International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
Article Received: *25 April 2023* ***Revised:*** *12 June 2023* ***Accepted:*** *30 July 2023*

BUGOPTIMIZE: Bugs dataset Optimization with Majority Vote Cluster-Based Fine-Tuned Feature Selection for Scalable Handling

Sayyed Jasmin Isahak
Research Scholar, Department of Computer Science and Engineering
Dr. A. P. J. Abdul Kalam University, Indore, MP,India
shaikh.jasmin.n@gmail.com

Dr. Manoj Eknath Patil
Research Supervisor, Department of Computer Science and Engineering
Dr. A. P. J. Abdul Kalam University, Indore, MP,India
mepatil@gmail.com

Abstract— Software bugs are prevalent in the software development lifecycle, posing challenges to developers in ensuring product quality and reliability. Accurate prediction of bug counts can significantly aid in resource allocation and prioritization of bug-fixing efforts. However, the vast number of attributes in bug datasets often requires effective feature selection techniques to enhance prediction accuracy and scalability. Existing feature selection methods, though diverse, suffer from limitations such as suboptimal feature subsets and lack of scalability. This paper proposes BUGOPTIMIZE, a novel algorithm tailored to address these challenges. BUGOPTIMIZE innovatively integrates majority voting cluster-based fine-tuned feature selection to optimize bug datasets for scalable handling and accurate prediction. The algorithm initiates by clustering the dataset using K-means, EM, and Hierarchical clustering algorithms and performs majority voting to assign data points to final clusters. It then employs filter-based, wrapper-based, and embedded feature selection techniques within each cluster to identify common features. Additionally, feature selection is applied to the entire dataset to extract another set of common features. These selected features are combined to form the final best feature set. Experimental results demonstrate the efficacy of BUGOPTIMIZE compared to existing feature selection methods, reducing MAE and RMSE in Linear Regression (MAE: 0.2668 to 0.2609, RMSE: 0.3251 to 0.308) and Random Forest (MAE: 0.1626 to 0.1341, RMSE: 0.2363 to 0.224), highlighting its significant contribution to bug dataset optimization and prediction accuracy in software development while addressing feature selection limitations. By mitigating the disadvantages of current approaches and introducing a comprehensive and scalable solution, BUGOPTIMIZE presents a significant advancement in bug dataset optimization and prediction accuracy in software development environments.

Keywords- *Software bugs, bug counts prediction, clustering, feature selection and majority voting*.

I. INTRODUCTION

Software bugs, those pesky anomalies lurking within the intricate software development framework, stand as an enduring challenge that casts significant shadows over product quality, reliability, and, ultimately, user satisfaction [1]. In an era where the complexity of software systems continues to soar to unprecedented heights, identifying and mitigating these bugs have become not just tasks but rather imperative missions for developers. As software becomes increasingly intertwined with various facets of daily life, from communication to commerce, the ramifications of unchecked bugs can ripple far and wide, underscoring the urgent need for diligent bug management strategies and practices.

Accurately predicting bug counts is a pivotal component of effective bug management, facilitating teams to allocate resources efficiently and prioritize their bug-fixing endeavours [2]. Nonetheless, this undertaking is far from trivial, primarily due to the myriad factors that influence the occurrence of bugs and the substantial volume of data encompassed within bug datasets. From the intricacies of software architecture to user interactions, numerous variables contribute to the unpredictable nature of bug manifestation, rendering precise forecasting a formidable challenge for software development teams. Despite advancements in data analysis techniques and machine learning algorithms, navigating through the complexity of bug prediction remains a

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
***Article Received:** 25 April 2023 **Revised:** 12 June 2023 **Accepted:** 30 July 2023*

nuanced endeavour, demanding continuous refinement and adaptation to evolving software landscapes.

In addressing the imperative of precise bug count prediction, proficient feature selection techniques become paramount. Feature selection serves as the linchpin in this endeavour, striving to discern the most pertinent and enlightening attributes within a dataset, thereby streamlining dimensionality and fortifying the efficacy of predictive models [3]. By meticulously selecting the most salient features, developers can distil the essence of the data, enabling algorithms to discern meaningful patterns and correlations with heightened accuracy. Consequently, feature selection optimizes computational resources and augments models' predictive prowess, furnishing invaluable insights that underpin informed decision-making in software development endeavours.

A diverse array of existing techniques for feature selection includes filter-based, wrapper-based, and embedded methods, each offering distinctive approaches to discerning feature relevance and enhancing prediction accuracy. While these methods provide invaluable insights into the significance of features and contribute substantially to refining predictive models, they are not immune to limitations. Filter-based techniques, for instance, prioritize feature selection independent of the chosen learning algorithm, facilitating efficient data preprocessing but potentially overlooking intricate relationships among features [4]. On the other hand, wrapper-based methods tailor feature selection to the specific predictive model employed, thereby capturing more nuanced interactions among variables but often incurring higher computational costs [5]. Embedded methods seamlessly integrate feature selection into the model construction process, optimizing both predictive accuracy and computational efficiency; however, they may lack the flexibility to accommodate diverse modelling techniques [6]. Despite these constraints, existing feature selection techniques represent indispensable tools in the arsenal of software developers, offering invaluable insights into data analysis and model refinement.

