Empirical Study of Software Defect Prediction: A Systematic Mapping

Le Hoang Son 1,2, Nakul Pritam 3, Manju Khari 4, Raghvendra Kumar 5, Pham Thi Minh Phuong 6 and Pham Huy Thong 7,8,*

1 Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam; sonlh@vnu.edu.vn
2 VNU Information Technology Institute, Vietnam National University, Hanoi 122400, Vietnam;
3 Leading Pseudo Code Labs; Delhi 110012, India; nakul.pritam@gmail.com
4 Department of Computer Science and Engineering, Advanced Communication Technologies and Research, Delhi 110012, India; manjukhari@yahoo.co.in
5 Department of Computer Science and Engineering, LNCT Group of College, Jabalpur 482001, MP, India; raghvendraagrawal@gmail.com
6 VNU University of Science, Vietnam National University, Hanoi 122400, Vietnam; phamthiminhphuong_t60@hus.edu.vn
7 Division of Data Science, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam
8 Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam
* Correspondence: phamhuythong@tdtu.edu.vn

Received: 3 January 2019; Accepted: 2 February 2019; Published: 13 February 2019

Abstract: Software defect prediction has been one of the key areas of exploration in the domain of software quality. In this paper, we perform a systematic mapping to analyze all the software defect prediction literature available from 1995 to 2018 using a multi-stage process. A total of 156 studies are selected in the first step, and the final mapping is conducted based on these studies. The ability of a model to learn from data that does not come from the same project or organization will help organizations that do not have sufficient training data or are going to start work on new projects. The findings of this research are useful not only to the software engineering domain, but also to the empirical studies, which mainly focus on symmetry as they provide steps-by-steps solutions for questions raised in the article.

Keywords: defect; machine learning; systematic literature mapping; verification; prediction; software metrics; security

1. Introduction

Defect Prediction in Software (DeP) is the process of determining parts of a software system that may contain defects [1]. Application of DeP models early in the software lifecycle allows practitioners to focus their testing manpower in a manner that the parts identified as “prone to defects” are tested with more rigor in comparison to other parts of the software system [2]. This leads to the reduction of manpower costs during development and also relaxes the maintenance effort [3,4]. DeP models are built using two approaches: First, by using measurable properties of the software system called Software Metrics and second, by using fault data from a similar software project. Once built, the DeP model can be applied to future software projects and hence practitioners can identify defects prone parts of a software system.

Initially, the models used for DeP were built using statistical techniques, but to make the model intelligent, i.e., capable of adapting to changing data in such a manner that as the development process matures the DeP model also matures; it is important that learning techniques are used while building DeP models. In the past, a large number of DeP studies have made use of machine learning techniques.
For instance, some authors used Decision Tree (DT) [5] and Artificial Neural Network (ANN) [6] to build DeP models using Object-Oriented (OO) metrics [7].

Although a large number of studies have been conducted in this domain, more efforts need to be explored. New algorithms are developed periodically, keeping in mind the shortcomings of the previous ones. The development processes also keep changing; this affects the type of data being maintained for a software project (defect data). Such factors require that new modeling techniques be built and benchmarked with the previous work to evaluate their efficacy. In short, the following challenges are encountered:

1. Difficulty in separating correct theories from the incorrect ones when the purpose of evaluation is practice.
2. Difficulty in the identification of quality literature from quality lacking literature.

To identify lacking areas and facilitate the building and benchmarking of new DeP models it is important to study the previous works in a systematic manner and extract meaningful information. Literature surveys and systematic mappings have attempted to answer various questions about DeP models in the past. The authors in Reference [6] study 106 papers published between 1991 and 2011 to identify software metrics that are important for software defect prediction [8–11]. They answered five research questions corresponding to various aspects of the software defect prediction problem with their focus on data used in model building. The authors [12] examined 36 defect prediction studies between 2000 and 2010. The focus is on independent variables and techniques used to build models. They also paid attention to whether feature selection on independent variables has been done or not. The authors in Reference [13] examined 148 studies between 2000 and 2009 with the focus on model building and performance evaluation techniques. Authors [14,15] reviewed 74 studies in 11 journals and seven conference proceedings. The focus of their study is on model building methods, metrics used, and datasets used. Other works can be seen in [16].

To the best of our knowledge, no systematic review, till date, brings all aspects of DeP together. To help researchers and practitioners, it is imperative that good quality literature surveys and reviews that are concerned with the literature available for DeP be conducted and their results made available to the software engineering community. In this systematic mapping, we address various questions, including the above-stated questions (Section 2.1). For this mapping, we collect all DeP studies between 1995 and 2018.

Remainder of the paper is organized as follows, in Section 2 describe the research questions addressed in this systematic mapping and the process for selection of primary studies, Section 3 gives the answers to the research questions identified in this work, Section 4 concludes this systematic mapping and also provides future directions extracted from this system mapping.

2. Review Method

The mapping method in this study is taken from Reference [1]. Figure 1 outlines the process diagram.

![Figure 1. The mapping process.](image-url)
The first step is to identify the need for conducting the systematic mapping and establishing a protocol for the mapping. After this, research questions that the systematic mapping attempts to address are formulated. Once the questions have been identified and evaluated, a search query is designed which is used to extract studies from digital libraries. The studies collected are then passed through a four-stage process:

1. Application of Inclusion-Exclusion Criteria.
2. Quality Analysis of Included Studies.
3. Extraction of Data from selected Studies.
4. Identification of means to report results.

2.1. Research Questions

The objective of this mapping is to establish a starting point for future research in DeP and simultaneously provide practitioners with a summary of all the work done in the area of DeP allowing them to pick a prediction policy that suits them. The research questions identified in this context are given in Table 1.

