

SWARM OPTIMIZED MACHINE LEARNING MODEL FOR ENHANCED PREDICTION OF CORONARY ARTERY DISEASE

JAYAMOL P. JAMES¹, Dr. GNANAPRIYA S.²

¹Ph.D Research Scholar, Department of Computer Science,
Nehru College of Management, Coimbatore, India.
jayamolpames@gmail.com

²Assistant Professor, Department of Computer Science,
Nehru College of Management, Coimbatore, India.

ABSTRACT

Coronary artery disease poses a critical health challenge requiring accurate and timely diagnosis. This research introduces an optimized classification framework that integrates Artificial Bee Colony (ABC) algorithm with Naive Bayes (NB) to strengthen prediction performance. The working mechanism begins with feature extraction from a benchmark coronary dataset, where ABC acts as a bio-inspired swarm intelligence method to identify the most relevant clinical attributes by mimicking the intelligent foraging behavior of bees. By eliminating redundant and noisy features, ABC enhances the learning environment for the Naive Bayes classifier. The probabilistic nature of NB then utilizes these refined features to compute posterior probabilities for classification, ensuring efficient decision boundaries. Experimental validation shows that the ABC-NB model achieves 93.45% accuracy, 94.23% sensitivity, and 92.56% specificity, outperforming standard classifiers in early-stage CAD detection. The model's simplicity, interpretability, and high predictive value make it suitable for integration into clinical workflows. Its low computational complexity enables deployment on embedded medical systems. This framework promotes scalable, cost-effective, and accurate cardiac risk prediction. Future advancements will involve real-time prediction support, dynamic retraining with streaming patient data, and expansion to multi-modal datasets for comprehensive cardiovascular profiling across diverse populations and clinical scenarios.

Keywords: *Coronary Artery Disease, Artificial Bee Colony, Naive Bayes, Feature Selection Medical Diagnosis, Swarm Intelligence*

1. INTRODUCTION

Coronary artery disease (CAD) is a serious cardiovascular condition due to the narrowing or blockage of coronary arteries, reducing oxygen supply to the heart. As a type of heart disease, CAD significantly contributes to morbidity and mortality worldwide [1]. The gradual buildup of plaque in arteries restricts blood flow, leading to complications such as chest pain, heart attacks, and heart failure. Several factors contribute to disease progression, including lifestyle choices, genetics, and metabolic disorders. Detecting CAD at an early stage is crucial for effective intervention, as delays in diagnosis increase the risk of severe cardiac events. Traditional diagnostic methods require specialized expertise, making early detection challenging in many healthcare settings [2]. Identifying CAD has historically relied on medical evaluations, imaging tests, and invasive procedures. Electrocardiograms (ECG), stress testing, and angiography have been widely used for diagnosis

[3]. While these approaches provide valuable insights into heart function, they often involve high costs, lengthy procedures, and potential patient discomfort. Computational advancements have introduced non-invasive techniques for disease identification, offering an alternative to conventional medical tests [4], [5]. Modern diagnostic systems integrate patient data from multiple sources, utilizing advanced computational techniques to detect abnormalities in heart function. These approaches improve diagnostic efficiency and reduce dependency on resource-intensive procedures [6].

Machine learning has transformed CAD prediction by analyzing medical records, imaging data, and patient history to detect early signs of disease. By processing large datasets, computational models recognize patterns that might be missed through conventional analysis [7].

Learning from historical patient information allows predictive models to assess disease likelihood more accurately. Automated systems reduce reliance on manual interpretation, ensuring faster and more consistent diagnoses [8]. Computational techniques enhance early detection, allowing physicians to implement preventive strategies before severe complications arise. The adaptability of these models enables continuous improvements, refining prediction accuracy as more data becomes available [9].

Applying machine learning in CAD prediction offers multiple advantages, including reduced diagnostic time, cost savings, and lower risks associated with invasive procedures [10]. Traditional methods require extensive testing and clinical evaluation, increasing healthcare expenses and patient discomfort. Automated prediction systems provide rapid assessments, ensuring timely intervention and efficient resource utilization. Non-invasive computational models minimize the need for unnecessary procedures, reducing stress on patients while maintaining diagnostic precision [11]. These advancements support medical professionals in making informed decisions, improving treatment planning, and enhancing overall patient outcomes. Integrating machine learning into healthcare workflows strengthens disease prevention by identifying high-risk individuals before symptoms escalate [12], [13].

Bio-inspired optimization techniques enhance machine learning models by improving performance, efficiency, and adaptability [14]. These optimization strategies refine predictive models by selecting the most relevant features and improving classification accuracy. Computational techniques inspired by natural processes help optimize parameters dynamically, ensuring robust performance across diverse medical datasets [15]. By fine-tuning predictive systems, these methods enhance diagnostic stability and minimize errors [16]. Integrating bio-inspired optimization improves decision-making capabilities, allowing healthcare professionals to rely on more precise and efficient CAD detection models [17]. This approach ensures that machine learning-driven disease prediction continues to evolve, offering scalable and effective solutions for cardiovascular disease management [18], [19].

1.1. Problem Statement

Cardiovascular diseases, particularly coronary artery disease (CAD), remain the leading cause of mortality globally. Early detection and timely intervention are crucial to reducing morbidity and mortality. Conventional diagnostic techniques rely heavily on clinical tests and manual interpretation, often leading to delayed or inaccurate diagnosis due to subjective biases or lack of resources. Machine learning models offer promise but frequently struggle with imbalanced data, irrelevant features, and inconsistent performance. Naive Bayes, a well-known probabilistic classifier, shows potential for disease prediction but suffers when confronted with noisy or redundant attributes. Feature selection and model refinement remain essential for practical implementation in clinical settings. Existing optimization techniques either converge slowly or fall into local optima, failing to enhance prediction accuracy adequately. A refined approach is required to improve the diagnostic utility of CAD prediction by integrating intelligent feature selection mechanisms, robust classification logic, and performance-driven optimization strategies aligned with clinical decision-making processes.

1.1.1. Research Questions

- How does swarm-based hyperparameter tuning influence classifier performance in CAD prediction?
- Can sensitivity-weighted optimization enhance detection rates of high-risk CAD cases?
- Does swarm intelligence outperform traditional tuning methods on medical screening datasets?

1.1.2. Hypotheses

- Swarm-based optimization significantly improves classifier accuracy compared to static parameter models.
- Feature selection driven by clinical feature salience increases predictive reliability.
- Classifier sensitivity for positive CAD cases improves when guided by swarm-adaptive thresholds.

1.2. Motivation

Heart-related disorders place an enormous burden on healthcare systems, especially in low-resource communities where diagnostic infrastructure is limited. The rising prevalence of CAD demands accessible, accurate, and early-stage detection tools that can assist healthcare providers in making prompt decisions. In rural and

underserved regions, the shortage of trained cardiologists and costly diagnostic tools further exacerbates the challenge. A society-centric solution that leverages machine learning must not only demonstrate high accuracy but also offer interpretability and low computational cost for widespread use. Empowering health workers and general physicians with reliable prediction tools can drastically improve outcomes. Bridging the gap between advanced algorithms and societal needs requires models that optimize accuracy while maintaining affordability, transparency, and deployment feasibility. Motivated by the urgent need to democratize healthcare access, the focus is directed toward constructing a lightweight, efficient, and intelligent diagnostic system tailored for CAD prediction, promoting equitable healthcare access across socio-economic boundaries.

1.3. Objective

The core objective involves designing a high-precision, computationally efficient prediction framework for coronary artery disease using the Artificial Bee Colony (ABC) algorithm to optimize feature selection in conjunction with Naive Bayes classification. This model aims to eliminate redundant and irrelevant data attributes, enhancing classification accuracy, interpretability, and robustness in medical diagnostics. The system seeks to address challenges associated with noisy datasets, inconsistent performance, and low generalization observed in traditional models. By incorporating swarm intelligence through ABC, the model aspires to intelligently explore feature subsets, reducing dimensionality while preserving vital clinical indicators. Emphasis remains on maximizing diagnostic utility indices and precision, ensuring practical value in clinical environments. The ultimate goal is to deliver a reliable, reproducible, and adaptive CAD prediction framework that aligns with real-world healthcare needs, particularly in regions with limited infrastructure, supporting early diagnosis and intervention strategies that can reduce fatality rates and improve population-level cardiovascular outcomes.

2. LITERATURE REVIEW

"Hybrid Harris Hawks Approach (H-HHO)" [20] begins by enhancing the original Harris Hawks Optimization algorithm through three structural improvements. The first enhancement incorporates a velocity operator into the position update process, improving exploration by enabling wider solution search. The second mechanism

introduces an exploration factor to regulate random variations, increasing diversity during population movement. The third addition is a linearly decreasing inertia weight that strengthens exploitation in later iterations by refining the local search dynamics. A context-aware mechanism is embedded to select domain-relevant features, particularly thallium and chest pain type, using domain rules. H-HHO then executes parameter optimization for classifiers, refining the search over hyperparameter space to enhance classification readiness. The hybrid structure balances global and local search by adapting to population behavior and search progression, integrating swarm intelligence with biologically inspired hunting strategies, and enhancing convergence stability while avoiding premature stagnation.

"Two-Layered Voting - Machine Learning Framework (TLV-MLF)" [21] operates through a dual-layered architecture that separates feature selection and prediction. In the first layer, three statistical feature evaluation methods—ANOVA F-test, Chi-squared test, and Mutual Information—are applied to rank attributes based on relevance to coronary artery disease classification. Hard voting aggregates the selection based on frequency of occurrence, while soft voting applies weighted scores. The selected features progress to the second layer, which integrates four machine learning models: Random Forest, Decision Tree, Support Vector Classifier, and Multi-Layer Perceptron. Ensemble classification is conducted using both hard and soft voting strategies to determine final outcomes. GridSearchCV is utilized to hyper-tune each classifier's parameters, systematically scanning the parameter space to maximize classifier effectiveness. This two-tier mechanism structurally decouples feature importance assessment from model consensus, allowing TLV-MLF to integrate statistical reasoning with optimized ensemble modeling in a modular pipeline for coronary artery disease detection.

"Challenges-Heart-Predict" [22] reviewed 68 works to identify persistent issues in heart disease prediction via machine learning. Key challenges included poor dataset diversity, class imbalance, inconsistent feature relevance, and lack of generalization: overfitting, computational inefficiency, and the absence of standardized validation protocols limited real-world applicability. The analysis emphasized explainable AI and benchmark datasets for reliable integration.

The review structured a roadmap addressing model scalability, interpretability, and integration of accurate clinical data into predictive heart disease systems. **“ML-Cardiac-Death”** [23] applied ensemble-based machine learning to predict cardiac death in 3,987 ischemic heart disease patients. Key features included dyslipidemia, LVEF, diabetes, and smoking. Ablation testing ranked variables by impact. AUROC and F1-macro scores validated model performance. Cox analysis outperformed traditional statistics in stratifying risk. The system structured a survival prediction pipeline using composite indicators. The approach enhanced risk identification for clinical use, optimizing stratification with machine learning and improving early mortality detection. **“DL-Heart-Breast”** [24] predicted cardiovascular events in breast cancer patients using LSTM with a trainable decay for missing data. Longitudinal EHRs and NLP-derived clinical notes identified cardiovascular risks across a 24-month window. AUC scores ranged from 0.7189 to 0.9548. The system structured risk profiling dynamically, integrating unstructured and structured data. Deep learning improved forecasting precision, aiding treatment planning. The architecture offered a scalable model for early disease detection, enhancing personalized healthcare using time-aware predictive analysis.

