JCTA Journal of Computing Theories and Applications ISSN:3024-9104

ResearchArticle

Enhancing Software Defect Prediction through Hybrid Multi-Filter Feature Selection and Imbalance Handling

Muhammad Khalid Maulana, Setyo Wahyu Saputro *, Mohammad Reza Faisal, Radityo Adi Nugroho, and As'ary Ramadhan

Department of Computer Science, Faculty of Mathematics and Natural Science, Lambung Mangkurat University, Banjarbaru 70714, Indonesia; e-mail : khaaliddd055@gmail.com; setyo.saputro@ulm.ac.id; reza.faisal@ulm.ac.id; radityo.adi@ulm.ac.id; as'ary.ramadhan@ulm.ac.id

* Corresponding Author : Setyo Wahyu Saputro

Abstract: Software Defect Prediction (SDP) aims to identify defective modules early in the software development lifecycle to improve software quality and reduce maintenance costs. However, SDP datasets commonly suffer from high dimensionality, feature redundancy, and class imbalance, which can degrade model performance and stability. This study proposes a hybrid feature selection framework to address these challenges and enhance prediction performance. The proposed approach integrates Combined Correlation and Mutual Information (CONMI), which combines the Pearson Correlation Coefficient (PCC) and Mutual Information (MI) to capture both linear and nonlinear feature relevance. The selected features are further refined through Top-K selection, correlation-based filtering to reduce multicollinearity, and Backward Elimination (BE) to obtain an optimal feature subset. To address class imbalance, SMOTE-Tomek is applied by combining over-sampling and data cleaning techniques. Experiments are conducted on twelve NASA MDP datasets using Logistic Regression (LR) and Naïve Bayes (NB) classifiers. The results show that the proposed framework consistently achieves the best performance, with Logistic Regression combined with SMOTE-Tomek obtaining the highest average AUC of 0.7923 ± 0.0714, while NB achieves 0.7554 ± 0.0580. Statistical analysis using a paired t-test indicates that the proposed method significantly outperforms MI+SMOTE-Tomek and BE+SMOTE-Tomek for Logistic Regression, whereas no significant differences are observed for NB. In addition to improving overall classification performance (AUC), the proposed approach also enhances minority class detection, as reflected in improved Recall and F1-score. Overall, the proposed hybrid framework provides an effective and reliable solution for software defect prediction, particularly for high-dimensional and imbalanced datasets.

Keywords: Backward Elimination; Class Imbalance; Feature Selection; Logistic Regression; Mutual Information; Naïve Bayes; Software Defect Prediction; SMOTE-Tomek.

Received: April, 2nd 2026
Revised: April, 22nd 2026
Accepted: April, 23rd 2026
Published: April, 24th 2026

1. Introduction

Software defects remain one of the most persistent challenges in software engineering, as undetected faults can propagate throughout the development lifecycle and ultimately result in system failures, increased maintenance costs, and reduced reliability. Software Defect Prediction (SDP) addresses this challenge by identifying defect-prone modules early in the development process, enabling development teams to allocate testing resources more effectively and reduce post-deployment correction costs [1]. Empirical evidence consistently shows that defects detected after release incur substantially higher remediation costs than those identified during earlier stages [2], highlighting the practical importance of accurate and efficient prediction methods [3].

Machine learning has become a dominant paradigm in SDP due to its ability to model complex and nonlinear relationships between software metrics and defect outcomes [4], [5]. Although advanced models such as Random Forest, XGBoost, and deep learning architectures have demonstrated strong performance in various domains, their application to SDP

presents several limitations. These models typically require large training datasets, offer limited interpretability, and incur higher computational costs—characteristics that are often misaligned with SDP datasets, which are generally small, high-dimensional, and imbalanced [6]. In contrast, Naïve Bayes (NB) and Logistic Regression (LR) remain well-suited for such conditions. NB efficiently handles high-dimensional data through its conditional independence assumption, while LR provides interpretable probabilistic outputs, allowing insights into the contribution of software metrics such as coupling and complexity [7]–[9]. Therefore, NB and LR are not merely baseline models but represent appropriate and practically relevant classifiers for SDP tasks.

In addition to high dimensionality and feature redundancy, SDP datasets commonly exhibit class imbalance, where defective modules are significantly outnumbered by non-defective ones [10]. This imbalance often leads to biased models that favor the majority class, resulting in poor detection of defective modules and degraded performance in Recall and F1-score. To address this issue, hybrid resampling techniques combining over-sampling and data cleaning strategies have been widely adopted. One commonly used approach is SMOTE-Tomek, which integrates the Synthetic Minority Over-sampling Technique (SMOTE) and Tomek Links. This method balances the dataset by generating synthetic minority samples while removing overlapping or ambiguous instances, thereby improving classification performance and minority class detection [11]. Prior studies have demonstrated that SMOTE-Tomek effectively reduces bias and enhances classification performance while maintaining overall model stability [12].

Despite advances in classification and data balancing, the effectiveness of SDP models remains highly dependent on feature quality. SDP datasets typically contain high-dimensional feature spaces with redundant, irrelevant, or highly correlated attributes, which increase model complexity and degrade generalization performance [13]. Multicollinearity is particularly problematic for LR, as correlated features distort coefficient estimates and reduce interpretability [14], [15]. Therefore, feature selection is not merely a preprocessing step but a critical component for improving model reliability and stability.

Feature selection techniques are generally categorized into filter, wrapper, and embedded methods [16], [17]. Wrapper methods such as Backward Elimination evaluate feature subsets using a target classifier and can produce highly optimized selections. However, they rely on sequential elimination, where removed features cannot be reconsidered, potentially leading to suboptimal subsets, especially in high-dimensional settings [18], [19]. Consequently, relying solely on wrapper-based methods is insufficient when the feature space contains significant redundancy and multicollinearity.

Filter-based methods provide a complementary approach due to their computational efficiency and model independence. Among these, Pearson Correlation Coefficient (PCC) and Mutual Information (MI) are widely used in SDP. PCC measures linear relationships between features and the target variable, offering a simple and interpretable relevance criterion [20]. In contrast, MI captures general statistical dependencies, including nonlinear relationships that PCC cannot detect [21], [22]. However, each method has limitations when used independently: PCC may overlook nonlinear relevance, while MI may be sensitive to noise and redundancy. Combining PCC and MI therefore enables a more comprehensive evaluation of feature relevance, capturing both linear and nonlinear dependencies within a unified framework.

Several studies have explored hybrid and multi-filter feature selection approaches for SDP and related domains, demonstrating that combining complementary feature evaluation strategies can improve robustness and generalization performance [23]–[30]. In addition, prior work has shown that integrating sampling techniques such as SMOTE-Tomek can enhance classification stability and minority class detection in imbalanced datasets [12], [31], [32]. However, these approaches are typically applied in isolation or lack a structured integration within a unified pipeline.

Despite these advancements, several limitations remain. First, many multi-filter approaches focus primarily on feature ranking without explicitly addressing multicollinearity among selected features. Second, methods relying on linear correlation alone may fail to capture nonlinear dependencies. Third, the integration of multi-filter selection with wrapper-based refinement remains limited, particularly in ensuring feature subset stability across diverse datasets. Finally, although imbalance handling techniques are widely used, they are often

applied independently from feature selection, lacking a unified framework that simultaneously considers feature relevance, redundancy reduction, and class imbalance.

To address these limitations, this study proposes a structured multi-filter feature selection framework termed Combined Correlation and Mutual Information (CONMI). CONMI integrates PCC and MI through an equal-weight combination to capture both linear and nonlinear feature relevance. The selected features are further refined using a Top-K strategy, followed by correlation-based filtering to reduce multicollinearity, and Backward Elimination to obtain an optimal feature subset. In addition, SMOTE-Tomek is incorporated to handle class imbalance prior to model training. This integrated pipeline ensures that the selected features are relevant, non-redundant, and stable, while also improving classification performance in imbalanced scenarios.

The proposed framework is evaluated using NB and LR classifiers on NASA Metrics Data Program (NASA MDP) datasets. Performance is assessed using AUC, Precision, Recall, and F1-score, along with statistical significance testing to validate the observed improvements.The main contributions of this study are summarized as follows::

- A CONMI-based multi-filter feature selection framework that captures both linear and nonlinear feature relevance in SDP.
- Integration of SMOTE-Tomek with structured feature selection to address class imbalance.
- A hybrid feature selection pipeline combining filter and wrapper methods for improved feature subset quality.
- Comprehensive evaluation using NB and LR across multiple datasets.
- Performance validation using multiple evaluation metrics and statistical significance testing.