<table>
<thead>
<tr>
<th>RQ#</th>
<th>Research Question</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1</td>
<td>What techniques have been used for building Defect Prediction in Software (DeP) models?</td>
<td>Identification of the various techniques used to build DeP models.</td>
</tr>
<tr>
<td>RQ2</td>
<td>What are the different data pre-processing techniques used in DeP models?</td>
<td>Identification of the techniques used to prepare the training data for DeP.</td>
</tr>
<tr>
<td>RQ2.1</td>
<td>What are the different techniques used for multi-co linearity analysis in DeP studies?</td>
<td>Identification of techniques to analyze the degree of multi-co-linearity among data.</td>
</tr>
<tr>
<td>RQ2.2</td>
<td>What are the different techniques used for feature sub selection in DeP studies?</td>
<td>Identification of techniques used to select relevant features in the input data.</td>
</tr>
<tr>
<td>RQ3</td>
<td>What is the kind of data used in DeP studies?</td>
<td>Evaluating the type of data used for DeP studies.</td>
</tr>
<tr>
<td>RQ3.1</td>
<td>Which data sets are used for DeP?</td>
<td>What are the popular datasets used in DeP studies?</td>
</tr>
<tr>
<td>RQ3.2</td>
<td>Which metrics are found to be significant predictors for DeP?</td>
<td>Identification of metrics that are found to be relevant for DeP.</td>
</tr>
<tr>
<td>RQ3.3</td>
<td>Which metrics are found to be insignificant predictors for DeP?</td>
<td>Identification of metrics that are found to be irrelevant for DeP.</td>
</tr>
<tr>
<td>RQ4</td>
<td>What are the methods used for performance evaluation in DeP models?</td>
<td>Identify the methods used to evaluate the performance of DeP models.</td>
</tr>
<tr>
<td>RQ4.1</td>
<td>What are the different performance measures used for DeP studies?</td>
<td>Identify the performance parameters that are used to evaluate how well a DeP model has worked.</td>
</tr>
<tr>
<td>RQ4.2</td>
<td>What are the different statistical tests of significance used in DeP studies?</td>
<td>Identify the statistical tests that have been used to evaluate the results of DeP models.</td>
</tr>
<tr>
<td>RQ5</td>
<td>What is the performance of various learning techniques across datasets?</td>
<td>Estimate the effectiveness of learning techniques for DeP.</td>
</tr>
<tr>
<td>RQ6</td>
<td>What is the performance of search-based techniques in defect prediction?</td>
<td>Estimate the effectiveness of search-based techniques for DeP.</td>
</tr>
<tr>
<td>RQ7</td>
<td>How many DeP studies attempt to predict security related defects and vulnerabilities?</td>
<td>Estimate the number of studies that work on security related defects and software vulnerabilities.</td>
</tr>
<tr>
<td>RQ8</td>
<td>How many DeP studies attempt to identify defects of varying severity levels?</td>
<td>Estimate the number of studies that work on defects of various severity levels.</td>
</tr>
<tr>
<td>RQ9</td>
<td>How many DeP studies perform cross-project DeP?</td>
<td>Estimate the number of studies that perform cross-project DeP.</td>
</tr>
</tbody>
</table>
2.2. Literature Review

Primary studies by definition correspond to the literature being mappings. To provide a strong mapping it is essential that selection of primary studies for mapping must be done carefully. While it is good that an exhaustive search is conducted for the selection of primary studies, in some cases it is not possible because of the number of primary studies available. In such cases the search criteria become important. For DeP studies, we can conduct an exhaustive search because the number of primary studies is not very large and since we are only concerned with those studies which are empirical in nature, the number shrinks further. We have selected the list of following digital libraries to perform our search:

1. IEEE Xplore
2. Springer Link
3. Science Direct
4. Wiley Online Library
5. ACM Digital Library
6. Google Scholar

The Search String: A search string is the combination of characters and words entered by a user into a search engine to find desired results. The information provided to the search engine of the digital library directly impacts the results provided by it. To ensure that all the primary studies that our mapping plans to address are covered we need to be careful in the selection and placement of keywords used in the search string.

```
software . ( defect + fault ) . ( software metrics + object oriented metrics + design metrics )
```

Here, '.' corresponds to the Boolean AND operation, and '+' Corresponds to the Boolean OR operation

The search string was executed on all six electronic databases mentioned above and the publication year was restricted to the range 1995–2018. The literature hence obtained was processed further using a carefully designed inclusion-exclusion criteria and quality analysis criteria as described in the following sections.

**The Inclusion-Exclusion Criteria**: The search results obtained by execution of the search string may still fetch some primary studies that either do not add value to the mapping or do not fall within the purview of what the mapping aims to accomplish.

Once all the primary studies have been obtained, a carefully designed inclusion-exclusion criteria are applied to the resultant set in order to eliminate entities that do not match the objectives of the mapping.

**Inclusion Criteria**:

- Empirical studies for DeP using software metrics.
- Studies that provide empirical analysis using statistical, search-based and machine learning techniques.

**Exclusion Criteria**:

- Literature Reviews and Systematic Reviews.
- Studies that do not use DeP as the dependent variable.
- Studies of non-empirical nature.
- If two studies by the same author(s) exist, where one is an extension of the previous work the former is discarded. But if the results differ, both are retained.
• **Review Committee**: We formed a review committee that comprises of two Assistant Professors and two senior researchers to mapping in order to rate all primary studies captured from the search. All studies were examined independently on the basis of the criteria defined in The Inclusion-Exclusion Criteria. Application of the inclusion-exclusion criteria resulted in 98 studies out of the total 156 studies being selected for quality analysis.

• **Quality Analysis**: Assessing the quality of a set of primary studies is a challenging task. A quality analysis questionnaire is prepared as part of this systematic mapping to assess the relevance of studies taking part in this mapping. The questionnaire takes into consideration suggestions given in Reference [5]. A total of 18 questions, given in Table 2, together form the questionnaire and each question can be answered as “Agree” (1 point), “Neutral” (0.5 points) and “Disagree” (0 points). Hence, a study can have a maximum 18 points and minimum 0 points.

• The same review committee enforces the quality analysis questionnaire.

**Table 2. Quality analysis questions.**

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>QQ1</td>
<td>Is the objective of the study clear?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ2</td>
<td>Does the study add value to the existing literature?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ3</td>
<td>Is the dataset size sufficient for this type of studies?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ4</td>
<td>Does the study perform multi-co linearity analysis?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ5</td>
<td>Does the study perform feature sub-selection?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ6</td>
<td>Are the independent variables clearly defined?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ7</td>
<td>Is the data collection procedure clearly defined?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ8</td>
<td>Does the study use statistical tests while evaluating results?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ9</td>
<td>Does the author provide sufficient detail about the experiment?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ10</td>
<td>Are the threats to validity given?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ11</td>
<td>Does the study provide parameters of test?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ12</td>
<td>Are the limitations of the study given?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ13</td>
<td>Does the study clearly define the performance parameters used?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ14</td>
<td>Are the learning techniques clearly defined?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ15</td>
<td>Are the results clearly stated?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ16</td>
<td>Does the abstract provide sufficient information about the content of the study?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ17</td>
<td>Is there a comparison among techniques? ML (Machine Learning) vs. ML?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QQ18</td>
<td>Is there a comparison among techniques? ML vs. Statistical?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Data Extraction**: To extract meaningful information for each study such that the research questions can be answered, the data extraction method should be objectively defined. We created the data form shown in Figure 2 and filled it for each of the primary study (SE) that passes the quality analysis criteria. The review committee filled the data extraction card and any conflicts raised during the data extraction process were resolved by taking suggestions from other researchers. The resultant data was converted to an Excel (.xlsx) Workbook for usage during the data synthesis process.

