Fall 10-8-2020

A Bibliometric Survey on the Reliable Software Delivery Using Predictive Analysis

Jalaj Pachouly
Symbiosis Institute of Technology (SIT), jalaj.pachouly.phd2019@sitpune.edu.in

Swati Ahirrao
Symbiosis Institute of Technology (SIT), swatia@sitpune.edu.in

Ketan Kotecha
Symbiosis Institute of Technology (SIT), director@sitpune.edu.in

A Bibliometric Survey on the Reliable Software Delivery Using Predictive Analysis

Jalaj Pachouly[1], Swati Ahirrao[2], Ketan Kotecha[3]

[1]*Research Scholar, Symbiosis Institute of Technology (SIT) affiliated to Symbiosis International (Deemed University), Pune, India.*

Email: jalaj.pachouly.phd2019@sitpune.edu.in

[2]*Ph.D. Guide and Associate Professor, Symbiosis Institute of Technology (SIT) affiliated to Symbiosis International (Deemed University), Pune, India.*

Email: swatia@sitpune.edu.in

[3]*Dr. Ketan Kotecha, Head, SCAAI, Symbiosis Institute of Technology (SIT) affiliated to Symbiosis International (Deemed University), Pune, India.*

Email: head@scaai.siu.edu.in, director@sitpune.edu.in

ABSTRACT

Delivering a reliable software product is a fairly complex process, which involves proper coordination from the various teams in planning, execution, and testing for delivering software. Most of the development time and the software budget's cost is getting spent finding and fixing bugs. Rework and side effect costs are mostly not visible in the planned estimates, caused by inherent bugs in the modified code, which impact the software delivery timeline and increase the cost. Artificial intelligence advancements can predict the probable defects with classification based on the software code changes, helping the software development team make rational decisions. Optimizing the software cost and improving the software quality is the topmost priority of the industry to remain profitable in the competitive market. Hence, there is a great urge to improve software delivery quality by minimizing defects and having reasonable control over predicted defects. This paper presents the bibliometric study for "**Reliable Software Delivery using Predictive analysis**" by selecting 450 documents from the Scopus database, choosing keywords like software defect prediction, machine learning, and artificial intelligence. The study is conducted for a year starting from 2010 to 2021. As per the survey, it is observed that Software defect prediction achieved an excellent focus among the researchers. There are great possibilities to predict and improve overall software product quality using artificial intelligence techniques.

Keywords: Software defect prediction, machine learning, artificial intelligence.

1. INTRODUCTION

Delivering the software product with good quality on the scheduled date is very crucial for a profitable business. One of the significant hurdles in this task is unidentified defects, usually hidden in the code, having the code changes for the new feature requirements, or fixing other existing code issues. As per the report [51] published in September 2018, in the Consortium for IT Software Quality (CISQ) states that more than 50% of the total software cost is consumed by the defect finding and fixing losses from the software failures post-deployment.

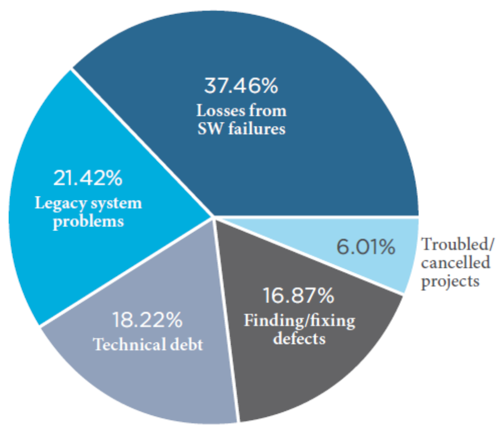


Figure 1: The Cost of Poor Quality Software in the US: A 2018 Report [51]

Another study carried out by Cambridge University's Judge Business School [53] found that software developers spent nearly 50% of the time finding and fixing the bugs. Further to this, they also indicated that 30% to 50% of every dollar is spent finding and fixing bugs. These are just a few numbers. Static code analysis can improve the software's quality by finding bugs based on statically defined rules specific to the language. However, it does not suffice the need for defect predictions for smooth planning and delivery. Most of the time, delivery-related decisions are taken by the leadership using a manual process and mostly based on their experience, which may not bring enough information on the table to justify the decisions and also consumes a lot of time. There is a great urge to have a tool that can optimize the release process, increase confidence in taken decisions, and ensure reliable software delivery with minimum bugs. Hence, presenting the bibliometric survey to improve the overall software delivery, focusing on software defect prediction, and forecasting other essential factors using artificial intelligence.

1.1 Bibliometric Analysis of Software Defect Prediction using Artificial Intelligence:

Software defect prediction is finding probable unidentified bugs in the proposed software delivery. Predictive analysis using artificial intelligence have proven their worth in many domains. Bibliometric analysis is a great way to get the current trends and understand what has been accomplished so far in software delivery using predictive analysis and exploring further literature to optimize the delivery process. The bibliometric term was first coined by Paul Otlet in 1934[55]. The bibliometric study uses the statistical tools and techniques to quantify the progress done in a specific research field using available articles, a paper published in various journals, books, etc.

Following are the high-level goals for the bibliometric study-

- To identify the various research publications in the field of research.
- Various languages in which research papers are published.
- The trend for publication over various years.
- The geographical study which contains the various countries where the research has conducted in the past.
- Trend line based on the source type.
- Theme diagram around the field of the research.
- Top authors who have contributed a lot in the field of the research.
- Publication trend based on the University or Organization.
- The publication received a citation.

This paper presents a bibliometric survey for software defect prediction to improve software delivery reliability. Section 2 highlights the collection of data related to software defect prediction. Section 3 shows the data which is extracted from the Scopus database. Three kinds of analysis, which are Network analysis, Statistical analysis, and Theme and Concept diagram analysis, are conducted in this section. Discussions from the analysis are shown in Section 4. Limitations are represented in Section 5, and the conclusion of the paper is drawn in Section 6. References are cited in the last section of the paper.

2. INITIAL COLLECTION OF DATA

There are various ways by which one can collect the research papers, articles like using the open-access database, paying the fee for individual paper as listed on the source site, or getting access to the papers using the institution or organization login. An excellent amount of the data related to published papers, including statistical information, is available on Google Scholar, Scopus, Research Gate, and Science Direct.

Scopus, a vast and prominent dataset with an excellent reputation in the research community, has chosen for doing bibliometric analysis.

2.1 Prominent keywords

The prominent keywords concerning software defect prediction analysis are divided into primary and secondary keywords.

Table 1. Shows search keywords used as a search strategy for selecting the data for this research. In the presented paper, research is limited to the years starting from 2010 to 2021; hence, it excluded the year less than 2010.

Table 1: Selection of search keywords for Software Defect Prediction

Primary keyword	"Software defect prediction".
Secondary keyword using (AND)	"Machine Learning".
Secondary keywords using (OR)	"Artificial Intelligence".

The exact query fired for the search for the documents in the Scopus dataset is:

```
(TITLE-ABS-KEY("software defect prediction")) AND ((machine
learning)) AND (artificial intelligence) AND ( EXCLUDE (
PUBYEAR,2009) OR EXCLUDE ( PUBYEAR,2008) OR EXCLUDE (
PUBYEAR,2007) OR EXCLUDE ( PUBYEAR,2006) OR EXCLUDE (
PUBYEAR,2005) )
```

2.2 Preliminary search results

The dataset used in this research paper is the Scopus database. The query, which is indicated in section 2.1, with relevant search keywords used as a search strategy, found the 450 publication on the Scopus database. The total result contains the papers where most of them are published and few, which is unpublished.

Table 2. Shows different types of publications in software defect prediction research. It is observed that 48.89% of the researchers have publicized their work in Articles, followed by conference paper, which contributes 47.56%. Conference review, book chapter, and review are the ones who are having meager contributions.

Table 2: Type of Publications in Software Defect Prediction

Type of Publications	**Number of Publications**	**Percentage**
Article	220	48.89%
Conference Paper	214	47.56%
Review	5	1.11%
Book Chapter	7	1.56%
Conference Review	4	0.89%
Total		100%

Dataset access information source: http://www.scopus.com (accessed on October 02, 2020)

The result from the search also analyzed for the type of language used for publishing documents. Table 3. Summarizes the contribution based on the language of published documents for Software defect prediction. As per the observation, English is the prominent language used by the researchers to publicize their papers and articles. Very few papers were written in the Chinese language.

Table 3: Languages trends used for publishing in Software Defect Prediction

Sr.No.	**Language used for Publishing**	**Count of Publications**
1	English	433
2	Chinese	17
Total		**450**

Data access information source: http://www.scopus.com (accessed on October 02, 2020)

2.3 Exploratory data highlights

Documents are collected on Software defect prediction using machine learning and artificial intelligence keywords from year starting 2010 to 2021. Table 4 indicates the trend of the yearly publication count on software defect prediction. Interpretation of this data indicates that most of the research contribution happened in the year 2018 and 2019. It is observed that contribution to the research was not significant from 2010 to 2015.

Table 4: Yearly publishing trends in Software Defect Prediction

Year	**Publication Count**
2021	1
2020	58
2019	101
2018	69
2017	44
2016	50
2015	27
2014	35
2013	16
2012	22
2011	15
2010	12

Data access information source: http://www.scopus.com (accessed on October 02, 2020)

Figure 2. Shows the result in the 2D line chart for publication count per year for Table 4. The Line chart represents the prominent year 2019, having the highest publication count of a total of 101 publicized documents in the area of Software defect prediction.

