>No.</th>
<th>Question</th>
<th>Yes</th>
<th>Partly/Not Relevant</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Research Questions</strong></td>
<td>QA1</td>
<td>Are the research questions clearly stated?</td>
<td>278(92.66%)</td>
<td>9(3%)</td>
<td>13(4.34%)</td>
</tr>
<tr>
<td></td>
<td>QA2</td>
<td>Are the research questions justified in the context?</td>
<td>267(89%)</td>
<td>17(5.67%)</td>
<td>16(5.33%)</td>
</tr>
<tr>
<td></td>
<td>QA3</td>
<td>Are all the research questions answered?</td>
<td>271(90.33%)</td>
<td>13(4.34%)</td>
<td>16(5.33%)</td>
</tr>
<tr>
<td><strong>Application Area</strong></td>
<td>QA4</td>
<td>Does the paper contribute significantly to the application area?</td>
<td>267(89%)</td>
<td>12(4%)</td>
<td>21(7%)</td>
</tr>
<tr>
<td><strong>Tool</strong></td>
<td>QA5</td>
<td>Is the usage of tools adequately described?</td>
<td>177(59%)</td>
<td>16(5.33%)</td>
<td>107(35.67%)</td>
</tr>
<tr>
<td></td>
<td>QA6</td>
<td>If the study involves tool development, then is the development of a tool explained clearly?</td>
<td>65(21.67%)</td>
<td>62(20.67%)</td>
<td>173(57.66%)</td>
</tr>
<tr>
<td></td>
<td>QA7</td>
<td>If the study involves tool development, then is its performance assessed?</td>
<td>76(25.33%)</td>
<td>59(19.67)</td>
<td>165(55%)</td>
</tr>
<tr>
<td></td>
<td>QA8</td>
<td>Is the tool benchmarked against existing tools?</td>
<td>54(18%)</td>
<td>119(39.67%)</td>
<td>127(42.33%)</td>
</tr>
<tr>
<td><strong>Dataset</strong></td>
<td>QA9</td>
<td>If the study involves data collection, are the data extraction methods defined?</td>
<td>223(74.33%)</td>
<td>17(5.67%)</td>
<td>60(20%)</td>
</tr>
<tr>
<td></td>
<td>QA10</td>
<td>If the study involves analysis, are the data sets adequately described?</td>
<td>258(86%)</td>
<td>18(6%)</td>
<td>24(8%)</td>
</tr>
<tr>
<td></td>
<td>QA11</td>
<td>Is the data available or can be collected by the method defined?</td>
<td>270(90%)</td>
<td>0(0%)</td>
<td>30(10%)</td>
</tr>
<tr>
<td><strong>Techniques</strong></td>
<td>QA12</td>
<td>Is the purpose of the technique defined clearly?</td>
<td>289(96.33%)</td>
<td>8(2.67%)</td>
<td>3(1%)</td>
</tr>
<tr>
<td></td>
<td>QA13</td>
<td>Is the result of the used technique clearly stated?</td>
<td>192(64%)</td>
<td>15(5%)</td>
<td>93(31%)</td>
</tr>
<tr>
<td></td>
<td>QA14</td>
<td>Is the result of the technique compared with other techniques?</td>
<td>167(55.67%)</td>
<td>44(14.67%)</td>
<td>89(29.66%)</td>
</tr>
<tr>
<td>Research Methodology</td>
<td>QA15</td>
<td>Is research methodology clearly described?</td>
<td>273(91%)</td>
<td>10(3.33%)</td>
<td>17(5.67%)</td>
</tr>
<tr>
<td>----------------------</td>
<td>------</td>
<td>------------------------------------------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>QA16</td>
<td>Are the threats to validity of the study clearly presented?</td>
<td>162(54%)</td>
<td>10(3.33%)</td>
<td>128(42.67%)</td>
<td></td>
</tr>
<tr>
<td>QA17</td>
<td>Does the results of the study are further added to the literature?</td>
<td>277(92.33%)</td>
<td>1(0.33%)</td>
<td>22(7.34%)</td>
<td></td>
</tr>
<tr>
<td>QA18</td>
<td>Does the study discuss the issues related to validity/reliability of their measures?</td>
<td>201(67%)</td>
<td>1(0.33%)</td>
<td>98(32.67%)</td>
<td></td>
</tr>
</tbody>
</table>

References for the 300 Reviewed Papers


International Workshop on Mining Software Repositories (MSR 2004), Edinburgh, Scotland, United Kingdom, 64-68.


Keir Mierle, Kevin Laven, Sam Roweis, Greg Wilson. Mining Student CVS Repositories for Performance Indicators. In Proceedings of 2nd International Workshop on Mining Software Repositories...


