JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 9 Number 1 | June 2024: 52-60
DOI: 10.15575/join.v9i1.1307

Improving with Hybrid Feature Selection in Software Defect Prediction

Muhammad Yoga Adha Pratama[1], Rudy Herteno[2], Mohammad Reza Faisal[3] , Radityo Adi Nugroho[4] , Friska Abadi[5]
[1,2,3,4,5]Department of Computer Science, University of Lambung Mangkurat, Kalimantan Selatan, Indonesia

Article Info

Article history:

Received February 21, 2024
Revised April 05, 2024
Accepted April 11, 2024
Available Online April 23, 2024

Keywords:

Particle Swarm Optimization
Feature Selection
Software Defect Prediction
Filter
Wrapper
Naïve Bayes

ABSTRACT

Software defect prediction (SDP) is used to identify defects in software modules that can be a challenge in software development. This research focuses on the problems that occur in Particle Swarm Optimization (PSO), such as the problem of noisy attributes, high-dimensional data, and premature convergence. So this research focuses on improving PSO performance by using feature selection methods with hybrid techniques to overcome these problems. The feature selection techniques used are Filter and Wrapper. The methods used are Chi-Square (CS), Correlation-Based Feature Selection (CFS), and Forward Selection (FS) because feature selection methods have been proven to overcome data dimensionality problems and eliminate noisy attributes. Feature selection is often used by some researchers to overcome these problems, because these methods have an important function in the process of reducing data dimensions and eliminating uncorrelated attributes that can cause noisy. Naive Bayes algorithm is used to support the process of determining the most optimal class. Performance evaluation will use AUC with an alpha value of 0.050. This hybrid feature selection technique brings significant improvement to PSO performance with a much lower AUC value of 0.00342. Comparison of the significance of AUC with other combinations shows the value of FS PSO of 0.02535, CFS FS PSO of 0.00180, and CS FS PSO of 0.01186. The method in this study contributes to improving PSO in the SDP domain by significantly increasing the AUC value. Therefore, this study highlights the potential of feature selection with hybrid techniques to improve PSO performance in SDP.

Corresponding Author:

Rudy Herteno,
Department of Computer Science,
University of Lambung Mangkurat,
Jl. Ahmad Yani KM 36, Banjarbaru, Kalimantan Selatan 70714
Email: rudy.herteno@ulm.ac.id

1. INTRODUCTION

Software defect prediction (SDP) is an inconvenient challenge because it has a negative effect on future software development. In SDP, it is one of the important aspects and needs more attention to ensure software quality and reliability [1]. Software that is detected to have defects will produce unexpected results during deployment [2]. The use of datasets in SDP often encounters cases such as high-dimensional data, noisy attributes, and imbalanced classes [3]. Particle Swarm Optimization (PSO) is used to solve optimization problems due to its simplicity, fast convergence, effectiveness, and excellent generalization ability. PSO is one of the most widely recognized Evolutionary Computation (EC) algorithms [4]. PSO adopts the social behavior of birds which is algorithmically translated to solve optimization problems where it is described as a flock of birds as a swarm of particles and each particle represents a candidate solution to find the best solution in a certain dimensional space [5]. PSO also has weaknesses that are very influential in the implementation process. In the case of PSO, high-dimensional

datasets also have the effect of causing convergence to be premature, making it less effective in handling noisy attributes [6] [7]. PSO often requires more iterations and produces much more complex models when applied to high-dimensional datasets [8]. In response to these challenges and in pursuit of enhancing the efficacy of PSO, a variety of methods have been introduced. These methods aim to either modify the PSO algorithm itself or integrate it with other metaheuristics. By capitalizing on the strengths of diverse algorithms, these approaches seek to enhance the overall performance of PSO [9].

This research is based on several research references that relate to the research to be carried out. This research will use feature selection from filter techniques such as Chi-Square and Correlation-Based Feature Selection. As well as selection from wrapper techniques such as Forward Selection. Then it will use machine learning from the Naïve Bayes classification algorithm. As in the research conducted by Iqbal in 2020. This research method involves such as multi filter and multi layer filter techniques. This research focuses on the realm of software defect prediction where the Area Under the Curve (AUC) result obtained is 0.817. In research conducted by Chakraborty in 2020, using a hybrid method called Hellinger net. This hybrid method combines several tree-based methods, such as Decision Tree and Random Forest which are integrated with the ensemble method in it. The result given in the average AUC value is 0.760. In research conducted by Harzevili in 2021, conducted research using MLMNB combined with statistical hypothesis testing. This method is one type of extension of naive Bayes classifier for research in SDP. The AUC result obtained is 0.690. In research conducted by Balogun in 2020, using multiple filters such as several machine learning (Decision Tree and Naïve Bayes), Chi-Square, ReliefF, Information Gain, and Rank Aggregation-Based Multi-Filter. The example taken is when using Naïve Bayes machine learning. The AUC result obtained is 0.746. Then the last research from Ding in 2020, using the Pruned Histogram-based isolation forest method to improve SDP performance. The AUC result obtained is 0.792. So the basis of the previous research above will be the basis of this research, namely by combining several feature selection techniques to produce a much higher AUC value.