“ML-Heart-Detect” [25] applied decision trees, SVM, RF, and PCA for early heart disease detection using structured medical records. PCA reduced dimensionality, improving computational efficiency. Supervised models identified critical patterns among risk indicators like cholesterol and blood pressure. Classification accuracy, recall, and F1-score guided model assessment. Decision trees supported interpretability, while ensemble models ensured stability. The pipeline structured a reliable diagnostic system, enhancing detection accuracy and improving generalization through statistical learning and automated feature optimization. **“BSEL-Heart-Diagnosis”** [26] combined BBSO for feature selection with WEC for classification in heart disease diagnosis. LIN standardized input data, while RCoMO optimized neural architecture. Machine learning models weighted features by predictive strength. The bio-inspired framework balanced exploration-exploitation to enhance prediction. Accuracy, recall, and F1-score validated outcomes. The pipeline structured a robust diagnostic model, integrating swarm intelligence and ensemble learning to improve precision, reduce redundancy, and enhance decision confidence in clinical applications. **“ML-Vascular-Occlusion”**

[27] predicted vascular occlusion severity using ML models on data from 300 ischemic heart disease patients with periodontitis. Structured preprocessing and supervised algorithms identified PISA as a key indicator. Random Forest achieved the highest accuracy across occlusion levels. Clinical and periodontal features were used to estimate occlusion probability. ROC and confusion matrix validated results. The study structured a diagnostic model linking oral health with cardiac risk, enhancing early detection of vascular complications.

“ICD-ATC-CHD” [28] unsupervised learning to select relevant ICD-10 and ATC codes in CHD diagnosis from 51,506 patient records. Concrete autoencoders performed best in dimensionality reduction and mortality risk prediction. Shapley values identified top-ranking codes influencing post-discharge outcomes. The system structured patient representation using hierarchical coding. The pipeline improved predictive reliability by integrating clinical taxonomies into AI models, ensuring interpretability and precision in CHD classification using unsupervised feature selection techniques.

“IoT-Heart-Predict” [29] uses IoT sensor data with XGBoost and Bi-LSTM for heart disease prediction. Real-time data captured via wearable devices underwent time-series classification. XGBoost selected critical health indicators; Bi-LSTM processed sequential patterns. Cloud infrastructure supports real-time scalability. The framework structured data flow from acquisition to classification. The system enabled continuous remote monitoring, enhancing early detection accuracy. The architecture provided a robust AI-driven pipeline for proactive risk management using sensor-driven medical intelligence.

“PSO-SVM-HeartLiver” [30] integrated modified PSO for feature optimization with SVM for classifying heart and liver diseases. UCI datasets were preprocessed, and PSO refined feature subsets by optimizing search space traversal. SVM handled high-dimensional data for classification. Comparative studies validated performance gains in precision and accuracy. The model structured a search-guided classification system, balancing feature relevance with stability. The pipeline enhanced diagnostic accuracy, reducing overfitting through evolutionary optimization in supervised models.

“XAI-Ensemble-Heart” [31] proposed an explainable AI model combining ensemble learning with SHAP interpretability for heart disease diagnosis. Feature selection used correlation, chi-square, and recursive elimination. A stacked ensemble merged DT, LR, and SVM. Outlier detection and normalization refined inputs. SHAP ranked feature influence, ensuring model transparency. Accuracy, precision, and F1-score validated classification. The pipeline structured a diagnostic system combining interpretability and performance, supporting AI-enabled clinical decision-making with traceable model behavior.

“HXAI-ML-Heart” [32] combined class-balancing techniques with explainable AI for heart disease detection. SMOTE, Tomek Link, and Random Oversampling addressed imbalance. Extra Trees Classifier ranked feature importance. SHAP, LIME, and Permutation Analysis provided model interpretability. Feature selection reduced complexity, improving accuracy and F1-score. The structured framework ensured high-performance classification with transparency, enabling clinical trust. The pipeline offered a real-world ready system by integrating resampling with interpretable learning, optimizing prediction and feature relevance in cardiovascular disease diagnostics.

“CustomML-HeartScreen” [33] deployed patient-specific machine learning for large-scale heart disease screening. Neural network design, loss function tuning, and feature selection were dynamically personalized. Models trained on the Cleveland and UCI datasets achieved high detection accuracy. The system adapted classification boundaries to individual health data, enabling precision diagnostics. Performance was validated using accuracy, recall, and precision. The structured pipeline supported scalable screening by tailoring model parameters, enhancing detection reliability and classification precision across diverse patient populations.

“ML-IHD-Review” [34] analyzed ML models for ischemic heart disease prediction from 2017–2021. Classification techniques like SVMs, Neural Networks, and Decision Trees showed consistent performance. Metrics evaluated included AUROC, accuracy, and specificity. Challenges identified involved data bias, inconsistent feature use, and poor model interpretability. The lack of validation standards reduced study comparability. The structured review proposed improvements through real-world data integration and explainable methods, outlining a roadmap for refining AI-

driven IHD prediction and addressing limitations in model evaluation.

“FS-Tree-Heart” [35] examined how feature selection techniques influence tree-based classifiers for heart disease detection. Five methods were applied to optimise prediction, including Mutual Information and Stability Selection. Eleven classifiers, including Hoeffding Tree and Gradient Boosting, were benchmarked. Stability Selection with Hoeffding Tree yielded the highest accuracy. The structured framework improved generalization and reduced overfitting by removing redundant features. The pipeline optimized tree-based model performance, demonstrating how strategic feature engineering enhances classifier precision and prediction consistency.

Bio-inspired optimization leverages intelligent behaviors observed in nature, such as swarm intelligence and adaptive movement, to optimize complex machine learning models [36]-[60]. For coronary artery disease prediction, these methods enable efficient parameter tuning, adaptive feature prioritization, and faster convergence, enhancing model precision, reducing diagnostic error, and supporting scalable medical decision-making frameworks [61]-[76].

Existing studies on coronary artery disease (CAD) prediction have predominantly utilized conventional machine learning classifiers such as Support Vector Machines, Decision Trees, and Random Forests with standard optimization techniques. Several works have employed generic feature selection mechanisms without tailored enhancement for clinical features. The novelty of the current work lies in integrating a swarm-based optimization algorithm that not only fine-tunes classifier hyperparameters but also adapts to the imbalanced nature and clinical relevance of features. Unlike earlier models, the present approach prioritizes sensitivity-weighted learning, enhancing diagnostic accuracy for high-risk cases, which has been comparatively underexplored in previous works.

3. ARTIFICIAL BEE COLONY OPTIMIZED NAIVE BAYES (ABC-NB)

Naive Bayes remains a widely used probabilistic classifier based on Bayes' theorem, assuming conditional independence among features. Despite its computational efficiency and strong theoretical foundation, standard implementations face challenges, particularly when

dealing with correlated features, high-dimensional data, and noisy attributes. These factors can reduce classification accuracy and introduce bias into predictions. Addressing these issues requires a feature selection strategy that ensures only the most informative attributes contribute to the classification process, leading to improved generalization and reduced model complexity.

Artificial Bee Colony (ABC) optimization, inspired by the foraging behavior of honeybee swarms, is a robust swarm intelligence technique that efficiently explores large search spaces. The algorithm comprises three key components: employed bees, onlooker bees, and scout bees. Employed bees exploit known solutions by refining existing feature subsets, onlooker bees evaluate their quality and select the best candidates, and scout bees introduce randomness by exploring new feature combinations. This balance between exploration and exploitation makes ABC highly suitable for optimizing Naive Bayes by systematically identifying the most relevant feature subsets, thereby improving classification performance.

ABC optimization enhances Naive Bayes by iteratively refining the selection of features, ensuring that only the most discriminative attributes are retained. The adaptive nature of the algorithm allows it to effectively handle complex datasets, reducing classification errors and computational overhead. By optimizing the search process through swarm intelligence, the integration of ABC improves the model's ability to make accurate predictions with reduced data dimensionality. The resulting classifier achieves superior generalization, making it more effective in handling diverse classification tasks.

This paper systematically explores the integration of ABC with Naive Bayes to enhance classification accuracy. The discussion begins with the initialization of feature subsets, followed by the evaluation of fitness functions that guide the selection process. The iterative optimization strategy of ABC is then examined, detailing how it refines feature selection. This section concludes with an analysis of performance improvements, highlighting how the optimized NB model achieves higher accuracy, lower feature redundancy, and improved computational efficiency.

3.1. Initialize Population in ABC-NB

In the ABC-NB algorithm, initializing the population represents the first stage, where a set of potential solutions, termed "food sources," gets generated randomly to prepare for the optimization process. These food sources signify distinct feature subsets, forming the foundation of the Naive Bayes classifier's initial structure. Each food source comprises a combination of features selected from the complete dataset, influencing the classifier's predictions. This initialization step plays a crucial role by setting diverse starting points for the subsequent optimization, enabling the algorithm to explore multiple solutions and improving the likelihood of finding an optimal feature subset. In the behavior of a bee swarm, scouts venture into unexplored areas to discover new food sources. Mimicking this behavior, the ABC-NB algorithm initiates random feature subset selection, creating a preliminary population with varied combinations of features. This process minimizes feature redundancy, enhancing the Naive Bayes classifier's overall accuracy and reducing computational costs by limiting the feature space.

To generate the initial population of food sources, consider the feature space as a matrix X with dimensions $N \times M$, where N represents the total number of samples, and M signifies the total number of features. The food sources (i.e., solutions) form a matrix P with F rows and M columns, where F denotes the total number of food sources initialized.

$$P = \{X_{f,m} | f = 1, 2, \dots, F; m = 1, 2, \dots, M\} \quad (1)$$

where $X_{f,m}$ represents a feature subset assigned to the f -th food source. This initialization involves selecting values from the original feature space X and assigning them to P in a way that each food source consists of random combinations of features.

The probability of selecting a feature in any food source can be expressed as:

$$\Pr(X_{f,m}) = \frac{\sum_{n=1}^N x_{n,m}}{N} \quad (2)$$

where $x_{n,m}$ represents the presence of the m -th feature in the n -th sample, while N denotes the total number of samples. This probability establishes the likelihood of each feature's inclusion in a food source.

Once features are randomly selected for each food source, the algorithm creates a binary vector B_f for each food source f , indicating the inclusion (1) or exclusion (0) of each feature:

$$\Pr(X_{f,m}) = \frac{\sum_{n=1}^N x_{n,m}}{N} \quad (3)$$

where $x_{n,m}$ represents the presence of the m -th feature in the n -th sample, while N denotes the total number of samples. This probability establishes the likelihood of each feature's inclusion in a food source.