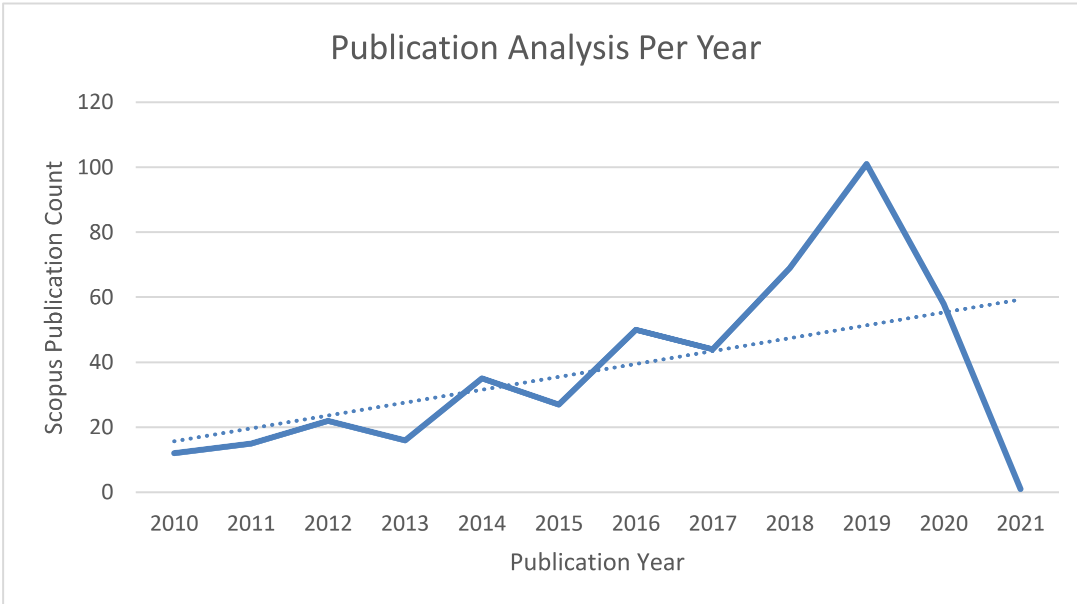


Figure 2: Yearly publishing trend in Software Defect Prediction
Data access information source: http://www.scopus.com (accessed on October 02, 2020)

2.4 Data Evaluation

Section 3 contains the detailed bibliometric analysis to understand the literature in their diversity and know more about the research and the researchers using the relevant keywords for software defect prediction. Different charts and graphs are created to showcase the depth and breadth of the research geographically and country-wise and based on affiliations to the organizations and the institutions. Theme and concept diagrams indicate the related concepts attached to the predictive analysis main topic for software delivery. Network diagrams are showcasing interesting relations like authors, citations, journals, etc.

3. BIBLIOMETRIC SURVEY

Three different methods are used to conduct the bibliometric survey for software defect prediction.

- The statistical analysis is majorly based on a country's contribution to the research area, contribution by subject area, author's affiliations, source type, authors, and source titles.
- Network analysis of research majorly based on geography, publication title, keywords, source title, year of publication, and collaboration among research authors, citation count, etc.
- Theme and concept diagrams, essential words with their weights across the literature, are extracted based on the query executed with the selected keywords.

3.1 Analysis based on geographic locations

Analysis for geographic location is carried out using the Google sheet tool, which needs input as two columns like Country name and the research paper count for that country. Once this data is provided to Google sheet, one can generate the geographical map based on the data, which shows the number of papers on specific geo-location hovering mouse on the map. There is a scale indicating the range of the publication count across the globe at the left bottom of the geographical map. In the generated geographical map, the green region indicated the location with the maximum number of published papers in the field of software defect prediction. According to the geographical map, the green region is China, with a maximum of 199 papers.

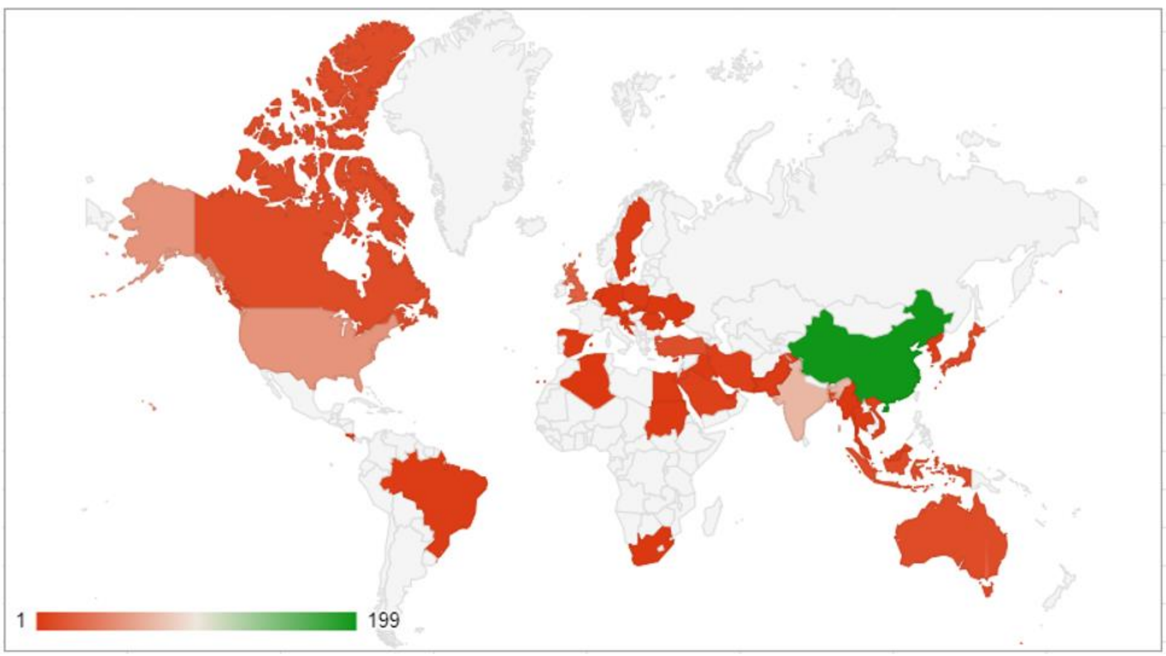


Figure 3. The geographical location of research Software Defect Prediction
Data access information source: http://www.scopus.com (accessed on October 02, 2020)
(Image Source: Google Sheet)

Figure 3 shows the top 10 contributing countries in the research of software defect prediction. The result is shown using the bar chart indicating that China contributes 44.22%, whereas the second contributor in India, with a 16% contribution in the research of Software defect prediction. Japan has the lowest contribution while considering the top 10 contributing countries.

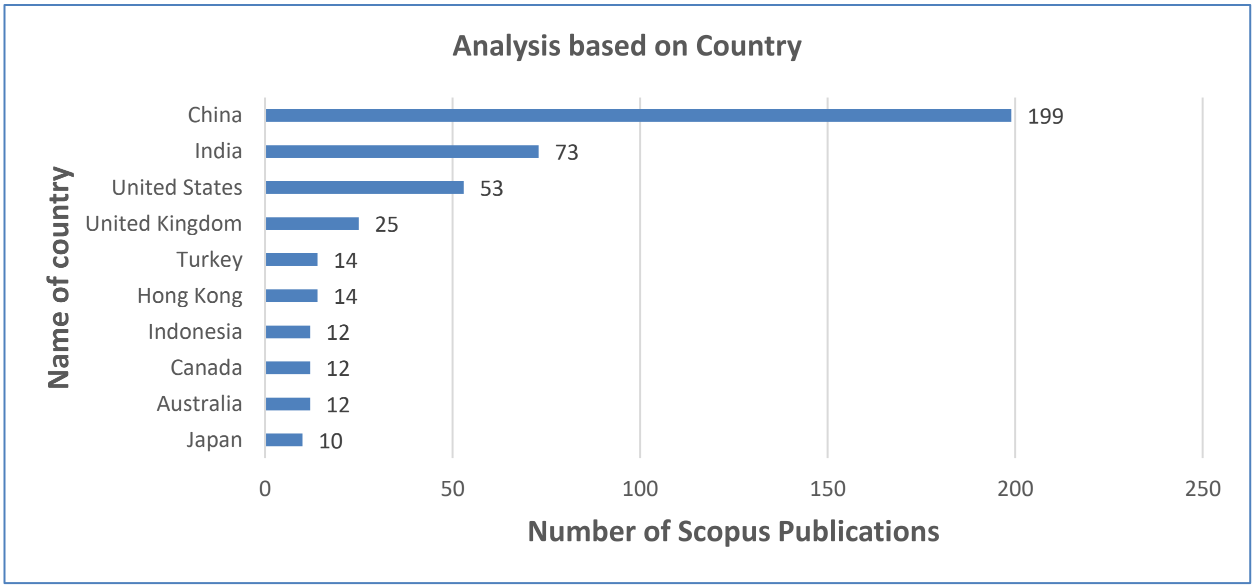


Figure 4: Ten topmost countries publishing papers on Software Defect Prediction
Data access information source: http://www.scopus.com (accessed on October 02, 2020)

3.2 Statistical analysis based on keywords

Table 5 indicates the top ten keywords for searching the Scopus database for software defect prediction analysis. By applying the keyword's right combination, one can select and filter the papers for the specific research area. Table 5 clearly shows that Software Defect Prediction is the most widely used keyword.

Table 5: Top ten keywords for Software Defect Prediction

Keywords	**Number of Publications**
Software Defect Prediction	158
Defects	103
Forecasting	88
Learning Systems	78
Defect Prediction	74
Computer Software Selection And Evaluation	72
Software Engineering	71
Software Testing	67
Machine Learning	64
Classification (of Information)	51

Data access information source: http://www.scopus.com (accessed on October 02, 2020)

3.3 Network Analysis

Network analysis shows the association among different attributes that add values in the computation. Network analysis shows the graphical diagram. Tool VOSviewer is used for generating various network analysis diagrams. Figures 5, 6, 7, 8 show the network analysis diagrams having various computable parameters for software defect prediction from the papers extracted from the Scopus database.

VOSviewer is a free tool that can be downloaded from the VOSviewer [55] website. VOSviewer can analyze the computable parameters using a bibliometric network. Input needs to be a comma-separated value file, also known as .csv file, to the VOSviewer. There are three kinds of visualization analysis using VOSviewer: Network visualization, Overlay visualization, and Density visualization.