In the process of overcoming the problem of high dimensionality and noisy attributes, one can use feature selection such as Chi-Square (CS). The CS method is one of the methods of the filter technique. CS has a contribution in helping to remove redundant and irrelevant attributes and select the most distinct features to minimize the data resulting in higher classification accuracy [10]. In addition, there are other filter techniques that can help overcome these problems such as Correlation-Based Feature Selection (CFS). This technique helps reduce the dimensionality of the data by identifying attributes that have a high correlation and also removing attributes that have no correlation with the target. This method is effective for reducing data dimensionality and noisy attributes so that prediction accuracy can be improved [11]. Theoretically, the research shows that applying filter methods in reducing dimensionality and removing noisy attributes by a number of classifiers has the potential to improve results in their prediction models [12]. After CS and CFS as filter techniques in this research, another wrapper technique of feature selection called Forward Selection (FS) is added. FS is a feature selection technique that begins with an empty set of features and progressively adds unused features. During the initial iteration, each feature is evaluated individually. Subsequently, in each iteration, one additional feature is added to the feature subset from the previous iteration, and the newly formed feature subset is re-evaluated. To minimize the number of evaluations needed, only the best feature subsets are retained after each iteration [13]. The process will be supported by the Naive Bayes (NB) classification machine learning algorithm. NB will work on the "naive" assumption that the independent effect of an attribute value on a given class is independent of other attribute values [14]. In NB, each attribute is treated equally, although in real-world applications, attributes can have different roles in discriminating classes [15].

This research focuses on solving PSO problems in SDP such as high-dimensional data and noisy attributes. The proposed method is feature selection with hybrid techniques. The techniques used are Filter and Wrapper, where the methods to be used in the Filter are CFS and CS, then from the Wrapper used is FS. By utilizing both feature selection techniques, a solution is created that can solve the problem of high-dimensional data and eliminate noisy attributes. In addition, this process also involves the classification machine learning algorithm of Naive Bayes to classify the classes that are considered the most optimal in order to get better results. We believe this combination of research methods can overcome the problem and improve the quality of AUC value in predicting software defects..

2. METHOD

The proposed research refers to a methodology of combining two techniques in feature selection. The techniques used are Filter and Wrapper. The filters used are CFS and CS, while the

p-ISSN: 2528-1682
e-ISSN: 2527-9165

wrapper used is FS. Then this hybrid technique will be integrated with the classification algorithm of Naive Bayes to find the most optimal class.