The remainder of this paper is organized as follows. Section 2 reviews related work in SDP, feature selection, and imbalance handling. Section 3 describes the proposed CONMI-based framework and experimental setup. Section 4 presents the experimental results and discussion. Section 5 compares the proposed method with existing approaches. Finally, Section 6 concludes the study and outlines directions for future work.

2. Related Works

Software Defect Prediction (SDP) has received considerable attention due to its role in identifying defect-prone modules early in the software development lifecycle, thereby reducing development costs and improving software quality. Machine learning techniques have been widely adopted in this domain due to their effectiveness in classification tasks. Among these, NB and LR remain prominent because of their simplicity, computational efficiency, and consistent performance on SDP datasets [33], [34]. For example, Oueslati and Manita [35] demonstrated that LR optimized with Fractional Chaotic Grey Wolf Optimizer (FCGWO), combined with SMOTE-based class balancing and cross-validation on NASA MDP datasets, achieves notable performance improvements. This reinforces that classical classifiers remain competitive and practically relevant when appropriately configured.

Handling class imbalance is another critical aspect in SDP. Hybrid resampling techniques that combine over-sampling and data cleaning have been shown to improve classification performance. Zhang et al. [31] applied SMOTE-Tomek to address imbalance in SDP datasets, achieving improved stability and generalization across NASA and PROMISE datasets. Similarly, Swana et al. [12] reported that SMOTE combined with Tomek Links reduces noise and overlapping samples, thereby enhancing classification performance. These studies confirm that SMOTE-Tomek is an effective strategy for improving minority class detection while maintaining overall model stability.

Beyond classifier selection and data balancing, feature selection plays a central role in SDP performance, as irrelevant and redundant features can degrade model accuracy and increase complexity [13]. Early feature selection approaches relied primarily on single filter methods, particularly linear correlation metrics, due to their simplicity and efficiency. However, their inability to capture nonlinear feature-target relationships limits their effectiveness [20], [21]. To address these limitations, hybrid feature selection approaches have been proposed by combining complementary techniques. For instance, Rahmayanti et al. [24] integrated Mutual Information (MI) with Recursive Feature Elimination (RFE) to capture nonlinear dependencies while refining feature subsets iteratively. Noor et al. [25] combined

correlation-based filtering with Backward Elimination to reduce irrelevant features before wrapper-based optimization. These approaches demonstrate that combining filter and wrapper methods can improve performance compared to single-method strategies.

More recently, multi-filter and ensemble-based frameworks have been introduced to enhance feature selection robustness. Balogun et al. [23] proposed Rank Aggregation-Based Multi-Filter Feature Selection (RMFFS), which aggregates rankings from multiple filter methods to reduce redundancy, and later introduced Enhanced Wrapper Feature Selection (EWFS) [26] to integrate multiple evaluation criteria. Rahman et al. [27] further explored weighted aggregation of correlation-based techniques to improve robustness in noisy and high-dimensional datasets, while Pratama et al. [28] demonstrated that combining statistical filters with wrapper methods effectively reduces noisy attributes. In related domains, Akazue et al. [29] and Asuai et al. [30] also showed that combining multiple feature evaluation strategies leads to more robust and generalisable feature subsets.

Despite these advancements, several limitations remain. First, many approaches relying on linear correlation metrics may overlook features with nonlinear relevance, leading to incomplete feature representation [25], [26]. Second, although multi-filter frameworks improve feature ranking, they often do not explicitly address multicollinearity among selected features, which is particularly problematic for LR, as correlated predictors distort coefficient estimates and reduce interpretability [14], [15] . Third, the integration of multi-filter selection with wrapper-based refinement remains limited, especially in ensuring feature subset stability across different SDP datasets [25]. Finally, although class imbalance handling techniques such as SMOTE-Tomek have been widely used, they are often applied independently from feature selection, without a unified framework that simultaneously considers feature relevance, redundancy reduction, and imbalance handling [31].

These limitations indicate the need for a more structured and integrated approach. In particular, there is a lack of frameworks that simultaneously: (1) capture both linear and nonlinear feature relevance, (2) explicitly mitigate multicollinearity, (3) integrate multi-filter selection with wrapper-based refinement, and (4) incorporate imbalance handling within a unified pipeline. Addressing these aspects collectively is essential for improving model stability and prediction performance in SDP. To address these gaps, this study proposes CONMI, a structured multi-filter feature selection framework. CONMI integrates PCC and MI to capture both linear and nonlinear feature-target relationships within a unified relevance measure. Correlation-based filtering is then applied to reduce multicollinearity, followed by Backward Elimination to refine the feature subset. In addition, SMOTE-Tomek is incorporated prior to model training to address class imbalance. This integrated design enables the proposed framework to simultaneously address feature relevance, redundancy, and imbalance, which are typically handled separately in existing approaches. Furthermore, NB and LR are adopted as reference classifiers due to their demonstrated effectiveness and stability on NASA MDP datasets [23], [34], [35], allowing a fair and practical evaluation of the proposed framework under realistic SDP conditions..

3. Proposed Method

This study proposes CONMI, a structured multi-filter feature selection framework designed to improve SDP performance. As illustrated in Figure 1, the framework consists of five sequential stages: (1) data preprocessing, (2) multi-filter feature scoring, (3) correlation-based multicollinearity filtering, (4) wrapper-based refinement using Backward Elimination, and (5) class imbalance handling using SMOTE-Tomek. The pipeline operates on NASA MDP datasets and produces an optimized feature subset, which is subsequently evaluated using NB and LR under Stratified 10-Fold Cross Validation.

In the multi-filter stage, PCC and MI scores are computed independently for each feature, normalized to a common scale, and combined using an equal-weight averaging scheme to produce the CONMI relevance score. Top-K features are then selected to reduce dimensionality. Subsequently, correlation-based filtering is applied to suppress multicollinearity among the selected features, followed by Backward Elimination to refine the subset based on classifier performance. To address class imbalance, SMOTE-Tomek is applied within each training fold prior to model training. Model performance is evaluated using Area Under the Curve (AUC), Precision, Recall, and F1-score, with statistical significance testing applied to validate the observed improvements.

