A Systematic Review of Feature Selection Techniques in Software Quality Prediction

Hadeel Alsolai[1,2], Marc Roper[2]
[1]*Computer Science and Information system*
Princess Nourah Bint Abdulrahman University
Riyadh, Saudi Arabia
[2]*Computer and Information Sciences*
University of Strathclyde
Glasgow, United Kingdom
Hadeel.alsolai@strath.ac.uk, Marc.roper@strath.ac.uk

Abstract— Background: Feature selection techniques are important factors for improving machine learning models because they increase prediction accuracy and decrease the time to create a model. Recently, feature selection techniques have been employed on software quality prediction problems with different results and no clear indication of which techniques are frequently used. Objective: This study aims to conduct a systematic review of the application of feature selection techniques in software quality prediction and answers eight research questions. Method: The review evaluates 15 papers in 9 journals and 6 conference proceedings from 2007 to 2017 using the standard systematic literature review method. Results: The results obtained from this study reveal that the filter feature selection method was the most commonly used in the studies (60%) and RELIEF was the most employed among this method, and a limited number of studies employed an ensemble method. Several studies used public datasets available in the PROMISE software project repository (60%). Most studies focused on software defect prediction (classification problem) using area under curve (AUC) as a primary evaluation measure, whereas only two studies focused on software maintainability prediction (regression problem) using mean magnitude of relative error (MMRE) as a primary evaluation measure. All selected studies performed k-fold cross-validation to evaluate model accuracy. Individual prediction models were mostly employed and ensemble models appeared only in three studies. Naive Bayes was the most investigated among individual models, whereas Random forest was the most investigated among ensemble models. Conclusion: Feature selection techniques used by selected primary studies have a positive impact on the performance of the prediction models. Further, both ensemble feature selection method and ensemble models have the ability for increasing prediction accuracy over single methods or individual models and have reported improvement in the prediction accuracy; however, the application of these techniques in software quality prediction is still limited.

Keywords— Systematic literature review, feature selection, software defect, software maintainability, prediction.

I. INTRODUCTION

Feature selection techniques have received increased attention in recent years owing to their ability to improve prediction accuracy and decrease the time to create the model. A number of studies have used various types of feature selection techniques for software quality prediction [1-15], and several literature review studies of feature selection techniques have investigated in the past [16, 17] in various domains, such as bioinformatics [18] and microarray data [19], but to the best of our knowledge there are no reviews investigating the application of feature selection techniques in software quality prediction. Therefore, this study contributes to both software engineering and machine learning area, since it explores the application of various types of feature selection techniques, along with different types of prediction models applied on software quality datasets.

Feature selection is considered one of the essential steps in data pre-processing [20] and plays a vital role to resolve the problem of high dimensionality dataset. This problem includes irrelevant variables, where one or more features (independent variables) have no impact on the target features (dependent variables), and redundant variables, where an independent variable is highly correlated with another and can be removed [21]. Feature selection methods can be classified into four main categories: the *filter method* which identifies the features without building machine learning models using heuristically determined relevant knowledge [22], the *wrapper method* which integrates features into a prediction model to select relevant features [23], the *embedded method* which takes advantage of both filter and wrapper methods, and selects the best subset during the creation of the model, and the *ensemble method* which combines the output of several feature selection methods. The ensemble method is based on the combination rules that include either the best subsets of attributes selection or the high ranking attributes [24]. Each method involves several types of techniques, and some of these techniques are described in Table II.

Software quality prediction is a critical factor for supporting decision making, reducing errors and utilising resources [25, 26]. The purpose of this study is to review the application of feature selection techniques in software quality prediction. We proposed eight research questions (RQs) to determine and analyse feature selection, datasets, software quality prediction types, evaluation measurements, and prediction models. The answers were extracted from 15 papers in 9 journals and 6 conference proceedings from 2007 to 20017 using the research method for conducting systematic reviews given by Kitchenham [27].

The remainder of this paper is organized as follows: Section II presents research method. Section III explains the research results. Section IV describes the research

978-1-7281-5532-6/19/$31.00 ©2019 IEEE

discussion. Section V concludes this paper with a summary and direction for future work.

II. Research Method

This section presents the research method in terms of the research questions, the research process, study selection and quality assessment. The research method used the guidelines and instructions of the systematic review proposed by Kitchenham [27]. This method is a well-known method and widely used in the software engineering field and aims to select, evaluate and analyse all relevant studies of the particular topic [27]. The steps of conducting the systematic literature review method are reported below.