Existing feature selection techniques, while invaluable in enhancing predictive accuracy, are not without their drawbacks. One common limitation lies in the propensity for these methods to yield suboptimal feature subsets, potentially overlooking crucial attributes or retaining irrelevant ones, thus impeding the efficacy of predictive models [7]. Moreover, scalability issues often arise, particularly with large-scale datasets, as the computational demands of feature selection techniques may become prohibitively burdensome, hampering their practical applicability in real-world scenarios. Additionally, the limited adaptability of these techniques to diverse datasets and problem domains poses a significant challenge, as the efficacy of feature selection may vary depending on the specific characteristics and nuances inherent to different data contexts. Consequently, while existing feature selection methods represent valuable tools in refining predictive models, addressing these disadvantages is crucial for advancing the field and fostering more robust and versatile approaches to feature selection in software development and data analysis contexts.

The inadequacies of current feature selection methodologies highlight the demand for a fresh approach that tackles these issues holistically. Consequently, the emergence of the BUGOPTIMIZE algorithm directly responds to this imperative, offering a groundbreaking solution designed to address the inherent limitations of existing techniques. By focusing on optimizing bug datasets for both scalability and precision in prediction, BUGOPTIMIZE represents a paradigm shift in feature selection methodologies within the domain of software development. This innovative algorithm promises to revolutionize bug management practices by providing developers with a robust tool that enhances the efficiency of bug handling and ensures more accurate bug count predictions, thus advancing state-of-the-art software quality assurance.

At the heart of its design, the BUGOPTIMIZE algorithm harnesses a sophisticated blend of cutting-edge methodologies, prominently featuring a unique approach known as majority voting cluster-based fine-tuned feature selection. This innovative strategy amalgamates various clustering algorithms and feature selection techniques, offering a comprehensive solution to circumvent the shortcomings that plague conventional methods. By leveraging the collective insights garnered from multiple clustering algorithms and fine-tuning feature selection through a majority voting framework, BUGOPTIMIZE introduces a novel paradigm in bug management and prediction. This multifaceted approach enhances the robustness of bug dataset optimization. It augments the accuracy of bug count predictions, laying the groundwork for more effective and efficient bug-handling practices in software development workflows.

The functionality of the BUGOPTIMIZE algorithm initiates with the clustering of the dataset, employing a diverse range of clustering algorithms to partition the data into distinct clusters. Implementing a majority voting mechanism allows data points to be assigned to their final clusters, ensuring a robust clustering outcome. Following this phase, a comprehensive suite of feature selection techniques is meticulously applied within each clustered group. This process identifies and extracts the most pertinent and informative features shared among data points within each cluster. By integrating these two pivotal steps—clustering and feature

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
Article Received: *25 April 2023* ***Revised:*** *12 June 2023* ***Accepted:*** *30 July 2023*

selection—BUGOPTIMIZE optimizes bug datasets with a precision surpassing conventional methodologies, thereby enhancing bug management and prediction efficacy in software development endeavours.

The distinctive aspect of the BUGOPTIMIZE algorithm lies in its holistic approach to feature selection, offering a comprehensive and scalable solution tailored specifically for bug dataset optimization and prediction accuracy within software development environments. Unlike conventional methods, BUGOPTIMIZE adopts a multifaceted strategy that seamlessly integrates clustering algorithms with sophisticated feature selection techniques, thereby addressing the limitations of existing approaches. This innovative fusion enhances the robustness of bug dataset optimization and significantly improves prediction accuracy, ultimately revolutionizing bug management practices. By embracing a holistic perspective, BUGOPTIMIZE empowers developers with a powerful toolset to navigate the complexities of software development, facilitating more efficient bug handling and fostering higher-quality software products.

The primary objective of this paper is to introduce the BUGOPTIMIZE algorithm, provide a comprehensive understanding of its methodology, validate its effectiveness through rigorous experimental evaluation, and conduct comparative analyses with existing feature selection methods. By meticulously detailing the inner workings of BUGOPTIMIZE and showcasing its performance in empirical studies, this paper seeks to make significant contributions to bug management and prediction in software development. The foremost contribution lies in the introduction of BUGOPTIMIZE, a pioneering algorithm specifically designed for bug dataset optimization. Through this novel approach, developers can expect to streamline bug-handling processes and enhance prediction accuracy, thus advancing state-of-the-art software quality assurance.

This paper aims to give developers and researchers a valuable tool to enhance bug count prediction accuracy within software development projects. It aspires to bridge the gap between theoretical research and practical application by introducing the BUGOPTIMIZE algorithm as a reliable solution. The scope of this paper extends across diverse software development environments where precise bug count prediction holds paramount importance, encompassing industries ranging from web development to enterprise software engineering. By delineating the potential applications and benefits of BUGOPTIMIZE, this paper aims to empower stakeholders within the software development community to adopt more effective bug management strategies, fostering higher-quality software products and improving overall project outcomes.

The organization of this paper is structured as follows: Section 2 provides an overview of existing feature selection techniques. Section 3 presents the methodology of the BUGOPTIMIZE algorithm. Section 4 discusses experimental results and comparative analysis of the BUGOPTIMIZE algorithm. Finally, Section 5 concludes the paper with a summary of key findings and directions for future research.

II. RELATED WORKS

Several studies have explored domain feature selection techniques to enhance predictive model performance.

Alazzam et al. [8] proposed a novel feature selection algorithm that draws inspiration from the intricate behaviours observed in pigeon populations, leveraging insights from avian flocking dynamics to enhance intrusion detection systems (IDS) efficacy. In their approach, pigeons' collective behaviours and adaptive strategies in navigating complex environments serve as a metaphor for optimizing the selection of pertinent features within IDS datasets. By harnessing principles inspired by avian flocking, the algorithm dynamically adjusts feature subsets, prioritizing those that exhibit patterns akin to the synchronized movements and cohesive decision-making observed in pigeon flocks. This innovative approach offers a unique perspective on feature selection, aiming to improve the detection capabilities of IDS by identifying and prioritizing relevant attributes that contribute to the accurate identification of intrusive activities amidst diverse and evolving network environments.