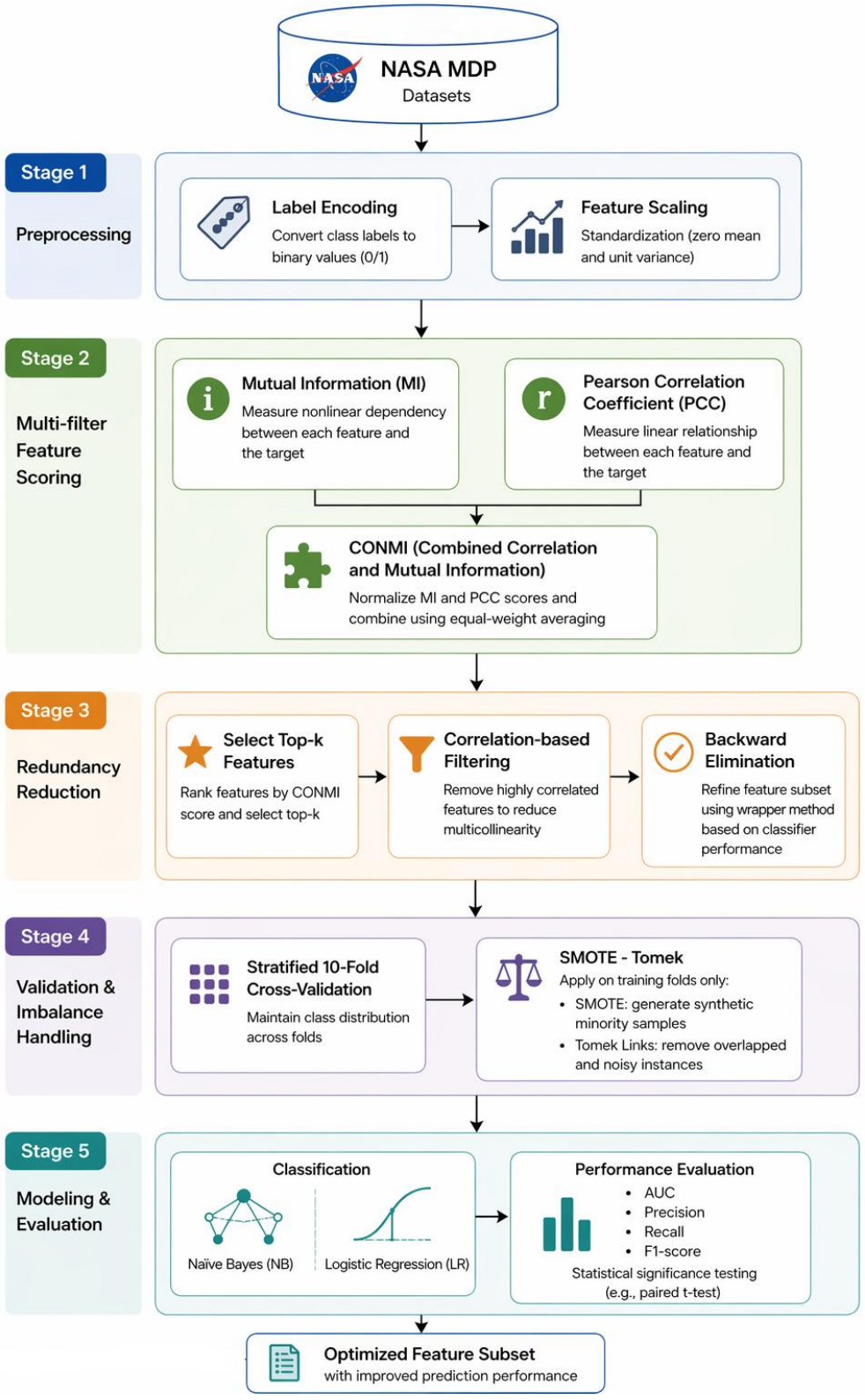

Figure 1. Proposed CONMI framework pipeline.

3.1. Dataset

This study utilizes datasets from the NASA Metrics Data Program (NASA MDP), which are widely used benchmarks in SDP research. These datasets contain software metrics such as Lines of Code (LOC), Cyclomatic Complexity, and Code Churn, representing characteristics of software modules. Each instance corresponds to a module labeled as defective or non-defective, forming a binary classification problem. A total of twelve datasets are used: CM1,

JM1, KC1, KC3, MC1, MC2, MW1, PC1, PC2, PC3, PC4, and PC5. These datasets are selected due to their diversity and widespread use in prior SDP studies. However, they present common challenges, including class imbalance and the presence of noisy or redundant features, which motivate the need for preprocessing and feature selection. A summary of dataset characteristics is provided in Table 1.

Table 1. NASA MDP dataset specifications

Dataset	Attributes	Instances	Non-Defects	Defects	Defective (%)
CM1	38	327	285	42	12.8
JM1	22	7720	6108	1612	20.9
KC1	22	1162	868	294	25.3
KC3	40	194	158	36	18.6
MC1	39	1952	1916	36	1.8
MC2	40	124	80	44	35.5
MW1	38	250	225	25	10.0
PC1	38	679	624	55	8.1
PC2	37	722	706	16	2.2
PC3	38	1053	923	130	12.3
PC4	38	1270	1094	176	13.9
PC5	39	1694	1236	458	27.0

3.2. Preprocessing

A preprocessing stage is performed to prepare the datasets prior to feature selection and classification. First, label encoding is applied to convert class labels into binary numerical values. Next, feature scaling is performed using standardization to ensure that each feature has zero mean and unit variance. This step is particularly important for LR to prevent features with larger magnitudes from dominating the learning process. Standardization is defined as:

$$x' = \frac{x - \mu}{\sigma} \tag{1}$$

where x is the original feature value, μ is the mean, and σ is the standard deviation. The scaling parameters are estimated from the training data and applied to the test data within each fold to avoid data leakage.

3.3. Mutual Information (MI)

MI is used to measure the statistical dependency between a feature and the target variable. It is defined as:

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{2}$$

where $p(x,y)$ is the joint probability distribution, and $p(x)$ and $p(y)$ are marginal probabilities.

In this study, MI is estimated using a nonparametric k-nearest neighbor approach, which avoids explicit density estimation and captures nonlinear dependencies. For each feature X_j, the MI score is computed and normalized:

$$MI_{norm}(j) = \frac{MI(j)}{MI(j) + \epsilon} \tag{3}$$

where $\epsilon = 10^{-9}$ ensures numerical stability. This normalization constrains MI values to the range $(0,1]$, allowing consistent integration with PCC.

3.4. Pearson Correlation Coefficient (PCC)

PCC measures the linear relationship between a feature and the target variable and is defined as:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (4)$$

For each feature X_j, the correlation score is computed as:

$$PCC(j) = |r(X_j, y)| \quad (5)$$

The absolute value ensures that both positive and negative relationships are treated equally.

While MI captures nonlinear dependencies, PCC provides a precise estimation of linear relationships. Their combination enables a more comprehensive evaluation of feature relevance, which forms the basis of the CONMI scoring mechanism.

3.5. Combined Correlation and Mutual Information (CONMI)

This study adopts CONMI to integrate linear and nonlinear feature relevance into a unified scoring mechanism. The CONMI score combines PCC and MI as follows:

$$CONMI(j) = \alpha \cdot PCC(j) + (1 - \alpha) \cdot MI_{norm(j)} \quad (6)$$

where $PCC(j) = |r(X_j, y)|$ and $MI_{norm(j)}$ denotes the normalized MI score.

In this study, equal weighting (α=0.5) is applied to ensure balanced contributions from both linear and nonlinear dependency measures. This design is conceptually grounded in the complementary nature of PCC and MI. PCC provides precise estimation of linear relationships, while MI captures general statistical dependencies independent of functional form. Since no prior assumption is made regarding the dominance of either type of dependency, equal weighting avoids introducing bias and ensures a balanced evaluation of feature relevance. This approach is consistent with prior studies that integrate PCC and MI as complementary measures [36], [37].

The CONMI score is computed independently for each feature and used to rank features in descending order. PCC does not require additional normalization since it is inherently bounded within $[0,1]$, ensuring compatibility with normalized MI values. After ranking, Top-K features are selected based on dataset dimensionality:

- Datasets with fewer than 30 features: Top-15
- Datasets with more than 30 features: Top-25

These thresholds balance dimensionality reduction and information retention. For datasets with limited features (e.g., JM1, KC1), selecting Top-15 preserves most informative attributes while reducing redundancy. Although empirically determined for NASA MDP datasets, this strategy is adaptable by adjusting K relative to dataset size. Similar Top-K selection strategies have been applied in prior MI-based feature selection studies [24].

3.6. Highly Correlated Attribute Filtering

To reduce redundancy, correlation-based filtering is applied to the selected features. A pairwise correlation matrix is computed, and feature pairs exceeding a threshold of 0.99 are identified. For each highly correlated pair:

- The feature with the lower CONMI score is removed
- The feature with the higher score is retained

This step reduces multicollinearity while preserving the most informative features, improving model stability and enhancing the effectiveness of subsequent wrapper-based refinement.

3.7. Backward Elimination

Backward Elimination is applied as a wrapper-based refinement step using LR as the evaluation model. The model is optimized using a quasi-Newton method (BFGS) with a maximum of 300 iterations. Feature significance is assessed using p-values, and features with p-values greater than 0.05 are iteratively removed until all remaining features satisfy the significance criterion. To ensure robustness, fallback strategies are applied in cases of numerical instability, including:

- removing highly correlated features
- removing low-variance features

3.8. Stratified 10-Fold Cross-Validation

To ensure reliable evaluation and reduce overfitting, Stratified 10-Fold Cross-Validation is employed. This approach preserves class distribution in each fold, which is essential for imbalanced SDP datasets. The dataset is partitioned into 10 folds, where 9 folds are used for training, 1 fold is used for testing. This process is repeated ten times, and final performance is reported as the average across all folds. Feature selection (CONMI, filtering, and Backward Elimination) is performed exclusively on the training data within each fold. The resulting feature subset is then applied to the corresponding test fold to prevent data leakage.

3.9. SMOTE-Tomek

To address class imbalance, SMOTE-Tomek is applied to the training data within each fold. This method combines SMOTE for synthetic sample generation and Tomek Links for noise removal. SMOTE generates synthetic samples as follows:

$$X_{new} = X_i + \lambda(X_n - X_i) \tag{7}$$

where x_i is a minority instance, x_n is one of its nearest neighbors, and $\lambda \in [0,1]$ is a random interpolation factor. Tomek Links are subsequently used to remove overlapping majority class instances, thereby improving class separability.

