

Feature selection and model performance Analysis

Ms Dimpy (Assistant professor)

Vanshika Dubey , Sakshi Kumari, Aarzia Juned

Artificial Intelligence And Data Science
Dr. Akhilesh Das Gupta Institute of Professional Studies

Abstract- Feature selection is an important step in machine learning that helps improve model performance by removing irrelevant and redundant data. This study compares different feature selection techniques and analyzes their impact on various models. Results show that proper feature selection reduces complexity, improves accuracy, and prevents overfitting, leading to better overall model performance.

Keywords-

- Feature Selection
- Machine Learning
- Model Performance
- Dimensionality Reduction
- Overfitting
- Model Evaluation

I. INTRODUCTION

In the field of machine learning, input features have a great impact on model performance. Large datasets tend to have unnecessary or repeated features which make them complex and inaccurate. Selecting features helps to efficiently select useful features for identification and development of models. This study analyzes the effects of other feature selection techniques for the overall performance of the model Related studies.

Related works : Software metrics are a large part of model inputs in Software Defect Prediction (SDP).

Further having this large number of metrics that give us redundant and irrelevant features negatively impacts model performance. In order to overcome this situation, many Feature Selection (FS) techniques have been designed to pick optimal subset of features. Past the findings of existing studies comparing these approaches.

A large-scale evaluation of two-eight FS methods with twenty-one classifiers using NASA and PROMISE datasets was done by Ghotra et al. Their findings show that the performance of correlation-based filter methods using Best First (BF) search is better than the performance of other methods.

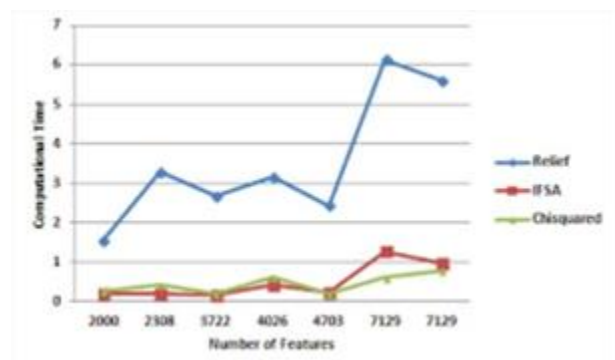
Nonetheless, they only benchmarked BF and GA search techniques and did not explore the results on other meta-heuristic approaches, such as BAT and Ant Search or algorithms, which may yield better results. Afzal and Torkar examined various methods of FS in 2011.

Techniques include IG, PCA, CFS, and wrapper approaches. Also include Naïve Bayes and Decision Tree classifier on five defect datasets. Evaluated Performance Based on AUC. It was found that FS enhances the performance of SDP, however no method is dominant.

Other studies looked into mixed and comparative methods. In a recent work, Gao et al. proposed a hybrid FS method that combines KS stats FS with Automatic Hybrid Search (AHS). It shows that the

methods could remove 85% of the features, and still achieve the same performance or a better performance than using full feature set. In addition, Akintola et al. and Rodriguez et al. showcased that the compact features subsets still retain or improve predictive accuracy. Furthermore, wrapper-based methods produced the most accurate classifiers, especially on small datasets.

Evaluations of FS methods across various search mechanisms are still lacking despite these contributions. That is why this study aims to perform a comparative performance analysis of various FS techniques using various classifiers and software defect datasets from NASA repository. The evaluation of models in terms of prediction accuracy is sufficient to select the best one. In addition, the stability of the selected model can be assessed using the coefficient of variation. Hence, using these stability measures, selected models can provide deeper insights into the effectiveness of FS methods in SDP. The four FFR methods with different functional characteristics are utilized and the top features are selected using $\log_2 N$, from the data set in the present study.



II. LITERATURE REVIEW

Feature selection (FS) is used heavily in machine learning to improve model performance by eliminating irrelevant and redundant features. Common types are filter methods (e.g. correlation, Information Gain), wrapper methods, and embedded methods. According to research, They help reduce the complexity and improve accuracy. Nonetheless,

performance diverges across datasets and models, and no single FS method is best overall.

Research shows that wrapper methods usually yield superior outcomes because they are evaluated based on the model. However, filter methods are faster and more scalable. A combination of both has been proposed to enjoy the advantage of each. In short, this work tells that feature selection if works effectively improves performance of the model. Also the method used depends on the dataset and problem at hand.