Meanwhile, Zhou et al. [9] focused on enhancing the efficiency of intrusion detection systems (IDS) by integrating advanced feature selection techniques and ensemble classifiers. Their research aimed to streamline the detection process by strategically selecting discriminative features from network traffic data, thus reducing computational overhead while preserving critical information for accurate intrusion detection. Leveraging ensemble classifiers, they sought to capitalize on the diverse perspectives of multiple base classifiers, aggregating outputs to improve detection accuracy and robustness against various intrusions and network anomalies. Zhou et al. endeavoured to equip cybersecurity professionals with sophisticated tools capable of swiftly and accurately identifying and neutralizing threats in complex network environments through this synergistic approach.

Ghosh et al. [10] conducted a comprehensive investigation into the prediction of cardiovascular disease, employing advanced machine learning algorithms augmented with Relief and LASSO feature selection techniques. Their research focused on harnessing the power of machine learning to develop predictive models capable of identifying

individuals at risk of cardiovascular ailments with high accuracy and reliability. By leveraging the Relief and LASSO feature selection methods, which are renowned for their ability to identify informative features while mitigating the effects of multicollinearity and overfitting, Ghosh et al. aimed to enhance the predictive performance of their models while ensuring interpretability and generalizability. Through meticulous experimentation and validation, their study sought to contribute valuable insights into the early detection and prevention of cardiovascular diseases, paving the way for developing more effective diagnostic and risk assessment tools in clinical practice.

Similarly, Wang et al. [11] embarked on pioneering research to combat Distributed Denial of Service (DDoS) attacks through the development of a dynamic detection method leveraging feature selection and feedback mechanisms. Their innovative approach aimed to address the ever-evolving nature of DDoS threats by dynamically adapting detection strategies based on real-time feedback from network traffic patterns. By integrating feature selection techniques, Wang et al. sought to identify key indicators of malicious activity within the network data while reducing computational complexity and false positive rates. Furthermore, incorporating feedback mechanisms enabled its detection system to continuously learn and refine its detection capabilities, enhancing its resilience against emerging DDoS attack vectors. Through rigorous experimentation and validation, their study contributed significant advancements in the field of DDoS mitigation, offering a robust framework for the proactive detection and mitigation of malicious network activities.

In addition to these approaches, Liu and Setiono [12] introduced a groundbreaking probabilistic wrapper approach to revolutionize feature selection and classification methodologies. Their innovative method represents a significant departure from traditional techniques by leveraging probabilistic reasoning to guide the feature selection process. By incorporating probabilistic models into the wrapper framework, Liu and Setiono sought to enhance the robustness and reliability of feature selection algorithms, enabling them to better adapt to the inherent uncertainties and complexities present in real-world datasets. This approach offers a principled and systematic means of evaluating feature subsets based on their likelihood of improving classification performance, thus enabling more informed and data-driven decision-making. Through empirical validation and comparative analysis, Liu and Setiono demonstrated the effectiveness and versatility of their probabilistic wrapper approach across diverse domains, paving the way for advancements in feature selection and classification methodologies.

El-Hasnony et al. [13] pioneered big data analytics with their enhanced feature selection model proposal. Recognizing the burgeoning importance of efficient data handling in big data, their model aimed to streamline the feature selection process, a critical step in data preprocessing and analysis. By leveraging advanced algorithms and techniques tailored specifically for large-scale datasets, El-Hasnony et al. sought to address the unique challenges posed by the volume, velocity, and variety of big data sources. Their model introduced novel methodologies for identifying and prioritizing relevant features, thus enabling more accurate and efficient data analysis while mitigating the computational burden associated with traditional feature selection approaches. Through rigorous experimentation and validation on real-world big data sets, their study provided compelling evidence of the efficacy and scalability of their proposed feature selection model, offering valuable insights and practical solutions for organizations grappling with the complexities of big data analytics.

Shafiq et al. [14] pioneered to tackle the pressing issue of identifying malicious traffic in Internet of Things (IoT) environments, employing sophisticated wrapper-based feature selection mechanisms. Recognizing the unique challenges posed by the interconnected nature and vast scale of IoT networks, their research aimed to develop robust methodologies for distinguishing between legitimate and malicious traffic patterns. By leveraging wrapper-based feature selection techniques, Shafiq et al. sought to identify the most discriminative and relevant features within IoT network data, thereby enhancing the effectiveness of intrusion detection and mitigation efforts. Their approach offered a systematic and data-driven framework for selecting features that exhibit strong discriminatory power, enabling more accurate and efficient identification of anomalous behaviour indicative of security threats in IoT environments. Through empirical validation and experimentation on real-world IoT datasets, Shafiq et al. provided compelling evidence of the efficacy and practical applicability of their proposed feature selection mechanisms, underscoring their significance in enhancing the security and resilience of IoT ecosystems against emerging cyber threats.