Visualization between the keywords and the source titles is shown in figure 5, extracted from the Scopus database. Circles represent the keywords that are extracted from the title of the source. The size of the circle indicates the keyword occurrence. There are links between the circle, which shows the association among the keyword, less distance means a strong association, and more distance means weak association. Closely related keywords are represented with the same colors. There are different colors to represent the different clusters. Labels represent the actual keyword, size of the circle, and the label depends on the weight of the keywords. The bigger label represents keywords with higher weight, and the circle and the keywords with low weight are relatively small than higher weights. Lines represent links between words. The default value of the lines is 1000, which can represent 1000 strong links between keywords. Circles that are closer to each other have strong relations between them. The threshold value of the keyword's minimum occurrence was set to 5, and analysis is done by selecting all 200 keywords that satisfied the threshold value and limiting the number of keywords to 25, as mentioned in figure 5 and figure 6.

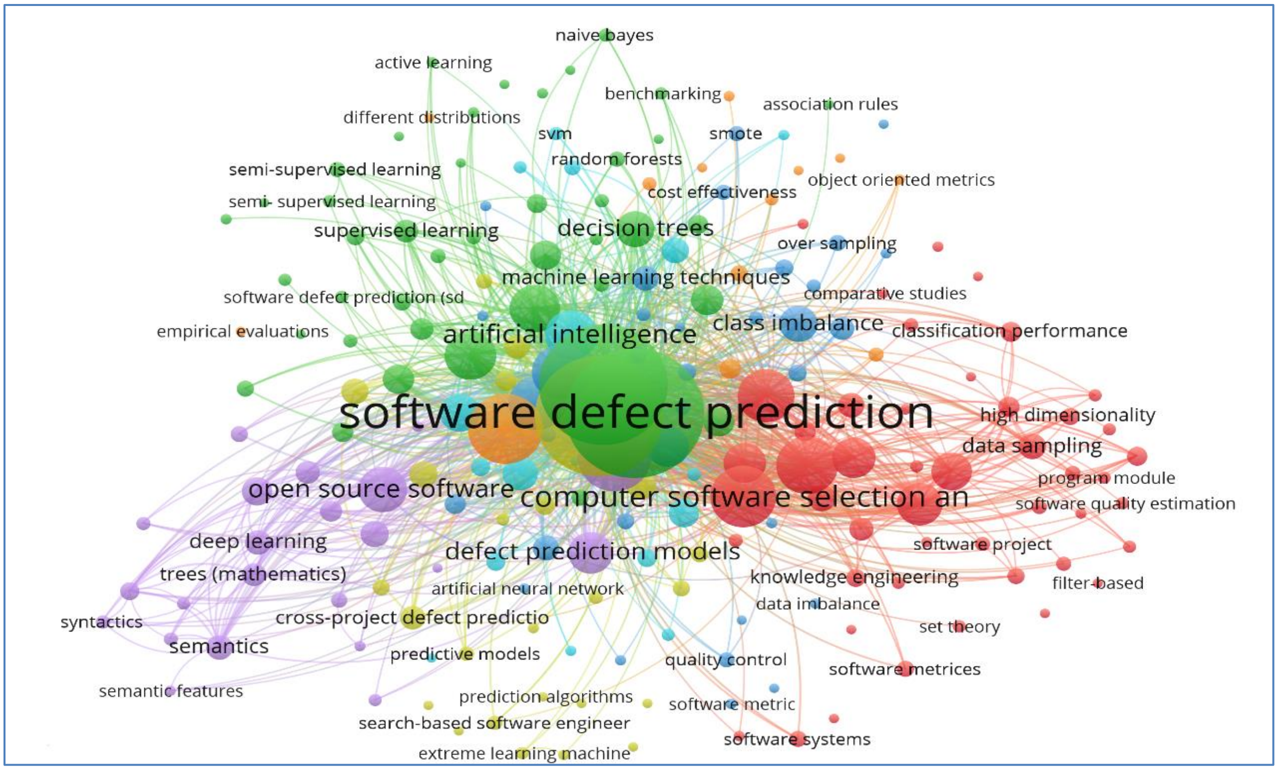


Figure 5: Network visualization diagram based on keywords and source title, with 200 keywords from Scopus dataset (accessed on October 02, 2020) (Image Source: https://www.vosviewer.com)

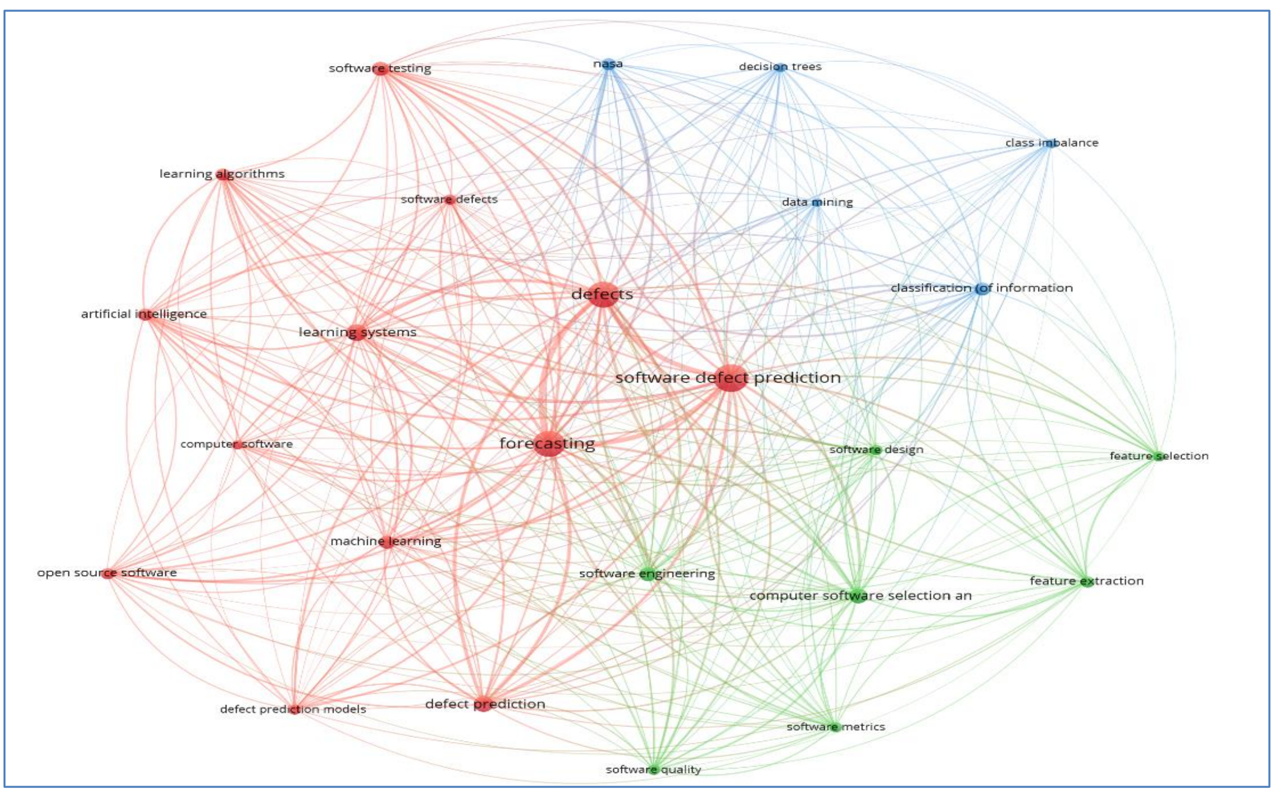


Figure 6: Network visualization diagram based on keywords and source title, with 25 keywords from Scopus dataset (accessed on October 02, 2020) (Image Source: https://www.vosviewer.com)

Figure 7 represents a cluster of co-authors and authors co-appearing among the same papers. The collaboration of their work is shown between the authors. The link represents the collaborative work of authors on the documents published. The author's threshold value having a minimum number of documents was set to 2 as a manual parameter, which resulted in 272 authors. The total strength of the co-authorship links with other authors is calculated and displayed in figure 7.

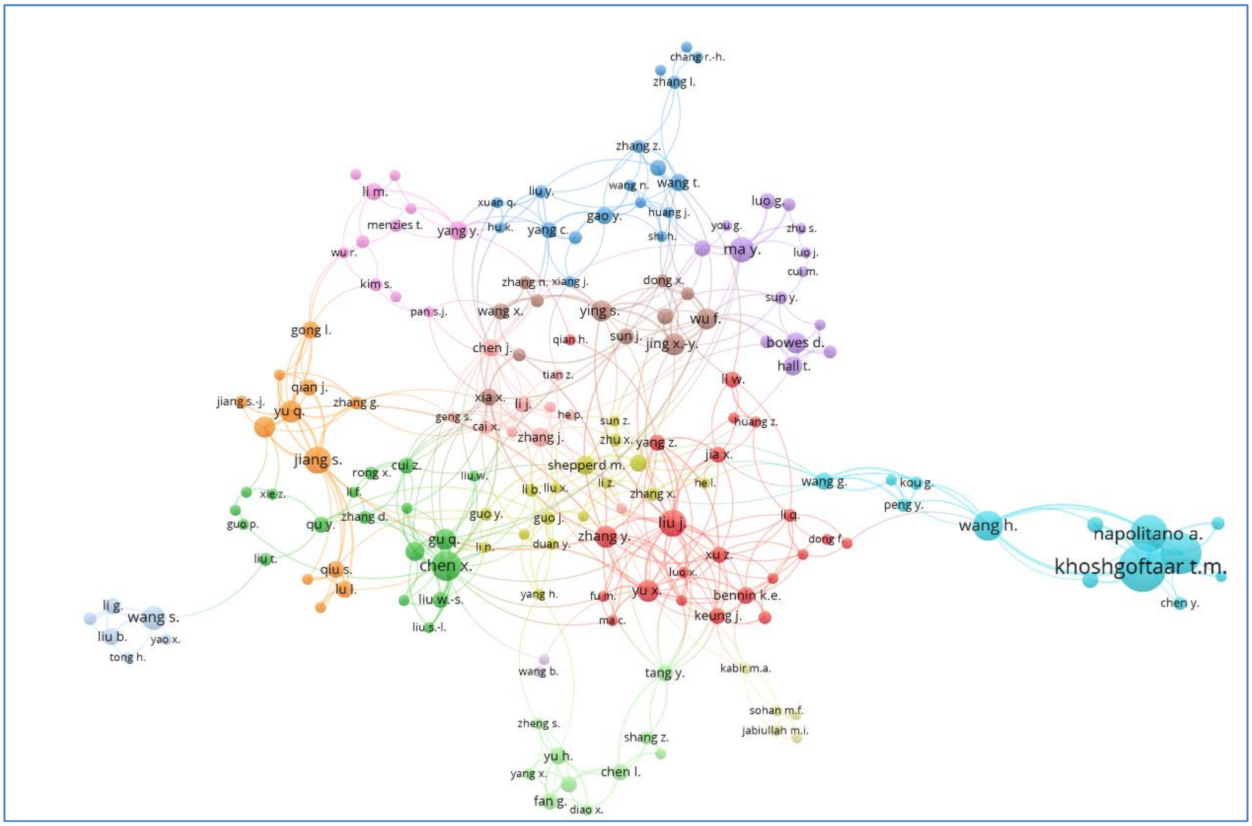


Figure 7: Network analysis diagram of co-authors and authors based on co-appearance among the same papers using Scopus dataset (accessed on October 02, 2020) (Image Source: https://www.vosviewer.com)

Figure 8 shows the visualization of the documents and the citations received by the document. The threshold value is set to two citations per document for this analysis, which retrieved 278 documents out of 450 documents, and accordingly, the citation link was calculated.