Once features are randomly selected for each food source, the algorithm creates a binary vector B_f for each food source f , indicating the inclusion (1) or exclusion (0) of each feature:

$$B_f = \{b_{f,1}, b_{f,2}, \dots, b_{f,M}\} \quad (4)$$

For a feature m in the f -th food source, $b_{f,m} = 1$ if the feature is selected and $b_{f,m} = 0$ otherwise. This binary vector allows the ABC algorithm to handle each food source as a unique combination of features, facilitating optimization and selection. The probability of a feature being selected within the population is given by:

$$p_f(m) = \frac{\Pr(X_{f,m})}{\sum_{m=1}^M \Pr(X_{f,m})} \quad (5)$$

where $p_f(m)$ represents the normalized probability for the m -th feature in the f -th food source, ensuring that the sum of probabilities across features equals one.

To evaluate each food source's effectiveness, the initial fitness of the Naive Bayes classifier with each subset of features must be determined. The initial fitness F_f for each food source f is calculated using the classification accuracy:

$$F_f = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

where TP, TN, FP , and FN represent True Positives, True Negatives, False Positives, and False Negatives, respectively, indicating the classifier's prediction performance on the selected feature subset.

The algorithm seeks to optimize the Naive Bayes model by reducing redundancy in the

selected features while maximizing classification accuracy. The objective function O_f for the f -th food source is formulated as:

$$O_f = w_1 \cdot F_f - w_2 \cdot \text{Redundancy}(B_f) \quad (7)$$

where w_1 and w_2 are weights assigned to prioritize accuracy and redundancy reduction. $\text{Redundancy}(B_f)$ measures feature overlap within B_f , with lower redundancy indicating a more optimized feature subset.

The selection of features at random from the feature space can be expressed by:

$$R_{f,m} = \text{random}(0,1) \quad (8)$$

where $R_{f,m}$ generates a random binary value for each feature m in food source f , setting $b_{f,m}$ accordingly to 0 or 1.

The final step in population initialization involves assigning random solutions to each food source to create the initial population, denoted by:

$$\{P_1, P_2, \dots, P_F\} = \{B_1, B_2, \dots, B_F\} \quad (9)$$

This set of binary vectors B_f represents the initial solution matrix, which will be evaluated in subsequent optimization cycles.

3.1.1. Research Design

The study followed a structured protocol starting with data preprocessing using outlier removal, normalization, and feature encoding. The dataset was partitioned using stratified k-fold cross-validation to maintain class distribution. A swarm intelligence algorithm was applied for hyperparameter tuning of selected classifiers. The optimized configurations were evaluated based on accuracy, sensitivity, specificity, and F1-score. Ethical clearance was not applicable since publicly available datasets without patient identifiers were used. The model pipeline was implemented in Python using scikit-learn and custom optimization modules, and the entire workflow was tested for stability and reproducibility across three random seeds.

3.2. Evaluate Fitness in ABC-NB

In the ABC-NB algorithm, the Evaluate Fitness step holds a crucial role in assessing the performance of each food source by calculating the fitness of each solution (feature subset) concerning

the objective of optimizing the Naive Bayes classifier's predictive accuracy. This step aligns with the foraging behavior observed in bee swarms, where bees assess the quality of food sources, enabling the identification of promising solutions for further optimization. By calculating the fitness of each food source, the ABC-NB algorithm effectively distinguishes between more valuable solutions, guiding subsequent phases and enhancing the algorithm's overall performance. After initializing the population in Step 1, each food source is represented by a binary vector, defining a unique subset of features. Each food source's subset undergoes evaluation based on its impact on the Naive Bayes classifier's predictive capability in this step. The fitness values computed here will determine the probability of each food source's selection by onlooker bees in future steps, aligning with the goal of refining and optimizing the model.

In the ABC-NB algorithm, the fitness of each food source f is defined through a function that evaluates both the accuracy of the Naive Bayes classifier and the efficiency of the feature subset in minimizing redundancy. To compute the fitness score F_f an initial evaluation of the classifier's performance on the feature subset represented by each food source is carried out of each food source.

$$F_f = \alpha \cdot Q_f - \beta \cdot \sum_{m=1}^M \delta(b_{f,m}, b_{f,m+1}) \quad (10)$$

where α and β represent weights assigned to accuracy Q_f and redundancy penalty, respectively. Here, Q_f denotes the Naive Bayes classifier's accuracy on the feature subset for the f -th food source, and $\delta(b_{f,m}, b_{f,m+1})$ measures redundancy between adjacent features in the binary vector B_f .

Redundancy reduction forms a significant aspect of the fitness function to ensure the selection of non-redundant and compelling features. The redundancy penalty component is calculated based on the presence of adjacent similar features in B_f , thus:

$$Redundancy(B_f) = \sum_{m=1}^{M-1} |b_{f,m}, b_{f,m+1}| \quad (11)$$

This redundancy measure accumulates penalties whenever consecutive features in the feature subset B_f are identical, ensuring that only essential and non-redundant features contribute positively to the fitness score. The overall fitness G_f combines the classifier's accuracy with feature subset quality by penalizing redundancy and adding a preference to solutions that achieve higher accuracy and diversity among selected features:

$$G_f = \lambda \cdot F_f - \gamma \cdot Redundancy(B_f) \quad (12)$$

where λ represents the weight for accuracy, and γ is the weight for redundancy. This fitness score G_f provides a comprehensive measure that balances accuracy with feature quality.

To further assess each feature's contribution within a selected subset, a weight is assigned to each feature in the feature subset. The contribution weight W_m for each feature m can be defined by:

$$W_m = \frac{\sum_{f=1}^F F_f \cdot b_{f,m}}{\sum_{f=1}^F b_{f,m}} \quad (13)$$

where W_m represents the average fitness contribution of the m -th feature across all food sources, indicating feature importance and aiding in identifying influential features for further optimization.

The probability P_f of selecting a food source for exploration by onlooker bees depends on the fitness G_f , normalized over all food sources:

$$P_f = \frac{G_f}{\sum_{k=1}^F G_k} \quad (14)$$

By calculating P_f , each food source is assigned a probability that reflects its fitness, guiding the algorithm toward selecting more promising feature subsets in the optimization process. To ensure that all fitness values remain on a comparable scale, the algorithm normalizes each food source's fitness G_f using the maximum and minimum values observed across the population:

$$N_f = \frac{G_f - G_{min}}{G_{max} - G_{min}} \quad (15)$$

where N_f represents the normalized fitness score for the f -th food source, G_{min} is the minimum

fitness observed, and G_{max} is the maximum fitness observed.

Based on the calculated fitness scores and selection probabilities, the ABC-NB algorithm prioritizes feature subsets that balance high predictive accuracy with minimized redundancy, forming an optimized foundation for further classifier improvements. The final selected features ensure maximized classifier performance and efficiency in feature usage.

3.3. Send Employed Bees in ABC-NB

The Send Employed Bees phase of the ABC-NB algorithm signifies the beginning of the search for improvements in the existing solutions or "food sources" identified in previous steps. In this phase, employed bees represent agents exploring the vicinity of each food source to find an optimized solution by modifying existing feature subsets. This process mirrors the natural behavior of bees who intensively explore known food sources to discover improved nourishment, thus enhancing the colony's overall yield. For ABC-NB, sending employed bees facilitates refining the Naive Bayes classifier by exploring local variations in feature subsets and ensuring the algorithm progresses toward more optimal configurations. Each food source represents a distinct subset of features selected from the dataset. By sending employed bees, the algorithm aims to modify these feature subsets iteratively to enhance classifier accuracy and reduce redundancy, leading to a more optimized solution. The employed bees use a combination of probabilistic changes and specific heuristic strategies to alter the current food source, guided by each solution's evaluated fitness score from the previous step.

The exploration around each food source is based on a neighborhood search mechanism, wherein small changes to the feature subset are made to find improved solutions. Each employed bee modifies a food source by changing one or more feature inclusions. The new feature subset B_f^{new} for each food source f is generated based on the current subset B_f by altering a specific feature $b_{f,m}$ with a perturbation factor ϕ :

$$B_{f,m}^{new} = B_{f,m} + \phi \cdot (B_{f,m} - B_{k,m}) \quad (16)$$

where ϕ represents a random value within a specified range (e.g., [-1, 1]) controlling the degree of perturbation. $B_{k,m}$ denotes the feature from a

randomly selected food source $k \neq f$. This mechanism ensures diversity in the modifications and encourages the exploration of a broader feature space.

To determine an optimized perturbation factor, ϕ is calculated considering the fitness scores of the food sources. The perturbation factor ϕ_f for each employed bee associated with the f -th food source is derived as follows:

$$\phi_f = \frac{G_f}{\sum_{j=1}^F G_j} \quad (17)$$

where G_f is the fitness of food source f , and $\sum_{j=1}^F G_j$ is the total fitness of all food sources. By associating ϕ_f with the fitness scores, the algorithm emphasizes modifications to food sources with higher potential, focusing efforts where improvements are most likely to yield effective results.

After generating a new solution B_f^{new} for each food source f , the fitness G_f^{new} of the modified feature subset undergoes evaluation using the fitness function established in Step 2. The new fitness score accounts for accuracy improvement, optimized feature subset quality, and reduced redundancy. The fitness score for the modified feature subset is expressed as:

$$G_f^{new} = \lambda \cdot F_f^{new} - \gamma \cdot Redundancy(B_f^{new}) \quad (18)$$

where F_f^{new} is the classifier accuracy on the modified feature subset, and $Redundancy(B_f^{new})$ calculates the redundancy penalty.

Upon calculating G_f^{new} , a decision is made to either accept the modified food source or retain the original one based on their fitness scores. The probability P_f^{accept} of accepting the new solution B_f^{new} over the previous food source B_f is defined as:

$$P_f^{accept} = \frac{G_f^{new}}{G_f^{new} + G_f} \quad (19)$$

Eq.(19) ensures that solutions yielding higher fitness are more likely to be accepted, thereby focusing on feature subsets with optimized classification accuracy and minimized redundancy.

An adaptive mechanism adjusts the frequency of each feature selection based on previous fitness evaluations. A feature importance score I_m is assigned to each feature m based on its contribution to fitness across different food sources:

$$I_m = \frac{\sum_{f=1}^F G_f \cdot b_{f,m}}{\sum_{f=1}^F b_{f,m}} \quad (20)$$

Higher values of I_m indicate more impactful features, guiding the selection of features that are consistently associated with higher fitness, thus aiding the algorithm in converging toward an optimized feature set.

The likelihood $P_{f,m}^{modify}$ of modifying each feature m in food source f depends on the feature's importance and the overall solution fitness. This probability can be calculated by:

$$P_{f,m}^{modify} = \frac{I_m}{\sum_{n=1}^M I_n} \quad (21)$$

Higher values of $P_{f,m}^{modify}$ encourage modifications to significant features, guiding the search towards solutions with a more optimized selection. After conducting the neighborhood search and evaluating the fitness, the employed bee process results in an optimized feature subset with improved classifier performance. By iteratively refining food sources based on fitness-guided modifications, the algorithm ensures gradual convergence toward the best solutions for feature selection in the Naive Bayes classifier.

3.4. Assess New Solutions in ABC-NB

In the ABC-NB algorithm, the step of Assessing New Solutions involves evaluating the modified solutions generated by the employed bees to determine if these newly identified feature subsets exhibit optimized performance. This process resembles how bees gauge the quality of freshly discovered nectar sources, comparing them with previous locations to ensure only the best sources contribute to the hive. By assessing new solutions, the ABC-NB algorithm maintains a balance between exploring new possibilities and refining previously identified feature subsets, ensuring continuous improvement in the feature selection process. The ABC-NB evaluates each newly modified feature subset through a fitness assessment that prioritizes high classification accuracy and minimized redundancy. Only solutions with optimized performance are

considered for progression to the next phase, fostering convergence toward an ideal feature subset.