III. METHODOLOGY

This section describes the FS methods, the respective search methods, classification algorithms, experimental setup, software defect dataset and performance metrics used in this study.

Filter Feature Ranking (FFR) methods evaluate and rank features independently of the prediction model using dataset characteristics such as statistical, probabilistic, and instance-based measures [29]. Each feature is assigned a score, and top-ranked features are selected accordingly. In this study, four FFR methods with different functional characteristics are used, and the top features are selected using $\log_2 N$, where N represents the total number of software metrics in the dataset. Table 1 summarizes the FFR methods applied. 3.1. Filter Feature Ranking Method

Table 1. List of Filter Feature Ranking (FFR) Methods.

Filter Feature Rank Method	Search Method	Characteristics	Reference
Information Gain Attribute Evaluator (IG)	Ranker Search	Probability-based	[18,32]
Relief Feature Attribute Evaluator (RFA)	Ranker Search	Instance-based	[29,32]
Gain Ratio Attribute Evaluator (GR)	Ranker Search	Probability-based	[29,32]
Clustering Variation Attribute Evaluator (CV)	Ranker Search	Statistics-based	[18,29]

3.2. Filter Feature Subset Selection Method

Feature Subset Selection (FSS), similar to FFR, evaluates and selects features based on specific properties [32]. However, it focuses on generating subsets of features that collectively improve predictive performance by considering feature

combinations. FSS relies heavily on search methods, which explore the feature space to identify optimal subsets, and its performance varies depending on the search strategy used [29]. In this study, two FSS methods are considered. Table 2 provides details of these methods, while Table 3 presents the different search methods along with their parameters.

Table 2. List of Filter Feature Subset Selection (FSS) Methods.

Filter Feature Subset Selection Method	Search Method	Reference
Correlation-based Feature Subset Selection (CFS)	Best First Search (BFS)	[32,36]
	Greedy Stepwise Search (GSS)	[29,36]
	Ant Search (AS)	
	Bat Search (BAT)	
	Firefly Search (FS)	
Consistency Feature Subset Selection (CNS)	Genetic Search (GS)	[32,36]
	PSO Search (PSOS)	
	Best First Search (BFS)	[32,36]
	Greedy Stepwise Search (GSS)	[29,36]
	Ant Search (AS)	
Consistency Feature Subset Selection (CNS)	Bat Search (BAT)	
	Firefly Search (FS)	
	Genetic Search (GS)	[32,36]
	PSO Search (PSOS)	

Table 3. Various Search Methods and Parameter Setting.

Search Methods	Parameter Settings
Best First Search	Direction = Bi-directional
Greedy Stepwise Search	Conservative Forward Selection = True; Search Backwards = False; NumToSelect = $\log_2 N$ (N = Number of Features)
Ant Search	AccelerateType = Accelerate; Chaotic co-efficient = 0.4 Chaotic Parameter Type = Chaotic Map; Parameter; Chaotic Type = logistic map; PopulationSize = 200; Pivomone = 2.0
Bat Search	AccelerateType = Accelerate; Chaotic co-efficient = 0.4; Chaotic Parameter Type = logistic Map; Parameter; Chaotic Type = logistic map; PopulationSize = 200; loudness = 0.5
Firefly Search	AccelerateType = Accelerate; Chaotic co-efficient = 0.4; Chaotic Parameter Type = logistic Map; Parameter; Chaotic Type = logistic map; PopulationSize = 200; absorption = 0.001; betaMin = 0.33
Genetic Search	PopulationSize = 200; MaxGeneration = 20; crossoverProb = 0.6
PSO Search	PopulationSize = 200; IndividualWeight = 0.34; InertiaWeight = 0.33; SocialWeight = 0.33

3.3. Classification Algorithms

For classification, four widely used algorithms—Naïve Bayes (NB), Decision Tree (DT), Kernel Logistic Regression (LR), and K-Nearest Neighbor (KNN)—were used to evaluate the effectiveness of FS methods. These classifiers were chosen to ensure

independence from any single technique and represent diverse characteristics: NB (Bayes-based), DT (tree-based), LR (function-based), and KNN (instance-based). This diversity helps analyze how different FS methods perform across various classifier types. Table 4 provides a brief description of these algorithms and their parameter settings.