A. Research questions (RQs)

This systematic literature review was performed to answer the following RQs which are designed to understand the range of feature selection techniques being employed and also gain an understanding of their effectiveness and applicability:

RQ1) What type of feature selection techniques have been proposed to predict software quality?

RQ2) What is the impact of the feature selection techniques on prediction accuracy?

RQ3) What are the specific techniques of feature selection used in software quality prediction?

RQ4) To what software quality prediction datasets have feature selection techniques been applied?

RQ5) With what type of software quality prediction problem (e.g. defect, maintainability) have feature selection techniques been used?

RQ6) What are the evaluation measures used to assess the capability of software quality prediction?

RQ7) What approach (e.g. hold-out, cross-validation) is used to assess the performance of software quality prediction models built on top of feature selection?

RQ8) What are the prediction models (i.e. individual, ensemble) used to predict software quality?

B. Research process

The search process was a manual search of the most common and widely used online digital libraries that publish peer-reviewed articles:

- IEEE eXplore (ieeexplore.ieee.org)
- Google scholar (scholar.google.com)
- Elsevier (sciencedirect.com)
- Springer (springerlink.com)
- Wiley online library (onlinelibrary.wiley.com)

Two types of publication were selected: either journal or conference. We applied the following research string to collect relevant studies of the application of feature selection techniques in software quality prediction:

> *(software maintainability OR maintainability index OR software defect OR software fault OR software quality) AND (feature selection OR selected features OR attribute selection OR data reduction) OR (estimate* OR predict* OR evaluate* OR forecast*)*

We performed the search without any restriction on the year. However, we found the papers that meet our criteria fell in the range 2007 to 2017. Therefore, the search was selected from 2007 to 2017. According to the best of our knowledge, there is no study outside our time frame.

C. Study selection

We applied the inclusion and exclusion criteria to identify relevant studies. We include peer-reviewed studies published between 2007 and 2017 and were written in the English language. Moreover, the search was limited to studies using feature selection techniques in software quality prediction (i.e. maintainability or defect). Therefore, we excluded any studies outside our defined time frame, research types and research topics or studies were not published peer-reviewed or written in English.

D. Quality assessment

We evaluated and selected relevant studies according to the following quality assessment questions:

QA1) Does the study use feature selection techniques?

QA2) Does the study aim to predict software quality?

QA3) Does the study use a public dataset for software quality prediction?

QA4) Does the study use suitable evaluation measures for both the machine learning problems (i.e. classification, regression and clustering) and software engineering?

QA5) Does the study use well-known validation techniques (e.g. n-fold cross validation or holdout)?

QA6) Does the study involve the construction of prediction models?

III. Results

A. Search results

Table I lists the details of the selected studies. 15 selected primary studies were chosen in this systematic literature review, which included 9 journals and 6 conference proceedings from 2007 to 20017. Most of the selected studies were downloaded from IEEE online digital library (5 papers), followed by Elsevier, Springer and Google scolar (3 papers in each library), whereas only one paper was downloaded from Wiley online library.

According to distribution of published papers over the years, three primary selected studies were published in 2009 and 2014, respectively, whereas two primary selected studies were published in each the following years: 2007, 2010 and 2017 (see Fig. 1).

TABLE I. SELECTED PRIMARY STUDIES.

Study ID	Ref	Year	Place published	Type
S1	[1]	2007	IEEE	Conference
S2	[2]	2007	IEEE	Conference
S3	[3]	2009	Elsevier	Journal
S4	[4]	2009	IEEE	Conference
S5	[5]	2009	IEEE	Conference
S6	[6]	2010	Google scholar	Journal
S7	[7]	2010	IEEE	Conference
S8	[8]	2011	Wiley online library	Journal
S9	[9]	2012	Elsevier	Journal
S10	[10]	2014	Google scholar	Conference
S11	[11]	2014	Google scholar	Journal
S12	[12]	2014	Springer	Journal
S13	[13]	2015	Elsevier	Journal
S14	[14]	2017	Springer	Journal
S15	[15]	2017	Springer	Journal

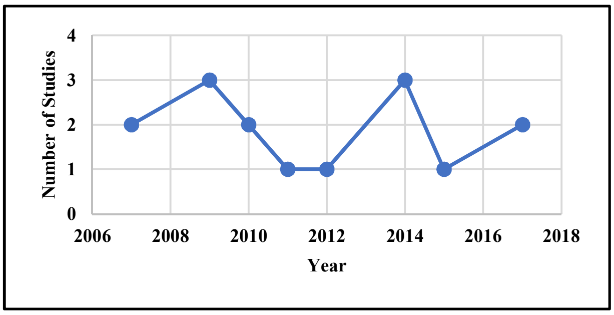

Fig. 1. Number of selected studies over the years.