Chen et al. [15] directed their research efforts towards a pivotal aspect of data analysis and classification by concentrating on selecting critical features through machine learning methods. Recognizing the fundamental role feature selection plays in the accuracy and efficiency of classification tasks, their study aimed to identify the subset of features most informative and relevant for distinguishing between different classes within a dataset. By leveraging a diverse array of machine learning algorithms and techniques, Chen et al. sought to systematically evaluate and prioritize features based

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
Article Received: *25 April 2023* ***Revised:*** *12 June 2023* ***Accepted:*** *30 July 2023*

on their discriminative power and contribution to the classification process. Their approach offered a moral and data-driven framework for feature selection, enabling more effective and interpretable classification models while mitigating the curse of dimensionality and improving computational efficiency. Through empirical validation and comparative analysis across various datasets and classification tasks, Chen et al. demonstrated the effectiveness and versatility of their feature selection methodologies, highlighting their significance in enhancing the performance and interpretability of machine learning models in diverse application domains.

Rostami et al. [16] pioneered research to enhance the feature selection process for medical datasets by integrating multi-objective Particle Swarm Optimization (PSO) techniques with node centrality measures. Recognizing the critical importance of feature selection in medical data analysis for disease diagnosis and prognosis tasks, their study sought to develop a robust framework to effectively identify the most informative and relevant features while accounting for multiple conflicting objectives. By leveraging PSO-based optimization algorithms, Rostami et al. aimed to efficiently search the feature space to identify subsets that optimize various criteria simultaneously, such as classification accuracy, model interpretability, and computational efficiency. Additionally, their integration of node centrality measures into the feature selection process aimed to leverage the inherent structural properties of medical datasets, further enhancing the discriminative power of selected features. Through rigorous experimentation and validation on real-world medical datasets, Rostami et al. provided compelling evidence of their proposed methodology's efficacy and practical applicability, highlighting its potential to improve the accuracy and interpretability of predictive models in medical decision-making scenarios.

Agrawal et al. [17] introduced a novel and innovative approach to feature selection by proposing S-shaped and V-shaped gaining-sharing knowledge-based algorithms. Recognizing the critical role of feature selection in optimizing predictive model performance and interpretability, their research aimed to develop algorithms that leverage domain knowledge and expertise to guide the selection process. The S-shaped and V-shaped gaining-sharing algorithms were designed to dynamically adjust feature subsets based on their contributions to predictive accuracy and model stability. Agrawal et al. sought to enhance the robustness and generalizability of feature selection methodologies across diverse application domains by incorporating insights from knowledge sharing and gain distribution principles. Through empirical validation and comparative analysis, their study demonstrated the effectiveness and versatility of the proposed algorithms in improving predictive model performance while offering valuable insights into the underlying mechanisms driving feature selection decisions. Overall, introducing the S-shaped and V-shaped gaining-sharing knowledge-based algorithms represents a significant advancement in feature selection, offering promising avenues for enhancing the efficiency and efficacy of predictive modelling tasks in various domains.

Despite the advancements in feature selection techniques, existing approaches exhibit several limitations. Common disadvantages include suboptimal feature subsets, scalability issues, limited adaptability to diverse datasets and problem domains, and dependency on specific domain knowledge or heuristics. These shortcomings underscore the need for a comprehensive and scalable solution to address these challenges effectively.

The BUGOPTIMIZE algorithm is specifically designed to tackle the limitations of existing feature selection techniques. By integrating innovative methodologies such as majority voting cluster-based fine-tuned feature selection, BUGOPTIMIZE offers a holistic approach to feature selection that overcomes the drawbacks of current methods. The algorithm aims to provide developers and researchers with a robust tool for optimizing bug datasets for scalable handling and accurate prediction, thereby advancing the state-of-the-art in bug count prediction and software development practices.

III. METHODOLOGY OF THE BUGOPTIMIZE ALGORITHM

The BUGOPTIMIZE algorithm represents an innovative approach tailored to enhance the management of bug datasets, prioritizing scalability and precision in predictive analysis. By integrating sophisticated clustering methodologies with fine-tuned feature selection techniques, BUGOPTIMIZE offers a comprehensive framework for optimizing bug datasets. The algorithm's methodology unfolds across multiple sequential steps, each meticulously crafted to systematically identify and extract the most relevant features from the input dataset. Through this iterative process, BUGOPTIMIZE aims to distil the complex bug dataset into a refined subset of attributes most indicative of bug occurrence and severity. By prioritizing informative features while ensuring scalability, the algorithm empowers developers and researchers with a robust tool for accurate bug prediction and efficient resource allocation. Algorithm 1 shows the proposed BUGOPTIMIZE algorithm.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
Article Received: *25 April 2023* ***Revised:*** *12 June 2023* ***Accepted:*** *30 July 2023*

Algorithm 1: BUGOPTIMIZE: Bugs dataset Optimization with Majority Vote Cluster-Based Fine-Tuned Feature Selection for Scalable Handling

Input : Bugs Dataset D, Number of clusters k
Output : Final set of best features
Step 1 : Cluster the dataset D using K-means, EM, and Hierarchical clustering algorithms.
Step 2 : Obtain cluster assignments for each data point from each clustering algorithm.
Step 3 : Perform majority voting to assign each data point to a final cluster.
Step 4 : Initialize empty arrays: best_features_Subset1[], best_features_Subset2[]
Step 5 : For each cluster C:
a. Apply filter-based selection for cluster C using CorrelationAttributeEval and Ranker search.
b. Apply wrapper-based selection for cluster C using ClassifierSubsetEval and GreedyStepwise search.
c. Apply embedded selection for cluster C using ReliefFAttributeEval and Ranker search.
d. Identify common features among the selected features of the three feature selection techniques.
e. Put identified common features into the best_features_Subset1 array.
Step 6 : Apply filter-based, wrapper-based, and embedded selection for the entire dataset D.
Step 7 : Identify common features among the selected features of the three feature selection techniques.
Step 8 : Put identified common features into the best_features_Subset2 array.
Step 9 : Extract the final best features from both the best_features_Subset1 array and best_features_Subset2 array.
Step 10 : Combine the best features obtained from step 9.
Step 11 : Extract a dataset consisting only of these selected best features.
Step 12 : Output the final set of best features.