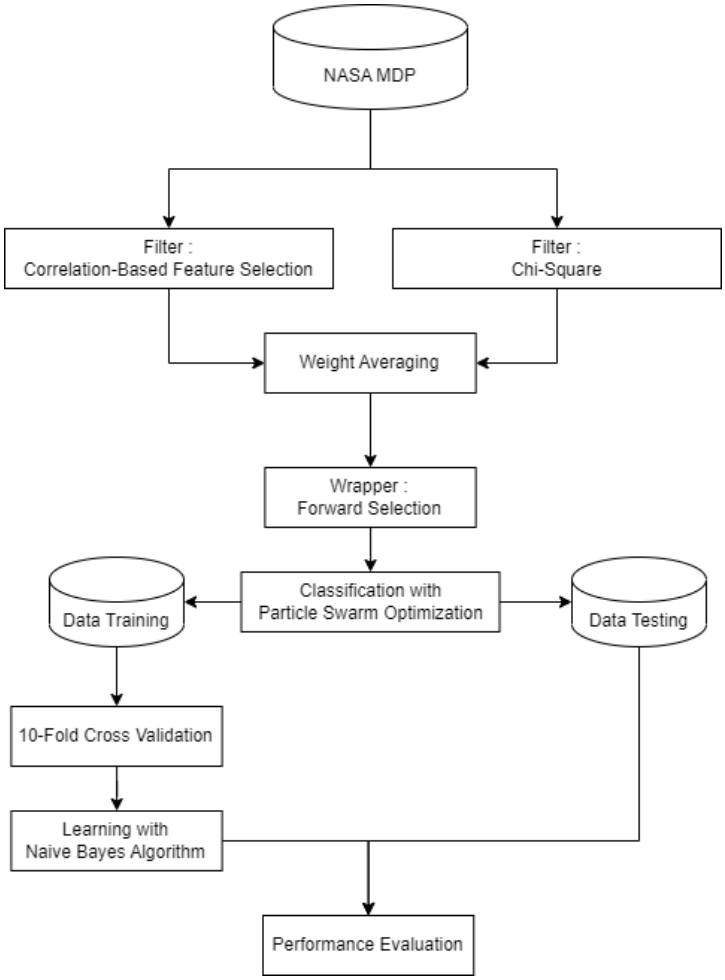

Figure 1. Research Flowchart

2.1 Data Collection

These experiments were conducted utilizing a set of 12 datasets sourced from NASA MDP D", meticulously chosen to represent diverse scenarios and complexities. Among other datasets, the NASA MDP dataset stands out, having been utilized in numerous research studies. However, it has also faced criticism for potentially containing erroneous data [16]. The data used includes CM1, JM1, KC1, KC3, MC1, MC2, MW1, PC1, PC2, PC3, PC4, PC5. This dataset can be downloaded from the following link https://github.com/klainfo/NASADefectDataset. This dataset originates from software projects in various domains and programming languages such as C, C++, and Java. However, the NASA MDP dataset is known to contain class imbalance [17], noisy attributes [18], and high-dimensional data [19]. Table 1 is presented, containing specifications about each dataset used.

Table 1. Nasa MDP Dataset Spesifications

Datasets	Attributes	Instances	Defects	Non-Defects
CM1	38	327	42	285
JM1	22	7782	1672	6110
KC1	22	1183	314	869
KC3	40	194	36	158
MC1	39	1988	46	1942
MC2	40	125	44	81

MW1	38	253	27	226
PC1	38	705	61	644
PC2	37	745	16	729
PC3	38	1077	134	943
PC4	38	1287	177	1110
PC5	39	1711	471	1240

2.2 Correlation-Based Feature Selection

Correlation-Based Feature Selection (CFS) is a form of feature selection that belongs to filter techniques. It works by selecting relevant features for the classification process in order to improve the quality of the classification model itself [20]. CFS uses correlation measures to evaluate the quality of a feature subset with the hypothesis that the optimal feature subset is one that contains features that are highly correlated with the class, but not correlated with each other. The form of correlation measure used is Pearson correlation where all variables have been normalized [15]. The selection of input attributes with the most significant impact on the target variable is determined by evaluating the association between the features and the target variable [21] [22] [23]. The coefficient size ranges from -1 to 1, which indicates a positive and negative form or no correlation between the feature and the target variable. This is a basic formula for calculating the Pearson Coefficient (1) [24]:

$$r = \frac{\sum(x_i - x)(y_i - y)}{\sqrt{\sum(x_i - x)^2 \ \sum(y_i - y)^2}} \quad (1)$$

2.3 Chi-Square

Chi-Square (CS) realizes a feature selection with filter technique that provides competitive results compared to other methods. In CS, each feature of the dataset is determined by identifying the features that depend most on the label. The features are sorted in descending order according to their importance [25]. The Chi-square independence test will be employed in this study to investigate the relationship between variables. One of the primary advantages of the Chi-square independence test is its versatility, as it can be utilized for analyzing both nominal and numerical data [26]. Below is a simple formula of CS that can be applied (2) [10]:

$$CS\ (t_k, ci) = \frac{N(AD - CB)^2}{(A+C)+(B+D)+(A+B)+(C+D)} \quad (2)$$

2.4 Wight Averaging

In this filtering process, the weight of the results given to each feature is in accordance with the significance level of the correlation relationship obtained from the CFS and CS methods. In more detail, the calculation of CFS as shown in the formula equation 1, after coming out the results obtained will be stored first. Then the calculation is done using CS with the formula of equation 2, the results will then be saved. Each of the results given from CFS and CS is weighted by taking the average of the results from these two methods. The results from both methods will be correlated again with the calculation using the FS method of the wrapper technique.