B. *Quality assessment results*

All the selected primary studies accept the inclusion and exclusion criteria and answer (YES) for all quality assessment questions. However, S15 used partial datasets (the datasets are not available, but were extracted from open source software systems). To the best of our knowledge, the selected primary studies are the only studies that meet these conditions.

IV. Discussion

Fig. 2 presents the distribution of the feature selection types and answers the **RQ1**. It clearly observed that most of the selected primary studies used the filter method (60%), whereas the wrapper and ensemble method were employed by a limited number of studies (20%). The ensemble method has been shown to be a very effective method and has outperformed other feature selection methods in three studies (S8, S9 and S15). S8 used a feature ranking to decrease the search space, then a feature subset selection to search for subsets of attributes. The results obtained from S8 reported that ensemble method achieved the best prediction accuracy among other feature selection methods used in this study. S9 integrated various feature selection methods and investigated 17 different ensemble types with a different number of combinations (2,3,4,6,8 and 10 feature selection methods). However, the results of S9 indicate that an ensemble of fewer combinations (e.g. two types of feature selection) outperformed combinations of several techniques. S15 selected the best subsets of attributes (i.e. dominant metrics) using seven feature selection methods that applied on 26 datasets. The results of S15 revealed that the prediction accuracy of these dominant metrics performed the best compared to other feature selection methods used in this study. To answer **RQ2**, all the feature selection techniques used by selected primary studies improved the prediction accuracy over using all features (i.e. without applying feature selection) and the ensemble methods produced a better prediction accuracy compared to single feature selection.

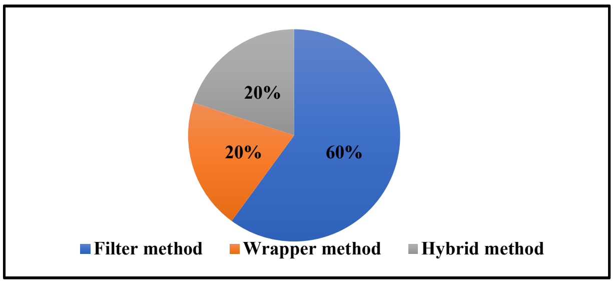

Fig. 2. The distribution of feature selection types.

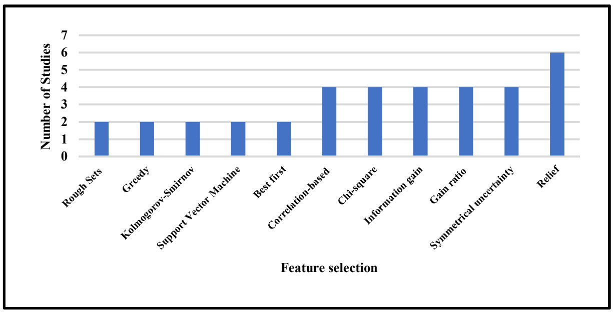

Fig. 3. The distribution of feature selection techniques.

Fig. 3 above illustrates the most frequent feature selection technique used by selected primary studies and this figure answers **RQ3.** RELIEF is the most feature selection technique that used by six studies. Table II lists a brief description of the feature selection techniques proposed in Fig. 3. It is worth mentioning that all these techniques are classified under filter method except support vector machine, which is wrapper method.

TABLE II. DESCRIPTION OF THE FEATURE SELECTION TECHNIQUES.