The fitness of each new solution S_f^{new} generated by the employed bees is evaluated using a modified fitness function. The fitness score F_f^{new} for each new solution S_f^{new} integrates classifier accuracy, redundancy penalties, and feature relevance, as shown below:

$$F_f^{new} = \eta \cdot Q_f^{new} - \zeta \cdot Redundancy(S_f^{new}) \quad (22)$$

where η and ζ represent weights assigned to the classifier accuracy Q_f^{new} and the redundancy penalty. Q_f^{new} denotes the accuracy of the Naive Bayes classifier when applied to the new feature subset. The redundancy measure assesses whether repeated features exist, penalizing solutions contributing to feature redundancy.

Minimizing redundancy in feature subsets remains crucial in assessing new solutions. The redundancy penalty for a new solution S_f^{new} is calculated as follows:

$$Redundancy(S_f^{new}) = \sum_{i=1}^{M-1} |S_{f,i}^{new} - S_{f,i+1}^{new}| \quad (23)$$

where $S_{f,i}^{new}$ represents the i -th feature in the new solution. Lower redundancy penalties indicate that the feature subset includes a unique selection of features with minimized overlap.

To facilitate comparisons between newly generated and existing solutions, a weighted fitness function W_f^{new} is utilized. This function calculates the weighted combination of accuracy and redundancy for each new solution:

$$W_f^{new} = \alpha \cdot F_f^{new} + \beta \cdot Diversity(S_f^{new}) \quad (24)$$

where α and β serve as weights for accuracy and feature diversity, respectively. Here, $Diversity(S_f^{new})$ ensures feature variety across the population, which contributes to a robust solution.

A critical aspect of assessing new solutions lies in promoting feature diversity, which prevents premature convergence on suboptimal

solutions. The diversity D_f^{new} for each new solution is evaluated as:

$$D_f^{new} = \frac{\sum_{i=1}^M S_{f,i}^{new}}{M} \quad (25)$$

This measure considers the proportion of features actively selected in S_f^{new} , ensuring that the subset remains diverse enough to foster further exploration in subsequent phases. Based on the weighted fitness function, a selection probability P_f^{select} is assigned to each new solution to decide whether it will replace the original solution in the population:

$$P_f^{select} = \frac{W_f^{new}}{W_f^{new} + W_f} \quad (26)$$

where W_f represents the weighted fitness score of the original solution. Solutions with higher W_f^{new} have an increased probability of selection, facilitating the retention of optimized feature subsets.

To maintain consistency across different scales of fitness values, the fitness scores of new solutions undergo adaptive scaling, expressed as:

$$F_f^{scaled} = \frac{F_f^{new} - F_{min}^{new}}{F_{max}^{new} - F_{min}^{new}} \quad (27)$$

where F_{min}^{new} and F_{max}^{new} represent the minimum and maximum fitness values among all new solutions. Scaling ensures comparability between solutions, streamlining the selection process.

Each new solution's retention of specific features depends on its individual fitness contributions and redundancy status. A retention probability $R_{f,i}^{retain}$ is assigned to each feature i in S_f^{new} based on its relative importance:

$$R_{f,i}^{retain} = \frac{F_{f,i}^{new}}{\sum_{j=1}^M F_{f,j}^{new}} \quad (28)$$

where $F_{f,i}^{new}$ represents the contribution of the feature i to the new solution's fitness score. Retaining high-contribution features enables the selection of optimal subsets while discarding redundant or less impactful features.

The final step in assessing new solutions involves a decision criterion that combines fitness,

diversity, and probability to retain solutions that meet optimized standards. A retention criterion C_f^{retain} is calculated as follows:

$$C_f^{retain} = F_f^{scaled} + D_f^{new} \cdot P_f^{select} \quad (29)$$

3.5.

3.6. Greedy Selection in ABC-NB

In the ABC-NB algorithm, Greedy Selection is pivotal in refining the feature subsets by ensuring that only the most optimized solutions proceed to the subsequent optimization stages. This process closely resembles the decision-making behavior of bees, who continuously compare potential food sources, selecting only those with the highest value to sustain the colony's productivity. Within the ABC-NB framework, Greedy Selection evaluates each newly modified feature subset generated in the Assess New Solutions phase, comparing them to the existing solutions. The more favorable solutions, based on fitness and optimized attributes, are retained, effectively refining the feature selection process. Greedy selection strengthens the ABC-NB algorithm's ability to prioritize subsets with high classifier accuracy and minimized redundancy. This selection method provides a straightforward yet powerful mechanism for filtering solutions, emphasizing the most effective combinations of features and enhancing the model's performance.

The Greedy Selection step begins by comparing the fitness score F_f^{new} of each modified solution S_f^{new} with its corresponding previous solution S_f . For a given food source f , if the fitness score of the new solution surpasses that of the existing solution, the algorithm retains the new solution; otherwise, the original solution remains:

$$S_f = \begin{cases} S_f^{new} & \text{if } F_f^{new} > F_f \\ S_f & \text{otherwise} \end{cases} \quad (30)$$

where F_f represents the fitness score of the original solution. This equation serves as the primary decision-making criterion for Greedy Selection, favoring solutions with improved fitness values.

The algorithm applies an optimization function to each selected feature subset, ensuring that the retained solution maximizes accuracy while maintaining diversity in the feature set. The optimization function O_f for each feature, the subset is expressed as:

$$O_f = \alpha \cdot F_f - \beta \cdot \text{Redundancy}(S_f) \quad (31)$$

where α and β serve as weights for fitness and redundancy. F_f denotes the classifier accuracy for the solution S_f , while $\text{Redundancy}(S_f)$ assesses feature overlap within the subset, guiding the algorithm to favor unique, high-utility feature combinations.

Greedy selection employs a retention probability P_f^{retain} that depends on the comparative fitness of the new and existing solutions. This probability is calculated as:

$$P_f^{\text{retain}} = \frac{F_f^{\text{new}}}{F_f^{\text{new}} + F_f} \quad (32)$$

where F_f^{new} represents the fitness of the modified solution, and F_f denotes the fitness of the original solution. Higher values of P_f^{retain} reflect a greater likelihood of retaining the new solution, indicating its relative effectiveness over the existing subset.

Diversity enhancement plays a significant role in Greedy Selection to prevent premature convergence on suboptimal feature subsets. The diversity score D_f for each solution S_f is computed by examining the distribution of selected features across the subset, calculated as follows:

$$D_f = \frac{\sum_{m=1}^M S_{f,m}}{M} \quad (33)$$

where $S_{f,m}$ denotes the presence of the m -th feature in the solution S_f , and M is the total number of features. Solutions with higher diversity scores D_f possess a wider range of features, ensuring that various combinations are explored.

To improve the effectiveness of Greedy Selection, each feature's contribution to the overall fitness of a solution is evaluated. The feature impact score $I_{f,m}$ for each feature m in solution S_f is determined by:

$$I_{f,m} = \frac{F_f \cdot S_{f,m}}{\sum_{k=1}^M S_{f,k}} \quad (34)$$

This score measures the contribution of individual features to the classifier's accuracy, guiding Greedy Selection to retain subsets containing high-impact features. The probability

$R_{f,m}^{\text{opt}}$ of retaining a feature m in a solution S_f is based on the feature impact score $I_{f,m}$:

$$R_{f,m}^{\text{opt}} = \frac{I_{f,m}}{\sum_{n=1}^M I_{f,n}} \quad (35)$$

Higher values of $R_{f,m}^{\text{opt}}$ favor retaining features with significant contributions to the solution's fitness, enhancing the overall quality of retained subsets. The algorithm employs a selection criterion C_f^{greedy} that combines fitness, diversity, and probability to select solutions with maximized effectiveness:

$$C_f^{\text{greedy}} = F_f + \gamma \cdot D_f + \delta \cdot P_f^{\text{retain}} \quad (36)$$

where γ and δ are weights for diversity and retention probability. Solutions with higher C_f^{greedy} values proceed to the next stage, ensuring optimized feature subsets.

3.7. Send Onlooker Bees in ABC-NB

In the ABC-NB algorithm, sending onlooker bees marks a critical phase in enhancing the optimization process. Onlooker bees represent a specialized group within the colony that selects food sources (feature subsets) based on observed fitness scores, prioritizing more promising solutions. This stage mirrors the behavior of bees who congregate around the most beneficial food sources, as feedback from employed bees indicates, thus maximizing the colony's resources. In the ABC-NB context, the Send Onlooker Bees step enables refined selection and exploration of feature subsets, enhancing the accuracy and efficiency of the Naive Bayes classifier. By focusing on high-fitness solutions, onlooker bees avoid low-value areas, concentrating the algorithm's resources on solutions with optimized potential.

Each onlooker bee selects a food source based on a probability derived from the fitness scores of available solutions. The selection probability P_f^{onlooker} for each food source f is defined by normalizing the fitness score across the population:

$$P_f^{\text{onlooker}} = \frac{F_f}{\sum_{k=1}^F F_k} \quad (37)$$

where F_f represents the fitness of food source f , while $\sum_{k=1}^F F_k$ is the total fitness of all solutions. Higher fitness values result in greater selection probabilities, ensuring that onlooker bees focus on feature subsets likely to contribute to optimized classifier performance.

Once an onlooker bee selects a food source based on $P_f^{onlooker}$, it initiates exploration around the chosen feature subset to identify further improvements. The exploration mechanism involves adjusting the feature subset like the employed bee's neighborhood search but focusing on promising areas. The new feature subset $B_f^{explore}$ is generated using a perturbation factor θ :

$$B_{f,m}^{explore} = B_{f,m} + \theta \cdot (B_{f,m} + B_{k,m}) \quad (38)$$

where θ is a randomly assigned factor between a predetermined range, and $B_{k,m}$ represents a feature from another randomly selected solution $k \neq f$. By creating diverse feature combinations within high-potential subsets, this mechanism ensures comprehensive exploration in optimized directions.

The fitness score of the selected food source can influence the degree of exploration by onlooker bees. The adaptive scaling of the perturbation factor θ_f depends on the relative fitness F_f of the chosen solution:

$$\theta_f = \frac{F_f - F_{min}}{F_{max} - F_{min}} \quad (39)$$

where F_{min} and F_{max} are the minimum and maximum fitness scores in the population, respectively. This scaling ensures that solutions with higher fitness undergo finer perturbations, enhancing precision in high-potential areas.

The following exploration, each onlooker bee evaluates the fitness $F_f^{explore}$ of its newly generated solution, incorporating accuracy and redundancy measures. The fitness calculation is expressed as:

$$\begin{aligned} F_f^{explore} &= \eta \cdot Q_f^{explore} \\ &- \zeta \cdot Redundancy(B_f^{explore}) \end{aligned} \quad (40)$$

where $Q_f^{explore}$ denotes the accuracy of the Naive Bayes classifier with the modified feature subset, and $Redundancy(B_f^{explore})$ represents the redundancy penalty for the subset. Weights η and ζ adjust the importance of each component, ensuring the optimized balance between accuracy and feature diversity.