Algorithm 1. Pseudocode of Correlation of Formula 1 and 2

```
Begin
    Dataset = fetch_dataset("NASA MDP D")
    Feature_weight_corr = Correlation_based_feature_selection(Dataset)
    Feature_weight_chi = Chi_square_feature_selection(Dataset)
    Average_feature_weight = (Feature_weight_corr + Feature_weight_chi) / 2
    Output_weight = Average_feature_weight
End
```

2.5 Forward Selection

This feature selection algorithm belongs to the wrapper technique whose working principle is to select the most relevant feature subset. The FS algorithm is one of the derivatives of the greedy search algorithm that reduces the dimension of the feature space into feature subspaces. This approach helps improve computational efficiency and reduces errors in removing irrelevant or noise-containing features [27]. Wrapper methods employ a learning algorithm, typically a classifier, to evaluate the effectiveness of features. Generally, wrapper methods exhibit superior performance in terms of accuracy compared to filter methods, albeit with the trade-off of being computationally more intensive. Conversely, filter methods are known for their efficiency, as they operate faster, but they may not always achieve the same level of accuracy as wrapper methods [28]. Forward greedy selection is a modification of the commonly used matching pursuit algorithm, also referred to as boosting in the machine learning literature that analyzes the performance of matching pursuit algorithms to achieve good approximations. While the forward greedy algorithm itself can yield good estimates, the selection of bases it employs is often inefficient [29]. FS has a basic rule, which is to greedily select additional predictors each step in reducing the squared error. This FS algorithm adds significant to the candidate pool and eliminates redundant candidate predictors step by step until all are potential [30]. The following is the formula for calculating FS (3) [31]:

$$\hat{j}_{t,FS} = arg\ min\ \left\| Res\left(Y \middle| X_{\hat{j}_{t \cup j}}\right)\right\|^2 = arg\ max \frac{\left|\langle Res\left(Y \middle| X_{\hat{j}_t}\right), Res\left(X_j \middle| X_{\hat{j}_t}\right)\rangle\right|}{\left\| Res\left(Y \middle| X_{\hat{j}_t}\right)\right\| \cdot \left\| Res\left(X_j \middle| X_{\hat{j}_t}\right)\right\|} \quad (3)$$

This FS will set $\hat{j}_{t+1} = \hat{j}_t \cup \hat{j}_{t,FS}$ and iterate until the model size has meets the predefined limit and the termination has been met [31].

2.6 Particle Swarm Optimization

PSO is one of the swarm-based algorithms used to find optimal solutions by simulating the movement behavior of a group of particles inspired by the movement of a flock of birds. This algorithm works by each particle representing a candidate solution to the problem to be solved. The particle's position in the search space is determined by the fitness function. Particles will continuously search for the optimal solution and approach better solutions in the search space. Through several filtering iterations, the particle that has the optimal solution is expected to be reached [32]. The following PSO formula can be used (4)(5) [32]:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (Pb_i(t) - X_i(t) + c_2 r_2 (Gb_i(t) - X_i(t)) \quad (4)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (5)$$

In PSO, there is the concept of Particle best (Pb) which is the optimization of the previous particle, and Global best (Gb) which is the optimization result of the entire particle population during iterations. In addition, there are acceleration factors, c1 and c2, which are used to set the learning step, and r1 and r2 are used to follow the distribution of random numbers in the search. The variable t indicates the number of iterations, and x is the inertia weight used to maintain a balance between exploration and exploitation capabilities in the search for solutions [32].

2.7 Performance Evaluation

To evaluate the findings in research focusing on PSO enhancement, this experiment using feature selection with hybrid techniques method, such as CFS, CS, and FS. Furthermore, these feature selection techniques will be integrated with the Naive Bayes machine learning classification algorithm. This approach aims to thoroughly understand and evaluate the significance of each approach used. The importance of the AUC (Area Under Curve) statistical value in assessing the classification performance for this research evaluation cannot be ignored [33]. The range of AUC values ranges from 0 to 1, where the higher the AUC value indicates better model performance [34]. The significance of the AUC output will be determined based on the average performance of the model. A significance level of 0.05 will be utilized, as it is a standard value commonly employed in testing research models, providing a confidence level of 95%. [35].

Algorithm 2. Pseudocode of Formula Correlation with Previous Formula

```
Begin
    Output_weight = Average_feature_weight
    Fs_Pso_model = forward_selection_with_pso(Output_weight)
```

```
    Training_data, Testing_data = split_dataset(Fs_Pso_model)
    Evaluation_metric = Cross_validation_naive_bayes(Training_data)
    Test_result = test_model(Training_data, Testing_data(Fs_Pso_model))
    Display_performance_evaluation(test_result)
End
```

3. RESULT AND DISCUSSION

This study aims to evaluate the effectiveness of feature selection with hybrid techniques with filter and wrapper techniques with integration using Naive Bayes in overcoming problems that often occur in PSO, such as noisy attributes, high-dimensional data, and premature convergence. This study will assess whether this method can improve the output performance of the feature selection process and whether the performance difference is significant based on the AUC value. This form of research evaluation provides new insights into the use of different approaches in feature selection to improve PSO performance in addressing frequently encountered problems and determine whether the feature selection with hybrid techniques can provide better and statistically significant results in improving PSO performance.