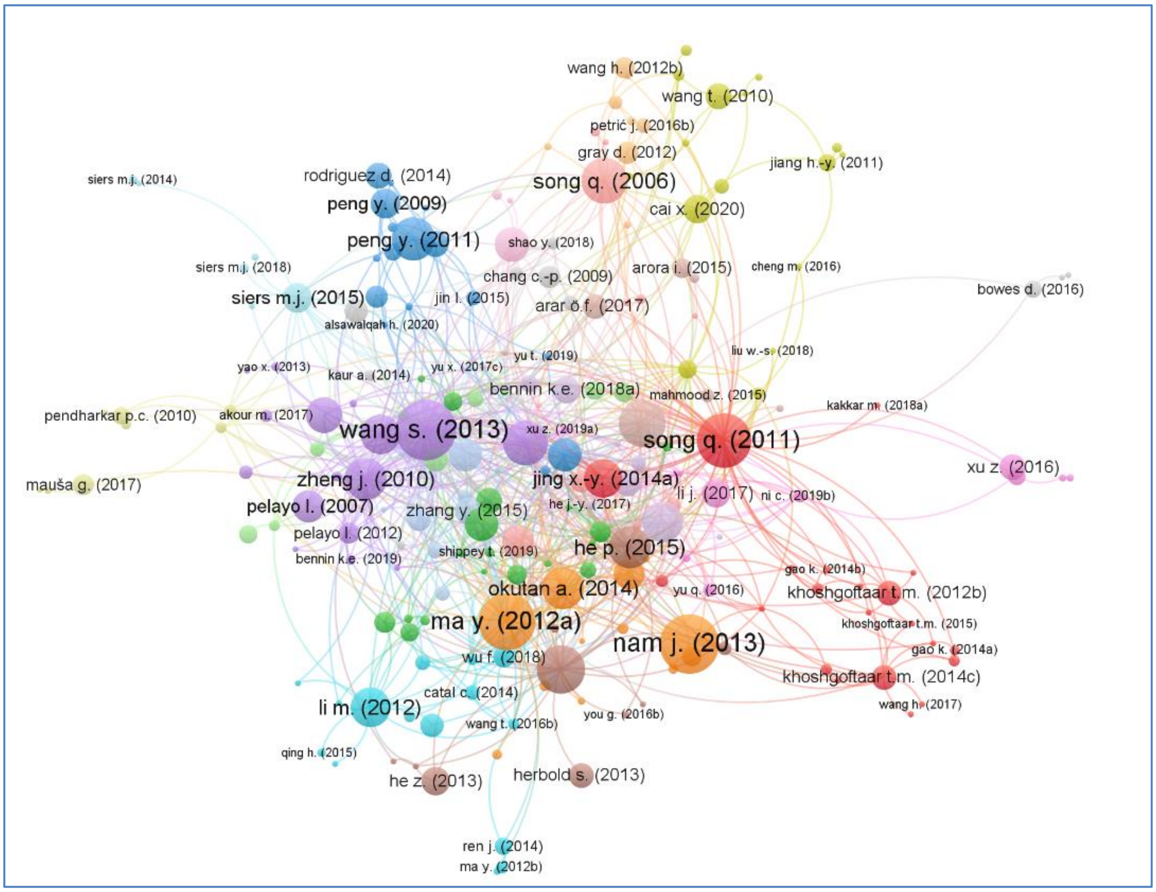


Figure 8: Network visualization of the document and the citations received by document using Scopus dataset (accessed on October 02, 2020) (Image Source: https://www.vosviewer.com)

3.4 Theme and Concept diagram

Themes and Concepts are useful to explore the area of the research in great depth. The theme is a more literal expression, whereas Concepts is the core idea, and generally, it is an abstract understanding of the experience. The concept can be part of many themes, and their collaboration can be used to generate more ideas in the research fields. Leximancer tool is a robust tool and can take many input formats like HTML, CSV, pdf, or text files and parse the documents to develop a theme and the concepts and showcase their association. Input is taken from CSV files in the current survey work, which got exported from the Scopus database having data of the 450 documents for Software defect prediction research. The bigger bubble indicates the theme's significance, as shown in figure 9, some of them are prediction. Machine, Classification, Software are diagnosed as a significant theme with higher weight.

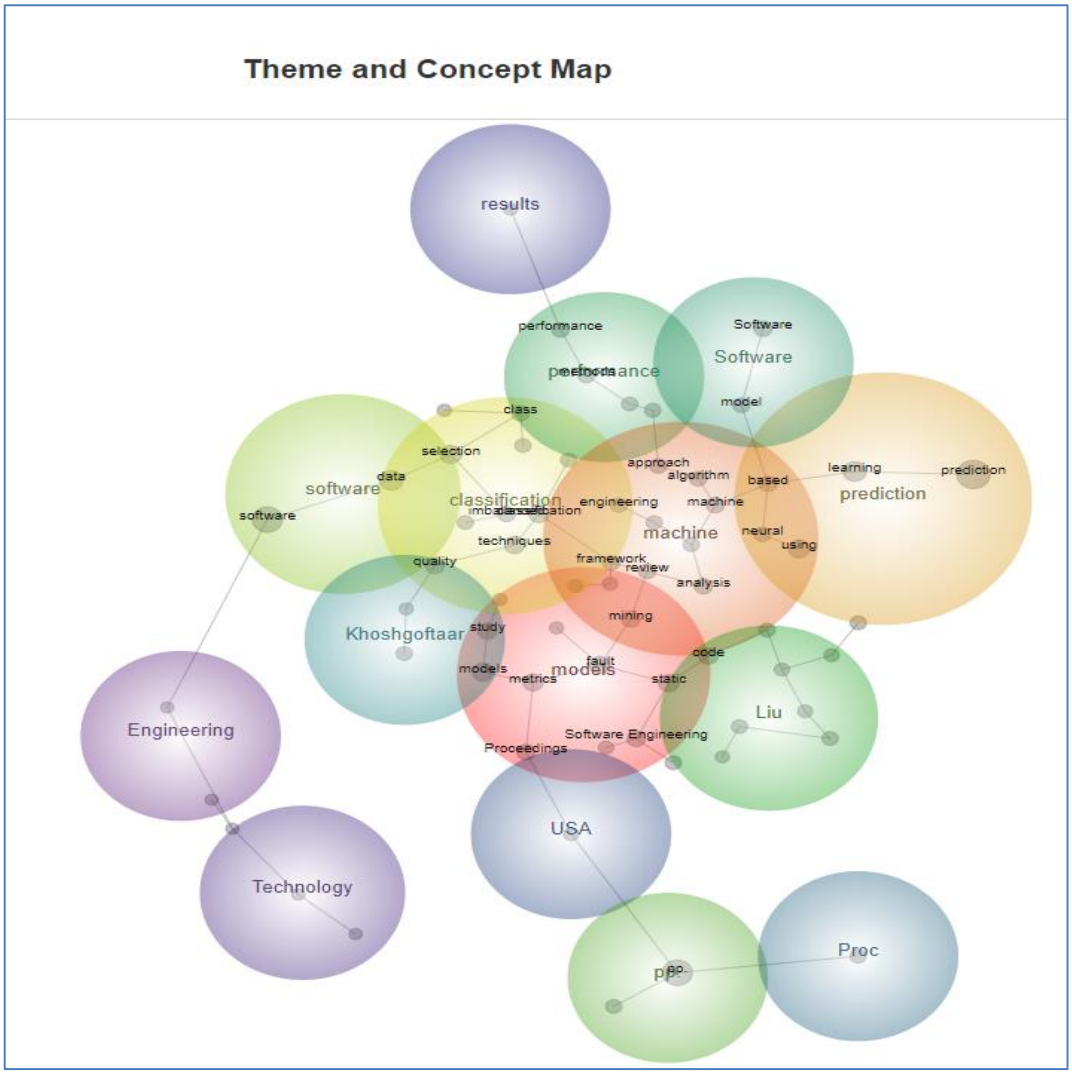


Figure 9: Theme and Concept diagram based on the Scopus database. (accessed on October 02, 2020)

(Image Source: https://lexiportal-app.leximancer.com/)

Figure 10 shows the Themes using the Leximancer tool based on the hits of the words in the dataset, exported as a CSV file from the Scopus dataset. Themes are shown in different colors according to their strength in the selected literature; circles with smaller sizes have a low score, whereas the theme with a higher score has the bigger circle. Overlap of the themes shows the similar concepts used in the themes, and their distance shows their association.