---

**Figure 2. Extraction Card.**
Data Synthesis: Data synthesis involves the accumulation of facts from the data collected during the data extraction process to build responses to the research questions [5]. In the data synthesis process, we identify SEs that have similar viewpoints regarding a particular research question. This allows for reaching conclusive answers to the research questions identified as part of the systematic mapping. Details of the techniques used to answer the selected research questions are given below:

1. RQ1: To answer this question we use a bar chart that shows the number of studies using machine learning; search-based techniques, and statistical techniques and threshold-based techniques.
2. RQ2: This question has two parts. The first part is concerned with multi-co linearity analysis techniques and the second part deals with feature sub-selection techniques used. To answer this question, we make use of bar charts and tables. The bar charts show the number of studies using a particular technique for multi-co linearity analysis or feature sub selection. The tables show what techniques were used in which SE.
3. RQ3: This research question deals with the data used in SEs. It has three parts, the first part deals with the various datasets used for DeP studies, the second part deals with the independent variables found significant in the identified SEs, and the third part deals with the independent variables found insignificant in the selected SEs. The first part and the second part are answered with the help of a bar chart/pie chart combination while the third part does not use any diagramming method.
4. RQ4: The fourth question deals with the performance measures and statistical tests used in DeP studies. We use a combination of bar charts/pie charts to address this question.
5. RQ5: This question makes use of a bar chart. The bars are used to denote the comparative performance of a learning technique.
6. RQ6: This question is addressed using a table. Since the number of studies is limited, tables are used to summarize which study uses what search-based technique.
7. RQ7: This question uses bar charts and tables to show the distribution of studies that address security related defects and vulnerabilities.
8. RQ8: This question does not use any diagramming method.
9. RQ9: This question does not use any diagramming method.

3. Results and Discussion

3.1. Description of SEs

This study is very useful for developers. Upon the application of quality analysis criteria, we have selected 98 primary studies that we call primary studies (SEs) using a mix of learning techniques, including machine learning, search-based techniques, statistical techniques and some hybrid frameworks. Most of the studies made use of public domain data and standard performance measures like accuracy, error rate, precision, recall and ROC analysis.

3.1.1. Quality Assessment Questions

Table 3 shows the results of the quality analysis. It shows the percentage and number of SEs that Agree to, Disagree to, and stands Neutral towards each quality question.
From the results of the quality assessment procedure, it is clear that QQ4, QQ8, and QQ18 have been answered negatively by most of the study. This is an important finding and strongly suggests that the available literature lacks the use of multi-co-linearity analysis procedures, statistical tests of significance and machine learning technique vs. statistical technique comparison. Also, the results of QQ5, QQ10, QQ11, and QQ17 are well distributed on the positive and negative end. We also found out that the abstracts of most studies were partially informative about what will the reader extract by reading the entire study. The majority of the studies have not given any limitations of their work clearly, but by reading the entire manuscript some limitations can be extracted.

The scores obtained by SEs in the quality analysis step are separated into four groups, Low, Average, High and Very High. The SEs that obtained High and Very High scores were considered for the mapping. Table 4 summarizes the percentage of SEs and the number of SEs in each of the four categories.

<table>
<thead>
<tr>
<th>Score</th>
<th>No. of Primary Study s</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High (13 and Above)</td>
<td>22</td>
<td>22.45%</td>
</tr>
<tr>
<td>High (9 to 12.5)</td>
<td>66</td>
<td>67.35%</td>
</tr>
<tr>
<td>Average (6.5 to 8.5)</td>
<td>7</td>
<td>7.14%</td>
</tr>
<tr>
<td>Low (6 and Below)</td>
<td>3</td>
<td>3.06%</td>
</tr>
</tbody>
</table>

We also assign a unique identifier to each SE. Table 5 presents the SE identifier along with its reference number. The identifiers allow for easy interpretation of results as they are used in all subsequent sections.
Table 5. Primary studies.