Name	ID	Description
Relief	S6 S7 S8 S9 S10 S12	Relief evaluates each attribute and computes their score by iteratively sampling. The score in Relief algorithm depends on the difference between attribute values of nearest-neighbour pairs.
Symmetrical uncertainty	S7 S8 S9 S10	Symmetrical uncertainty selects attributes by computing the symmetrical uncertainty that deals with the problem of information gain bias. This problem takes attributes with more values using the normalization that ranges from 0 to 1.
Gain ratio	S7 S8 S9 S10	Gain ratio selects attributes using a decision tree that takes the number and size of branches. The gain ratio is considered an updated version of the information gain that decreases its bias with respect to the intrinsic information during splitting.
Information gain	S7 S8 S9 S10	Information gain computes how much the amount of information generated by feature items. It uses the same concept of gain ratio without considering the intrinsic information.
Chi-square	S7 S8 S9 S10	Chi-square is computed between each attribute and the dependent variable (i.e. target) and is used only for classification problem (i.e. categorical feature). The Chi-square identifies whether the relationship between these categorical features of the sample would reflect their real relationship in the population.
Correlation-based	S1 S3 S12 S13	Correlation-based feature selection takes the whole attributes, then it sorts these attributes according to their degree of correlation to the dependent variable (i.e. target).
Best first	S12 S15	Best first begins with the empty group of attributes and searches forward or begins with the full group of attributes and searches backwards.
Support vector machine	S5 S6	Support vector machine model can be used in the wrapper method to perform feature selection.
Kolmogorov-Smirnov	S4 S8	Kolmogorov-Smirnov selects attributes according to Kolmogorov-Smirnov score statistic. It calculates the maximum variation between the empirical distribution function of the attributes in each class, where the greater the distance between the distribution functions indicates the better the attribute identifies between two classes.
Greedy	S13 S15	Greedy can be employed either forward or backward search across attributes. It searches continually until removing or inserting attributes affect negatively in the prediction accuracy.
Rough sets	S4 S14	Rough Sets is described as a formal approximation of a conventional set. It uses a theory, which is depended on the classical set.

Fig. 4 shows the most frequent datasets used by selected primary studies and this figure answers **RQ4**. The results indicate that most of the datasets were gathered from the PROMISE software project repository (60%), that contains several public datasets which are appropriate for software quality prediction [28]. CM1, JM1, KC1, KC2, and PC1 are the most frequently used together as a group of datasets appearing in the selected primary studies (i.e. S1, S2, S3, S6, S7, S9, S11, S12 and S13), which are located in PROMISE repository and extracted from Java systems. These datasets are extracted from NASA projects for defect prediction with including two classes (FALSE if the system does not report defects or TRUE if the system report defects). Further, these datasets include 22 attributes and were reduced to 7 attributes in S1 and S2, and in the range from 4 to 13 attributes in the remaining studies using different feature selection methods. Other selected primary studies used a group of the datasets from PROMISE repository (i.e. S4, S5, S8 and S10). These datasets considered the second group of the datasets that most frequently used and were collected from telecommunications software system for defect prediction. This system contains over 10 million lines of code and the datasets has 42 attributes that were reduced to 6 or 4 attributes in all studies after applying feature selection. Finally, the other datasets used in the prediction of software maintainability, one of them is publicly available as a table appendix published in [29] (i.e. S14), and the second one is a private dataset collected from real-world Java systems (i.e. S15).

Fig. 5 presents the distribution of the types of software quality prediction and answers **RQ5**. These types include 87% software defect and 13% software maintainability (i.e. S14 and S15). This figure provides evidence of the restricted feature selection used by software maintainability studies.

With respect to **RQ6**, all studies of software defect prediction used Area under carve (AUC) as a primary evaluation measure for the classification problem, whereas software maintainability prediction used mean magnitude of relative error (MMRE) as a primary evaluation measure for the regression problem. Regarding **RQ7**, all the studies regardless of their types (i.e. classification or regression problems) used ten- fold cross validation to evaluate the prediction models.

Finally, the most frequently used prediction models (either individual or ensemble) that have been used in selected primary studies are presented in Fig.6 and this figure answers **RQ8**. Seven models are used in total, six of them are individual models, and one is an ensemble model (i.e. Random Forest). Naive Bayes is the most common individual models used by eleven studies (i.e. S1, S3, S4, S5, S6, S8, S9, S10, S11, S12 and S13), followed by Support Vector Machine and then K Nearest Neighbours. Moreover, all the selected primary studies used individual models, whereas three of them employed ensemble models as well (i.e. S3, S6 and S13). The results obtained from this analysis reveals that the application of ensemble models in software quality prediction with feature selection was limited compared with individual models. These studies concluded that the ensemble models yield improved prediction accuracy over most of the investigated models and provided better results than individual models.

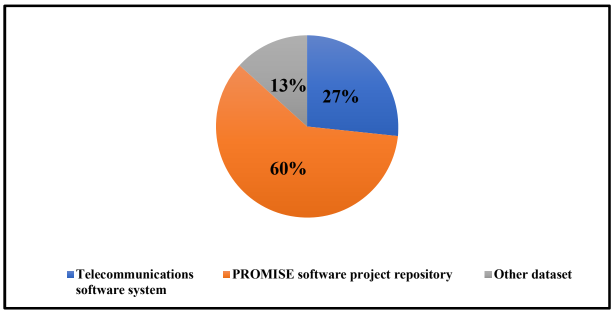

Fig. 4. The distribution of datasets.