The configuration used is summarized in Table 2 :

Table 2. SMOTE-Tomek configuration

Parameter	Value
SMOTE k_neighbors	5
Tomek Links	Default
Random state	42

SMOTE-Tomek is applied strictly to the training fold after data splitting to prevent data leakage. The test fold contains only original instances, ensuring unbiased evaluation.

3.10. Classification

After obtaining the optimized feature subset, SMOTE-Tomek is applied to the training data, followed by classification using NB and LR within each cross-validation fold.

3.10.1. Naïve Bayes (NB)

NB is a probabilistic classifier based on Bayes' theorem:

$$P(A|B) = \frac{P(B|A)\ P\ (A)}{P(B)} \tag{8}$$

Gaussian NB is used due to the continuous nature of software metrics.

Table 3. Naïve Bayes Configuration

Parameter	Value
Classifier	Gaussian NB
Parameters	Default

3.10.2. Logistic Regression (LR)

LR models the relationship between input features and a binary outcome:

$$z = \beta_0 + \beta_1\ X_1 + \beta_2\ X_2 + \ldots + \beta_n\ X_n \tag{9}$$

LR produces probabilistic outputs and interpretable coefficients, making it suitable for SDP.

Table 4. Logistic regression configuration

Parameter	Value
Solver	Default
Penalty	L2
Max iterations	1000
Random state	42

LR is also used in Backward Elimination for feature significance evaluation.

3.11. Performance Evaluation

Model performance is evaluated using multiple complementary metrics to provide a comprehensive assessment under class imbalance conditions.

- AUC measures overall discrimination capability
- Precision evaluates prediction accuracy for defective modules
- Recall measures the ability to detect actual defective modules
- F1-score provides a balanced measure of Precision and Recall

All metrics are computed for each fold and reported as mean ± standard deviation. To validate robustness, statistical significance testing is conducted using paired tests across folds, ensuring that observed differences are not due to random variation.

4. Results and Discussion

This section presents the experimental results and analysis of the proposed feature selection framework. Experiments were conducted on twelve NASA MDP datasets, including JM1, PC1, KC1, MC1, PC5, MW1, KC3, CM1, PC2, PC3, PC4, and MC2. The evaluation was performed using NB and LR under Stratified 10-Fold Cross Validation. Performance was assessed using AUC, along with Precision, Recall, and F1-score to provide a comprehensive evaluation.

4.1. Experimental Environment

All experiments were conducted in a Python-based environment on a cloud-based notebook platform to ensure reproducibility and consistent execution. To ensure fair comparison, all feature selection methods were evaluated under identical experimental settings. The evaluated methods include:

- Baseline (no feature selection),
- PCC (Top-K based on PCC),
- MI (Top-K based on MI),
- BE (applied directly),
- CONMI+TopK, and
- CONMI+TopK+BE (proposed method).

All methods were evaluated using Stratified 10-Fold Cross Validation with NB and LR. LR was configured with a maximum of 1000 iterations and L2 regularization, while NB used Gaussian assumptions with default parameters. The CONMI score employed equal weighting (α=0.5), and Top-K selection used K=15 (features <30) and K=25 otherwise. Backward Elimination used a significance threshold of 0.05 with a maximum of 300 iterations. SMOTE-Tomek was applied within each training fold using k=5 and random state = 42. All experiments used a fixed random seed (42) to ensure reproducibility.

4.2. Experimental Results

Table 5 presents the per-dataset results of the proposed CONMI+TopK+BE method without SMOTE-Tomek, while Table 6 reports the corresponding results with SMOTE-Tomek. All results are reported as mean ± standard deviation (SD) over Stratified 10-Fold Cross Validation for both NB and LR.

From Table 5, LR generally achieves higher AUC values than NB across most datasets. For example, LR outperforms NB on PC1 (0.8697 vs 0.7818), PC2 (0.8807 vs 0.8189), PC3 (0.8241 vs 0.7930), and PC4 (0.9025 vs 0.7920). This indicates that the proposed feature

selection framework is particularly beneficial for LR, which is sensitive to feature redundancy and multicollinearity. However, despite achieving higher AUC, LR consistently produces lower Recall compared to NB. For instance, in PC3, NB achieves a Recall of 0.5538 compared to 0.1308 for LR, while in MW1, NB achieves 0.5000 and LR only 0.1333. This suggests that LR tends to produce more conservative predictions, prioritizing overall discrimination over minority class detection.

Table 5. Per-Dataset Results of CONMI+TopK+BE Without SMOTE-Tomek (Mean ± SD, 10-fold CV)

Dataset	Classifier	AUC	F1	Precision	Recall
JM1	NB	0.6282 ± 0.0210	0.2867 ± 0.0332	0.4800 ± 0.0472	0.2047 ± 0.0265
	LR	0.6916 ± 0.0186	0.1745 ± 0.0304	0.5639 ± 0.0823	0.1036 ± 0.0197
PC1	NB	0.7818±0.1304	0.3512±0.1774	0.3152±0.1391	0.4167±0.2518
	LR	0.8697±0.0709	0.2710±0.1794	0.4750±0.2839	0.2033±0.1616
KC1	NB	0.6870±0.0627	0.2164±0.0944	0.5799±0.1710	0.1363±0.0653
	LR	0.6968±0.0705	0.1581±0.0718	0.6433±0.2660	0.0920±0.0462
MC1	NB	0.8221±0.1059	0.2238±0.1629	0.1881±0.1542	0.3167±0.2438
	LR	0.8199±0.1157	0.0400±0.1200	0.1000±0.3000	0.0250±0.0750
PC5	NB	0.7255±0.0369	0.1468±0.0560	0.5920±0.1349	0.0850±0.0341
	LR	0.7324±0.0310	0.0865±0.0599	0.4929±0.2284	0.0480±0.0348
MW1	NB	0.8074±0.1161	0.4205±0.2590	0.3750±0.2275	0.5000±0.3333
	LR	0.8144±0.1060	0.1567±0.2468	0.2000±0.3317	0.1333±0.2082
KC3	NB	0.7478±0.1190	0.3483±0.2600	0.4417±0.3250	0.3083±0.2556
	LR	0.7740±0.1314	0.2838±0.2461	0.4833±0.4246	0.2250±0.2206
CM1	NB	0.7343±0.1335	0.2388±0.2164	0.2257±0.2017	0.2800±0.2804
	LR	0.7844±0.0948	0.1650±0.2249	0.2833±0.3948	0.1200±0.1646
PC2	NB	0.8189±0.1385	0.0583±0.1181	0.0367±0.0737	0.1500±0.3202
	LR	0.8807±0.1209	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000
PC3	NB	0.7930±0.0716	0.4198±0.0859	0.3412±0.0754	0.5538±0.1182
	LR	0.8241±0.0548	0.2008±0.1275	0.5200±0.2769	0.1308±0.0913
PC4	NB	0.7920±0.0652	0.3267±0.1048	0.3894±0.0896	0.3069±0.1338
	LR	0.9025±0.0272	0.4784±0.1260	0.7657±0.1495	0.3637±0.1212
MC2	NB	0.7059±0.1265	0.4071±0.2268	0.5333±0.3263	0.3650±0.2314
	LR	0.7009±0.1167	0.4411±0.1942	0.6517±0.2950	0.3650±0.2025

Severe imbalance effects are also evident. In PC2, LR achieves a high AUC (0.8807) but yields zero Recall and F1-score, indicating complete failure in detecting defective instances. A similar pattern is observed in MC1, where Recall remains extremely low (0.0250) despite relatively high AUC (0.8199). These results highlight the limitation of classification performance when imbalance is not explicitly addressed. After applying SMOTE-Tomek (Table 6), a substantial improvement in Recall is observed across nearly all datasets. For example, LR Recall increases significantly in PC2 (0.0000 → 0.7500), MC1 (0.0250 → 0.7417), and MW1 (0.1333 → 0.7333). Similarly, NB Recall improves in PC3 (0.5538 → 0.8154) and PC2 (0.1500 → 0.8500), demonstrating the effectiveness of SMOTE-Tomek in improving minority class detection. While Recall improves considerably, Precision generally decreases, reflecting an increase in false positives after resampling. However, F1-score improves in most cases (e.g., PC3 and PC4 for LR), indicating a more balanced classification performance.

Performance variability is reflected in standard deviation values. Smaller datasets such as PC1, MW1, KC3, and MC1 exhibit higher variability, while larger datasets such as JM1 and PC4 demonstrate more stable performance. This suggests that dataset size influences model stability and generalization. To provide an overall comparison, Table 7 presents the average performance across all datasets and methods.