<table>
<thead>
<tr>
<th>Primary Study</th>
<th>SE1</th>
<th>SE2</th>
<th>SE3</th>
<th>SE4</th>
<th>SE5</th>
<th>SE6</th>
<th>SE7</th>
<th>SE8</th>
<th>SE9</th>
<th>SE10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>[17,18]</td>
<td>[19,20]</td>
<td>[21]</td>
<td>[22,23]</td>
<td>[24–26]</td>
<td>[27,28]</td>
<td>[29,30]</td>
<td>[31]</td>
<td>[32]</td>
<td>[33]</td>
</tr>
<tr>
<td>Primary Study</td>
<td>SE11</td>
<td>SE12</td>
<td>SE13</td>
<td>SE14</td>
<td>SE15</td>
<td>SE16</td>
<td>SE17</td>
<td>SE18</td>
<td>SE19</td>
<td>SE20</td>
</tr>
<tr>
<td>Reference</td>
<td>[34]</td>
<td>[35,36]</td>
<td>[37–39]</td>
<td>[40]</td>
<td>[41]</td>
<td>[42]</td>
<td>[43–45]</td>
<td>[46]</td>
<td>[47,48]</td>
<td>[49]</td>
</tr>
<tr>
<td>Primary Study</td>
<td>SE21</td>
<td>SE22</td>
<td>SE23</td>
<td>SE24</td>
<td>SE25</td>
<td>SE26</td>
<td>SE27</td>
<td>SE28</td>
<td>SE29</td>
<td>SE30</td>
</tr>
<tr>
<td>Reference</td>
<td>[50]</td>
<td>[51–56]</td>
<td>[57]</td>
<td>[58–60]</td>
<td>[61]</td>
<td>[62]</td>
<td>[63]</td>
<td>[64,65]</td>
<td>[66]</td>
<td>[67–69]</td>
</tr>
<tr>
<td>Primary Study</td>
<td>SE31</td>
<td>SE32</td>
<td>SE33</td>
<td>SE34</td>
<td>SE35</td>
<td>SE36</td>
<td>SE37</td>
<td>SE38</td>
<td>SE39</td>
<td>SE40</td>
</tr>
<tr>
<td>Reference</td>
<td>[70]</td>
<td>[71]</td>
<td>[72]</td>
<td>[73–75]</td>
<td>[76]</td>
<td>[77]</td>
<td>[78]</td>
<td>[79]</td>
<td>[80,81]</td>
<td>[82]</td>
</tr>
<tr>
<td>Primary Study</td>
<td>SE41</td>
<td>SE42</td>
<td>SE43</td>
<td>SE44</td>
<td>SE45</td>
<td>SE46</td>
<td>SE47</td>
<td>SE48</td>
<td>SE49</td>
<td>SE50</td>
</tr>
<tr>
<td>Reference</td>
<td>[83,84]</td>
<td>[85]</td>
<td>[86]</td>
<td>[87]</td>
<td>[88]</td>
<td>[89]</td>
<td>[90]</td>
<td>[91]</td>
<td>[92]</td>
<td>[93,94]</td>
</tr>
<tr>
<td>Primary Study</td>
<td>SE51</td>
<td>SE52</td>
<td>SE53</td>
<td>SE54</td>
<td>SE55</td>
<td>SE56</td>
<td>SE57</td>
<td>SE58</td>
<td>SE59</td>
<td>SE60</td>
</tr>
<tr>
<td>Reference</td>
<td>[95,96]</td>
<td>[97,98]</td>
<td>[99]</td>
<td>[100]</td>
<td>[101]</td>
<td>[102]</td>
<td>[103]</td>
<td>[104]</td>
<td>[105]</td>
<td>[106]</td>
</tr>
<tr>
<td>Primary Study</td>
<td>SE61</td>
<td>SE62</td>
<td>SE63</td>
<td>SE64</td>
<td>SE65</td>
<td>SE66</td>
<td>SE67</td>
<td>SE68</td>
<td>SE69</td>
<td>SE70</td>
</tr>
<tr>
<td>Reference</td>
<td>[107]</td>
<td>[108]</td>
<td>[109]</td>
<td>[110]</td>
<td>[111]</td>
<td>[112]</td>
<td>[113]</td>
<td>[114]</td>
<td>[115,116]</td>
<td>[117,118]</td>
</tr>
<tr>
<td>Primary Study</td>
<td>SE71</td>
<td>SE72</td>
<td>SE73</td>
<td>SE74</td>
<td>SE75</td>
<td>SE76</td>
<td>SE77</td>
<td>SE78</td>
<td>SE79</td>
<td>SE80</td>
</tr>
<tr>
<td>Reference</td>
<td>[119,120]</td>
<td>[121,122]</td>
<td>[123]</td>
<td>[124,125]</td>
<td>[126]</td>
<td>[127]</td>
<td>[128–130]</td>
<td>[131]</td>
<td>[132,133]</td>
<td>[134–136]</td>
</tr>
<tr>
<td>Primary Study</td>
<td>SE81</td>
<td>SE82</td>
<td>SE83</td>
<td>SE84</td>
<td>SE85</td>
<td>SE86</td>
<td>SE87</td>
<td>SE88</td>
<td>SE89</td>
<td>SE90</td>
</tr>
<tr>
<td>Reference</td>
<td>[137]</td>
<td>[138]</td>
<td>[139]</td>
<td>[140,141]</td>
<td>[142,143]</td>
<td>[144]</td>
<td>[145]</td>
<td>[146,147]</td>
<td>[148]</td>
<td>[149]</td>
</tr>
<tr>
<td>Primary Study</td>
<td>SE91</td>
<td>SE92</td>
<td>SE93</td>
<td>SE94</td>
<td>SE95</td>
<td>SE96</td>
<td>SE97</td>
<td>SE98</td>
<td>Reference</td>
<td>[150,151]</td>
</tr>
</tbody>
</table>

Table 6. Quality scores of primary studies obtaining at least 13 points.

<table>
<thead>
<tr>
<th>Primary Study</th>
<th>Quality Score</th>
<th>Primary Study</th>
<th>Quality Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE8</td>
<td>13</td>
<td>SE31</td>
<td>14</td>
</tr>
<tr>
<td>SE14</td>
<td>14</td>
<td>SE57</td>
<td>14.5</td>
</tr>
<tr>
<td>SE24</td>
<td>17.5</td>
<td>SE41</td>
<td>13.5</td>
</tr>
<tr>
<td>SE35</td>
<td>13</td>
<td>SE75</td>
<td>1.3</td>
</tr>
<tr>
<td>SE38</td>
<td>16.5</td>
<td>SE58</td>
<td>14</td>
</tr>
<tr>
<td>SE51</td>
<td>15</td>
<td>SE86</td>
<td>14.5</td>
</tr>
<tr>
<td>SE54</td>
<td>14</td>
<td>SE98</td>
<td>15.5</td>
</tr>
<tr>
<td>SE63</td>
<td>15</td>
<td>SE2</td>
<td>13.5</td>
</tr>
<tr>
<td>SE80</td>
<td>13.5</td>
<td>SE66</td>
<td>13</td>
</tr>
<tr>
<td>SE84</td>
<td>14</td>
<td>SE45</td>
<td>15</td>
</tr>
<tr>
<td>SE96</td>
<td>15.5</td>
<td>SE33</td>
<td>14</td>
</tr>
</tbody>
</table>

3.1.2. Publication Source

After quality analysis, 98 studies are left, and 10 studies are discarded. Table 7 provides the details of the studies published in journals and conferences of high repute. It can be noted that IEEE Transactions on Software Engineering has the highest percentage of selected studies (11.22%) followed by the Journal of Systems and Software (10.20%).
### Table 7. Summary of top publications.