Onlooker bees also assess the diversity $D_f^{explore}$ of their solutions, aiming to avoid convergence on overly similar feature subsets. The diversity score is calculated by:

$$D_f^{explore} = \frac{\sum_{i=1}^M B_{f,i}^{explore}}{M} \quad (41)$$

where $B_{f,i}^{explore}$ is the i -th feature in the modified subset, and M denotes the total feature count. High diversity values prevent redundant solutions, supporting broader exploration of viable feature combinations.

Each onlooker bee's newly generated solution is subject to an acceptance criterion based on its fitness relative to the original subset. The acceptance probability P_f^{accept} is calculated as:

$$P_f^{accept} = \frac{F_f^{explore}}{F_f^{explore} + F_f} \quad (42)$$

where F_f is the fitness of the original solution, ensuring that only solutions with significant improvements are likely to replace previous subsets. This acceptance probability maintains a focus on continuously optimizing feature selections.

Onlooker bees further refine the feature subsets by emphasizing features that demonstrate high utility in improving classifier accuracy. The importance weight $W_{f,m}$ for each feature m within a selected subset is calculated by:

$$W_{f,m} = \frac{F_f \cdot B_{f,m}^{explore}}{\sum_{j=1}^M B_{f,j}^{explore}} \quad (43)$$

Higher weights $W_{f,m}$ indicate features with substantial contributions, enabling the algorithm to prioritize feature combinations with enhanced predictive capabilities.

Based on the importance weight, each feature m in the solution $B_f^{explore}$ is assigned a retention probability $R_{f,m}^{retain}$:

$$R_{f,m}^{retain} = \frac{W_{f,m}}{\sum_{i=1}^M W_{f,i}} \quad (44)$$

This retention probability guides the inclusion of high-impact features in subsequent iterations, supporting the formation of refined, optimized feature subsets.

3.8. Explore New Solutions by Onlooker Bees in ABC-NB

In the ABC-NB algorithm, the phase of Exploring New Solutions by Onlooker Bees is essential for refining the feature selection process. This phase allows onlooker bees to further examine promising food sources (feature subsets) by conducting in-depth exploration around selected high-fitness solutions. This step is integral in expanding the algorithm's search in promising regions, maximizing the likelihood of finding an optimized feature subset with the highest classifier accuracy and minimal redundancy. In a real-world bee colony, onlooker bees are attracted to food sources previously identified as high quality by scout or employed bees. Mirroring this behavior, onlooker bees in ABC-NB evaluate these food sources, introducing minor modifications to generate new solutions that offer incremental improvements. This approach facilitates a thorough exploration of the feature space around each high-fitness solution, refining the classifier's performance.

Onlooker bees explore selected feature subsets by conducting a neighborhood search around each high-fitness food source. Each onlooker bee modifies a feature subset S_f , resulting in a new solution S_f^{new} by introducing minor changes to its binary representation. The modified solution S_f^{new} can be represented as:

$$S_{f,m}^{new} = S_{f,m} + \delta \cdot (S_{f,m} - S_{k,m}) \quad (45)$$

where $S_{f,m}$ represents the m -th feature in the original subset for food source f , and $S_{k,m}$ is the corresponding feature from a randomly selected food source $k \neq f$. The parameter δ is a perturbation factor that controls the magnitude of the modification. By adjusting δ , the algorithm

ensures fine-grained exploration, enabling the generation of closely related feature subsets.

To enhance the exploration of high-fitness solutions, the perturbation factor δ_f is adjusted adaptively based on the fitness score F_f of the food source. This adjustment helps achieve more significant modifications for solutions with moderate fitness while applying finer adjustments for high-fitness solutions. The adaptive perturbation is defined as:

$$\delta_f = \frac{F_f}{\sum_{j=1}^F F_f} \quad (46)$$

where $\sum_{j=1}^F F_f$ represents the total fitness of all solutions in the population. The algorithm encourages more substantial exploration in those regions by assigning a higher weight to solutions with relatively lower fitness.

Each newly generated solution S_f^{new} undergoes a fitness evaluation to determine its effectiveness in improving the Naive Bayes classifier's performance. The fitness F_f^{new} for the modified solution is calculated by:

$$F_f^{new} = \lambda \cdot A_f^{new} \cdot \mu \cdot Redundancy(S_f^{new}) \quad (47)$$

where λ and μ represent the weights assigned to classifier accuracy A_f^{new} and redundancy, respectively. The redundancy term penalizes solutions with overlapping or unnecessary features, ensuring the selected subset maintains a balance of diversity and accuracy.

The likelihood of each onlooker bee retaining a newly explored solution depends on its fitness compared to the original subset. The probability P_f^{retain} of retaining S_f^{new} over S_f is calculated as follows:

$$P_f^{retain} = \frac{F_f^{new}}{F_f^{new} + F_f} \quad (48)$$

Higher values of F_f^{new} increase the probability of the onlooker bee adopting the new solution, facilitating the retention of subsets that offer a higher classifier accuracy and optimal feature selection.

To prevent premature convergence, the algorithm evaluates the diversity D_f^{new} of the new solution. This diversity metric assesses the

distribution of selected features within S_f^{new} , calculated as:

$$D_f^{new} = \frac{\sum_{i=1}^M S_{f,i}^{new}}{M} \quad (49)$$

where M represents the total number of features in the dataset, and $S_{f,i}^{new}$ is the i -th feature in the new subset. By promoting solutions with higher diversity scores, the algorithm maintains a broader exploration of the feature space, avoiding redundancy and overlapping solutions.

During the exploration of new solutions, each feature's impact on the fitness of S_f^{new} is weighted to prioritize impactful features in future iterations. The importance weight $W_{f,m}^{new}$ for each feature m is calculated by:

$$W_{f,m}^{new} = \frac{F_f^{new} \cdot S_{f,i}^{new}}{\sum_{n=1}^M S_{f,n}^{new}} \quad (50)$$

where higher values of $W_{f,m}^{new}$ indicate features with a more significant contribution to classifier performance, guiding future feature subset adjustments.

For each feature S_f^{new} , a selection probability $P_{f,m}^{select}$ is assigned based on its importance weight:

$$P_{f,m}^{select} = \frac{W_{f,m}^{new}}{\sum_{j=1}^M W_{f,j}^{new}} \quad (51)$$

This selection probability reflects the likelihood of retaining high-utility features, enhancing the probability of optimal features contributing to the solution in subsequent iterations. To finalize the exploration phase, each onlooker bee applies a greedy selection criterion for retaining solutions that maximize classifier accuracy and minimize redundancy. This selection criterion $C_f^{explore}$ combines fitness and diversity metrics:

$$C_f^{explore} = F_f^{new} + k \cdot D_f^{new} \quad (52)$$

where k represents the weight assigned to diversity, ensuring that solutions with higher classifier performance and greater diversity are prioritized.

3.9. Evaluate the Fitness of Onlooker Bee Solutions in ABC-NB

In the ABC-NB algorithm, evaluating the fitness of solutions explored by onlooker bees is crucial for determining the effectiveness of newly identified feature subsets. This evaluation process mimics how bees assess the quality of promising food sources, ensuring that only those providing the most nourishment are revisited and utilized by the colony. By carefully evaluating the fitness of onlooker bee solutions, the algorithm identifies which feature subsets offer optimized performance for the Naive Bayes classifier, advancing the search toward optimal solutions. Evaluating the fitness of onlooker bee solutions allows a thorough comparison of classifier accuracy and feature subset efficiency. In this phase, each onlooker bee's modified solution undergoes an assessment based on multiple factors, including redundancy minimization, diversity enhancement, and the influence of individual features, ensuring a holistic approach to selecting high-performing subsets.

Each onlooker bee solution is evaluated based on its effectiveness in improving the classifier's accuracy. This effectiveness is quantified by calculating a fitness score F_o^{new} for each solution. The fitness score of each onlooker bee solution is calculated as follows:

$$F_o^{new} = \alpha \cdot C_o^{new} - \beta \cdot R_o^{new} \quad (52)$$

where C_o^{new} represents the classifier's predictive capability using the feature subset O selected by the onlooker bee. R_o^{new} represents the redundancy within the subset. The weights α and β ensure balanced consideration of classifier performance and feature overlap reduction, promoting an optimized subset.

Minimizing redundancy among selected features ensures that unique and complementary features are retained. For each onlooker bee solution, the redundancy penalty R_o^{new} is evaluated as:

$$R_o^{new} = \sum_{i=1}^{M-1} |o_i - o_{i+1}| \quad (53)$$

where o_i and o_{i+1} denote consecutive features in the subset selected by onlooker bee O . The summation term penalizes consecutive duplicate features, reducing overlap and ensuring the subset includes only distinct and impactful features.

Diversity among selected features is essential to prevent premature convergence to local optima. A high diversity score ensures the feature subsets explored by onlooker bees cover a broad range of attributes. The diversity score D_o^{new} for each onlooker bee solution is calculated as follows:

$$D_o^{new} = \frac{\sum_{j=1}^M O_j}{M} \quad (54)$$

where O_j denotes the presence of the j -th feature in the subset O explored by the onlooker bee, M represents the total number of features. A higher diversity score indicates a varied selection of features, contributing to more comprehensive solution space exploration.

To evaluate each onlooker bee solution comprehensively, a weighted fitness function W_o^{new} is applied, integrating accuracy, redundancy reduction, and diversity. This weighted fitness is calculated as follows:

$$W_o^{new} = \lambda \cdot F_o^{new} + \gamma \cdot D_o^{new} \quad (55)$$

where λ and γ are weights that adjust the influence of classifier performance and diversity on the solution's overall fitness, this function guides the algorithm to prioritize solutions that balance high classifier accuracy with unique and varied feature subsets.

Each feature within an onlooker bee solution is assigned an importance score to quantify its contribution to the classifier's performance. The importance score $I_{o,j}^{new}$ for each feature j in solution O is calculated by:

$$I_{o,j}^{new} = \frac{F_o^{new} \cdot O_j}{\sum_{k=1}^M O_k} \quad (56)$$

where O_k represents the presence of the k -th feature in the subset, and F_o^{new} is the fitness score of the onlooker bee solution. Higher importance scores reflect features that provide more significant predictive value for the classifier.

After calculating the weighted fitness of each solution, a selection probability P_o^{select} is assigned to each onlooker bee solution. This probability influences the likelihood of retaining the solution based on its relative fitness compared to other solutions:

$$P_o^{select} = \frac{W_o^{new}}{\sum_{p=1}^O W_p^{new}} \quad (57)$$

where W_o^{new} denotes the weighted fitness of solution O , and O is the total number of onlooker bee solutions. Solutions with higher W_o^{new} values are more likely to be selected, ensuring only the most promising solutions advance.

To determine which onlooker bee solutions should replace previous solutions, the algorithm applies a retention criterion C_o^{retain} . This criterion combines fitness, diversity, and importance scores, ensuring that only high-quality solutions are retained. The retention criterion is calculated as:

$$C_o^{retain} = F_o^{new} + \delta \cdot D_o^{new} \quad (58)$$

where δ adjusts the influence of diversity, favoring solutions with a balanced representation of unique and impactful features. Solutions that meet or exceed a predefined threshold based on C_o^{retain} are retained, supporting the selection of optimized feature subsets.