The BUGOPTIMIZE algorithm systematically optimizes bug datasets for scalable handling and accurate prediction by leveraging a sophisticated combination of clustering and fine-tuned feature selection techniques. Initially, the algorithm clusters the input bug dataset using K-means, EM, and Hierarchical clustering algorithms and then assigns each data point to a final cluster through majority voting. Subsequently, within each cluster, the algorithm applies filter-based, wrapper-based, and embedded feature selection methods to identify common features among the subsets obtained. Furthermore, feature selection is conducted on the entire dataset to ensure comprehensive coverage of important attributes. The algorithm then extracts the final set of best features by combining common features identified from cluster-specific and dataset-wide selections. Finally, a new dataset comprising these selected features is generated, culminating in the output of the final set of best features optimized for bug prediction in software development environments. Figure 1 shows the system architecture of the BUGOPTIMIZE algorithm.

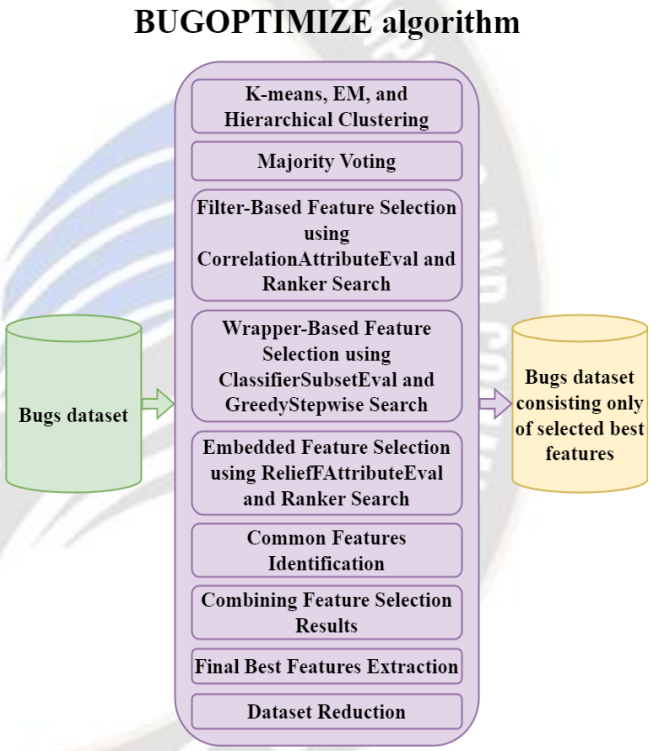

Figure 1: System architecture of the BUGOPTIMIZE algorithm

3.1 Dataset Clustering

In the initial phase of the BUGOPTIMIZE algorithm, the dataset undergoes a comprehensive clustering process utilizing three prominent algorithms: K-means, Expectation-Maximization (EM), and Hierarchical clustering. This step is pivotal as it partitions the input bug dataset into distinct clusters based on the inherent similarities among data points, thereby capturing intricate patterns and structures embedded within the dataset. By applying these diverse clustering techniques, BUGOPTIMIZE endeavours to ensure robust and comprehensive clustering, enabling subsequent analyses to

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
Article Received: *25 April 2023* ***Revised:*** *12 June 2023* ***Accepted:*** *30 July 2023*

effectively capture the underlying dynamics and relationships within the bug dataset.

3.1.1 K-means clustering

In the BUGOPTIMIZE algorithm, the K-means clustering technique is pivotal in partitioning the bug dataset into clusters. Leveraging the capabilities of the WEKA tool, the K-means algorithm iteratively assigns data points to clusters based on their proximity to the cluster centroids. By minimizing the within-cluster sum of squared distances, K-means effectively identifies clusters with compact and well-separated boundaries, facilitating extracting meaningful insights from the bug dataset.

3.1.2 Expectation-Maximization (EM) Clustering

Utilizing the Expectation-Maximization (EM) algorithm within BUGOPTIMIZE enables the identification of latent structures within the bug dataset. Through the iterative application of the EM algorithm, the WEKA tool probabilistically assigns data points to clusters based on the likelihood of their membership. By iteratively refining cluster parameters to maximize the probability of the observed data, EM clustering accommodates datasets with complex distributions and overlapping clusters, enhancing the algorithm's ability to capture nuanced patterns and relationships present in bug datasets.

3.1.3 Hierarchical clustering

The Hierarchical clustering technique employed in the BUGOPTIMIZE algorithm facilitates the creation of a hierarchical representation of clusters within the bug dataset. Leveraging the hierarchical clustering capabilities of the WEKA tool, this technique recursively merges data points into clusters based on their pairwise distances or similarities. By constructing a dendrogram that illustrates the hierarchical relationships between clusters, Hierarchical clustering offers insights into the nested structures and subgroupings present within the bug dataset, aiding in the identification of clusters at varying levels of granularity.