The results given in Table 2 below, display the output AUC difference values of 48 trials on 12 datasets from NASA MDP D". From the analysis, it can be seen that the performance of PSO improves after combining the feature selection approach between filter techniques from CFS and CS with wrapper techniques from FS, compared to using PSO. It is important to note that the potential of feature selection with hybrid techniques can be said to obtain improved PSO performance optimization in handling noisy attributes and high-dimensional data. The following is a further explanation of each column of table 2 below:

1. In the column with PSO method, the AUC value tends to be unbalanced in each dataset due to the performance in PSO is too high dimensional data and a lot of noisy data so that the average AUC value can be said to be inconsistent even at 0.855 with the lowest value at 0.672.
2. In the column with FS PSO, there is an imbalance in the consistency of performance values on each dataset due to the lack of correlation in handling PSO problems and there is a decrease in the average AUC value at 0.798 with the lowest value at 0.484.
3. In the column with the CFS FS PSO method, again experiencing a decrease in the performance of the consistency of values on each dataset which is again due to the lack of good handling of problems in PSO and the average AUC value has decreased at 0.767 with the lowest value at 0.561.
4. In the column with CS FS PSO method, there is a slight improvement although not significant, the problem handling here is still not meaningful because the average AUC value is at 0.786 with the lowest value at 0.540.
5. In the column with the combined CFS CS FS PSO hybrid method, there is a significant improvement in performance compared to the others. Even when compared to PSO, there is an improvement that is classified as significantly improved because it touches the average value of AUC at 0.877 which proves successful in improving and handling problems in PSO with the lowest AUC average value marked only at 0.679.

Table 2. Final AUC Perfomance

Datasets	PSO	FS PSO	CFS FS PSO	CS FS PSO	CFS CS FS PSO
CM1	0.895	0.819	0.828	0.882	0.892
JM1	0.672	0.663	0.637	0.656	0.679
KC1	0.749	0.759	0.700	0.727	0.755
KC3	0.879	0.896	0.868	0.861	0.944
MC1	0.887	0.777	0.716	0.653	0.916
MC2	0.891	0.901	0.788	0.837	0.944
MW1	0.928	0.954	0.820	0.985	0.980
PC1	0.894	0.874	0.813	0.876	0.914
PC2	0.913	0.484	0.561	0.540	0.940
PC3	0.872	0.826	0.811	0.842	0.879
PC4	0.901	0.883	0.893	0.862	0.916
PC5	0.773	0.745	0.767	0.713	0.765
Average	**0.855**	**0.798**	**0.767**	**0.786**	**0.877**

Table 3 displays the performance quality based on the inter-testing methods. This comparison compares the proposed method, namely feature selection with hybrid techniques, with a combination of other methods. This is evidenced in columns two and three, in column two shows the value of improving the quality of significance between several combined methods with feature selection with hybrid techniques proposed in this study. While the third column shows whether it is significant or not. This means that if the AUC value obtained is below the alpha AUC value of 0.050, then it can be said that there is a "significance" performance improvement. Conversely, if the AUC value is above the alpha value, it can be said that the performance improvement is "not significance". Here the focus is on improving and fixing the problems found in PSO, if you look at the comparison of the CFS CS FS PSO method which is the proposed method with PSO, it gets a superior value, which is around 0.00342 and this value is below the predetermined alpha value so that it can be said that the quality of performance has improved "significance". Then if you look at other comparisons between the proposed method of this research and other methods, such as comparison with FS PSO getting a value of 0.02535 getting quality performance "significance", comparison with CFS FS PSO getting a value of 0.00180 getting quality performance "significance", comparison with CS FS PSO getting a value of 0.01186 getting quality performance "significance". Judging from the overall quality of performance, all get "significance" which means successful in solving the problems that exist in PSO with an increase in the quality of performance on the AUC value.

Table 3. T-Test Result

Method Comparison	T-Test (α = 0.050)	Performance Quality
CFS CS FS PSO – PSO	**0.00342**	**Significance**
CFS CS FS PSO – FS PSO	**0.02535**	**Significance**
CFS CS FS PSO – CFS FS PSO	**0.00180**	**Significance**
CFS CS FS PSO – CS FS PSO	**0.01186**	**Significance**

This comparison compares several methods related to feature selection and overcoming problems in SDP. The AUC result obtained in this study is 0.877 with the hybrid feature selection method. This can prove that the AUC value generated from the method in this study gets results that tend to be high compared to other studies. Table 4 serves to compare some previous research results that are relevant to this research, namely in terms of improving and overcoming problems found in SDP.