Table 6. Per-Dataset Results of CONMI+TopK+BE With SMOTE-Tomek (Mean ± SD, 10-fold CV)

Dataset	Classifier	AUC	F1	Precision	Recall
JM1	NB+SMOTE	0.6337±0.0211	0.3137±0.0319	0.4745±0.0430	0.2345±0.0260
	LR+SMOTE	0.6914±0.0188	0.4343±0.0221	0.3583±0.0224	0.5521±0.0272
PC1	NB+SMOTE	0.7990±0.1155	0.3150±0.1681	0.2634±0.1303	0.4167±0.2518
	LR+SMOTE	0.8809±0.0662	0.3901±0.0951	0.2693±0.0617	0.7400±0.2507
KC1	NB+SMOTE	0.6901±0.0638	0.2902±0.0878	0.6056±0.1195	0.1940±0.0684
	LR+SMOTE	0.6968±0.0705	0.3731±0.1006	0.5505±0.1136	0.2859±0.0915
MC1	NB+SMOTE	0.8017±0.0979	0.0969±0.0501	0.0543±0.0283	0.4667±0.2392
	LR+SMOTE	0.8243±0.1004	0.1013±0.0283	0.0545±0.0156	0.7417±0.1766
PC5	NB+SMOTE	0.7324±0.0310	0.2108±0.0533	0.5698±0.1125	0.1310±0.0366
	LR+SMOTE	0.7324±0.0310	0.3542±0.0525	0.5307±0.0492	0.2688±0.0537
MW1	NB+SMOTE	0.8135±0.1109	0.3764±0.1696	0.2675±0.1287	0.6667±0.2887
	LR+SMOTE	0.8240±0.1059	0.4180±0.0919	0.3058±0.0824	0.7333±0.2380
KC3	NB+SMOTE	0.7343±0.1415	0.4890±0.2365	0.6250±0.2795	0.4500±0.2843
	LR+SMOTE	0.7742±0.1346	0.5352±0.1825	0.5333±0.2225	0.5833±0.2528
CM1	NB+SMOTE	0.7391±0.1296	0.3048±0.1978	0.2915±0.2674	0.4000±0.2588
	LR+SMOTE	0.7759±0.0886	0.4104±0.1329	0.3082±0.1091	0.6350±0.1975
PC2	NB+SMOTE	0.8431±0.1381	0.1472±0.0483	0.0809±0.0276	0.8500±0.2291
	LR+SMOTE	0.8728±0.1508	0.1297±0.0709	0.0713±0.0393	0.7500±0.4031
PC3	NB+SMOTE	0.7835±0.0786	0.4021±0.0528	0.2674±0.0369	0.8154±0.1043
	LR+SMOTE	0.8291±0.0540	0.4399±0.0824	0.3154±0.0632	0.7308±0.1252
PC4	NB+SMOTE	0.7919±0.0749	0.3835±0.0705	0.3933±0.0709	0.4101±0.1401
	LR+SMOTE	0.9019±0.0237	0.5475±0.0484	0.4179±0.0531	0.8056±0.0671
MC2	NB+SMOTE	0.7028±0.1358	0.5111±0.1535	0.6583±0.2428	0.4750±0.1861
	LR+SMOTE	0.7034±0.1169	0.5267±0.1325	0.5767±0.1908	0.5400±0.1881

Table 7. Average Performance (Mean ± SD) Across All Methods and Datasets

Method	Classifier	AUC	F1-Score	Precision	Recall
Baseline	NB	0.7371 ± 0.0512	0.2905 ± 0.1044	0.3458 ± 0.1873	0.3880 ± 0.2021
	LR	0.7715 ± 0.0657	0.2571 ± 0.1600	0.4287 ± 0.2276	0.2049 ± 0.1470
	NB+SMOTE	0.7400 ± 0.0480	0.3104 ± 0.1155	0.3410 ± 0.1964	0.5008 ± 0.2181
	LR+SMOTE	0.7631 ± 0.0723	0.3892 ± 0.1435	0.3128 ± 0.1418	0.6153 ± 0.0878
PCC	NB	0.7448 ± 0.0517	0.3048 ± 0.1109	0.3543 ± 0.1739	0.3551 ± 0.1771
	LR	0.7731 ± 0.0687	0.2330 ± 0.1397	0.4195 ± 0.2350	0.1776 ± 0.1149
	NB+SMOTE	0.7469 ± 0.0518	0.3221 ± 0.0953	0.3509 ± 0.1731	0.4441 ± 0.1741
	LR+SMOTE	0.7721 ± 0.0707	0.3871 ± 0.1352	0.3152 ± 0.1442	0.6338 ± 0.0855
MI	NB	0.7434 ± 0.0618	0.2969 ± 0.1185	0.3562 ± 0.1864	0.3638 ± 0.2008
	LR	0.7720 ± 0.0647	0.2456 ± 0.1362	0.4466 ± 0.2125	0.1913 ± 0.1249
	NB+SMOTE	0.7432 ± 0.0624	0.3147 ± 0.1180	0.3464 ± 0.1827	0.4682 ± 0.2022
	LR+SMOTE	0.7670 ± 0.0656	0.3840 ± 0.1444	0.3076 ± 0.1453	0.6237 ± 0.0774
BE	NB	0.7411 ± 0.0604	0.2788 ± 0.1235	0.3479 ± 0.1864	0.3288 ± 0.1930
	LR	0.7681 ± 0.0734	0.1850 ± 0.1560	0.3602 ± 0.2649	0.1362 ± 0.1241
	NB+SMOTE	0.7372 ± 0.0586	0.3342 ± 0.1441	0.2967 ± 0.1702	0.6479 ± 0.1990
	LR+SMOTE	0.7630 ± 0.0724	0.3835 ± 0.1404	0.2950 ± 0.1314	0.6687 ± 0.1018
CONMI+TopK	NB	0.7397 ± 0.0506	0.3014 ± 0.1044	0.3547 ± 0.1764	0.3647 ± 0.1755
	LR	0.7756 ± 0.0622	0.2429 ± 0.1460	0.4476 ± 0.2382	0.1839 ± 0.1191
	NB+SMOTE	0.7420 ± 0.0525	0.3123 ± 0.0963	0.3436 ± 0.1785	0.4525 ± 0.1733
	LR+SMOTE	0.7697 ± 0.0609	0.3911 ± 0.1455	0.3135 ± 0.1478	0.6303 ± 0.0720

Table 7 (Continued)

CONMI+TopK +BE	NB	0.7537 ± 0.0572 0.2870 ± 0.1090 0.3748 ± 0.1597 0.3020 ± 0.1377
	LR	0.7908 ± 0.0706 0.2045 ± 0.1389 0.4313 ± 0.2253 0.1507 ± 0.1140
	NB+SMOTE	0.7554 ± 0.0580 0.3201 ± 0.1196 0.3793 ± 0.1991 0.4592 ± 0.2159
	LR+SMOTE	0.7923 ± 0.0714 0.3884 ± 0.1363 0.3577 ± 0.1679 0.6139 ± 0.1715

4.3. Performance Analysis and Discussion

An implicit ablation analysis can be derived from the comparative results in Table 7, highlighting the contribution of each component in the proposed pipeline. The aggregated results in Table 7 indicate that the proposed CONMI+TopK+BE consistently achieves superior AUC performance across both NB and LR, with and without SMOTE-Tomek. The best performance is obtained when combined with LR and SMOTE-Tomek, demonstrating that integrating multi-filter feature selection with wrapper-based refinement enhances discriminative capability beyond individual feature selection methods. From a component perspective, the results provide clear evidence of the contribution of each stage in the proposed pipeline. PCC and MI, when used independently, yield only moderate improvements, indicating that relying solely on linear or nonlinear dependency measures is insufficient. The CONMI mechanism improves performance by combining both perspectives, producing a more balanced and informative feature ranking. Further improvement is achieved through Top-K selection, which reduces dimensionality while preserving relevant features. Additional improvement is observed when Backward Elimination is applied, as it refines the feature subset by removing statistically insignificant features, thereby reducing noise and improving generalization.