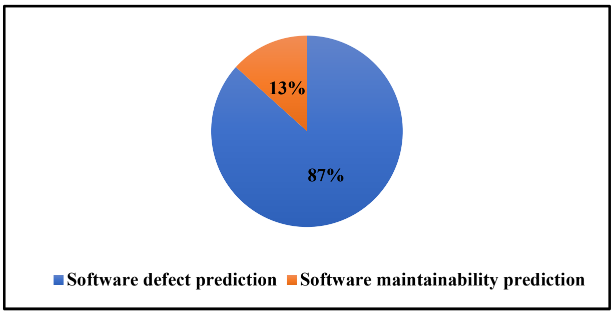

Fig. 5. The distribution of software quality prediction types.

V. CONCLUSION

This study has conducted a systematic literature review to investigate the application of feature selection techniques in software quality prediction. The research process was conducted on five online digital libraries to choose peer-reviewed papers published in either conferences or journals. Fifteen studies have been chosen between 2007 and 2017, and eight research questions have been addressed and analysed. The selected primary studies have reported the effectiveness of feature selection in improving prediction accuracy of software quality. However, these studies have used a wide variety of filter methods and a limited number of ensemble (hybrid) methods. Moreover, these studies have performed several individual models and a few ensemble models, whereas the ensemble models outperform individual models.

The results concluded that among feature selection, the filter selection method (60%), in particular RELIEF, has been used by most of the selected studies (6 studies). All the selected primary studies confirmed that feature selection techniques have the ability to improve the performance of

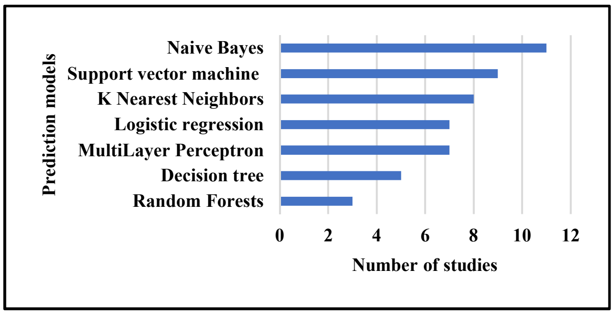

Fig. 6. Most frequently prediction models used.

machine learning models. Additionally, most of the selected studies used public datasets hosted in the PROMISE software project repository (60%). 87% of software quality prediction types were carried out for software defect prediction, compared to 13% (2 studies) that were carried out by software maintainability prediction. AUC was the main evaluation measure for the classification problem, whereas MMRE was the main evaluation measure for the regression problems. Further, all selected studies performed k-fold cross-validation. Individual prediction models were created by all studies, whereas ensemble models were created only in three studies. Among prediction models, Naive Bayes was the most individual model used by selected studies, whereas Random forest was the most ensemble model used by them.

In future investigations, it might be possible to analyse the improvements in accuracy obtained using feature selection techniques in the selected primary studies. Additionally, this study seeks to answer eight RQs, which will help to determine the research gaps. Therefore, we will employ feature selection techniques with respect to ensemble methods on software maintainability datasets using different machine learning models that include ensemble models.

ACKNOWLEDGMENTS

We gratefully acknowledge Princess Nourah bint Abdulrahman University and the University of Strathclyde to achieve this research and provide the different facilities in carrying out this research.

REFERENCES

[1] D. Rodríguez, R. Ruiz, J. Cuadrado-Gallego, and J. Aguilar-Ruiz, "Detecting fault modules applying feature selection to classifiers," in *2007 IEEE International Conference on Information Reuse and Integration*, 2007, pp. 667-672: IEEE.

[2] D. Rodriguez, R. Ruiz, J. Cuadrado-Gallego, J. Aguilar-Ruiz, and M. Garre, "Attribute selection in software engineering datasets for detecting fault modules," in *33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO 2007)*, 2007, pp. 418-423: IEEE.

[3] C. Catal and B. Diri, "Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem," *Information Sciences,* vol. 179, no. 8, pp. 1040-1058, 2009.

[4] K. Gao, T. M. Khoshgoftaar, and H. Wang, "An empirical investigation of filter attribute selection techniques for software quality classification," in *2009 IEEE International Conference on Information Reuse & Integration*, 2009, pp. 272-277: IEEE.