Table 4. Comparison with Previous Research Methods

Previous Research Method	Average AUC	Proposed Research Method	Average AUC
MLP FS ROS [36]	0.817	Hybrid Feature Selection	**0.877**
Hellinger Net [37]	0.760	Hybrid Feature Selection	**0.877**
MLMNB [38]	0.690	Hybrid Feature Selection	**0.877**
RMFFS NB CS [39]	0.746	Hybrid Feature Selection	**0.877**
PHBIF [40]	0.792	Hybrid Feature Selection	**0.877**

4. CONCLUSION

This study shows that the use of the proposed method, namely hybrid feature selection with a combination of filter and wrapper techniques integrated with the Naive Bayes classification algorithm can provide significant improvements in improving the performance of the PSO algorithm. In PSO itself there are problems such as high dimensionality, noisy attributes, and premature convergence. The filter techniques used are filtering methods such as CFS and CS. Then combined with wrapper techniques such as FS. It can be seen from several combinations of methods that have been carried out by researchers that several combinations other than the proposed method show inconsistent values on each NASA MDP dataset which makes the assumption that there is no match in their use in terms of improving and overcoming problems. Then after a combination of filter and wrapper techniques that produce quite consistent values and on each dataset. Comparison of the AUC significance value between the hybrid feature selection method compared to several other combinations shows a significant improvement in performance quality with an AUC value that is below the alpha value of 0.050. The significant value obtained from using the hybrid feature selection method reaches a value of 0.00342 compared to using only PSO. Another comparison between such as with FS getting a value of 0.02535, with CFS FS getting a value of 0.00180, and with CS FS getting a value of 0.01186. In addition, the average value resulting from the use of methods with hybrid feature selection gets a value of 0.877. The selection of methods in

this study provides evidence that Improving with Hybrid Feature Selection with a combination of filter and wrapper techniques integrated with Naive Bayes can significantly improve quality while solving the problem. Suggestions for future research can focus on further exploration of hybrid combinations of different technique approaches and their application in different application domains to focus on quality improvement and problem solving.

ACKNOWLEDGEMENTS

This research was supported by the University of Lambung Mangkurat (ULM). We thank our colleagues from the Departement of Computer Science.

REFERENCES

[1] M. J. Hernández-Molinos, A. J. Sánchez-García, R. E. Barrientos-Martínez, J. C. Pérez-Arriaga, and J. O. Ocharán-Hernández, "Software Defect Prediction with Bayesian Approaches," *Mathematics*, vol. 11, no. 11, Jun. 2023, doi: 10.3390/math11112524.

[2] C. Ni, X. Chen, F. Wu, Y. Shen, and Q. Gu, "An empirical study on pareto based multi-objective feature selection for software defect prediction," *Journal of Systems and Software*, vol. 152, pp. 215–238, Jun. 2019, doi: 10.1016/j.jss.2019.03.012.

[3] B. Khan *et al.*, "Software Defect Prediction for Healthcare Big Data: An Empirical Evaluation of Machine Learning Techniques," *J Healthc Eng*, vol. 2021, 2021, doi: 10.1155/2021/8899263.

[4] H. Xie, L. Zhang, C. P. Lim, Y. Yu, and H. Liu, "Feature selection using enhanced particle swarm optimisation for classification models," *Sensors*, vol. 21, no. 5, pp. 1–40, Mar. 2021, doi: 10.3390/s21051816.

[5] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle Swarm Optimization: A Comprehensive Survey," *IEEE Access*, vol. 10, pp. 10031–10061, 2022, doi: 10.1109/ACCESS.2022.3142859.

[6] M. Cai, "An Improved Particle Swarm Optimization Algorithm and Its Application to the Extreme Value Optimization Problem of Multivariable Function," *Comput Intell Neurosci*, vol. 2022, 2022, doi: 10.1155/2022/1935272.

[7] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, Aug. 2022, doi: 10.1007/s11831-021-09694-4.

[8] J. Divasón, A. Pernia-Espinoza, and F. J. Martinez-de-Pison, "HYB-PARSIMONY: A hybrid approach combining Particle Swarm Optimization and Genetic Algorithms to find parsimonious models in high-dimensional datasets," *Neurocomputing*, vol. 560, Dec. 2023, doi: 10.1016/j.neucom.2023.126840.