<table>
<thead>
<tr>
<th>Publication Name</th>
<th>Type</th>
<th>Number</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM/IEEE International Symposium on Empirical Software Engineering</td>
<td>Conference</td>
<td>1</td>
<td>1.02%</td>
</tr>
<tr>
<td>Advances in Engineering Software</td>
<td>Journal</td>
<td>1</td>
<td>1.02%</td>
</tr>
<tr>
<td>Applied Soft Computing</td>
<td>Journal</td>
<td>1</td>
<td>1.02%</td>
</tr>
<tr>
<td>Automated Software Engineering</td>
<td>Journal</td>
<td>1</td>
<td>1.02%</td>
</tr>
<tr>
<td>Empirical Software Engineering</td>
<td>Journal</td>
<td>6</td>
<td>6.12%</td>
</tr>
<tr>
<td>Engineering Applications of Artificial Intelligence</td>
<td>Journal</td>
<td>1</td>
<td>1.02%</td>
</tr>
<tr>
<td>Expert Systems with Applications</td>
<td>Journal</td>
<td>2</td>
<td>2.04%</td>
</tr>
<tr>
<td>IEEE International Symposium on Software Reliability</td>
<td>Conference</td>
<td>3</td>
<td>3.06%</td>
</tr>
<tr>
<td>IEEE Software</td>
<td>Journal</td>
<td>1</td>
<td>1.02%</td>
</tr>
<tr>
<td>IEEE Transactions on Software Engineering</td>
<td>Journal</td>
<td>11</td>
<td>11.2%</td>
</tr>
<tr>
<td>Information and Software Technology</td>
<td>Journal</td>
<td>2</td>
<td>2.04%</td>
</tr>
<tr>
<td>Information Sciences</td>
<td>Journal</td>
<td>3</td>
<td>3.06%</td>
</tr>
<tr>
<td>Journal of Systems and Software</td>
<td>Journal</td>
<td>10</td>
<td>10.2%</td>
</tr>
<tr>
<td>LNCS-Springer</td>
<td>Journal</td>
<td>1</td>
<td>1.02%</td>
</tr>
<tr>
<td>Software Quality Journal</td>
<td>Journal</td>
<td>2</td>
<td>2.04%</td>
</tr>
</tbody>
</table>

3.1.3. Publication Year

Figure 3 shows the distribution of DeP studies from the year 1995 to 2018. It shows that the research in DeP picked up in the year 2005 when five studies were conducted. Out of the total 98 studies selected for this mapping, only 12 were conducted till 2004. In 2007, 2008, 2009, 2010, 2011, 2012 and 2013 the number of DeP studies conducted is 11, 15, 10, 15, 9, 12, and 5 respectively. These account for 77 out of the selected 98 studies. For 2018, we could find only one quality publication till date, but the data for 2018 should not be considered complete. One of the reasons for the surge in DeP studies after 2005 could be the availability of the public domain data on the PROMISE repository.

3.2. RQ1: What Techniques Have Been Used for Building DeP Models?

In this section, we elaborate on the distribution of model building techniques used in DeP studies. As shown in Figure 4 the most used learning method is the Decision Tree, 44 out of 98 studies selected for this systematic mapping use some variant of the decision tree method. Techniques like C4.5, J48, CART, and Random Forest come under the decision tree class. Bayesian learners [65] i.e., Naïve Bayes, Bayes Net, etc. have been used by 39 studies. Regression, Discriminant analysis and Threshold based classification have been performed in 35, 8 and 4 studies respectively. Both support vector machine and neural network have been used in 21 studies.
3.3. RQ2: What Are the Different Data Pre-Processing Techniques Used in DeP Models?

Research question #2 summarizes all the techniques that deal with data pre-processing. This study answers this question in two parts; the first part summarizes all methods that are used to analyze the degree of multi-co-linearity among features and the second part deals with the techniques used to select relevant features from the dataset.

3.3.1. What Are the Different Techniques Used for Multi-Co-Linearity Analysis in DeP Studies?

Analysis of data multi-co-linearity is an important step of data pre-processing, because it ascertains that the data being used for the study is not collinear. Collinear data does not add value to the study as one variable can be used to predict another variable hence reducing the independence among variables. This affects the final results in two ways.

The predictive capability of independent variables cannot be assessed effectively as at least one independent variable ‘x’ is highly dependent on another independent variable ‘y’. If the variable ‘y’ a significant predictor of defects, then the independent variable ‘x’ can also present itself as a significant predictor even if ‘x’ by itself does not carry any meaningful information that is different to the information available with ‘y’.

The final results reported by the prediction model are affected. Table 8 shows all studies that apply multi-co-linearity procedures and the corresponding procedure applied. Only seven out of 98 studies have stated in their work that they make use of multi-co-linearity procedures.

Table 8. Studies applying multi co-linearity procedures.

<table>
<thead>
<tr>
<th>Primary Study</th>
<th>VIF</th>
<th>Condition No.</th>
<th>PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE4</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE17</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SE26</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SE31</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SE41</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE50</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE98</td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5 shows the graphical representation of the distribution of multi-co-linearity analysis techniques used across seven studies. We can see that the Condition Number method is the most popular in DeP studies, four studies make use of this method. VIF (variable inflation factor) and PCA (principal component analysis) were applied in two studies each.
3.3.2. What are Different Techniques Used for Feature Sub Selection in DeP Studies?

In this section, we determine the number of studies that state the use of feature sub-selection techniques and what feature sub-selection techniques are widely used. It is important that we conduct feature sub-selection on the input data before it is supplied to the learning algorithm, because the data could contain redundant and irrelevant features. Out of the 98 studies selected for this systematic mapping, there are 44 studies that use feature sub-selection methods, i.e. exactly 50 percent of the studies use feature sub-selection methods. Figure 6 shows the distribution of feature sub-selection techniques across all 44 studies. The most frequently used technique for feature selection is correlation-based feature selection (CFS), 19 studies make use of CFS. After CFS, the most used technique is Principal Component Analysis (PCA). It should be noted that while the number of studies that apply multi-co-linearity analysis is only seven (7.14%) the number of studies that apply feature sub selection is 44 (50%).

The reports corresponding to RQ2.1 show that only 7.14% of all studies make use of multi-co-linearity analysis techniques. This is a very small number and reflects the fact the very little emphasis has been laid on the analysis of multi-co-linearity among data elements present in datasets used for DeP studies in the past.
Since DeP studies are heavily dependent on data and their primary aim is to identify metrics that are relevant to DeP, the result of our analysis towards RQ2 is not very encouraging. Every DeP study must ensure that the data used for model building is free from multi-co-linearity and only consists of relevant metrics so that highly accurate estimates of metric relevance and model performance can be made.

3.4. RQ3: What Is the Kind of Data Used in DeP Studies?

In this question, we deal with various questions relating to data used in a DeP model. We answer this question in three parts. In the first part, we analyze existing studies to find out what type of data has been used in model training and testing. In the second part, we examine all studies to find out which metrics were found relevant and what percentage of studies is reporting a particular metric as relevant for DeP. And finally, we examine all the studies to find out which metrics were found unsuitable for DeP.