3.2 Majority Voting for Cluster Assignment

In the subsequent stage of the BUGOPTIMIZE algorithm, termed Majority Voting for Cluster Assignment, each data point undergoes assignment to a definitive cluster through a consensus-based mechanism. This process involves aggregating cluster assignments derived from the outcomes of K-means, EM, and Hierarchical clustering algorithms, leveraging the diversity of clustering approaches to counteract potential biases and uncertainties inherent in individual algorithms. The algorithm aims to achieve robust and reliable cluster assignments by employing a majority voting scheme, thereby laying a strong foundation for subsequent analyses. Through this approach, BUGOPTIMIZE seeks to enhance the accuracy and stability of cluster assignments, enabling more effective extraction of insights and patterns from the bug dataset.

3.3 Fine-tuned Feature Selection within Clusters

Within each cluster identified through the clustering process, the BUGOPTIMIZE algorithm fine-tuned feature selection to identify the most informative attributes. This step is crucial for optimizing the model's predictive performance and reducing the dataset's dimensionality. To achieve this, the algorithm employs a comprehensive suite of feature selection techniques, encompassing filter-based, wrapper-based, and embedded methods, leveraging the capabilities of the WEKA tool.

3.3.1 Filter-based Feature Selection

In the filter-based feature selection phase, the algorithm evaluates the relevance of individual features by employing correlation-based evaluation measures such as CorrelationAttributeEval, which are available in the WEKA tool. This approach quantifies the correlation between each attribute and the target variable, enabling the identification of attributes that exhibit strong predictive power. Additionally, search methods like Ranker provided by WEKA rank feature based on their correlation scores, allowing the algorithm to prioritize the most relevant attributes for inclusion in the final feature subset.

3.3.2 Wrapper-based Feature Selection

Wrapper-based feature selection techniques are employed to refine feature subsets within each cluster further using the WEKA tool. This approach involves the iterative evaluation of feature subsets using a specific classifier, such as ClassifierSubsetEval, available in WEKA, combined with a search method like GreedyStepwise. By iteratively adding or removing features from the subset and evaluating their impact on model performance, wrapper-based selection aims to identify the optimal feature subset that maximizes predictive accuracy. The GreedyStepwise search method systematically explores the space of feature subsets, selecting or deselecting features based on their contribution to model performance.

3.3.3 Embedded Feature Selection

Embedded feature selection techniques, such as ReliefFAttributeEval provided by the WEKA tool, are utilized within each cluster to refine the feature subset further. Unlike filter-based and wrapper-based methods, embedded selection evaluates feature relevance within the context of the classification model itself. ReliefFAttributeEval, for example, assesses the importance of features by considering their relevance to the classification task while accounting for feature interactions. This technique identifies features that contribute most significantly to the model's predictive accuracy and ranks them accordingly using Ranker search methods available in WEKA.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
Article Received: *25 April 2023* ***Revised:*** *12 June 2023* ***Accepted:*** *30 July 2023*

By leveraging a combination of filter-based, wrapper-based, and embedded feature selection techniques within each cluster using the WEKA tool, the BUGOPTIMIZE algorithm effectively identifies the most informative attributes for bug prediction. This comprehensive approach ensures that the final feature subset is optimized for predictive accuracy and generalization across different clusters within the bug dataset.

3.4 Identification of Common Features

In the subsequent stage of the BUGOPTIMIZE algorithm, termed "Identification of Common Features," the algorithm systematically identifies features that exhibit consistent relevance and informativeness across various feature selection techniques. Following the application of filter-based, wrapper-based, and embedded methods within each cluster, the algorithm aggregates the selected subsets to identify common features. These common features, representing attributes that consistently contribute to the predictive accuracy and robustness of the model, are collated into an array for further analysis. By focusing on features that demonstrate high relevance across different selection techniques, BUGOPTIMIZE ensures that only the most informative attributes are retained for subsequent processing and model building, enhancing the efficiency and effectiveness of bug prediction in software development environments.

3.5 Feature Selection for Entire Dataset

In complement to the cluster-specific feature selection process, the BUGOPTIMIZE algorithm conducts feature selection on the entirety of the dataset to extract an additional set of common features. This step is instrumental in ensuring comprehensive coverage of important attributes that may not have been adequately captured within individual clusters. By encompassing the entire dataset, BUGOPTIMIZE safeguards against potential oversights or biases from focusing solely on cluster-specific feature selection. Through this holistic approach, the algorithm guarantees that all relevant features, regardless of their distribution across clusters, are considered in the final feature selection process. Consequently, BUGOPTIMIZE facilitates the identification of a robust set of attributes that collectively contribute to the accuracy and reliability of bug prediction models in software development contexts.

3.6 Extraction of Final Best Features

In the culminating step of the BUGOPTIMIZE algorithm, termed "Extraction of Final Best Features," the algorithm amalgamates the common features identified through cluster-specific and dataset-wide feature selection procedures to compile the ultimate set of best features. By integrating insights from diverse feature selection techniques across the entire dataset, BUGOPTIMIZE ensures a comprehensive representation of the most informative attributes relevant to bug prediction. Subsequently, these meticulously selected features are extracted to generate a new dataset exclusively comprising the most influential attributes. This consolidation process aims to distil the dataset to its essence, retaining only those attributes deemed indispensable for accurate bug prediction. Through this meticulous curation of features, BUGOPTIMIZE facilitates the creation of a refined dataset optimized for predictive modelling, thereby bolstering the efficacy and precision of bug prediction in software development scenarios. Finally, the algorithm outputs the final set of best features, providing developers and researchers with a refined dataset optimized for scalable handling and accurate prediction of bug counts.

This methodology of the BUGOPTIMIZE algorithm enables efficient feature selection and dataset optimization, contributing to improved bug prediction accuracy in software development environments.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

This section delves into the results and discussions stemming from an investigation into the effectiveness of the BUGOPTIMIZE algorithm. This algorithm's primary objective is to enhance the precision of bug count predictions by meticulously selecting pertinent features from a designated bug dataset. The bug dataset under scrutiny comprises 18 attributes, each playing a distinct role in understanding and managing reported software bugs.