[9] B. J. Solano-Rojas, R. Villalón-Fonseca, and R. Batres, "Micro Evolutionary Particle Swarm Optimization (MEPSO): A new modified metaheuristic," *Systems and Soft Computing*, vol. 5, Dec. 2023, doi: 10.1016/j.sasc.2023.200057.

[10] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved Chi-square for Arabic text classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 2, pp. 225–231, Feb. 2020, doi: 10.1016/j.jksuci.2018.05.010.

[11] M. P. A. Starmans, S. R. van der Voort, J. M. C. Tovar, J. F. Veenland, S. Klein, and W. J. Niessen, "Radiomics," in *Handbook of Medical Image Computing and Computer Assisted Intervention*, Elsevier, 2019, pp. 429–456. doi: 10.1016/B978-0-12-816176-0.00023-5.

[12] W. Shafqat, S. Malik, K. T. Lee, and D. H. Kim, "Pso based optimized ensemble learning and feature selection approach for efficient energy forecast," *Electronics (Switzerland)*, vol. 10, no. 18, Sep. 2021, doi: 10.3390/electronics10182188.

[13] M. Reif and F. Shafait, "Efficient feature size reduction via predictive forward selection," *Pattern Recognit*, vol. 47, no. 4, pp. 1664–1673, Apr. 2014, doi: 10.1016/j.patcog.2013.10.009.

[14] K. Orphanou, A. Dagliati, L. Sacchi, A. Stassopoulou, E. Keravnou, and R. Bellazzi, "Incorporating repeating temporal association rules in Naïve Bayes classifiers for coronary heart disease diagnosis," *J Biomed Inform*, vol. 81, pp. 74–82, May 2018, doi: 10.1016/j.jbi.2018.03.002.

[15] S. Chen, G. I. Webb, L. Liu, and X. Ma, "A novel selective naïve Bayes algorithm," *Knowl Based Syst*, vol. 192, Mar. 2020, doi: 10.1016/j.knosys.2019.105361.

[16] R. Ferenc, P. Gyimesi, G. Gyimesi, Z. Tóth, and T. Gyimóthy, "An automatically created novel bug dataset and its validation in bug prediction," *Journal of Systems and Software*, vol. 169, Nov. 2020, doi: 10.1016/j.jss.2020.110691.

[17] S. Mcmurray and A. H. Sodhro, "A Study on ML-Based Software Defect Detection for Security Traceability in Smart Healthcare Applications," *Sensors*, vol. 23, no. 7, Apr. 2023, doi: 10.3390/s23073470.

[18] H. Alsawalqah *et al.*, "Software defect prediction using heterogeneous ensemble classification based on segmented patterns," *Applied Sciences (Switzerland)*, vol. 10, no. 5, Mar. 2020, doi: 10.3390/app10051745.

[19] H. Wei, C. Hu, S. Chen, Y. Xue, and Q. Zhang, "Establishing a software defect prediction model via effective dimension reduction," *Inf Sci (N Y)*, vol. 477, pp. 399–409, Mar. 2019, doi: 10.1016/j.ins.2018.10.056.

[20] M. Shobana *et al.*, "Classification and Detection of Mesothelioma Cancer Using Feature Selection-Enabled Machine Learning Technique," *Biomed Res Int*, vol. 2022, 2022, doi: 10.1155/2022/9900668.

[21] P. Bathla and R. Kumar, "A hybrid system to predict brain stroke using a combined feature selection and classifier," *Intelligent Medicine*, Aug. 2023, doi: 10.1016/j.imed.2023.06.002.

[22] J. Linja, J. Hämäläinen, P. Nieminen, and T. Kärkkäinen, "Feature selection for distance-based regression: An umbrella review and a one-shot wrapper," *Neurocomputing*, vol. 518, pp. 344–359, Jan. 2023, doi: 10.1016/j.neucom.2022.11.023.

[23] N. García-Pedrajas and G. Cerruela-García, "MABUSE: A margin optimization based feature subset selection algorithm using boosting principles," *Knowl Based Syst*, vol. 253, Oct. 2022, doi: 10.1016/j.knosys.2022.109529.

[24] B. Sen Peng, H. Xia, Y. K. Liu, B. Yang, D. Guo, and S. M. Zhu, "Research on intelligent fault diagnosis method for nuclear power plant based on correlation analysis and deep belief network," *Progress in Nuclear Energy*, vol. 108, pp. 419–427, Sep. 2018, doi: 10.1016/j.pnucene.2018.06.003.

[25] W. BinSaeedan and S. Alramlawi, "CS-BPSO: Hybrid feature selection based on chi-square and binary PSO algorithm for Arabic email authorship analysis," *Knowl Based Syst*, vol. 227, Sep. 2021, doi: 10.1016/j.knosys.2021.107224.