3.4.1. Which Data Sets Are Used for SFP?

In this question, we attempt to summarize the various datasets that are used by DeP studies in the past. A wide variety of public domain datasets like NASA, Eclipse, Apache have been used by multiple researchers to build DeP models; at the same time, there are instances where studies have used data from systems built by students while some have used industrial systems to generate data. Figure 7 shows the distribution of datasets used for building DeP models. The most popular datasets for DeP model building come from NASA MDP program, a total of 42 studies out of 98 use data provided under this program. Eclipse dataset has been used by 10 studies while Apache datasets are used by four studies. The data from PROMISE repository has been used by 17 studies. The Mozilla foundation has also provided data to four studies, three of these studies are addressing security-related defects. The “Others” category includes all industrial software systems, legacy systems and data sources which were not stated in the studies.

![Figure 7. Sources of data used across DeP studies.](image)

3.4.2. Which Metrics Are Found to Be Significant Predictors for DeP?

In this part, we identify the metrics that are found to be significant predictors of software defects. Figure 8 clearly shows that the most significant predictor of defects is found to be LOC, a total of 26 studies have reported LOC as a significant predictor of defects. Coupling metric (CBO) and RFC have been reported as significant by 25 and 22 studies, respectively. Complexity metric WMC is also reported significant by 15 studies. Only nine studies found DIT to be a significant predictor of software defects and NOC is found to be significant in six studies.
3.4.3. Which Metrics Are Found to Be Insignificant Predictors for DeP?

In this part, we identify metrics that are not found suitable for software defect prediction. It has been widely recognized that the inheritance metrics are the weakest predictors of defects. In this mapping, we observed that both the Chidamber and Kemerer inheritance metrics, DIT and NOC, have been widely reported insignificant predictors for software defect prediction. DIT has been reported insignificant by 23 studies while NOC is reported weak by 28 studies. Some studies have reported other metrics like LCOM3 and LCOM as weak predictors of defects. LCOM3 and LCOM have been reported insignificant by 7 and 11 studies respectively.

3.5. RQ4: What Are the Methods Used for Performance Evaluation in DeP Models?

In this question, we summarize the various performance evaluation methods used in DeP studies. This question is answered in two parts, in this first part we analyze the various performance measures used to capture the performance of a DeP model and in the second part, we summarize the statistical tests used to conduct a comparative analysis of performance measures.

3.5.1. What are Different Performance Measures Used for DeP Studies?

There are a large number of performance measures available for use. Some standard performance measures are Accuracy, Sensitivity, Specificity, and Area under the ROC Curve, Precision and Recall. From Figure 9 it is clear that many of these popular performance measures have been used in DeP studies. The most popular measure is the Recall which is used by 39 out of 98 studies. ROC Area, Accuracy and Precision have been used in 28, 25 and 27 studies, respectively.
3.5.2. What are Different Statistical Tests of Significance Used in DeP studies?

Here, we analyze how many studies have made use of statistical tests in their evaluation of the results. By using such methods, the researcher can give a mathematical foundation to the evaluation process. Some advantages of using statistical tests are:

1. Statistical tests are standardized and allow for suppression of author bias.
2. Every statistical test follows a mathematical procedure and hence the reader(s) can better understand how the results were processed. The author is not required to illustrate the entire result processing algorithm if it chooses to use a standard statistical test.

Based on the above facts, we strongly suggest that statistical tests should be part of all research studies that involve large datasets. In the 98 primary studies selected for this mapping, 12 statistical tests have been used by 18 studies (Figure 10). The Wilcoxon test has been used by five studies. Wilcoxon test is a two sample non-parametric test. Such tests are used to evaluate the difference between two samples (for defect prediction studies, the sample is the result(s) generated by an algorithm). The drawback of two sample tests is that does not allow comparison of multiple techniques at once. For this reason, we need k sample tests. Examples of k sample tests are Friedman test, *t*-Test, etc. The analysis of 98 primary studies suggests that k sample tests have been rarely used in literature. The Friedman test has been used in three studies and the *t*-Test has been used in only one study. Two sample tests are better suited for post hoc analysis of results.

Figure 10. Statistical Tests used by DeP Studies.

3.6. RQ5: What Is the Performance of Various Learning Techniques across Datasets?

In this question, we attempt to approximate the learning ability of various learning techniques. To answer this question, we follow a systematic approach consisting of the following steps:

a. Identification of performance measure(s) to examine the predictive capability of the proposed models.
b. Identification of datasets to consider for performance validation.
c. A collection of the performance related data corresponding to the identified performance measure(s) for the identified datasets.
d. Comparison of performance of each technique pair for all datasets identified.
e. Reporting the results.

While a good number of performance measures have been studied in this mapping, we selected the Area under the ROC Curve for performance evaluation, because it is found to suppress the effects of data imbalance on the resultant model. Any dataset that has an unequal distribution of defective and non-defective classes can be considered to be imbalanced [4]. Since the data distribution is skewed in imbalanced datasets, learning from these datasets requires a different approach [4]. ROC curves [8] aim at maximizing both the sensitivity (True Positive Rate) and 1-Specificity (True Negative Rate) of
the model and, therefore, overcome any issue relating to data imbalance. For the final analysis, we imposed the following restrictions:

Only those techniques were selected that were used in five or more studies. This ensures that only those techniques are evaluated which have been widely used in literature. Techniques that are used in fewer studies cannot be compared to studies that have been evaluated in varied experimental setups. Factors such as researcher bias, selection of test parameters, etc., affect the performance of a technique greatly. Summarizing the results of techniques reported in at least five studies will diminish the impact of such factors.

Only those datasets were considered which have been evaluated using ROC analyzes in at least two studies. The objective of this constraint is to eliminate those datasets that have been used only once and could get included in the analysis severely impacting the overall performance of a learning algorithm.

Finally, we assess the performance of 10 learning techniques over 11 datasets (KC1 is used at method level, as well as a class level). Details of the techniques and datasets are given in Table 9.

Table 9. Datasets and learning techniques evaluated in RQ8.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Learning Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR1, CM1, Eclipse, jEdit, JM1, KC1 (Method and Class Level), PC1, PC2, PC3, PC4</td>
<td>Artificial Neural Network [27,164], Bayes Net, C4.5, Classification and Regression Trees (CART), Discriminant Analysis, KStar, Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine</td>
</tr>
</tbody>
</table>

The relation given below summarizes the comparison procedure,

for each [(a, b): where a and b are learning techniques and a ≠ b]
compare AUC(a) and AUC(b)
if(AUC(a)>AUC(b))
    set 1
else
    set 0
    calculate sum(1) and sum(0) as ‘better_than’ and ‘not_better_than’

Summarily, for each pair of algorithms, we check the performance of the AUC measure at the dataset level. Only when the AUC value is available for both techniques (for a given dataset), a comparison is made. Figure 11 provides a detailed description of the performance analysis.