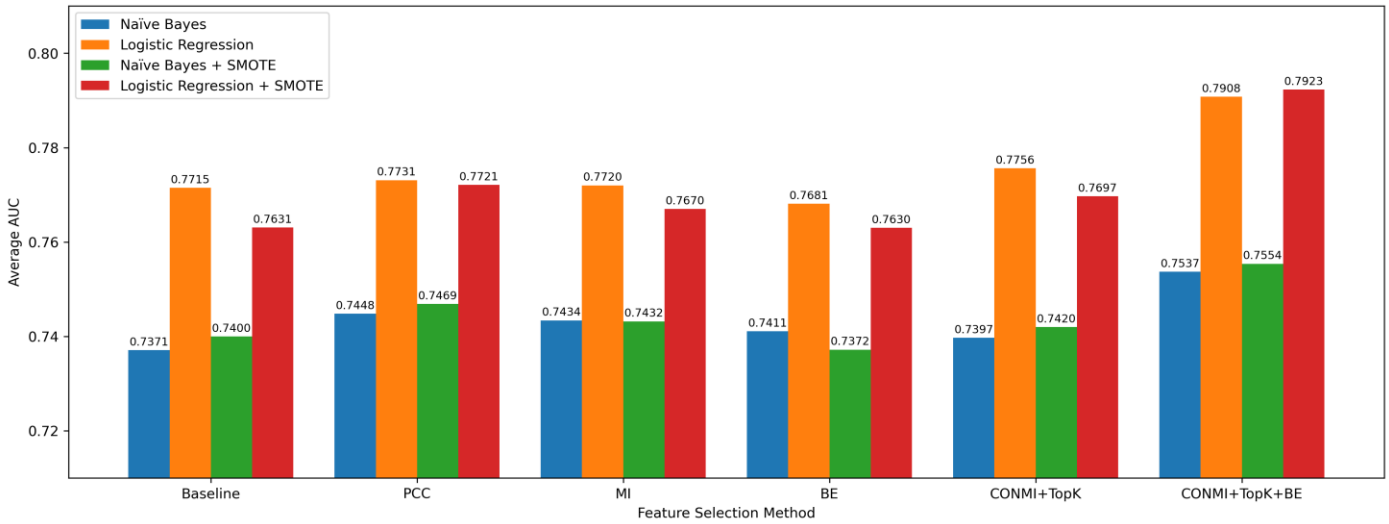

Figure 2. Average AUC Comparison Across Feature Selection Methods (12 NASA MDP Datasets)

A critical observation relates to the role of correlation filtering. Removing this stage results in numerical instability during LR training due to multicollinearity, leading to singular matrix errors. This indicates that correlation filtering is not merely a performance enhancement step but a necessary component for ensuring model stability. A notable trade-off between AUC and Recall is consistently observed. Without SMOTE-Tomek, LR achieves higher AUC but significantly lower Recall, indicating a bias toward majority class prediction. After applying SMOTE-Tomek, Recall improves substantially while AUC remains relatively stable, reinforcing the complementary roles of feature selection and imbalance handling. From a classifier perspective, LR benefits more from the proposed framework than NB. This is consistent with the nature of LR, which is sensitive to feature redundancy and multicollinearity, and therefore gains more from structured feature selection. In contrast, NB assumes conditional independence among features and is less sensitive to feature interactions, resulting in smaller but consistent improvements. These findings suggest that the proposed framework not only improves classification performance but also enhances robustness across different datasets and classifiers. In practical SDP applications, this implies that model configuration

should be aligned with operational objectives—whether prioritizing defect detection (Recall-oriented) or minimizing false positives (Precision-oriented). To further illustrate these findings, Figure 2 presents the average AUC comparison across all methods and classifiers.

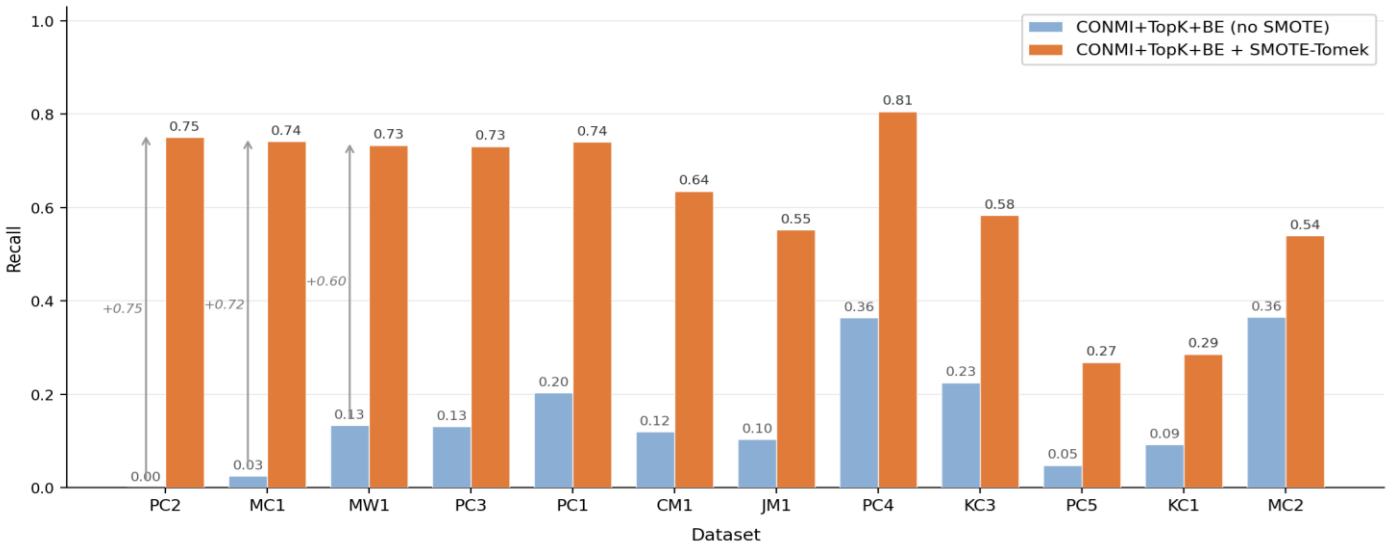

Figure 3. Recall comparison: with vs without smote-tomek per dataset (logistic regression)

Figure 2 shows that CONMI+TopK+BE consistently achieves the highest AUC, with additional improvements when combined with SMOTE-Tomek. Additionally, Figure 3 illustrates the impact of SMOTE-Tomek on Recall for LR. Figure 3 highlights that SMOTE-Tomek significantly improves Recall across most datasets, reinforcing the importance of integrating imbalance handling into the feature selection and classification pipeline.

4.4. Statistical Significance Analysis

To further validate the effectiveness of the proposed method, statistical significance testing was conducted using a paired t-test with AUC as the primary evaluation metric. While additional metrics such as Recall, Precision, and F1-score are reported in Section 4.2 to provide a comprehensive performance perspective, this analysis focuses on AUC to assess overall discriminative capability. The results are summarized in Table 8.

Table 8. Statistical Significance Testing (Paired t-test, $\alpha = 0.05$, AUC-based).

Method Comparison	Classifier	p-value	Significance ($p < 0.05$)
CONMI+TopK+BE+SMOTE vs Baseline+SMOTE	LR	0.0620	Not Significant
CONMI+TopK+BE+SMOTE vs PCC+SMOTE	LR	0.1393	Not Significant
CONMI+TopK+BE+SMOTE vs MI+SMOTE	LR	0.0487	Significant
CONMI+TopK+BE+SMOTE vs BE+SMOTE	LR	0.0123	Significant
CONMI+TopK+BE+SMOTE vs CONMI+TopK+SMOTE	LR	0.0634	Not Significant
CONMI+TopK+BE+SMOTE vs Baseline+SMOTE	NB	0.2482	Not Significant
CONMI+TopK+BE+SMOTE vs PCC+SMOTE	NB	0.1241	Not Significant
CONMI+TopK+BE+SMOTE vs MI+SMOTE	NB	0.3109	Not Significant
CONMI+TopK+BE+SMOTE vs BE+SMOTE	NB	0.1024	Not Significant
CONMI+TopK+BE+SMOTE vs CONMI+TopK+SMOTE	NB	0.0768	Not Significant

The paired t-test results in Table 8 indicate that the proposed CONMI+TopK+BE combined with SMOTE-Tomek achieves consistently higher AUC values compared to all baseline methods across both LR and NB. For LR, statistically significant improvements are observed when compared with MI+SMOTE ($p = 0.0487$) and BE+SMOTE ($p = 0.0123$). These results

suggest that the integration of multi-filter feature selection with wrapper-based refinement provides measurable gains over individual feature selection strategies. However, comparisons with Baseline, PCC, and CONMI+TopK do not reach statistical significance, indicating that while performance improvements are observed, they are not uniformly strong across all comparisons.