[26] F. Uğurlu, S. Yıldız, M. Boran, Ö. Uğurlu, and J. Wang, "Analysis of fishing vessel accidents with Bayesian network and Chi-square methods," *Ocean Engineering*, vol. 198, Feb. 2020, doi: 10.1016/j.oceaneng.2020.106956.

[27] S. Shafiee, L. M. Lied, I. Burud, J. A. Dieseth, M. Alsheikh, and M. Lillemo, "Sequential forward selection and support vector regression in comparison to LASSO regression for spring wheat yield prediction based on UAV imagery," *Comput Electron Agric*, vol. 183, Apr. 2021, doi: 10.1016/j.compag.2021.106036.

[28] A. Got, A. Moussaoui, and D. Zouache, "Hybrid filter-wrapper feature selection using whale optimization algorithm: A multi-objective approach," *Expert Syst Appl*, vol. 183, Nov. 2021, doi: 10.1016/j.eswa.2021.115312.

[29] P. Shekhar and A. Patra, "A forward–backward greedy approach for sparse multiscale learning," *Comput Methods Appl Mech Eng*, vol. 400, Oct. 2022, doi: 10.1016/j.cma.2022.115420.

[30] H. Zhao, Z. Gao, F. Xu, Y. Zhang, and J. Huang, "An efficient adaptive forward–backward selection method for sparse polynomial chaos expansion," *Comput Methods Appl Mech Eng*, vol. 355, pp. 456–491, Oct. 2019, doi: 10.1016/j.cma.2019.06.034.

[31] J. Wieczorek and J. Lei, "Model selection properties of forward selection and sequential cross-validation for high-dimensional regression," *Canadian Journal of Statistics*, vol. 50, no. 2, pp. 454–470, Jun. 2022, doi: 10.1002/cjs.11635.

[32] B. Fu, Y. He, Q. Guo, and J. Zhang, "An improved competitive particle swarm optimization algorithm based on de-heterogeneous information," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 6, Jun. 2023, doi: 10.1016/j.jksuci.2022.12.012.

[33] R. Malhotra, R. Kapoor, P. Saxena, and P. Sharma, "SAGA: A Hybrid Technique to handle Imbalance Data in Software Defect Prediction," in *ISCAIE 2021 - IEEE 11th Symposium on Computer Applications and Industrial Electronics*, Institute of Electrical and Electronics Engineers Inc., Apr. 2021, pp. 331–336. doi: 10.1109/ISCAIE51753.2021.9431842.

[34] M. H. Murad, A. K. Balla, M. S. Khan, A. Shaikh, S. Saadi, and Z. Wang, "Thresholds for interpreting the fragility index derived from sample of randomised controlled trials in cardiology: a meta-epidemiologic study," *BMJ Evid Based Med*, vol. 28, no. 2, pp. 133–136, 2023, doi: 10.1136/bmjebm-2021-111858.

[35] N. S. Mohamed *et al.*, "Impact factors of orthopaedic journals between 2010 and 2016: trends and comparisons with other surgical specialties," *Ann Transl Med*, vol. 6, no. 7, pp. 114–114, Apr. 2018, doi: 10.21037/atm.2018.03.02.

[36] A. Iqbal and S. Aftab, "A classification framework for software defect prediction using multi-filter feature selection technique and MLP," *International Journal of Modern Education and Computer Science*, vol. 12, no. 1, pp. 18–25, 2020, doi: 10.5815/ijmecs.2020.01.03.

[37] T. Chakraborty and A. K. Chakraborty, "Hellinger Net: A Hybrid Imbalance Learning Model to Improve Software Defect Prediction," *IEEE Trans Reliab*, vol. 70, no. 2, pp. 481–494, Jun. 2021, doi: 10.1109/TR.2020.3020238.

[38] N. S. Harzevili and S. H. Alizadeh, "Analysis and modeling conditional mutual dependency of metrics in software defect prediction using latent variables," *Neurocomputing*, vol. 460, pp. 309–330, Oct. 2021, doi: 10.1016/j.neucom.2021.05.043.

[39] A. O. Balogun *et al.*, "Empirical analysis of rank aggregation-based multi-filter feature selection methods in software defect prediction," *Electronics (Switzerland)*, vol. 10, no. 2, pp. 1–16, Jan. 2021, doi: 10.3390/electronics10020179.

[40] Z. Ding and L. Xing, "Improved software defect prediction using Pruned Histogram-based isolation forest," *Reliab Eng Syst Saf*, vol. 204, Dec. 2020, doi: 10.1016/j.ress.2020.107170.