It can be seen that Random Forest is the best performing algorithm. Its performance is inferior to Discriminant Analysis at two datasets and against KStar and CART, it performs better in all cases. The worst learner based on this analysis is the CART algorithm. The CART algorithm is better compared to SVM in three cases while BayesNet, Discriminant Analysis and Naïve Bayes reported better AUC than CART in all comparisons. Not far behind Random Forest is the Discriminant Analysis with better results than CART and KStar in all comparisons, but in comparison to Random Forest, it is lagging against the other techniques. On the downside, KStar did better than CART and the comparison shows that in all comparisons except the ones against CART, KStar performed poorly.

Other learning techniques, as can be seen in the graph, had an average performance. The average, maximum and minimum ROC area for the above learning techniques are given in Table 10.
Figure 11. Performance comparison of learning techniques.

Table 10. Search-based techniques used in the literature.

<table>
<thead>
<tr>
<th>Learning Technique</th>
<th>Minimum ROC Area</th>
<th>Maximum ROC Area</th>
<th>Average ROC Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artificial NN</td>
<td>0.66</td>
<td>0.94</td>
<td>0.76</td>
</tr>
<tr>
<td>Bayes Net</td>
<td>0.66</td>
<td>0.90</td>
<td>0.78</td>
</tr>
<tr>
<td>C4.5</td>
<td>0.50</td>
<td>0.93</td>
<td>0.72</td>
</tr>
<tr>
<td>CART</td>
<td>0.61</td>
<td>0.87</td>
<td>0.73</td>
</tr>
<tr>
<td>Discriminant Analysis</td>
<td>0.73</td>
<td>0.88</td>
<td>0.81</td>
</tr>
<tr>
<td>KStar</td>
<td>0.60</td>
<td>0.83</td>
<td>0.69</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.49</td>
<td>0.89</td>
<td>0.73</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.63</td>
<td>0.85</td>
<td>0.75</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.73</td>
<td>0.97</td>
<td>0.82</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>0.50</td>
<td>0.92</td>
<td>0.71</td>
</tr>
</tbody>
</table>

3.7. RQ6: What Is the Performance of Search-Based Techniques in Defect Prediction?

Evolutionary computation techniques are learning techniques based on the theory of biological evolution. Techniques of this kind have been successfully used in various software engineering domains like Test Case Prioritization, Automated Test Data Generation [90,91], etc. In the last five years, considerable work has been done on evaluating the applicability of these techniques to the defect prediction problem. The authors in [115] argued that the performance measures of a model can be used as fitness function for search-based techniques and these techniques are better suited for software engineering problems given their superior ability to handle noisy data, better tolerance to missing data and ability to search globally with less probability of returning the local optimal solution.

The most studied techniques on the basis of Table 11 are the Artificial Immune Recognition System (AIRS), Ant Colony Optimization (ACO) and Genetic Programming (GP); all three techniques have been evaluated by two studies. Evolutionary Programming (EP), Evolutionary Subgroup Discovery (ESD), Genetic Algorithm (GA), and Gene Expression Programming (GeP) and Particle Swarm Optimization (PSO) have been studied only once.
Table 11. Search-based techniques used in literature.

<table>
<thead>
<tr>
<th>Primary Study</th>
<th>ACO</th>
<th>AIRS</th>
<th>EP</th>
<th>ESD</th>
<th>GA</th>
<th>GP</th>
<th>GeP</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE22</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE45</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE57</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE59</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE72</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE76</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SE90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SE92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

To evaluate the performance of a technique, it is important that a good number of independent studies be conducted and the results of all such studies are considered. Out of eight techniques, only three have been used more than once. Hence, for the performance evaluation we consider only these three techniques.

3.7.1. Ant Colony Optimization

In this mapping, two studies have evaluated the performance of ACO. One study uses a derivative of ACO, i.e., AntMiner+ while the other uses the traditional ACO algorithm. Both the studies have reported the prediction accuracy of the resultant models. The study using AntMiner+ has a prediction accuracy of 90.87 while the study using conventional ACO has a prediction accuracy of 64 percent.

3.7.2. Artificial Immune Recognition System

The AIRS technique reports an average recall of 74 percent in SE22 on the KC1 dataset which is very good while SE57 reports a recall of 82 percent on telecommunication software.

3.7.3. Genetic Programming

In SE57, the GP algorithm has reported a recall of 98 percent which is very high. It is evident that search-based techniques have a good predictive capability and this domain should be explored further. Algorithms like Artificial Bee Colony (ABC) have been used widely and in most cases the ABC algorithm has produced better results than other search-based techniques. It is necessary that the performance of the ABC algorithm be evaluated for defect prediction.

3.8. RQ7: How Many DeP Studies Attempt to Predict Security Related Defects and Vulnerabilities?

The question related to the identification of security related defects has not been discussed in any systematic mapping to the best of the authors’ knowledge. In this study, we estimate the number of studies which take steps in this direction. We try to visit every aspect of security related defects in our response to this question. It was found that only 3% have tried to address security related defects and vulnerabilities.

In Table 12 we present the different datasets that have been used in these three studies. Out of three, two studies use datasets provided by Mozilla Foundation, one study uses data given by Red Hat and one study uses data from an unspecified source.
Table 12. Primary study used to answer research questions.

<table>
<thead>
<tr>
<th>Primary Study</th>
<th>Dataset Used</th>
<th>Independent Variables Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE42</td>
<td>Mozilla Based</td>
<td>Traditional Source Metrics (McCabe Cyclomatic Complexity, LOC)</td>
</tr>
<tr>
<td>SE79</td>
<td>Mozilla Based, Red Hat Enterprise Linux Based</td>
<td>Traditional Source Metrics (Complexity, Coupling, Code Churn, Comments, Developer Activity, Developer Cluster, Contribution Centrality)</td>
</tr>
</tbody>
</table>

Table 13 gives the details about the learning methods used for the three studies and it can be seen that all three studies have used statistical techniques. There is no use of machine learning or search-based approaches for security related defect prediction.

Table 13. Primary studies used to answer research questions.

<table>
<thead>
<tr>
<th>Primary Study</th>
<th>ML</th>
<th>ST</th>
<th>SB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE42</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SE43</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SE79</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ML—Machine Learning ST—Statistical Technique | SB—Search-Based.