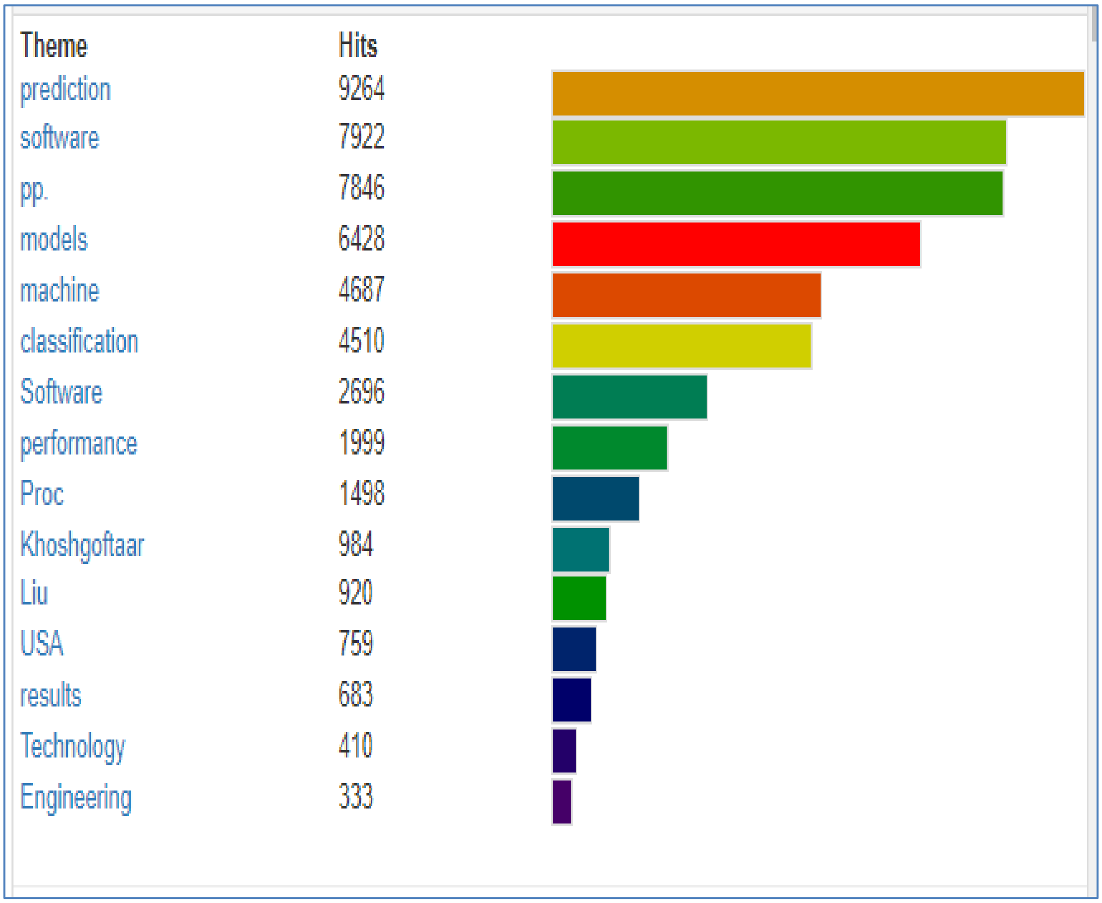
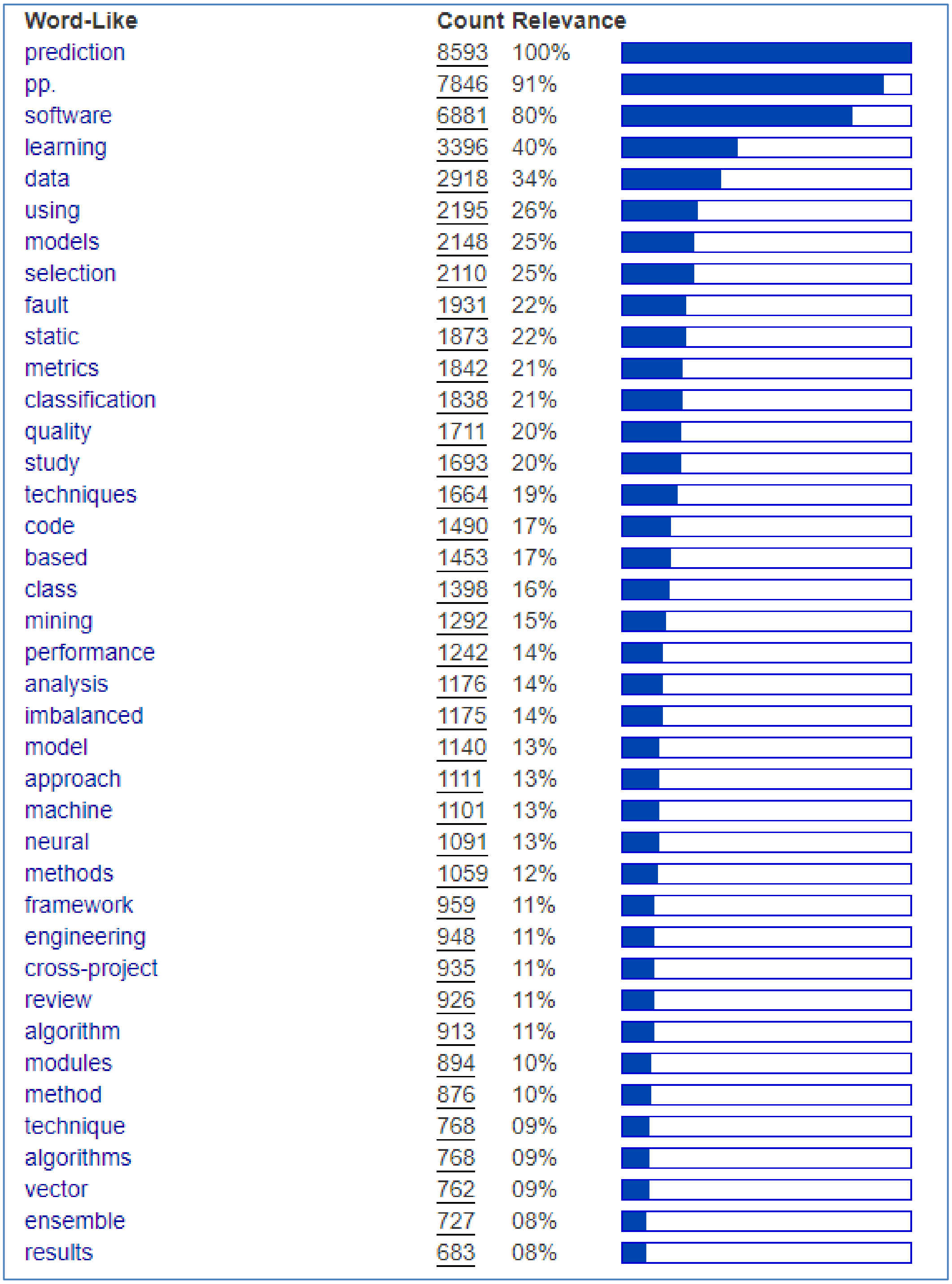


Figure 10. Theme calculated from the Scopus dataset based on the number of hits (accessed on October 02, 2020) (Image Source: https://lexiportal-app.leximancer.com/)

Figure 11 shows the rank of the words based on their relevance in the selected literature. As per the figure 11, it can be observed that Software, Prediction, and Learning are the highest among others from the perspective of relevance, which also matches the aim of the study.



Word-Like	Count	Relevance
prediction	8593	100%
pp.	7846	91%
software	6881	80%
learning	3396	40%
data	2918	34%
using	2195	26%
models	2148	25%
selection	2110	25%
fault	1931	22%
static	1873	22%
metrics	1842	21%
classification	1838	21%
quality	1711	20%
study	1693	20%
techniques	1664	19%
code	1490	17%
based	1453	17%
class	1398	16%
mining	1292	15%
performance	1242	14%
analysis	1176	14%
imbalanced	1175	14%
model	1140	13%
approach	1111	13%
machine	1101	13%
neural	1091	13%
methods	1059	12%
framework	959	11%
engineering	948	11%
cross-project	935	11%
review	926	11%
algorithm	913	11%
modules	894	10%
method	876	10%
technique	768	09%
algorithms	768	09%
vector	762	09%
ensemble	727	08%
results	683	08%

Figure 11. Ranked Word-like concepts using the Scopus database (accessed on October 02, 2020) (Image Source: https://lexiportal-app.leximancer.com/)

3.5 Statistical analysis based on Subject areas

Figure 12 shows the distribution of the publications in various disciplines extracted for Software defect prediction publications. It can be easily concluded that most of the research is conducted in the Computer Science area, followed by Engineering and Mathematics and Material science areas. Some research has been carried out in the area of Decision Science and Energy.