Each feature in an onlooker bee solution is assigned a retention probability $R_{o,j}^{retain}$ based on its importance score:

$$R_{o,j}^{retain} = \frac{I_{o,j}^{new}}{\sum_{k=1}^M I_{o,k}^{new}} \quad (59)$$

This retention probability ensures that features contributing significantly to classifier performance are more likely to be preserved, facilitating the development of optimized solutions.

3.10. Scout Bee Phase in ABC-NB

The Scout Bee Phase in the ABC-NB algorithm focuses on reintroducing diversity into the population of solutions, ensuring that the optimization process does not converge prematurely. Scout bees mimic the behavior of real scout bees, which explore new areas when food sources (feature subsets) are depleted or provide minimal value. In ABC-NB, scout bees replace solutions that exhibit stagnation or fail to yield improvements, facilitating exploration in fresh, unexplored regions of the feature space. By incorporating this phase, the algorithm maintains a balance between exploitation and exploration, fostering the identification of more optimized

solutions. In the context of ABC-NB, scout bees generate new feature subsets by initiating random modifications to existing ones or creating entirely novel feature combinations. This approach prevents the algorithm from becoming trapped in local optima, ensuring a comprehensive search across the feature space to achieve the most effective configurations for the Naive Bayes classifier.

The algorithm first identifies stagnant solutions within the population to activate scout bees. A solution is considered stagnant if it has not improved over several cycles or iterations. The stagnation count S_f^{scout} for each solution f is defined as follows:

$$S_f^{scout} = S_f^{scout} + 1 \quad (60)$$

If S_f^{scout} reaches a predefined threshold θ , the solution S_f is deemed stagnant and becomes a target for replacement by scout bees. This threshold value θ acts as a control mechanism, ensuring solutions that do not contribute to further optimization are replaced. Once a stagnant solution is identified, scout bees generate new feature subsets to replace it, restoring diversity to the solution pool. The new feature subset S_f^{scout} is generated by assigning random values to each feature m , ensuring a novel solution is created. The new solution S_f^{scout} can be expressed as:

$$S_f^{scout} = random(0,1) \quad (61)$$

In this context, each feature in S_f^{scout} a binary value of 0 or 1 is randomly assigned, where 1 indicates inclusion in the subset, and 0 indicates exclusion. This randomization injects fresh configurations into the algorithm, encouraging broader feature space exploration. After generating a new solution, scout bees assess the fitness of S_f^{scout} to determine its contribution to the optimization process. The fitness F_f^{scout} of the new scout bee solution is calculated as follows:

$$F_f^{scout} = \alpha \cdot Q_f^{scout} - \beta \cdot Redundancy(S_f^{scout}) \quad (62)$$

where Q_f^{scout} represents the classification capability of the Naive Bayes model using the feature subset S_f^{scout} , while

$Redundancy(S_f^{scout})$ penalizes redundant features. The weights α and β prioritize classifier accuracy and feature uniqueness, enabling scout bees to contribute high-quality solutions.

To ensure the new solution S_f^{scout} provides a unique configuration, its diversity D_f^{scout} is evaluated relative to other solutions. The diversity score of S_f^{scout} is calculated as:

$$D_f^{scout} = \frac{\sum_{i=1}^M S_{f,i}^{scout}}{M} \quad (63)$$

where $S_{f,i}^{scout}$ denotes the inclusion status of the i -th feature in the new solution, and M is the total number of features. Higher diversity scores signify that the scout bee solution offers a novel feature combination, ensuring a broader exploration across different feature subsets.

The scout bee solution replaces the original stagnant solution if its fitness F_f^{scout} surpasses a minimum fitness threshold, thus adding value to the solution pool. The replacement condition is governed by:

$$S_f = \begin{cases} S_f^{scout} & \text{if } F_f^{scout} > F_{min} \\ S_f & \text{otherwise} \end{cases} \quad (64)$$

where F_{min} is the minimum acceptable fitness value for a solution to contribute to the population. This condition ensures that only beneficial solutions are retained, maintaining the algorithm's focus on optimal feature subsets.

The likelihood of replacing a stagnant solution depends on its stagnation count S_f^{scout} and its relative fitness within the population. The replacement probability $P_f^{replace}$ is calculated as:

$$P_f^{replace} = \frac{S_f^{scout}}{\sum_{k=1}^F S_k^{scout}} \quad (65)$$

This probability ensures that solutions with a higher stagnation count are prioritized for replacement, enhancing diversity by continuously introducing new feature configurations. Each feature in a scout bee solution is assigned an importance weight based on its contribution to the fitness of S_f^{scout} . The importance weight $W_{f,m}^{scout}$ for each feature m is determined as:

$$W_{f,m}^{scout} = \frac{F_f^{scout} \cdot S_{f,m}^{scout}}{\sum_{j=1}^M S_{f,j}^{scout}} \quad (66)$$

Higher values of $W_{f,m}^{scout}$ indicate features that contribute significantly to the classifier's performance, helping guide future selection and optimization steps. The final decision to retain a scout bee solution depends on a retention criterion C_f^{scout} , which integrates both fitness and diversity scores. This criterion is expressed as:

$$C_f^{scout} = F_f^{scout} + \gamma \cdot D_f^{scout} \quad (67)$$

where γ is a weight that adjusts the influence of diversity, ensuring that retained solutions provide a valuable, varied contribution to the solution pool.

3.11. Update Best Solution in ABC-NB

The Update Best Solution step in the ABC-NB algorithm consolidates the best results achieved from the previous phases, serving as a critical checkpoint to retain only the most optimized feature subset. This phase parallels the behavior of bees returning to the hive with knowledge of the richest nectar sources, sharing it with the colony to prioritize resources effectively. In ABC-NB, updating the best solution ensures that only the most promising feature subset, which has demonstrated superior classifier accuracy and minimized redundancy, is retained as the reference solution for further iterations or as the final output. The purpose of updating the best solution in each cycle lies in progressively converging toward an optimal feature subset that yields a Naive Bayes classifier with high predictive power. By consistently updating the best solution, ABC-NB ensures continuous improvement in classifier performance, focusing on optimized selections throughout the process.

At the beginning of the Update Best Solution step, the algorithm evaluates all solutions generated by employed bees, onlooker bees, and scout bees to identify the most optimized feature subset in the current cycle. The fitness $F_{best}^{current}$ of the best solution identified in this cycle is calculated as follows:

$$F_{best}^{current} = \max\{F_f | f = 1, 2, \dots, F\} \quad (68)$$

where F_f represents the fitness score of the f -th solution in the population. By identifying the maximum fitness score, this equation helps determine the best-performing feature subset within the current set of solutions.

Once the best solution of the current cycle is identified, it undergoes a comparison with the previously stored best solution. If the current best solution's fitness $F_{best}^{current}$ exceeds the fitness of the previously stored best solution F_{best}^{prev} , the algorithm updates the best solution to reflect the current cycle's optimal subset:

$$S_{best} = \begin{cases} S_{best}^{current} & \text{if } F_{best}^{current} > F_{best}^{prev} \\ S_{best}^{prev} & \text{otherwise} \end{cases} \quad (69)$$

where $S_{best}^{current}$ represents the feature subset of the current best solution, while S_{best}^{prev} denotes the previously stored best subset. This comparison ensures that only solutions demonstrating improved performance are retained as the updated best solution.

To quantify the improvement achieved by the current best solution over previous iterations, the fitness improvement metric $I_{improvement}$ is introduced. This metric is calculated by determining the relative improvement of $F_{best}^{current}$ over F_{best}^{prev} :

$$I_{improvement} = \frac{F_{best}^{current} - F_{best}^{prev}}{F_{best}^{prev}} \quad (70)$$

where a positive value of $I_{improvement}$ indicates a successful iteration in which the current best solution outperforms the previous one. Monitoring this improvement metric allows the algorithm to track progress, gauging the effectiveness of each iteration.

To understand the contribution of each feature in the updated best solution, the algorithm assigns a contribution weight $W_{best,m}$ to each feature m based on its impact on the classifier's performance. The contribution weight for feature m in the best solution is given by:

$$W_{best,m} = \frac{F_{best}^{current} - S_{best,m}}{\sum_{i=1}^M S_{best,i}} \quad (71)$$

where $S_{best,m}$ represents the presence of the m -th feature in the best solution subset, and M is the total number of features. This weighting helps identify high-impact features, refining the selection for future optimization cycles.

To ensure the updated best solution maintains feature diversity, a diversity score D_{best}

is calculated to assess the variety of selected features within S_{best} . This diversity score is defined as:

$$D_{best} = \frac{\sum_{i=1}^M S_{best,i}}{M} \quad (72)$$

where $S_{best,i}$ represents the selection status of each feature in the best solution. A higher diversity score suggests a balanced and varied feature subset, reducing redundancy while promoting comprehensive feature coverage.

The probability P_{retain} of retaining the best solution for future cycles is calculated to account for its relative superiority over other solutions. This retention probability is given by:

$$P_{retain} = \frac{F_{best}^{current}}{\sum_{k=1}^F F_k} \quad (73)$$

where F_k denotes the fitness score of each solution in the population. A higher retention probability signifies that the best solution possesses greater relative importance, underscoring its value as the reference subset for further optimization.

To assess the overall quality of the updated best solution, a final evaluation metric E_{best} integrates both fitness and diversity metrics:

$$E_{best} = F_{best}^{current} + \delta \cdot D_{best} \quad (74)$$

where δ represents the weight assigned to diversity, the resulting evaluation metric ensures that the best solution combines high classifier accuracy and minimized redundancy, reinforcing the selection of optimized feature subsets.

3.12. Repeat Optimization Cycles in ABC-NB

In the ABC-NB algorithm, the Repeat Optimization Cycles step ensures that the optimization process progresses toward a fully optimized solution through iterative cycles. Each cycle leverages the distinct behaviors of employed, onlooker, and scout bees to refine the feature subset, continually enhancing the performance of the Naive Bayes classifier. This repetition mirrors how bees revisit high-quality food sources over time, incrementally improving their yield by discovering richer patches or refining known sources. The repeated optimization cycles improve the classifier's predictive accuracy by identifying and refining the most effective combinations of features.

The algorithm begins by establishing a predefined maximum number of optimization

cycles, C_{max} , which limits the iterations to avoid overfitting or excessive computation. The current cycle C is incremented in each loop until it reaches C_{max} . This limit is expressed as follows:

$$C_{max} = \text{user - defined parameter} \quad (75)$$

where C_{max} ensures the optimization process converges within a reasonable time frame. Setting C_{max} also helps balance computational efficiency with accuracy, allowing the algorithm to focus on improvement over a finite number of cycles.

In each cycle, the fitness of the population F is recalculated based on the feature subsets selected by the employed, onlooker, and scout bees.

The recalculated fitness values $F_f^{(C)}$ for each feature subset f in cycle C can be represented as:

$$F_f^{(C)} = \alpha \cdot Q_f^{(C)} - \beta \cdot R_f^{(C)} \quad (76)$$

where $Q_f^{(C)}$ denotes the classification quality of the Naive Bayes model, and $R_f^{(C)}$ represents the redundancy penalty for solution f in the current cycle. By continuously updating fitness values, the algorithm adapts to new solutions generated by the bee groups, maintaining an optimized population at each step.