Table 14 shows various statistical tests that are used to evaluate the results of studies working on security related defects and system vulnerabilities. It is encouraging to see that only one out of three studies did used statistical tests.

Table 14. Primary studies used to answer research questions.

<table>
<thead>
<tr>
<th>Primary study</th>
<th>Statistical Test Applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE42</td>
<td>Wilcoxon Signed Rank Test</td>
</tr>
<tr>
<td>SE43</td>
<td>-</td>
</tr>
<tr>
<td>SE79</td>
<td>Weltch-t-Test, Bonferoni Test</td>
</tr>
</tbody>
</table>

3.9. RQ8: How Many DeP Studies attempt to Identify Defects of Varying Severity Levels?

Severity is a very important attribute of a defect. When a defect is detected in a software system, the severity assigned to it helps project managers identify the level of impact it can have on the system. A DeP model that can estimate the severity of a possible defect is very helpful to practitioners, because it makes them aware of the modules which can have defects of high severity, moderate severity and low severity and the order of action ideally should be testing of modules with high severity defects first, then modules with defects having moderate severity level and finally those modules which have defects of low severity. Out of the total 98 studies, only four studies (SE20, SE41, SE50, and SE62) have attempted to build DeP models that identify multiple severity levels.

3.10. RQ9: How Many DeP Studies Perform Cross-Project DeP?

Cross-project defect prediction is a very important aspect of DeP in software. There are software organizations that do not have enough historical data that can be used in DeP in current projects. Cross-project DeP uses historical data from different projects of a similar nature and this data can be used by software organizations lacking historical data to build and train DeP models. These trained models can then be used to identify defect prone parts of running projects. Cross-project defect prediction means that the DeP model trained on one set of data and is validated on the data obtained
from different software. Essentially, this is what makes a DeP model applicable in a real-life situation. A DeP model is useful only if it can predict defects in future projects, but to apply a DeP model to future projects it must be trained on past project data. To test the effectiveness of such a model and to simulate this real-life application scenario, cross-project defect prediction models are constructed. These models are trained on data from project X and tested on data from project Y. In literature, most of the studies use the same data for training and validation using various types of cross-validation techniques like hold-out, k-cross-validation, etc. Only six studies (SE46, SE54, SE56, SE84, SE85, and SE96) of the total 98 studies have attempted to build cross-project DeP models.

It is important that a good amount of work in future is done on cross-project DeP. The work should cover a diverse range of software projects so that the applicability of a DeP model to projects on which is not trained is thoroughly evaluated.

4. Conclusions

In this systematic mapping, we address nine research questions corresponding to different stages of development of a DeP model. We explored each aspect of the process ranging from data collection; data pre-processing, and techniques used to build DeP models to measures used to evaluate model performance and statistical evaluation schemes used to mathematically validate the results of a DeP model. We also looked into the existing literature to find out how many studies address security related defects and software vulnerabilities, how many cross-project DeP studies have been conducted and how many studies address defects of varying severity levels. To achieve these goals, we carried out an elaborate search on six digital libraries and collect 156 studies which are then processed using a carefully designed inclusion-exclusion criteria and quality analysis questionnaire. Out of the total 156 studies, we selected 98 studies for addressing nine research questions formed for this systematic mapping.

The research questions in this systematic mapping were constructed by taking into account the following definition of an ideal DeP model: “An ideal DeP model should be able to classify defects on the basis of severity, should detect security-related defects and system vulnerabilities and should have the ability to detect defects on systems on which it was not trained”.

Results of this study showed that the existing literature has covered some of the parts of the DeP process fairly well, for example, the sizes of datasets used are large, and nearly all machine learning methods have been examined. But when taking into account the overall approach and effectiveness of DeP studies, there are a lot of shortcomings. Studies have not made much use of multi-co-linearity analysis techniques and only half of the studies selected for mapping have used feature sub-selection techniques. There are only three studies that perform DeP for security related defects and system vulnerabilities, only 6% of the total studies perform cross-project DeP and only four studies have worked on defects of varying severity levels. The number of studies that use statistical tests for model comparison is also limited at 23 studies. Overall, we conclude that although the literature available for software defect prediction has high-quality work, it lacks in the overall methodology that is used to construct defect prediction models.

The following future guidelines are provided on the basis of the results of this study:

1. Datasets used for DeP studies should undergo thorough pre-processing that includes multi-co-linearity analysis and feature sub selection.
2. Most of the researches in defect prediction involve data obtained from open source software systems. Few studies use industrial datasets. It is important that industrial data be used for building defect prediction models so that the models can be generalized.
3. Future studies should make extensive comparisons between search-based techniques, machine learning techniques and statistical techniques. This mapping found out that few search-based techniques like AIRS and GP have good performance in predicting defects, but that number of studies that support this finding is very small. Machine learning and statistical techniques,
on the other hand, have been evaluated extensively. To reach conclusions it is important that search-based techniques are also extensively evaluated [173–187].

4. Since the amount of data is large, it is necessary that statistical evaluation of the performance of resultant DeP models be performed to reach the conclusion that is backed by mathematical principals.

5. Security is an important aspect of any software system. It is necessary that researchers explore the applicability of DeP models specifically to security related defects and vulnerabilities.

6. Severity is a basic component of any defect detected in a software system. If the testing resources are limited in number, then it makes sense to test those modules first that have a possibility of carrying high severity defects. It is essential that more studies that address security related defects be conducted.

7. For organizations looking to start new projects, it is important that they can make use of training data from existing DeP models. In reality, a DeP model is useful only if it can apply the rules learned from project A to project B. to explore the possibility of this application, cross-project DeP models should be built, and the focus should be on identification of metrics that provide maximum information about defects.

Author Contributions: N.P.: Conceived of the presented idea, Verified the analytical methods submitted the second author; M.K.: Select literature and discuss the findings, Monitored the results; L.H.S.: Started the literature findings, Conceived the overview and background study, Supervised the findings of this work; R.K.: Contributed to the overview and background study, Explained various issues; P.H.T.: Checked and verified the mathematical models, Supervised the findings of this work, Contributed to conclude the paper; P.T.M.P.: Contributed to the overview and background study, Discussion of experimental results.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank the Editor-in-Chief and the anonymous reviewers for their valuable comments and suggestions.

Conflicts of Interest: The authors declare that they do not have any conflict of interest.

References


86. Basili, V.R.; Briand, L.; Melo, W.L. *A Validation of Object-Oriented Design Metrics as Quality Indicators*; Technical Report; University of Maryland, Department of Computer Science: College Park, MD, USA, 1995.