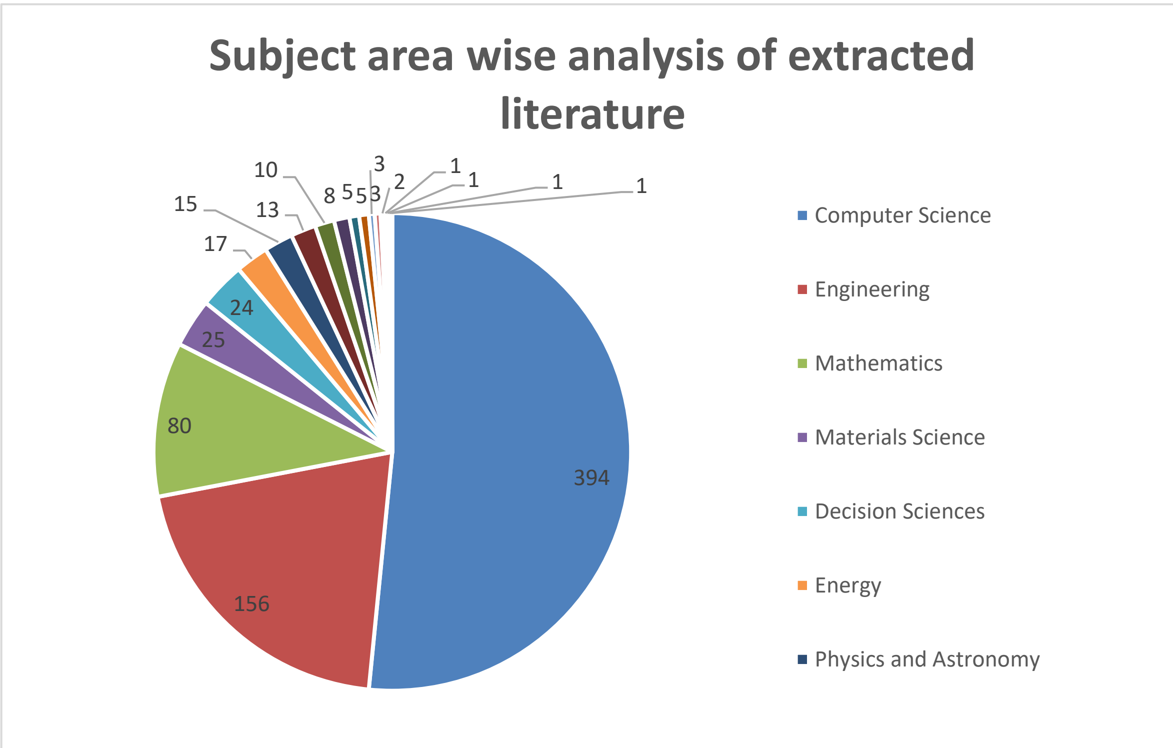


Figure 12: Subject area wise analysis of extracted literature for Software Defect Prediction. Data access information source: http://www.scopus.com (accessed on October 02, 2020)

3.6 Statistical analysis based on Affiliations

Affiliation statistic shows the contribution based on the universities and organizational affiliations. Figure 13 shows the top ten universities contributed to software defect prediction. The Wuhan University of China shows maximum contribution towards the research in the field of Software Defect Prediction, followed by Florida Atlantic University. Amity University, Noida form India, is at fourth rank.

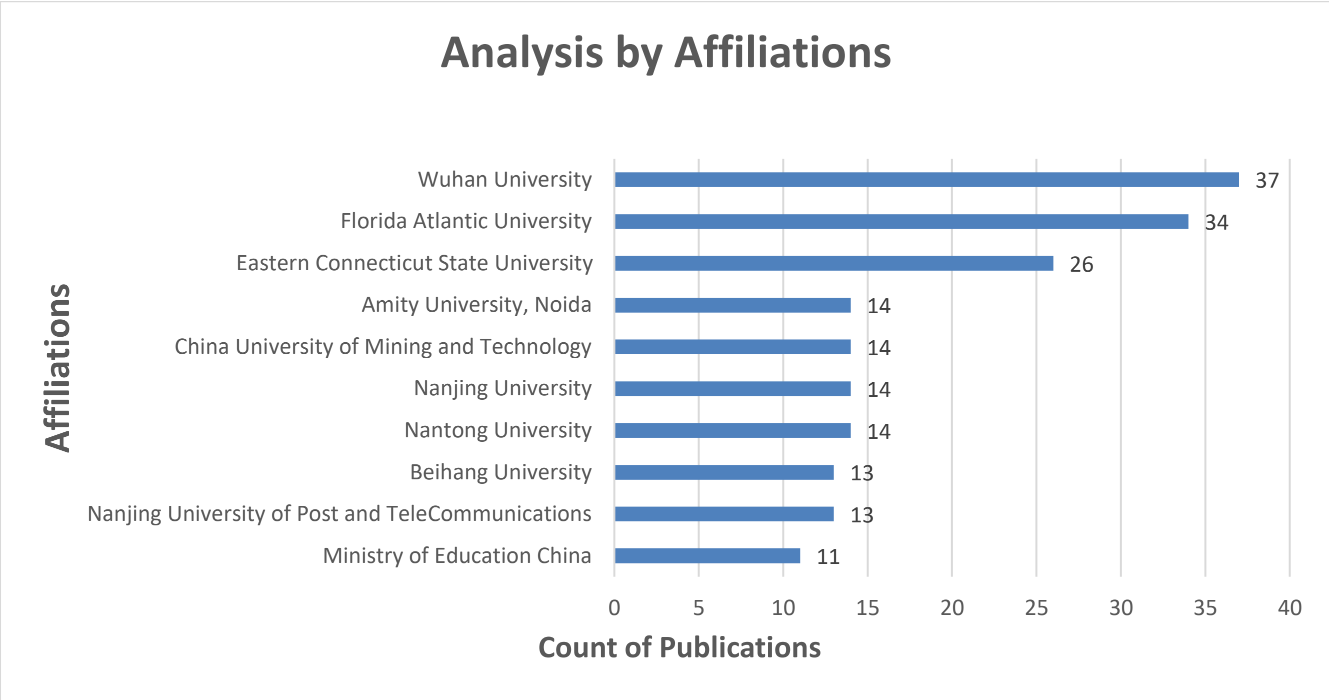


Figure 13: Affiliation statistics for Software Defect Prediction

Data access information source: http://www.scopus.com (accessed on October 02, 2020)

3.7 Statistical analysis based on Authors

Figure 14 shows the top ten authors with the maximum contribution in the area of Software Defect Prediction. The top contributing author belongs to Florida Atlantic University, USA. The top 3 contributing authors are from the USA.

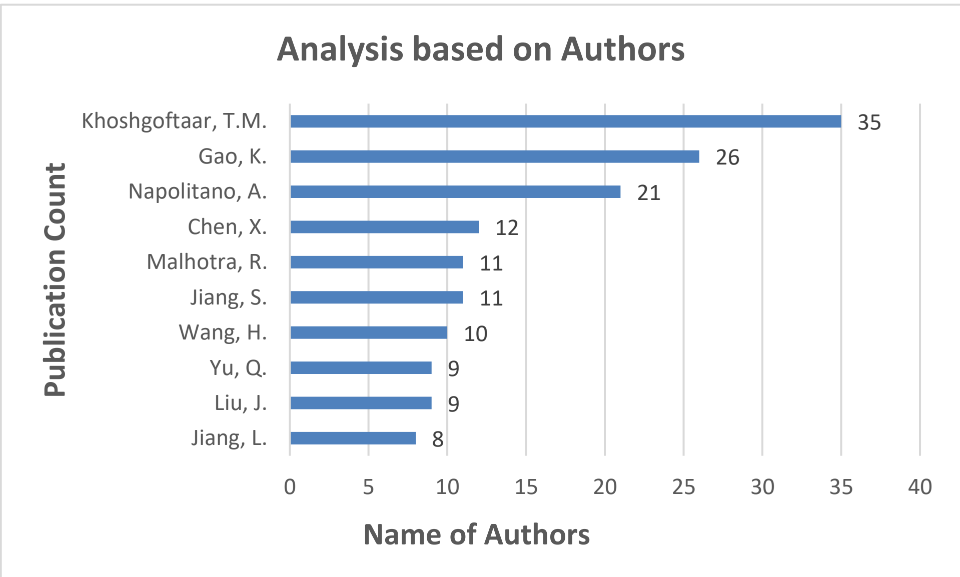


Figure 14: Top ten authors contributing to the research area of Software Defect Prediction
Data access information source: http://www.scopus.com (accessed on October 02, 2020)

3.8 Statistical analysis based on Source Types

Source types of scholarly articles mean where the original research work is published. It can be clearly stated from figure 15 that 49% of the publications are from Articles followed by 48% of publications in Conference proceedings. It has been observed that review publications are relatively low for the software defect prediction.

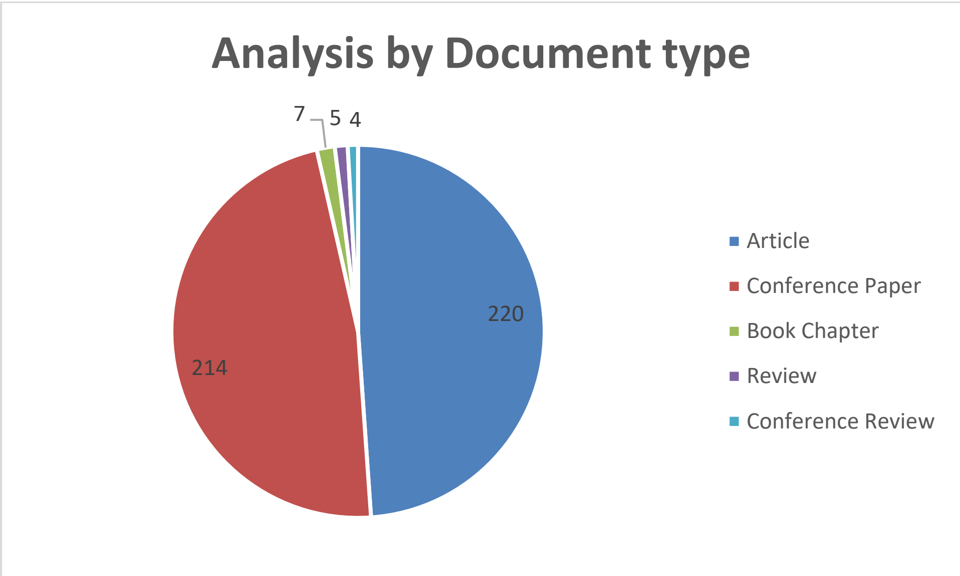


Figure 15: Source types for publications in Software Defect Prediction
(Source: http://www.scopus.com (accessed on October 02, 2020)

3.9 Analysis based on publication citations

Table 6 shows citations count based on years, extracted from publications extracted in the area of Software Defect Prediction. To date, the total citation count of 450 publications is 6440. Citation counts are low up to 2017, while the maximum number of citation are observed in 2019, followed by 2018.

Table 6: Analysis based on citations for publications in Software Defect Prediction

Year	<2016	2016	2017	2018	2019	2020	>2020	Total
No. of Citations	766	601	750	1123	1869	1316	15	6440

Data access information source: http://www.scopus.com (accessed on October 02, 2020)

The top ten publication titles extracted from the Scopus database that received the maximum number of citations to date are represented in Table 7. It can be observed that the research work with the title 'Using class imbalance learning for software defect prediction' gets the maximum number of citations in this field of Software defect prediction.