Table 4. Classification Algorithms.

Classifier	Description	Parameter Setting
Naive Bayes (NB)	Algebra Theorem based classification technique	NumDiscountPlace = 2; NumOfFold = Normal Dist.
Decision Tree (DT)	A Tree based classification technique	Confidence factor = 0.25; MinNumObj = 2
Kernel Logistic Regression (LR)	A Function-based classification technique	Kernel = PolyKernel (E = 1.0, C = 25000); lambda = 0.01; Quadratic penalty = BFGS Optimization
K Nearest Neighbor (KNN)	An Instance learning-based classification technique	K = 1; NNSearch = LinearNNSearch based on Euclidean Distance

3.4. Experimental Setup

This section discusses the experimental setup of the comparative performance analysis of FS methods as depicted in Figure 1. The experimental setup can be described based on three major steps.

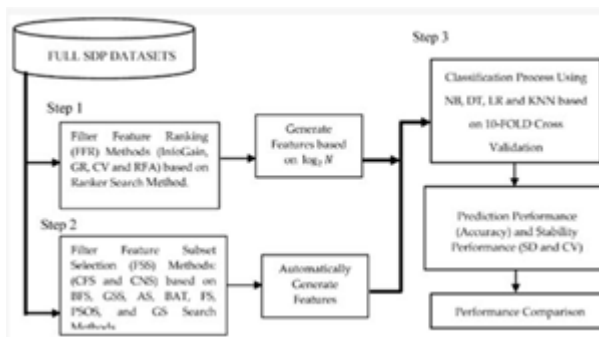


Figure 1. Experimental Setup

Step 1: Four FFR methods (IG, RFA, GR, CV with Ranker Search) were applied to the original datasets to rank features. From each ranked list, $\log_2 N$ features were selected, specifically the top 6 features. The $\log_2 N$ criterion (where $N \geq 21$) was adopted based on prior studies [26,27,28,29,32]. This step resulted in reduced datasets.

Step 2: Two FSS methods (CFS, CNS) combined with 7 search methods (BFS, BAT, FS, AS, GSS, PSOS, GS) produced 14 FSS methods. These methods generated feature subsets with high predictive potential using parameters defined in Table 3, resulting in reduced datasets.

Step 3: Four classifiers (NB, DT, LR, KNN) were applied to both reduced and full datasets.

Experiments used 10-fold cross-validation (CV) to reduce bias, overfitting, and class imbalance [29,37]. Due to randomness in FSS, each experiment was repeated 10 times. In total, 2900 experiments were conducted: (4 FFR × 5 datasets × 4 classifiers) + (14 FSS × 5 datasets × 4 classifiers × 10 runs) + (4 classifiers × 5 datasets).

3.5. Software Defect Datasets

The datasets used in this study were obtained from the NASA MDP repository [38]. Since these datasets are known to be noisy and require preprocessing [26,28,29,32], the cleaned version provided by Shepperd et al. [39] was used. Table 5 presents details of the cleaned datasets, including the number of

Table 5. Software Defect Datasets.

S.No.	Datasets	Language	Number of Features	Number of Modules
1.	CM1	C	37	327
2.	KC1	C++	21	1162
3.	KC3	Java	39	194
4.	MW1	C	37	250
5.	PC2	C	37	722

3.6. Performance Evaluation Metrics

The performance of the models is evaluated using accuracy, which measures the percentage of correctly classified instances based on True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN):