Ahmed E. Hassan, Member, IEEE, Audris Mockus, Member, IEEE, Richard C. Holt, Member, IEEE, and Philip M. Johnson, Member, IEEE. 2005. Guest Editors’ Introduction: Special Issue on Mining Software Repositories. In IEEE TRANSACTIONS ON SOFTWARE ENGINEERING journal (2005).


Margaret-Anne St. The Evolution of the Social Programmer. In the Proceedings of the 2012 IEEE
International Working Conference on Mining Software Repositories (MSR 2012), Zurich, Switzerland, 140.


[177] Bogdan Dit, Andrew Holtzhauser, Denys Poshvyanky and Huzefa Kagdi. 2013. A Dataset from Change


References


Arvinder Kaur- Dr.(Mrs.) Arvinder Kaur is a Dean and Professor in the University School of Information Technology, Guru Gobind Singh Indraprastha University, Delhi. She obtained her B.E(Electrical Engineering) & M.E. (Computer Engineering) from Thapar Institute of Engg. & Technology, Patiala, Punjab. She Obtained Ph.D. from GGS Indraprastha University Delhi. She was 4th topper in B.E.(Electrical Engineering) and ranked 2nd in M.E.(Computer Engineering) with C.G.P.A. of 9.81. Prior to joining Guru Gobind Singh Indraprastha University, she worked with Dr. B.R. Ambedkar Regional Engineering College, Jalandhar (1993-2000) and Thapar Institute of Engg. & Tech., Patiala. She has also worked for Gabriel India Ltd., Parwanoo(H.P) as Engineer (Research & Development). She has published more than 100 papers in International/ National journals & Conferences. Her seven research papers have been published as Book chapters in various Springer publications. Her Arnetminer Index is 7. Her research interests include Software Engineering, Object-Oriented Software Engineering, Software Metrics, Applications of Artificial Intelligence in Software Engineering, Software Project Management and Software Metrics. She was awarded Career award for young Teachers from All India Council for Technical Education (AICTE), Delhi for three years from 2006 of Rs. 10.5 lakh. She has been awarded Research Award consecutively for four years (2010, 2011 ,2012, 2013) by Guru Gobind Singh Indraprastha University. She has visited, & delivered lectures in International conferences at Kingston College, London ( U.K.), University of Tor Vergata Rome (Italy), Las Vegas (U.S.A.), Hongkong & Phuket (Thailand). She is a lifetime member of ISTE & CSI. She has chaired technical sessions in National as well as International conferences. She was co-ordinator for Ph.D programme of University School of Information & Communication Technology, & chairperson Staff Deverlopment Cell, Guru Gobind Singh Indraprastha University, Delhi.

Kamaldeep Kaur is with University School of Information, Communication and Technology, Guru Gobind Singh Indraprastha University. She has 20 years of experience in software development, consulting and teaching computer science and software engineering. She obtained her doctorate and M.Tech degree in Information Technology from Guru Gobind Singh Indraprastha University. Her research interests include software engineering, software quality management, software metrics, software testing, Internet of Things, Cloud computing and Computer Architecture. She is a lifetime member of ISTE. She has published more than 20 research papers in scopus indexed journals and IEEE conferences. She has successfully completed UGC major project “Design and development of Techniques for measurement and prediction of...
software quality” in 2016. She is also Head of Microprocessor Lab and also Head of IEEE Women in Engineering Chapter at University School of Information and Communication Technology, Guru Gobind Singh Indraprastha University.

**Deepti Chopra** is pursuing her Ph.D. from University School of Information and Communication Technology, Guru Gobind Singh Indraprastha University. She has been working as Assistant Professor in the Department of Computer Science at Indraprastha College for Women, University of Delhi since 2016. She obtained her M.Tech (I.T.) and B.Tech (CSE) degrees from Guru Gobind Singh Indraprastha University. Her research interests include software engineering, data mining, software metrics, mining software repositories, text mining and software quality.

**Harguneet Kaur** is pursuing Ph.D as a Full Time Scholar with University School of Information, Communication and Technology, Guru Gobind Singh Indraprastha University. She had 2.5 years of experience as a Assistant Professor in Department of Information Technology in Guru Tegh Bahadur Institute of Technology. She had 1.5 year of industrial experience in Infosys Technology Limited, Bangalore as a System Engineer. She obtained her M.Tech and B.Tech degree in Information Technology from Guru Gobind Singh Indraprastha University. Her research interests include software engineering, software quality management, software metrics, software testing. She has published research papers in scopus indexed journals and IEEE conferences.