[5] T. M. Khoshgoftaar and K. Gao, "Feature selection with imbalanced data for software defect prediction," in *2009 International Conference on Machine Learning and Applications*, 2009, pp. 235-240: IEEE.

[6] N. Gayatri, S. Nickolas, A. Reddy, S. Reddy, and A. Nickolas, "Feature selection using decision tree induction in class level metrics dataset for software defect predictions," in *Proceedings of the world congress on engineering and computer science*, 2010, vol. 1, pp. 124-129.

[7] T. M. Khoshgoftaar, K. Gao, and N. Seliya, "Attribute selection and imbalanced data: Problems in software defect prediction," in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, 2010, vol. 1, pp. 137-144: IEEE.

[8] K. Gao, T. M. Khoshgoftaar, H. Wang, and N. Seliya, "Choosing software metrics for defect prediction: an investigation on feature selection techniques," *Software: Practice and Experience,* vol. 41, no. 5, pp. 579-606, 2011.

[9] H. Wang, T. M. Khoshgoftaar, and A. Napolitano, "Software measurement data reduction using ensemble techniques," *Neurocomputing,* vol. 92, pp. 124-132, 2012.

[10] K. Gao, T. M. Khoshgoftaar, and R. Wald, "Combining feature selection and ensemble learning for software quality estimation," in *The Twenty-Seventh International Flairs Conference*, 2014.

[11] S. Agarwal and D. Tomar, "A feature selection based model for software defect prediction," *assessment,* vol. 65, 2014.

[12] A. Okutan and O. T. Yıldız, "Software defect prediction using Bayesian networks," *Empirical Software Engineering,* vol. 19, no. 1, pp. 154-181, 2014.

[13] I. H. Laradji, M. Alshayeb, and L. Ghouti, "Software defect prediction using ensemble learning on selected features," *Information and Software Technology,* vol. 58, pp. 388-402, 2015.

[14] L. Kumar and S. K. Rath, "Software maintainability prediction using hybrid neural network and fuzzy logic approach with parallel computing concept," *International Journal of System Assurance Engineering and Management,* journal article vol. 8, no. 2, pp. 1487-1502, 2017.

[15] B. R. Reddy and A. Ojha, "Performance of Maintainability Index prediction models: a feature selection based study," *Evolving Systems,* pp. 1-26.

[16] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence,* vol. 97, no. 1, pp. 245-271, 1997/12/01/ 1997.

[17] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence,* vol. 97, no. 1, pp. 273-324, 1997/12/01/ 1997.

[18] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics,* vol. 23, no. 19, pp. 2507-2517, 2007.

[19] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Advances in bioinformatics,* vol. 2015, 2015.

[20] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[21] M. Karagiannopoulos, D. Anyfantis, S. Kotsiantis, and P. Pintelas, "Feature selection for regression problems," *Proceedings of the 8th Hellenic European Research on Computer Mathematics & its Applications, Athens, Greece,* vol. 2022, 2007.

[22] C. H. Ooi, M. Chetty, and S. W. Teng, "Differential prioritization in feature selection and classifier aggregation for multiclass microarray datasets," *Data Mining and Knowledge Discovery,* journal article vol. 14, no. 3, pp. 329-366, June 01 2007.

[23] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Machine Learning Proceedings 1994*: Elsevier, 1994, pp. 121-129.

[24] V. Bolón-Canedo and A. Alonso-Betanzos, "Ensembles for feature selection: A review and future trends," *Information Fusion,* vol. 52, pp. 1-12, 2019.

[25] A. De Lucia, E. Pompella, and S. Stefanucci, "Assessing effort estimation models for corrective maintenance through empirical studies," *Information and Software Technology,* vol. 47, no. 1, pp. 3-15, 2005.

[26] S. Shafi, S. M. Hassan, A. Arshaq, M. J. Khan, and S. Shamail, "Software quality prediction techniques: A comparative analysis," in *2008 4th International Conference on Emerging Technologies*, 2008, pp. 242-246.

[27] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University,* vol. 33, no. 2004, pp. 1-26, 2004.

[28] B. Cukic, "Guest Editor's Introduction: The Promise of Public Software Engineering Data Repositories," *IEEE Software,* vol. 22, no. 6, pp. 20-22, 2005.

[29] W. Li and S. Henry, "Object-oriented metrics that predict maintainability," *The Journal of Systems & Software,* vol. 23, no. 2, pp. 111-122, 1993.