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \times 100\%$$

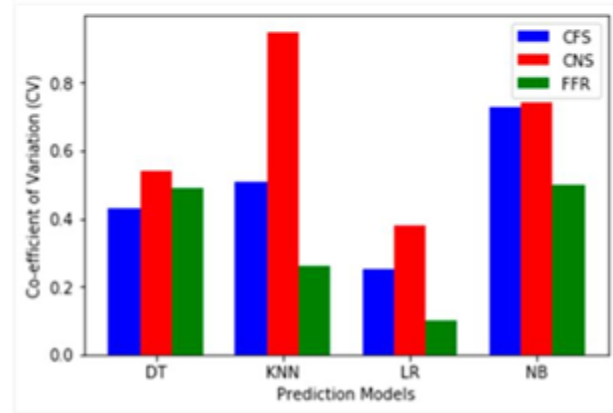
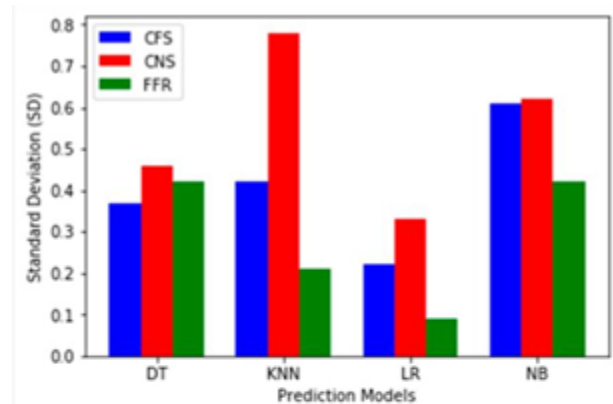
To assess performance stability, the Coefficient of Variation (C.V) is used, defined as the ratio of standard deviation (SD) to average (AVE) [15,40]:

$$C.V = \frac{SD}{AVE} \times 100\%$$

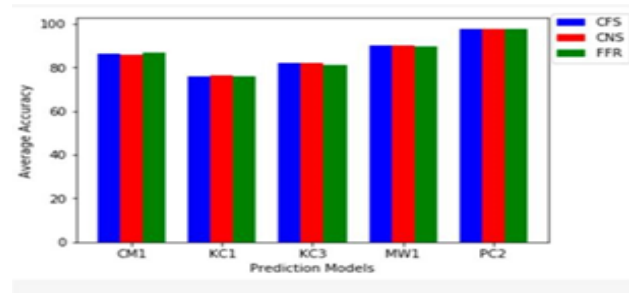
Higher C.V values indicate less stable prediction models.

IV. EXPERIMENTAL RESULTS

Experimental results show that feature selection significantly improves model performance, reducing feature sets by up to 72% while experiencing only marginal accuracy reductions (approx. 0.6%–0.8%). Key findings indicate that wrapper methods like RFE are highly effective for accuracy, while filter methods like Information Gain are faster, though no single method outperforms all others across all datasets. Performance Stability of Prediction Models (SD).



Performance of Feature Selection Method on each Dataset



Key Experimental Findings

Performance Impact: Feature selection, including techniques described on Medium and GeeksforGeeks, generally enhances accuracy by removing noise and reduces computational cost.

Dimensionality Reduction: Experiments frequently show massive reductions in feature space (e.g., from thousands down to a few dozen) while maintaining high classification accuracy, particularly in genetic algorithm-based studies.

Method Comparison: Studies in ScienceDirect and IEEE find that univariate methods (e.g., chi-square) work well for high-dimensional data, while multivariate methods (e.g., wrapper techniques) are better for smaller, complex datasets.

Best Practices: Combining methods, such as mRMR followed by a wrapper, often yields the best results.

Common Results by Technique

L1/L2 Regularization: Proven effective for sparse data, reducing feature sets by 72% with minimal (0.6%–0.8%) accuracy loss in logistic regression applications.

RFE: Often results in better performance by evaluating feature combinations, though it is computationally more expensive.

Information Gain/Chi-Square: Effective, fast filters for identifying relevant features, but may miss feature interactions.

Data Characteristics

Experiments Indicate Microarray Dataset is High Dimensional The key focus here is on feature

selection and also data selection. Gaussian code filtering may enhance accuracies (e.g., from 0.93 to 0.98).

V. APPLICATIONS

Feature selection, a crucial data preprocessing step, improves machine learning model performance by removing irrelevant, redundant, or noisy data. Its

key applications include enhancing accuracy, reducing overfitting, improving model interpretability, speeding up training times, and optimizing resource usage in fields like bioinformatics, text mining, cybersecurity, and image recognition.

VI. THREAT TO VALIDITY

This section discusses potential threats to the validity of the study, as emphasized by Wohlin et al. [44].

External validity: This concerns the generalizability of results. The study used 5 software defect datasets from the NASA repository, which vary in size and characteristics. However, results may not generalize to other datasets, although the experimental setup and parameter settings support replication.

Internal validity: This relates to the choice of models and FS methods. Factors such as dataset quality, classifier selection, and noise can affect results [45]. In this study, 4 classifiers and 18 FS methods (based on 2 FS techniques and 7 search methods) were used. Future work can explore additional methods.