In contrast, for NB, none of the comparisons reach statistical significance. This suggests that the proposed feature selection and imbalance handling framework does not produce statistically distinguishable improvements in AUC for this classifier. This behavior is consistent with the independence assumption of NB, which reduces its sensitivity to feature interaction and feature selection refinements.

It is important to note that the absence of statistical significance in AUC does not necessarily imply the absence of practical improvement. As shown in Section 4.2, Recall and F1-score improve substantially under the proposed framework, particularly when combined with SMOTE-Tomek and LR. These improvements are more pronounced than those observed in AUC, indicating that the primary benefit of the proposed approach lies in enhancing minority class detection rather than solely improving ranking performance.

4.5. Comparison with State-of-the-Art Methods

To further assess the effectiveness of the proposed framework, its performance is compared with several prior studies that employ the same NASA MDP datasets. This comparison provides a clearer benchmark of how the proposed method performs relative to existing approaches under comparable experimental settings. The results are summarized in Table 9.

Table 9. Comparison with State-of-the-Art Methods on NASA MDP Datasets (AUC)

Reference	AUC
Suntoro et al. [38]	0.7520
Aryanti et al. [39]	0.7496
Rahmayanti et al. [24]	0.7855
Balogun et al. [23]	0.7460
Proposed Method	0.7923

The results in Table 9 show that the proposed method achieves the highest AUC among the compared approaches. Although the improvement is moderate, it is consistent and indicates that the integration of multi-filter feature selection (CONMI), Top-K reduction, Backward Elimination, and SMOTE-Tomek provides complementary benefits. Prior studies such as Rahmayanti et al. [24] and Balogun et al. [23] primarily focus on hybrid or multi-filter strategies without explicitly addressing multicollinearity or integrating imbalance handling within a unified framework. In contrast, the proposed method combines these components in a structured pipeline, contributing to improved feature quality and model robustness. In addition to AUC, improvements in Recall and F1-score (Section 4.2) further suggest enhanced minority class detection. These results demonstrate that the proposed approach offers a consistent and competitive improvement while maintaining methodological simplicity.

5. Conclusions

This study proposes a structured hybrid framework for SDP that integrates multi-filter feature selection (CONMI), dimensionality reduction (Top-K), correlation-based filtering, and wrapper-based refinement (BE), combined with SMOTE-Tomek for imbalance handling. The experimental results across twelve NASA MDP datasets demonstrate that the proposed approach consistently improves classification performance, achieving the highest average AUC (0.7923 ± 0.0714) when combined with LR and SMOTE-Tomek. Statistical analysis further indicates significant improvements over selected baselines (MI and BE) for LR, while more moderate gains are observed for NB. The findings support the central premise of this study: combining complementary feature selection strategies with imbalance handling leads to more informative, stable, and discriminative feature subsets. In particular, the integration of linear and nonlinear relevance measures (PCC and MI), together with redundancy reduction and wrapper-based refinement, contributes to improved model robustness. Additionally,

the use of SMOTE-Tomek enhances minority class detection, as reflected in improved Recall and F1-score, which are critical in imbalanced SDP scenarios.

From a practical perspective, the proposed framework provides a systematic and interpretable pipeline for handling high-dimensional and imbalanced datasets. This contributes to the field by demonstrating that relatively simple and explainable models, when combined with structured feature selection and data balancing, can achieve competitive performance without relying on more complex or computationally intensive approaches. Nevertheless, this study has several limitations. The evaluation is restricted to NASA MDP datasets and two classical classifiers, which may limit generalizability. The Top-K selection strategy remains heuristic, and alternative imbalance handling techniques or classifier families were not explored.

Future work may focus on adaptive or data-driven feature selection strategies, cost-sensitive learning approaches, and evaluation on cross-project defect prediction settings. Extending the framework to additional classifiers, including ensemble or deep learning models, may further clarify its generalizability and applicability. In summary, the proposed framework offers a consistent and effective approach for SDP by jointly addressing feature relevance, redundancy, and class imbalance within a unified pipeline.

Author Contributions: Conceptualization: M.K.M. and S.W.S.; Methodology: M.K.M.; Software: M.K.M.; Validation: M.K.M., S.W.S., and M.R.F.; Formal analysis: M.K.M.; Investigation: M.K.M.; Resources: S.W.S.; Data curation: M.K.M.; Writing—original draft preparation: M.K.M.; Writing—review and editing: S.W.S., M.R.F., R.A.N., and A.R.; Visualization: M.K.M.; Supervision: S.W.S.; Project administration: S.W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used in this study are publicly available NASA software defect dataset via link : https://github.com/klainfo/NASADefectDataset.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] M. Jorayeva, A. Akbulut, C. Catal, and A. Mishra, "Machine Learning-Based Software Defect Prediction for Mobile Applications: A Systematic Literature Review," *Sensors*, vol. 22, no. 7, p. 2551, Mar. 2022, doi: 10.3390/s22072551.

[2] A. Rahim, Z. Hayat, M. Abbas, A. Rahim, and M. A. Rahim, "Software Defect Prediction with Naïve Bayes Classifier," in *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, Jan. 2021, pp. 293–297. doi: 10.1109/IBCAST51254.2021.9393250.

[3] J. B. Awotunde, S. Misra, A. E. Adeniyi, M. K. Abiodun, M. Kaushik, and M. O. Lawrence, "A Feature Selection-Based K-NN Model for Fast Software Defect Prediction," in *Lecture Notes in Computer Science*, 2022, pp. 49–61. doi: 10.1007/978-3-031-10542-5_4.

[4] M. Nevendra and P. Singh, "A Survey of Software Defect Prediction Based on Deep Learning," *Arch. Comput. Methods Eng.*, vol. 29, no. 7, pp. 5723–5748, Nov. 2022, doi: 10.1007/s11831-022-09787-8.

[5] A. Khalid, G. Badshah, N. Ayub, M. Shiraz, and M. Ghouse, "Software Defect Prediction Analysis Using Machine Learning Techniques," *Sustainability*, vol. 15, no. 6, p. 5517, Mar. 2023, doi: 10.3390/su15065517.

[6] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Inf. Fusion*, vol. 81, pp. 84–90, May 2022, doi: 10.1016/j.inffus.2021.11.011.

[7] R. Wibowo, M. A. Soeleman, and A. Affandy, "Hybrid Top-K Feature Selection to Improve High-Dimensional Data Classification Using Naïve Bayes Algorithm," *Sci. J. Informatics*, vol. 10, no. 2, pp. 113–120, Apr. 2023, doi: 10.15294/sji.v10i2.42818.

[8] Y. Hu *et al.*, "Beyond Comparing Machine Learning and Logistic Regression in Clinical Prediction Modelling: Shifting from Model Debate to Data Quality," *J. Med. Internet Res.*, vol. 27, p. e77721, Nov. 2025, doi: 10.2196/77721.

[9] M. N. Aisy, S. A. Wulandari, and D. R. I. M. Setiadi, "A Probabilistic Feature-Augmented GRU-Attention Model for Chronic Disease Prediction on Imbalanced Data," *J. Futur. Artif. Intell. Technol.*, vol. 2, no. 2, pp. 282–293, Jul. 2025, doi: 10.62411/faith.3048-3719-100.

[10] N. A. A. Khleel and K. Nehéz, "A novel approach for software defect prediction using CNN and GRU based on SMOTE Tomek method," *J. Intell. Inf. Syst.*, vol. 60, no. 3, pp. 673–707, Jun. 2023, doi: 10.1007/s10844-023-00793-1.

[11] D. R. I. M. Setiadi, K. Nugroho, A. R. Muslikh, S. W. Iriananda, and A. A. Ojugo, "Integrating SMOTE-Tomek and Fusion Learning with XGBoost Meta-Learner for Robust Diabetes Recognition," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 1, pp. 23–38, May 2024, doi: 10.62411/faith.2024-11.