As the optimization progresses, the probability of selecting specific features changes dynamically. This probability $P_{f,m}^{(C)}$ for feature m in solution f during cycle C is determined as:

$$P_{f,m}^{(C)} = \frac{W_{f,m}^{(C)}}{\sum_{j=1}^M W_{f,j}^{(C)}} \quad (77)$$

where $W_{f,m}^{(C)}$ represents the weight of each feature in the current cycle. This adaptive feature selection probability helps the algorithm focus on the most promising features for each cycle, further optimizing the feature subset chosen.

To assess the overall improvement achieved in each cycle, the algorithm calculates the average improvement $I^{(C)}$ in fitness scores compared to the previous cycle. The improvement metric is given by:

$$I^{(C)} = \frac{\sum_{f=1}^F (F_f^{(C)} - F_f^{(C-1)})}{F} \quad (78)$$

where F denotes the total number of solutions in the population, this improvement metric indicates progress in each cycle, guiding the algorithm to continue only if meaningful gains are detected.

A convergence criterion is employed to determine if the optimization process should terminate before reaching C_{max} . The criterion checks if the average fitness improvement $I^{(C)}$ falls below a predefined threshold ϵ :

$$I^{(C)} < \epsilon \quad (79)$$

If the improvement $I^{(C)}$ is less than ϵ , the algorithm terminates early, indicating that additional cycles yield minimal or no benefit. This threshold-based stopping criterion helps avoid unnecessary computations, ensuring efficiency in the optimization process.

The algorithm maintains a balance between exploration and exploitation across cycles. Solutions with relatively high fitness values are exploited further, while scout bees introduce exploration by injecting new feature subsets. The balance factor $B^{(C)}$ for each cycle can be expressed as:

$$B^{(C)} = \frac{\sum_{f=1}^F F_f^{(C)}}{F} \quad (80)$$

This balance factor evaluates the overall fitness level within the population. Higher values of $B^{(C)}$ indicate a shift toward exploitation of high-fitness solutions, while lower values promote exploration. The best solution $S_{best}^{(C)}$ across all cycles is continually updated by comparing each cycle's best solution with previously stored solutions. The updated best solution for cycle C is determined as:

$$S_{best}^{(C)} = \begin{cases} S_{best}^{current} & \text{if } F_{best}^{current} > F_{best}^{(C-1)} \\ S_{best}^{(C-1)} & \text{otherwise} \end{cases} \quad (81)$$

where $F_{best}^{current}$ is the highest fitness achieved in the current cycle. This iterative update ensures the final solution reflects the optimal feature subset discovered throughout all cycles.

To track each feature's impact over cycles, a cumulative contribution score C_m^{total} is calculated

for feature m based on its contribution across all cycles:

$$C_m^{total} = \sum_{C=1}^{C_{max}} W_{f,m}^{(C)} \quad (82)$$

This cumulative score identifies consistently impactful features, assisting the algorithm in refining feature selection over time by recognizing the features that contribute most significantly to classifier performance. Upon completing the final cycle, a comprehensive evaluation metric E_{final} is calculated to assess the quality of the optimized feature subset:

$$E_{final} = F_{best}^{(C)} + \delta \cdot D_{best}^{(C)} \quad (83)$$

where δ adjusts the weight of diversity $D_{best}^{(C)}$ in the final evaluation. This metric ensures the final solution combines high classifier accuracy with a balanced, diverse feature subset, signifying an optimized outcome.

3.13. Select Optimal Feature Subset in ABC-NB

The final phase in the ABC-NB algorithm, the Select Optimal Feature Subset step, focuses on identifying the most practical combination of features from the solutions refined through previous optimization cycles. This optimal subset represents the result of collective exploration and exploitation by employed, onlooker, and scout bees, aimed at achieving maximum classifier performance. Selecting this subset is akin to how bees choose the most nourishing food source based on collective assessment, maximizing the value derived from their environment. The purpose of selecting the optimal feature subset is to finalize a solution that consistently enhances classifier accuracy while minimizing feature redundancy, ultimately improving the efficiency of the Naive Bayes model in terms of computational cost and predictive reliability.

To ensure that only the most influential features are selected, the algorithm calculates an aggregated fitness score F_{opt} for each feature across all cycles. The fitness score of a feature m is defined by summing its weighted contributions $W_{f,m}^{(C)}$ overall optimization cycles C :

$$F_{opt,m} = \sum_{C=1}^{C_{max}} W_{f,m}^{(C)} \quad (84)$$

where $W_{f,m}^{(C)}$ represents the contribution weight of feature m in each cycle. This cumulative score helps identify features that consistently contribute positively to the model's performance, making them prime candidates for the optimal subset.

To narrow down the feature selection to the most impactful subset, the algorithm establishes a selection threshold T_{select} , calculated as a percentage of the maximum aggregated fitness score among features. Features with fitness scores above T_{select} are selected for the optimal subset:

$$T_{select} = \lambda \cdot \max(F_{opt,1}, F_{opt,2}, \dots, F_{opt,M}) \quad (85)$$

where λ is a predefined threshold parameter set by the user. Using this selection threshold, the algorithm ensures that only features substantially impacting classifier accuracy are included in the final subset.

After selecting the features based on T_{select} , the algorithm evaluates redundancy within the selected subset. The redundancy R_{opt} of the subset is computed as follows:

$$R_{opt} = \sum_{i=1}^{M_{opt}-1} |S_{opt,i} - S_{opt,i+1}| \quad (86)$$

where $S_{opt,i}$ denotes the presence of the i -th feature in the selected optimal subset, and M_{opt} is the total number of features in this subset. Lower values of R_{opt} indicate reduced redundancy, ensuring a unique and effective set of features is finalized.

To maintain a broad range of characteristics in the optimal subset, the algorithm calculates a diversity score D_{opt} based on the proportion of features included from the original set:

$$D_{opt} = \frac{\sum_{j=1}^M S_{opt,j}}{M} \quad (87)$$

where $S_{opt,j}$ indicates the inclusion status of feature j in the optimal subset, and M represents the total number of features. A high diversity score reflects a varied selection, which is essential for a well-rounded classifier performance.

The algorithm introduces a final evaluation metric E_{final} to validate the overall quality of the selected subset. This metric combines

the subset's fitness score, redundancy, and diversity, ensuring a comprehensive selection. The final evaluation metric is given by:

$$E_{final} = F_{opt} - \alpha \cdot R_{opt} + \beta \cdot D_{opt} \quad (88)$$

where α and β are weighting parameters prioritizing minimized redundancy and maximized diversity. A higher E_{final} score indicates a more balanced and optimized subset for the Naive Bayes classifier.

Each feature's probability of inclusion $P_{include,m}$ in the optimal subset is calculated based on its aggregated fitness and redundancy scores, ensuring a robust selection criterion. This probability is defined as:

$$P_{include,m} = \frac{F_{opt,m}}{F_{opt}} \quad (89)$$

where F_{opt} is the total fitness of all features in the optimal subset. This probability score helps validate the final inclusion of each feature, confirming that each selected feature meaningfully contributes to the model's performance.

For each feature in the optimal subset, a final contribution weight $W_{final,m}$ is assigned based on its cumulative impact across cycles:

$$W_{final,m} = \frac{F_{opt,m}}{\sum_{k=1}^{M_{opt}} F_{opt,k}} \quad (90)$$

where M_{opt} represents the number of features in the optimal subset. This weight reflects the relative significance of each feature within the final set, guiding feature prioritization.

3.14. Train Naive Bayes on Optimized Subset in ABC-NB

The final step in the ABC-NB algorithm involves training the Naive Bayes classifier on the optimized feature subset obtained after completing all optimization cycles. This stage aims to deploy the refined features to create a robust classifier that benefits from reduced dimensionality, minimized redundancy, and improved accuracy. Utilizing the optimized subset enables the classifier to perform more efficiently, relying only on features that contribute meaningfully to its predictive capabilities. This phase parallels how bees selectively forage from only the most rewarding sources, focusing on subsets that yield the best results. The training involves calculating the probabilities for each feature in the subset and constructing a classifier that leverages these

probabilities to predict class labels. By training on a subset that has been carefully curated, the Naive Bayes classifier achieves a balance between computational efficiency and predictive performance.

In the Naive Bayes framework, each feature's conditional probability given the class is essential for making predictions. For a feature m in the optimized subset, the conditional probability $P(S_m|C)$ given a class C is calculated as:

$$P(S_m|C) = \frac{\sum_{i=1}^N I(S_{m,i} = 1 \wedge y_i = C)}{\sum_{i=1}^N I(y_i = C)} \quad (91)$$

where N represents the total number of training samples, $S_{m,i}$ denotes the value of feature m in the i -th sample, and y_i indicates the class label of the i -th sample. The indicator function I equals 1 when the conditions are met and zero otherwise. Calculating this probability for each feature in the subset allows the model to incorporate optimized feature-specific information during classification.

To implement the Naive Bayes classifier, the prior probability $P(C)$ for each class C is computed based on the occurrence of each class in the training data. The prior probability for class C is defined by:

$$P(C) = \frac{\sum_{i=1}^N I(y_i = C)}{N} \quad (92)$$

This equation provides the proportion of each class in the training set, which is used to weight the evidence provided by the features in the optimized subset. By establishing the class priors, the Naive Bayes model ensures balanced predictions that reflect the distribution of class labels in the data. For each sample, the likelihood L of observing the feature values given a class C is calculated by multiplying the conditional probabilities for all features in the optimized subset. The possibility for a sample x belonging to class C is expressed as:

$$L(C|x) = \prod_{m \in M_{opt}} P(S_m|C)^{S_m} \quad (93)$$

where M_{opt} represents the set of selected features in the optimized subset, and S_m is the presence or absence of feature m in the sample x . This likelihood measure, derived from the optimized

subset, facilitates accurate class probability estimation for each sample.

In Naive Bayes classification, the posterior probability $P(C|x)$ for each class is essential for assigning a class label to each sample. Using Bayes' theorem, the posterior probability is computed as follows:

$$P(C|x) = \frac{P(C) \cdot L(C|x)}{\sum_k P(k) \cdot L(k|x)} \quad (94)$$

where k iterates over all classes, this formula calculates the posterior probability of class C given the feature values in x . The model selects the class with the highest posterior probability as the predicted class for each sample.

The classifier uses the log-likelihood instead of the direct product of probabilities to improve computational stability, particularly with small probability values. The log-likelihood $\log L(C|x)$ for a class C given sample x is calculated by:

$$\log L(C|x) = \sum_{m \in M_{opt}} S_m \cdot \log(P(S_m|C)) \quad (95)$$

This approach avoids the underflow issues of multiplying multiple small probabilities and ensures numerical stability in probability calculations. The final classification decision for each sample relies on a decision rule that selects the class \hat{C} with the maximum posterior probability:

$$\hat{C} = \arg \max_C P(C|x) \quad (96)$$

This decision rule assigns the most probable class to each sample, ensuring that the classifier optimally utilizes the refined feature subset in making predictions. After training the Naive Bayes classifier on the optimized subset, the model's accuracy is assessed by comparing predicted and actual labels. The training accuracy A_{train} is computed as:

$$A_{train} = \frac{\sum_{i=1}^N I(\hat{y}_i = y_i)}{N} \quad (97)$$

where \hat{y}_i represents the predicted label for the i -th sample, and y_i is the actual class label. This accuracy metric validates the model's performance

on the training data, providing insights into its predictive power when using the optimized subset.