Table 7: An analysis of top ten publication based on citations in Software Defect Prediction

Publication Title	Citations received by the Publications yearly							
	<2016	2016	2017	2018	2019	2020	>2020	Total
Using class imbalance learning for software defect prediction	25	26	39	46	74	40	-	250
Transfer defect learning	36	36	35	41	58	36	-	242
Transfer learning for cross-company software defect prediction	41	26	38	39	53	29	-	226
A general software defect proneness prediction framework	66	31	27	31	38	15		208
Dealing with noise in defect prediction	52	31	17	34	37	18	-	189
Automatically learning semantic features for defect prediction		3	19	38	75	43	1	179
An investigation on the feasibility of cross-project defect prediction	29	26	25	34	37	15		166
Researcher bias: The use of machine learning in software defect prediction	14	26	21	27	43	24		155
Software defect prediction using ensemble learning on selected features	5	15	21	26	47	27		141
Ensemble of software defect predictors: An AHP-based evaluation method	92	8	6	5	14	9		134

Data access information source: http://www.scopus.com (accessed on October 02, 2020)

3.10 Statistical analysis based on source titles

Statistics based on the top ten source titles from retrieved literature are represented in figure 16 for software defect prediction publications. It is observed that the maximum numbers of publications are done in source title from "Lecture Notes in Computer Science Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics". However, ACM International Conference Proceedings ranks second, and IEEE Access comes $4^{th,}$ and IEEE Transactions on Reliability used less.

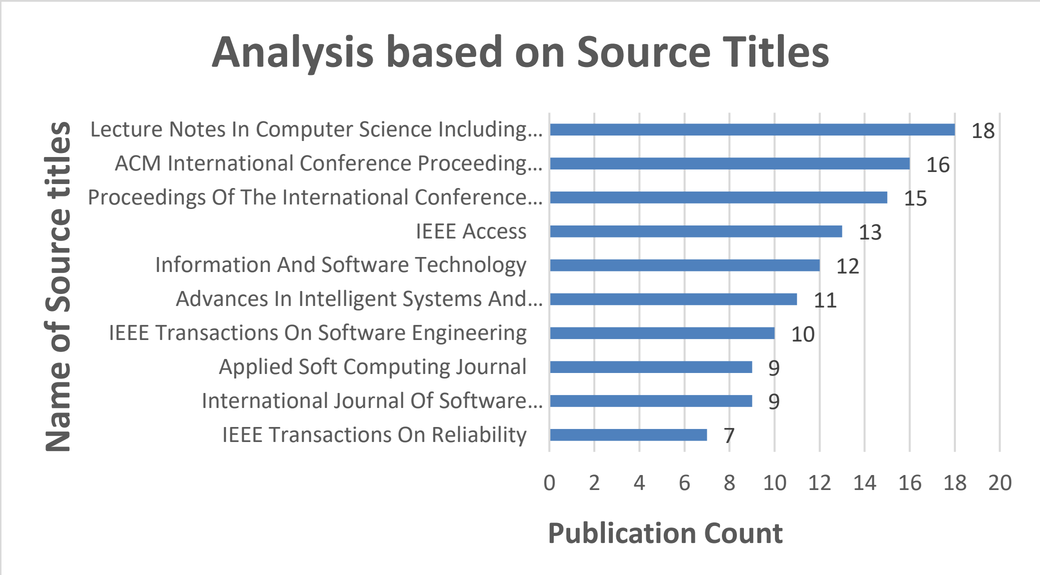


Figure 16: Source statistics for publications in Software Defect Prediction
Data access information source: http://www.scopus.com (accessed on October 02, 2020)

3.11 Analysis based on Funding Sponsors

Statistical analysis based on Funding sponsors in the Software Defect Prediction research area is shown in figure 17. The top 10 funding sponsors are considered based on the statistics. It can be observed that the National Natural Science Foundation of China is the highest funding foundation; also, the National Basic Research Program of China stands second.

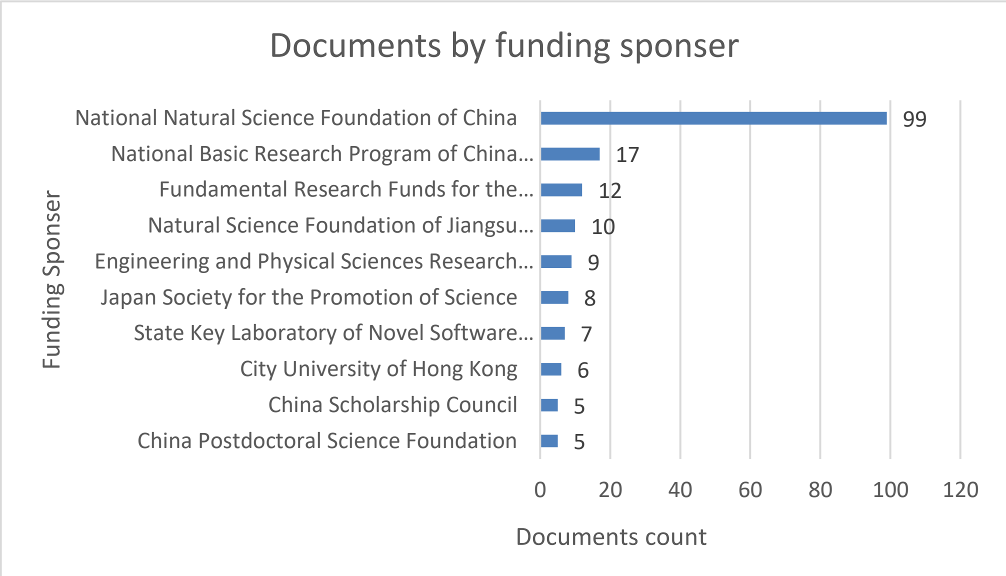


Figure 17: Funding Sponsors statistics in Software Defect Prediction Research Area
Data access information source: http://www.scopus.com (accessed on October 02, 2020)

4. DISCUSSION BASED ON THE RESEARCH STUDY

Research in software defect prediction is getting good momentum. Many studies were conducted to predict the bugs and other prominent factors that impact the delivered software quality. The present study majorly focuses primarily on the usage of Artificial Intelligence techniques in the field of Software defect prediction, which is the topmost concern for the successful delivery of the software product. As per the study, it is observed that most of the research work is presented majorly by articles and followed by the conference papers. Articles and conference papers are an excellent medium to communicate the research idea to the community. Peer feedback also improves the quality of the paper and the research related questions. It is found that English is the most preferred language for writing the research papers, followed by the Chinese language. It is observed that initial years have less research published from the range of the years for the current study, and then it keeps getting incremented year by year and reaches the maximum peak in the year 2019. It is observed that the top country, which contributes to the research area, is China, followed by India. Out of 450 selected papers, China contributed 199 papers, followed by India with 73 papers. Out of 450 papers, only 9 are review papers, which are quite a few numbers, which also motivated us to write this Bibliometric review. Table 5 shows the top 10 significant keywords like Software Defect Prediction, Defects, Forecasting, Learning Systems, Defect Prediction, Computer Software Selection, and Evaluation, Software Engineering, Software Testing, Machine Learning, Classification (of Information).

Further studies can explore those keywords to get their maximum advantage for any newly proposed research. The statistic shows that Amity University Noida from India is among the top 10 affiliations, which is a great indicator that India has the right presence in the software defect prediction field. India is the second most country that contributed to the maximum number of research papers after China out of 450 selected research documents for the study. Citation count is highest in the year 2019, reflecting the great interest in the research.

5. LIMITATIONS OF THE PRESENT STUDY

The current study was conducted with a limitation where the paper has considered only the Scopus database for selecting the state of the art literature. In contrast, more work could be taken into consideration, like Google Scholar and Web of Science. The listing of the existing research changes dynamically as per the arrangement of the keywords. Current research took care of the defined order of the keyword by the research authors. Hence, it could also be tried with various combinations and add or exclude a few synonyms representing a similar meaning as per the research are. The research paper considered publication with a limited set of years from 2010 to 2021 and did not consider the research before 2010.

6. CONCLUSION

In this bibliometric study, the significance of software defect prediction is presented using the Scopus dataset by selecting the 450 relevant documents based on the keyword: software defect prediction, machine learning, and artificial intelligence. Based on the survey, it is found that Software Defect Prediction is one of the hot topics of research among the research community, especially in China and India. The presented bibliometric study will surely help and motivate the young researchers to get a comprehensive understanding of the work done in the last ten years in the software defect prediction. Based on the survey, it can be stated confidently that more and more predictive techniques will immerge using artificial intelligence techniques and improve the software quality to make the software industry more mature.

REFERENCES

[1] Prabha, C. L. (2020). Software Defect Prediction Using Machine Learning Techniques. Icoei, 728–733.

[2] Hasanpour, A., Farzi, P., Tehrani, A., & Akbari, R. (2020). Software Defect Prediction Based On Deep Learning Models: Performance Study. http://arxiv.org/abs/2004.02589

[3] Wongpheng, K., & Visutsak, P. (2020). Software Defect Prediction using Convolutional Neural Network. 240–243.

[4] Yu, Q., Jiang, S., Qian, J., Bo, L., Jiang, L., & Zhang, G. (2020). Process metrics for software defect prediction in object-oriented programs. 283–292. https://doi.org/10.1049/iet-sen.2018.5439

[5] Technology, I., & Conference, A. C. (2020). Software Defect Prediction via Transformer. Itnec, 874–879.

[6] Sheng, L., Lu, L., & Lin, J. (2020). An adversarial discriminative convolutional neural network for cross-project defect prediction. IEEE Access, 8, 55241–55253. https://doi.org/10.1109/ACCESS.2020.2981869

[7] Ardimento, P., & Mele, C. (2020). Using BERT to Predict Bug-Fixing Time. IEEE Conference on Evolving and Adaptive Intelligent Systems, 2020-May. https://doi.org/10.1109/EAIS48028.2020.9122781

[8] Immaculate, S. D. (2019). Machine Learning Algorithms. 2019 International Conference on Data Science and Communication (IconDSC), 1–7.