[12] E. F. ; Swana *et al.*, "Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset," *Sensors*, vol. 22, no. 9, p. 3246, Apr. 2022, doi: 10.3390/S22093246.
[13] B. Mumtaz, S. Kanwal, S. Alamri, and F. Khan, "Feature Selection Using Artificial Immune Network: An Approach for Software Defect Prediction," *Intell. Autom. Soft Comput.*, vol. 29, no. 3, pp. 669–684, 2021, doi: 10.32604/iasc.2021.018405.
[14] J. Y.-L. Chan *et al.*, "Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review," *Mathematics*, vol. 10, no. 8, p. 1283, Apr. 2022, doi: 10.3390/math10081283.
[15] M. Cuartas, E. Ruiz, D. Ferreño, J. Setién, V. Arroyo, and F. Gutiérrez-Solana, "Machine learning algorithms for the prediction of non-metallic inclusions in steel wires for tire reinforcement," *J. Intell. Manuf.*, vol. 32, no. 6, pp. 1739–1751, Aug. 2021, doi: 10.1007/s10845-020-01623-9.
[16] R. Deng, Y. Liu, L. Luo, D. Chen, and X. Li, "Unsupervised Feature Selection using Pseudo Label Approximation," in *2021 13th International Conference on Machine Learning and Computing*, Feb. 2021, pp. 498–502. doi: 10.1145/3457682.3457758.
[17] M. N. Juybari, P. Baraldi, A. Palermo, A. E. Milani, A. Marzani, and E. Zio, "Wrapper Selection of Features for Fault Diagnostics of Truss Structures," in *Book of Extended Abstracts for the 32nd European Safety and Reliability Conference*, 2022, pp. 1867–1874. doi: 10.3850/978-981-18-5183-4_S02-06-619-cd.
[18] B. T. Pham *et al.*, "Performance assessment of artificial neural network using chi-square and backward elimination feature selection methods for landslide susceptibility analysis," *Environ. Earth Sci.*, vol. 80, no. 20, p. 686, Oct. 2021, doi: 10.1007/s12665-021-09998-5.
[19] J. R. Busenbark, H. (Elle) Yoon, D. L. Gamache, and M. C. Withers, "Omitted Variable Bias: Examining Management Research With the Impact Threshold of a Confounding Variable (ITCV)," *J. Manage.*, vol. 48, no. 1, pp. 17–48, Jan. 2022, doi: 10.1177/01492063211006458.
[20] P. Chen, F. Li, and C. Wu, "Research on Intrusion Detection Method Based on Pearson Correlation Coefficient Feature Selection Algorithm," *J. Phys. Conf. Ser.*, vol. 1757, no. 1, p. 012054, Jan. 2021, doi: 10.1088/1742-6596/1757/1/012054.
[21] F. Macedo, R. Valadas, E. Carrasquinha, M. R. Oliveira, and A. Pacheco, "Feature selection using Decomposed Mutual Information Maximization," *Neurocomputing*, vol. 513, pp. 215–232, Nov. 2022, doi: 10.1016/j.neucom.2022.09.101.
[22] B. M. Kessels, R. H. B. Fey, and N. van de Wouw, "Mutual information-based feature selection for inverse mapping parameter updating of dynamical systems," *Multibody Syst. Dyn.*, vol. 64, no. 3, pp. 437–464, Jul. 2025, doi: 10.1007/s11044-024-10015-3.
[23] A. O. Balogun *et al.*, "Empirical Analysis of Rank Aggregation-Based Multi-Filter Feature Selection Methods in Software Defect Prediction," *Electronics*, vol. 10, no. 2, p. 179, Jan. 2021, doi: 10.3390/electronics10020179.
[24] R. Rahmayanti, R. Herteno, S. W. Saputro, T. H. Saragih, and F. Abadi, "Comparative Study of Filter, Wrapper, and Hybrid Feature Selection Using Tree-Based Classifiers for Software Defect Prediction," *Indones. J. Electron. Electromed. Eng. Med. Informatics*, vol. 8, no. 1, pp. 1–16, Dec. 2025, doi: 10.35882/ijeeemi.v8i1.294.
[25] Muhammad Noor, Radityo Adi Nugroho, Setyo Wahyu Saputro, Rudy Herteno, and Friska Abadi, "Optimization of Backward Elimination for Software Defect Prediction with Correlation Coefficient Filter Method," *J. Electron. Electromed. Eng. Med. Informatics*, vol. 6, no. 4, pp. 397–404, Sep. 2024, doi: 10.35882/jeeemi.v6i4.466.
[26] A. O. Balogun *et al.*, "Software Defect Prediction Using Wrapper Feature Selection Based on Dynamic Re-Ranking Strategy," *Symmetry (Basel).*, vol. 13, no. 11, p. 2166, Nov. 2021, doi: 10.3390/sym13112166.
[27] M. N. M. Rahman, R. A. Nugroho, M. R. Faisal, F. Abadi, and R. Herteno, "Optimized multi correlation-based feature selection in software defect prediction," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 22, no. 3, p. 598, Jun. 2024, doi: 10.12928/telkomnika.v22i3.25793.
[28] M. Y. A. Pratama, R. Herteno, M. R. Faisal, R. A. Nugroho, and F. Abadi, "Improving with Hybrid Feature Selection in Software Defect Prediction," *J. Online Inform.*, vol. 9, no. 1, pp. 52–60, Apr. 2024, doi: 10.15575/join.v9i1.1307.
[29] M. I. Akazue, I. A. Debekeme, A. E. Edje, C. Asuai, and U. J. Osame, "UNMASKING FRAUDSTERS: Ensemble Features Selection to Enhance Random Forest Fraud Detection," *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 201–211, Dec. 2023, doi: 10.33633/jcta.v1i2.9462.
[30] C. Asuai *et al.*, "Enhancing DDoS Detection via 3ConFA Feature Fusion and 1D Convolutional Neural Networks," *J. Futur. Artif. Intell. Technol.*, vol. 2, no. 1, pp. 145–162, Jun. 2025, doi: 10.62411/faith.3048-3719-105.
[31] J. Zhang, D. Li, W. E. Wong, and S. Wang, "A Hybrid Sampling and Multi-Objective Optimization Approach for Enhanced Software Defect Prediction," *arXiv*. Oct. 13, 2024. [Online]. Available: http://arxiv.org/abs/2410.10046
[32] M. S. Masari, M. A. Danladi, I. L. Onyinye, and L. K. Tohomdet, "Android Malware Detection Using Machine Learning with SMOTE-Tomek Data Balancing," *J. Comput. Theor. Appl.*, vol. 3, no. 3, pp. 302–313, Jan. 2026, doi: 10.62411/jcta.15084.
[33] T. Wahyuningsih, D. Manongga, I. Sembiring, and S. Wijono, "Comparison of Effectiveness of Logistic Regression, Naive Bayes, and Random Forest Algorithms in Predicting Student Arguments," *Procedia Comput. Sci.*, vol. 234, pp. 349–356, 2024, doi: 10.1016/j.procs.2024.03.014.
[34] S. K, J. V. K, H. S, and K. V, "Defect Prediction Model for Software Projects using Naïve Bayesian Classifier," *Int. J. Eng. Trends Technol.*, vol. 71, no. 9, pp. 170–177, Sep. 2023, doi: 10.14445/22315381/IJETT-V71I9P216.
[35] R. Oueslati and G. Manita, "Software Defect Prediction Using Integrated Logistic Regression and Fractional Chaotic Grey Wolf Optimizer," in *Proceedings of the 19th International Conference on Evaluation of Novel Approaches to Software Engineering*, 2024, pp. 633–640. doi: 10.5220/0012704600003687.
[36] H. Gong, Y. Li, J. Zhang, B. Zhang, and X. Wang, "A new filter feature selection algorithm for classification task by ensembling pearson correlation coefficient and mutual information," *Eng. Appl. Artif. Intell.*, vol. 131, p. 107865, May 2024, doi: 10.1016/j.engappai.2024.107865.
[37] H. Zhou, X. Wang, and R. Zhu, "Feature selection based on mutual information with correlation coefficient," *Appl. Intell.*, vol. 52, no. 5, pp. 5457–5474, Mar. 2022, doi: 10.1007/s10489-021-02524-x.

[38] J. Suntoro, F. W. Christanto, and H. Indriyawati, "Software Defect Prediction Using AWEIG+ADACOST Bayesian Algorithm for Handling High Dimensional Data and Class Imbalance Problem," *Int. J. Inf. Technol. Bus.*, vol. 5, no. 1, pp. 27–32, Nov. 2022, doi: 10.24246/ijiteb.512018.27-32.

[39] A. K. Aryanti, R. Herteno, F. Indriani, R. A. Nugroho, and M. Muliadi, "Implementation of Copeland Method on Wrapper-Based Feature Selection Using Random Forest For Software Defect Prediction," *Indones. J. Electron. Electromed. Eng. Med. Informatics*, vol. 7, no. 1, pp. 90–101, Feb. 2025, doi: 10.35882/2pgffc67.