The first attribute, Bug_ID, acts as a unique identifier assigned to each bug within the dataset, facilitating precise tracking and reference. Subsequently, the Description attribute provides a textual exposition of each bug, offering crucial insights into its nature, behaviour, and specific characteristics. This textual information aids in comprehensively understanding the reported issues.

Further, attributes such as Software_Process_Model and Project_Phase shed light on the broader context within which bugs arise. They delineate the software development process model employed, whether Agile or Waterfall and the particular phase of the development lifecycle during which bugs are identified, such as Requirements or Testing. Understanding these contextual elements aids in contextualizing bug occurrences and prioritizing resolution efforts accordingly. Additionally, attributes like Developer(s)_involved and Team_Lead allocate responsibility within the development team, delineating who addresses reported bugs and oversees the resolution process—

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
Article Received: *25 April 2023* ***Revised:*** *12 June 2023* ***Accepted:*** *30 July 2023*

date_reported chronicles the timing of bug reports, facilitating lifecycle tracking and timeline management.

Severity and Priority attributes provide crucial insights into the urgency and criticality of reported bugs. Resolution_status, on the other hand, Resolution_status offers a snapshot of the current state of bug resolution efforts, categorizing bugs as Open, In Progress, Resolved, or Closed. Attributes such as Module, Operating_System, Test_Environment, Code_Repository, and Test_Framework provide contextual information regarding the environment and conditions under which bugs manifest, aiding in targeted bug resolution and system understanding.

Furthermore, Code_Reviewer and Unit_Test_Coverage attributes contribute to ensuring code quality and comprehensiveness of testing efforts, thereby bolstering bug resolution effectiveness. Finally, the Bug_count attribute encapsulates the frequency of bug occurrences within the dataset, providing a quantitative measure of the prevalence of reported issues.

By meticulously examining these 18 attributes, the BUGOPTIMIZE algorithm endeavours to refine bug count prediction accuracy, enhancing software quality and reliability.

The dataset covers various bug scenarios, offering a realistic basis to assess the effectiveness of the BUGOPTIMIZE algorithm in feature selection. Implemented in Java during experimentation, BUGOPTIMIZE benefits from Java's adaptability for swift algorithm development. Java's robustness and flexibility make it an ideal platform for constructing a comprehensive bug dataset feature selection tool. Utilizing the Weka tool further streamlines implementation, given its widespread use in experimental research within data mining and machine learning. Evaluating the BUGOPTIMIZE algorithm involves employing Linear Regression and Random Forest algorithms to compare bug count prediction accuracy before and after feature selection, measured using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics.

Mean Absolute Error (MAE):

MAE calculates the average absolute variance between predicted and actual values in a dataset, offering a simple indication of error magnitude regardless of direction. The formula for MAE can be expressed as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x}_i| \quad (1)$$

Where:

- N denotes the total number of observations in the dataset.
- x_i represents the actual value.
- $\hat{x}_i$ represents the predicted value.

Root Mean Squared Error (RMSE):

RMSE is more comprehensive, giving heavier penalties for larger errors. It computes the square root of the average squared deviations between predicted and actual values. The RMSE formula can be expressed as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2} \quad (2)$$

Where:

- N represents the total number of observations in the dataset.
- x_i denotes the actual value.
- $\hat{x}_i$ represents the predicted value.

Lower values of MAE and RMSE indicate higher accuracy. Table 1 presents the comparative analysis before and after feature selection by BUGOPTIMIZE.

Table 1: Comparative Evaluation Before and After Feature Selection by BUGOPTIMIZE

Algorithm	**Before** BUGOPTIMIZE **feature selection**		**After** BUGOPTIMIZE **feature selection**	
	MAE	**RMSE**	**MAE**	**RMSE**
Linear Regression	0.2668	0.3251	0.2609	0.308
Random Forest	0.1626	0.2363	0.1341	0.224

Applying the BUGOPTIMIZE algorithm to the bug dataset notably improved the accuracy of bug count prediction, as illustrated in Figure 2.

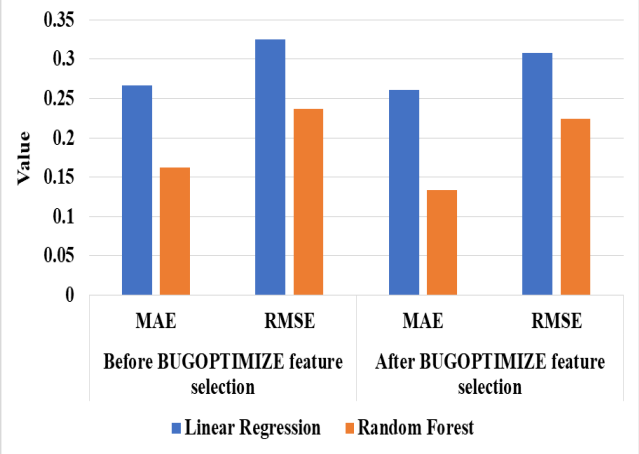

Figure 2: Comparative Assessment Before and After Feature Selection by BUGOPTIMIZE

The results analysis (Table 1 and Figure 2) vividly demonstrates the impact of the BUGOPTIMIZE feature selection algorithm on the performance of bug prediction

models. Before applying BUGOPTIMIZE, both Linear Regression and Random Forest algorithms exhibited certain Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) levels. However, following the implementation of the BUGOPTIMIZE feature selection, notable improvements were observed in the predictive accuracy of both algorithms. For Linear Regression, the MAE decreased from 0.2668 to 0.2609, while the RMSE decreased from 0.3251 to 0.308. Similarly, the MAE decreased from 0.1626 to 0.1341 for Random Forest, and the RMSE decreased from 0.2363 to 0.224. These reductions in error metrics indicate that BUGOPTIMIZE feature selection effectively enhances the performance of bug prediction models, resulting in more accurate and reliable predictions. The comparative analysis underscores the significance of BUGOPTIMIZE in optimizing feature selection processes, ultimately leading to improved bug prediction outcomes in software development scenarios.