[9] Hammouri, A., Hammad, M., Alnabhan, M., & Alsarayrah, F. (2018). Software Bug Prediction using machine learning approach. International Journal of Advanced Computer Science and Applications, 9(2), 78–83. https://doi.org/10.14569/IJACSA.2018.090212

[10] Lal, H., & Pahwa, G. (2017). Root cause analysis of software bugs using machine learning techniques. Proceedings of the 7th International Conference Confluence 2017 on Cloud Computing, Data Science and Engineering, 105–111. https://doi.org/10.1109/CONFLUENCE.2017.7943132

[11] Rhmann, W., Pandey, B., Ansari, G., & Pandey, D. K. (2020). Software fault prediction based on change metrics using hybrid algorithms: An empirical study. Journal of King Saud University - Computer and Information Sciences, 32(4), 419–424. https://doi.org/10.1016/j.jksuci.2019.03.006

[12] Grundy, J. (2019). Towards Effective AI-powered Agile Project M anagement. 41–44. https://doi.org/10.1109/ICSE-NIER.2019.00019

[13] Choetkiertikul, M., Dam, H. K., Tran, T., Pham, T., Ghose, A., & Menzies, T. (2019). A Deep Learning Model for Estimating Story Points. IEEE Transactions on Software Engineering, 45(7), 637–656. https://doi.org/10.1109/TSE.2018.2792473

[14] Zhang, J. I. E., Member, G. S., Wu, J., & Member, S. (2020). CDS : A Cross – Version Software Defect Prediction Model with Data Selection. 110059–110072. https://doi.org/10.1109/ACCESS.2020.3001440

[15] Choetkiertikul, M., Dam, H. K., Tran, T., & Ghose, A. (2017). Predicting the delay of issues with due dates in software projects Predicting the delay of issues with due dates in software. Empirical Software Engineering, January 2018. https://doi.org/10.1007/s10664-016-9496-7

[16] Li, N., Shepperd, M., & Guo, Y. (2020). A Systematic Review of Unsupervised Learning Techniques for Software Defect Prediction.

[17] Qiao, L., Li, X., Umer, Q., & Guo, P. (2020). Neurocomputing. Neurocomputing, 385, 100–110. https://doi.org/10.1016/j.neucom.2019.11.067

[18] Bezemer, M. K. C. (n.d.). The Impact of Feature Reduction Techniques on Defect Prediction Models.

[19] Chen, J., Hu, K., Yu, Y., Chen, Z., Xuan, Q., Liu, Y., & Filkov, V. (n.d.). Software Visualization and Deep Transfer Learning for Effective Software Defect Prediction.

[20] Jiang, L. I., Jiang, S., Dong, Y. U. E., & Yu, Q. (2020). Which Process Metrics Are

Significantly Important to Change of Defects in Evolving Projects : An Empirical Study? 93705–93722. https://doi.org/10.1109/ACCESS.2020.2994528

[21] Extensive, A., & Study, E. (n.d.). SS symmetry Impact of Feature Selection Methods on the Predictive Performance of Software Defect Prediction Models : An Extensive Empirical Study. Ml.

[22] Deshpande, B. S., Kumar, B., & Kumar, A. (2020). Object Oriented Design Metrics for Software Defect Prediction : An Empirical Study. May.

[23] Sun, Z., Zhang, J., Sun, H., & Zhu, X. (2020). Collaborative filtering based recommendation of sampling methods for software defect prediction. Applied Soft Computing Journal, 90, 106163. https://doi.org/10.1016/j.asoc.2020.106163

[24] Ding, Z., & Xing, L. (2020). Improved software defect prediction using Pruned Histogram-based isolation forest. Reliability Engineering and System Safety, 204(May), 107170. https://doi.org/10.1016/j.ress.2020.107170

[25] Morasca, S., & Lavazza, L. (2020). On the assessment of software defect prediction models via ROC curves. 3977–4019.

[26] Shao, Y., Liu, B., Wang, S., & Li, G. (2020). Knowledge-Based Systems Software defect prediction based on correlation weighted class association rule mining. Knowledge-Based Systems, 196, 105742. https://doi.org/10.1016/j.knosys.2020.105742

[27] Alsawalqah, H., Hijazi, N., Eshtay, M., & Faris, H. (2020). Applied sciences Software Defect Prediction Using Heterogeneous Ensemble Classification Based on Segmented Patterns. https://doi.org/10.3390/app10051745

[28] Jin, C. (2020). Sample set. Soft Computing, 1. https://doi.org/10.1007/s00500-020-05159-1

[29] Yuan, Z. (2020). ALTRA : Cross-Project Software Defect Prediction via Active Learning and Tradaboost. 8, 30037–30049.

[30] Ren, J., & Liu, F. (2020). Applied sciences A Novel Approach for Software Defect prediction Based on the Power Law Function.

[31] Iqbal, A., & Aftab, S. (2020). A Classification Framework for Software Defect Prediction Using Multi-filter Feature Selection Technique and MLP. February, 18–25. https://doi.org/10.5815/ijmecs.2020.01.03

[32] Education, I. J. M., Science, C., & Khan, M. Z. (2020). Hybrid Ensemble Learning Technique for Software Defect Prediction. February, 1–10. https://doi.org/10.5815/ijmecs.2020.01.01

[33] Xe, L. W., Dwd, E., & Hihfw, R. I. R. U. (2019). $ *lw+xe edvhg ’dwd &roohfwlrq 0hwkrg iru 6riwzduh ’hihfw 3uhglfwlrq. 100–108. https://doi.org/10.1109/DSA.2019.00020

[34] Jadhav, R. B., Joshi, S. D., Thorat, U. G., & Joshi, A. S. (2020). Software Defect Prediction Utilizing Deterministic and Probabilistic Approach for Optimizing Performance through Defect Association Learning. 8(6), 6–11.

[35] Cao, H. (2020). A Systematic Study for Learning-Based Software Defect Prediction A Systematic Study for Learning-Based Software Defect Prediction. https://doi.org/10.1088/1742-6596/1487/1/012017

[36] Ronchieri, E., Canaparo, I. M., Belgiovine, M., & Salomoni, D. (2019). Software Defect Prediction on Unlabelled Dataset with Machine Learning Techniques. 2019–2020.

[37] Amasaki, S. (2020). Cross-version defect prediction : use historical data, cross-project data, or both ? Cross-Version Defect Prediction : Use Historical Data, Cross-Project Data, or the Both ? February. https://doi.org/10.1007/s10664-019-09777-8

[38] Afric, P., Sikic, L., Kurdija, A. S., & Silic, M. (2020). REPD : Source Code Defect Prediction as Anomaly Detection. May. https://doi.org/10.1016/j.jss.2020.110641

[39] Yucalar, F., Ozcift, A., Borandag, E., & Kilinc, D. (2020). Engineering Science and Technology, an International Journal Multiple-classifiers in software quality engineering : Combining predictors to improve software fault prediction ability. Engineering Science and Technology, an International Journal, 23(4), 938–950.

https://doi.org/10.1016/j.jestch.2019.10.005
[40] Ahluwalia, A., Di, M., & Falessi, D. (n.d.). On the Need of Removing Last Releases of Data When Using or Validating Defect Prediction Models.
[41] Florence, R. J. L. (2019). Software defect prediction techniques using metrics based on neural network classifier. Cluster Computing, 22(s1), 77–88. https://doi.org/10.1007/s10586-018-1730-1
[42] Malhotra, R., & Kamal, S. (2019). Neurocomputing An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data. Neurocomputing, 343, 120–140. https://doi.org/10.1016/j.neucom.2018.04.090
[43] Ni, C., Chen, X., Wu, F., Shen, Y., & Gu, Q. (2019). The Journal of Systems and Software An empirical study on pareto based multi-objective feature selection for software defect prediction. The Journal of Systems & Software, 152, 215–238. https://doi.org/10.1016/j.jss.2019.03.012
[44] Wang, K., Liu, L., Yuan, C., & Wang, Z. (2020). Software defect prediction model based on LASSO – SVM. Neural Computing and Applications, 1. https://doi.org/10.1007/s00521-020-04960-1
[45] Meilong, S., He, P., Xiao, H., Li, H., & Zeng, C. (2020). An Approach to Semantic and Structural Features Learning for Software Defect Prediction. 2020.
[46] Sohan, F., Kabir, A., Rahman, M., Mahmud, S. M. H., & Bhuiyan, T. (2020). Training Data Selection Using Ensemble Dataset Approach for SDPSoftware Defect Prediction Dataset Approach for Software Defect (Issue July). Springer International Publishing. https://doi.org/10.1007/978-3-030-52856-0
[47] Nalini, C., & Krishna, T. M. (2020). An Efficient Software Defect Prediction Model Using Neuro Evalution Algorithm based on Genetic Algorithm. 102, 135–138.
[48] Tabassum, S., Minku, L. L., Feng, D., & Cabral, G. G. (n.d.). An Investigation of Cross-Project Learning in Online Just-In-Time Software Defect Prediction.
[49] Jureczko, M., Nguyen, N. T., Szymczyk, M., & Unold, O. (2019). Towards implementing defect prediction in the software development process. 1, 1–5.
[50] Software Engineering Approach to Bug Prediction Models Using Machine Learning as a Service (MLaaS). (2019). July 2018.
[51] The Cost of Poor Quality Software in the US: A 2018 Report (2018, September 26).
Retrieved from
https://www.it-cisq.org/the-cost-of-poor-quality-software-in-the-us-a-2018-report/The-Cost-of-Poor-Quality-Software-in-the-US-2018-Report.pdf
[52] Software fail watch 5th edition. (2018,).
Retrieved from
https://www.tricentis.com/resources/software-fail-watch-5th-edition/
[53] Cambridge University study states software bugs cost economy $312 billion per year (2013, January 08).
Retrieved from
Financial Content: Cambridge University study states software bugs cost economy $312 billion per year.

[54] Otlet, P. (1934), Traité De Documentation: Le Livre Sur Le Livre, Théorie Et Pratique, Editiones Mundaneum: Mons, Belgium

[55] VOSviewer download website - https://www.vosviewer.com/download