Construct validity: This focuses on evaluation metrics. Accuracy was used to measure performance, while Coefficient of Variation (C.V) assessed stability. Other metrics such as AUC and F-measure could also be considered.

VII. FUTURE SCOPE

- Conceiving ways that have more robustness and adaptability.
- Methods for Performing Feature Selection. uniformly throughout various datasets.
- Greater utilization of hybrid and ensemble methods.
- Techniques Integrating Filter and Wrapper allocative techniques
- Enhance software defect prediction Performance.
- Developed adaptive and dynamic feature Selection methods.

- Examine optimal solutions through metaheuristic optimization.
- Algorithms like Genetic Algorithms (GA). the optimization of the particle swarm.
- Concentrate on a feature that is scalable and able to provide real-time feedback.
- Techniques for Selecting Data for Applications .
- Research on the feature interaction and.
- Selection techniques taking dependency into account.
- Using multiple evaluation metrics (e.g., AUC, F-measure, and other performance evaluation metrics allows improved performance assessment).
- Emerging applications of feature selection methods.
- Fields such as cybersecurity, healthcare, and. Internet of things.

VIII. CONCLUSION

The importance of feature selection on the performance of Software Defect Prediction (SDP) models has been demonstrated in this study. The findings from the feature selection techniques used in this study show that feature selection can reduce dimensionality but typically keeps the accuracy of prediction models the same or with very slight improvements in accuracy. The FFR and FSS feature selection methods showed good performance; however, the actual performance of each method will vary based on the dataset being used, the classifier chosen to build the predictive model, and the specific search strategy that is being used to find the best subset of features.

Wrapper-based feature selection methods typically produced better predicting models than filter-based feature selection methods, but wrapper methods typically take longer to compute than filter methods. No feature selection method was consistently better than the other methods used in the various prediction scenarios. As such, it is critical to carefully choose a feature selection method based on the available problem characteristics. Thus, feature selection provides improved efficiency, reduced overfitting, and improved predictability in the final predictive models.

REFERENCES

1. 2026 Liu, K. Et al a survey on rough feature Selection recent advances and Challenges IEEE/CAA journal of automatic silica
- 2025 Waseem, Q. et al., "Feature Selection for IoT Device Classification", Springer (Innovative Infrastructure Solutions) 2025 "Feature Selection for Mixed Data: Challenges and Solutions", Elsevier (Neurocomputing)
- 2024 Song, X. et al., "Evolutionary Computation for Feature Selection: A Survey", Elsevier (Swarm & Evolutionary Computation)
- 2024 Wu, D. et al., "Multi-metric Latent Feature Analysis", IEEE Transactions on Services Computing
- 2024 Xia, D. et al., "Fuzzy Entropy-based Feature Selection", IEEE Transactions on Fuzzy Systems
- 2023 Wan, J. et al., "Interactive Feature Selection via Fuzzy Methods", IEEE Transactions on Cybernetics
- 2023 Xu, W. et al., "Entropy-based Feature Selection Method", IEEE Transactions on Emerging Topics in Computational Intelligence
- 2023 Gong, Y. et al., "Genetic Algorithm for High-Dimensional Feature Selection", IEEE/CAA Journal
- 2022 Sang, B. et al., "Incremental Feature Selection using Conditional Entropy", IEEE Transactions on Fuzzy Systems
- 2022 Yuan, Z. et al., "Entropy-based Feature Selection for Mixed Data", IEEE Transactions on Systems, Man, and Cybernetics
- 2022 Li, J. et al., "Feature Selection: A Data Perspective", ACM Computing Surveys
- 2021 Bolón-Canedo, V. et al., "Recent Advances in Feature Selection", Elsevier (Knowledge-Based Systems)
- 2021 Nogueira, S. et al., "On Stability of Feature Selection Algorithms", Journal of Machine Learning Research
- 2021 Saeys, Y. et al., "Feature Selection for High-Dimensional Data", Oxford University Press (Bioinformatics)

16. 2020 Chandrashekar, G. et al., "Feature Selection Methods Review", Elsevier (Computers & Electrical Engineering)
17. 2019 Zhang, R. et al., "Feature Selection with Multi-view Data", Elsevier (Information Fusion)