V. CONCLUSION AND FUTURE WORK

In conclusion, BUGOPTIMIZE presents a comprehensive solution for optimizing bug datasets and enhancing bug count prediction accuracy in software development environments. Through its innovative combination of majority voting cluster-based fine-tuned feature selection techniques, BUGOPTIMIZE effectively addresses the challenges associated with existing feature selection methods, resulting in significant reductions observed in both Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) compared to existing methods. Specifically, in Linear Regression, MAE decreased from 0.2668 to 0.2609 and RMSE from 0.3251 to 0.308, while for Random Forest, MAE decreased from 0.1626 to 0.1341 and RMSE from 0.2363 to 0.224. These findings underscore the substantial contribution of BUGOPTIMIZE to bug dataset optimization and prediction accuracy in software development environments. Future work could involve adapting and extending BUGOPTIMIZE to other areas, such as healthcare, finance, or manufacturing, where predictive modelling and feature selection are critical for decision-making. By leveraging the strengths of BUGOPTIMIZE in different domains, it is possible to enhance prediction accuracy and optimize datasets for various applications, extending its impact beyond software development. Overall, BUGOPTIMIZE lays the foundation for future research endeavours to advance predictive modelling and feature selection techniques across diverse domains.

REFERENCES

[1] Rodríguez-Pérez, G., Robles, G., Serebrenik, A., Zaidman, A., Germán, D. M., & Gonzalez-Barahona, J. M. (2020). How bugs are born: a model to identify how bugs are introduced in software components. Empirical Software Engineering, 25, 1294-1340.

[2] Jahanshahi, H., Cevik, M., & Başar, A. (2020). Predicting the number of reported bugs in a software repository. In Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings 33 (pp. 309-320). Springer International Publishing.

[3] Pandey, S. K., Mishra, R. B., & Tripathi, A. K. (2020). BPDET: An effective software bug prediction model using deep representation and ensemble learning techniques—Expert Systems with Applications, 144, 113085.

[4] Siddiqi, M. A., & Pak, W. (2020). Optimizing filter-based feature selection method flow for the intrusion detection system. Electronics, 9(12), 2114.

[5] Le, T. M., Vo, T. M., Pham, T. N., & Dao, S. V. T. (2020). A novel wrapper–based feature selection for early diabetes prediction enhanced with a metaheuristic. IEEE Access, 9, 7869-7884.

[6] Rahman, M. M., Usman, O. L., Muniyandi, R. C., Sahran, S., Mohamed, S., & Razak, R. A. (2020). A review of machine learning feature selection and classification methods for autism spectrum disorder. Brain sciences, 10(12), 949.

[7] Potharlanka, J. L. (2024). Feature importance feedback with Deep Q process in ensemble-based metaheuristic feature selection algorithms. Scientific Reports, 14(1), 2923.

[8] Alazzam, H., Sharieh, A., & Sabri, K. E. (2020). A feature selection algorithm for intrusion detection system based on pigeon-inspired optimizer. Expert systems with applications, 148, 113249.

[9] Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer networks, 174, 107247.

[10] Ghosh, P., Azam, S., Jonkman, M., Karim, A., Shamrat, F. J. M., Ignatious, E., ... & De Boer, F. (2021). Efficient prediction of cardiovascular disease using machine learning algorithms with relief and LASSO feature selection techniques. IEEE Access, 9, 19304-19326.

[11] Wang, M., Lu, Y., & Qin, J. (2020). A dynamic MLP-based DDoS attack detection method using feature selection and feedback. Computers & Security, 88, 101645.

[12] Liu, H., & Setiono, R. (2022). Feature selection and classification–a probabilistic wrapper approach. In Industrial and engineering applications or artificial intelligence and expert systems (pp. 419-424). CRC Press.

[13] El-Hasnony, I. M., Barakat, S. I., Elhoseny, M., & Mostafa, R. R. (2020). Improved feature selection model for big data analytics. IEEE Access, 8, 66989-67004.

[14] Shafiq, M., Tian, Z., Bashir, A. K., Du, X., & Guizani, M. (2020). IoT malicious traffic identification using wrapper-based feature selection mechanisms. Computers & Security, 94, 101863.

[15] Chen, R. C., Dewi, C., Huang, S. W., & Caraka, R. E. (2020). Selecting critical features for data classification based on machine learning methods. Journal of Big Data, 7(1), 52.

[16] Rostami, M., Forouzandeh, S., Berahmand, K., & Soltani, M. (2020). Integration of multi-objective PSO-based feature selection and node centrality for medical datasets. Genomics, 112(6), 4370-4384.

[17] Agrawal, P., Ganesh, T., Oliva, D., & Mohamed, A. W. (2022). S-shaped and v-shaped gaining-sharing knowledge-based algorithm for feature selection. Applied Intelligence, 1-32.