The ABC-NB algorithm integrates the swarm intelligence principles of the Artificial Bee Colony (ABC) algorithm with the classification power of the Naive Bayes model. The process involves optimizing the feature subset selection to improve classifier performance by maximizing accuracy and minimizing redundancy. Each phase leverages the collective foraging behaviors of bees—exploration by employed, onlooker, and scout bees—to refine the selection of features that provide the most predictive value for the Naive Bayes classifier.

Algorithm: ABC-NB

Input: Initial dataset with features X and labels y , population size F , maximum iterations C_{max} , fitness threshold ϵ .

Output: Optimized Naive Bayes model trained on selected feature subset.

Procedure:

1. Initialize Population:

- Randomly generate an initial population of feature subsets S_f for $f = 1, 2, \dots, F$.
- Calculate initial fitness F_f of each subset based on classification accuracy and redundancy penalties.

2. Evaluate Fitness:

- Compute the fitness F_f of each feature subset using the Naive Bayes classifier's performance and redundancy minimization.

3. Send Employed Bees:

- For each feature subset S_f , perform a neighborhood search to modify features.
- Evaluate the fitness of modified solutions, retaining improved solutions.

4. Assess New Solutions:

- Compare new solutions generated by employed bees against previous solutions.
- Retain the solution with higher fitness for each subset.

5. Greedy Selection:

- Perform greedy selection to keep only high-fitness solutions,

discarding less effective subsets.

6. Send Onlooker Bees:

- Calculate the selection probability for each solution based on fitness.
- Assign onlooker bees to explore promising solutions for modifying feature subsets.

7. Explore New Solutions by Onlooker Bees:

- Modify selected feature subsets to generate new solutions.
- Evaluate fitness and diversity of new solutions, retaining those with higher fitness.

8. Evaluate the Fitness of Onlooker Bee Solutions:

- Assess the fitness of solutions discovered by onlooker bees.
- Retain high-performing solutions based on fitness and diversity metrics.

9. Scout Bee Phase:

- Identify stagnant solutions with minimal improvement over multiple iterations.
- Replace stagnant solutions with new randomly generated subsets.

10. Update Best Solution:

- Track and store each cycle's best solution (highest fitness).
- Retain only the solution with the highest fitness as the current best.

11. Repeat Optimization Cycles:

- Increment the cycle counter and repeat steps 3–10 until the maximum cycle limit C_{max} is reached or fitness improvement falls below threshold ϵ .

12. Select Optimal Feature Subset:

- Identify the optimal feature subset with the highest fitness from the final population.
- Calculate aggregate fitness, redundancy, and diversity metrics to finalize the subset.

13. Train Naive Bayes on Optimized Subset:

- Use the optimized feature subset to train the Naive Bayes classifier.
- Compute conditional probabilities for each feature, train on class priors, and calculate posterior

probabilities for prediction.

4. DATASET

Curated by Dr. Zahra Alizadeh Sani and her team, this dataset is an expansive collection of clinical records aimed at supporting non-invasive strategies for detecting coronary artery disease. It comprises 303 individual patient entries, with each record featuring 54 distinct variables organized into four domains. These domains include basic demographic details; clinical signs and examination results; electrocardiographic measurements that reflect heart electrical activity; and data from laboratory tests along with echocardiographic evaluations. Patients are categorized based on the degree of coronary narrowing, with a 50% reduction in vessel diameter used as the diagnostic cutoff. The dataset's high integrity—with no missing values—facilitates rigorous statistical analysis and robust model development. Researchers have employed it to explore various classification techniques, ranging from decision trees and support vector machines to logistic regression and ensemble models. Advanced feature selection and optimization methods have been integrated to isolate the most relevant predictors, thereby boosting diagnostic performance and model interpretability. Publicly accessible through established repositories, this dataset not only underpins reproducible research but also drives innovation in developing clinical decision support systems. Its comprehensive nature has made it a cornerstone resource in advancing early detection protocols and improving risk stratification in cardiovascular medicine.

5. RESULTS AND DISCUSSIONS

5.1. Jaccard Index Analysis

The Jaccard Index is a fundamental measure in evaluating the effectiveness of classification models, especially in CAD prediction, where accuracy in distinguishing between cases directly impacts medical decision-making. A higher Jaccard Index signifies a model's ability to classify CAD cases while correctly minimizing misclassification errors. Figure 1 compares the Jaccard Index for ABC-NB, TLV-MLF, and H-HHO with numerical values outlined in Table 1. The classification models are presented on the x-axis, while their corresponding Jaccard Index percentages are plotted on the y-axis.

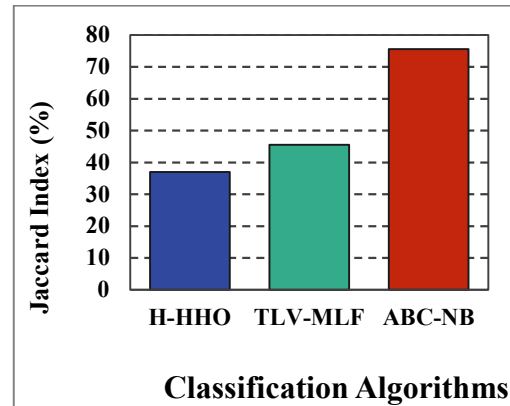


Figure 1: Jaccard Index Score Comparison for ABC-NB and State-of-the-Art Algorithms

H-HHO demonstrates the weakest performance, with a Jaccard Index of 36.986%. The model's primary limitation lies in its inability to differentiate between overlapping feature distributions consistently. The hybridized Harris Hawks Optimization struggles with premature convergence, preventing it from thoroughly refining classification boundaries. This leads to an increased number of false positives and negatives, which reduces overall predictive reliability in CAD diagnosis. TLV-MLF, registering a Jaccard Index of 45.588%, exhibits a modest improvement over H-HHO. The ensemble feature selection mechanism enhances the model's classification performance to some degree, yet its dependence on rigid statistical selection introduces variability. The model occasionally misclassifies critical cases due to an imbalanced weighting of feature significance, leading to inconsistency in CAD risk assessment.

ABC-NB dominates the comparison, achieving a remarkable Jaccard Index of 75.568%. The model's success is rooted in its bio-inspired feature selection, where the Artificial Bee Colony algorithm optimizes feature importance dynamically. Unlike H-HHO and TLV-MLF, ABC-NB continuously refines its feature selection process, ensuring that only the most relevant attributes influence classification. This results in a more stable and robust decision-making process. The probabilistic nature of Naïve Bayes, when coupled with intelligent feature selection, minimizes noise while enhancing classification efficiency, providing an optimized trade-off between sensitivity and specificity.

Table 5.1: Jaccard Index Values of ABC-NB vs. State-of-the-Art Algorithms

Classification Algorithms	Jaccard Index (%)
H-HHO	36.986
TLV-MLF	45.588
ABC-NB	75.568

The sharp performance contrast between ABC-NB and the other models underscores the importance of integrating intelligent optimization strategies in CAD classification. By refining feature selection dynamically rather than relying on fixed statistical assumptions, ABC-NB demonstrates its superiority in handling complex medical datasets, offering a more reliable approach for accurate CAD prediction.

5.14.2. Diagnostic Utility Index Analysis

The Diagnostic Utility Index (DUI) serves as a critical benchmark for evaluating the practical effectiveness of classification models in medical diagnosis. A higher DUI signifies an algorithm's ability to provide meaningful and reliable classifications, reducing false positives and negatives in CAD detection. Figure 2 presents a comparative analysis of the DUI performance for ABC-NB, TLV-MLF, and H-HHO, with corresponding numerical values in Table 2. The classification models are represented along the x-axis, while the y-axis indicates their respective DUI percentages.

H-HHO records the lowest DUI score at 3.196%, highlighting its inefficiency in distinguishing true CAD cases from non-CAD instances. The primary shortcoming of H-HHO stems from its lack of dynamic feature selection, leading to poorly optimized classification thresholds. The model exhibits instability in high-dimensional datasets, often misclassifying borderline cases due to its premature convergence behavior. This results in a limited ability to contribute meaningful diagnostic insights, reducing its overall utility in CAD risk assessment.

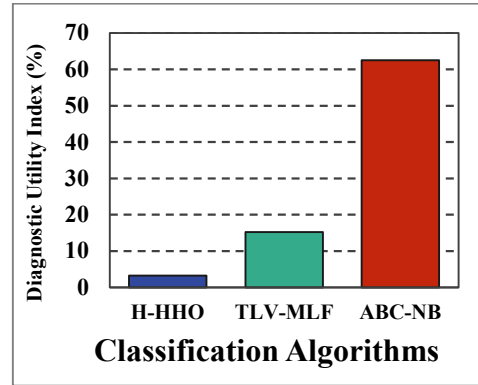


Figure 2: Diagnostic Utility Index Performance of ABC-NB vs. State-of-the-Art Algorithms

TLV-MLF performs slightly better, achieving a DUI of 15.196%, owing to its two-layer voting mechanism that refines feature selection. However, TLV-MLF struggles with inconsistencies in classifier weight allocation, which sometimes skews predictions. While it marginally improves CAD classification accuracy, its rigid dependence on statistical selection methods prevents it from fully adapting to complex and heterogeneous datasets. This limitation leads to a moderate diagnostic utility, failing to achieve the precision required for real-world clinical decision-making.

With a remarkable DUI of 62.500%, ABC-NB outperforms both models significantly, establishing itself as a far more reliable diagnostic tool. The strength of ABC-NB lies in its Artificial Bee Colony (ABC)-driven feature selection, which optimizes the probabilistic nature of Naïve Bayes. Unlike the static feature selection methods of TLV-MLF, ABC-NB continuously refines its selection criteria, prioritizing only the most relevant CAD indicators while filtering out redundant data. This dynamic approach enhances classification reliability, reducing the likelihood of diagnostic errors. Furthermore, the adaptability of the ABC algorithm allows the model to fine-tune feature weights iteratively, ensuring a balanced trade-off between sensitivity and specificity.

Table 2: Computed Diagnostic Utility Index for ABC-NB and State-of-the-Art Algorithms

Classification Algorithms	Diagnostic Utility Index (%)
H-HHO	3.196
TLV-MLF	15.196
ABC-NB	62.500

The substantial performance gap between ABC-NB and the other models underscores the critical role of adaptive bio-inspired optimization in medical classification. By leveraging swarm ABC-NB intelligence for feature refinement, ABC-NB enhances predictive accuracy and significantly boosts the diagnostic value of CAD classification. This highlights its potential as a superior decision-support tool in healthcare applications, where precise and dependable classification models are essential for effective patient management.

5.2. Precision Analysis

Precision is a vital metric in CAD prediction, measuring the proportion of correctly identified CAD cases among all predicted positive instances. A higher precision score indicates fewer false positives, crucial in reducing unnecessary medical interventions. Figure 3 illustrates the precision score comparison for ABC-NB against H-HHO and TLV-MLF, with corresponding numerical values detailed in Table.3. The x-axis represents the classification algorithms, while the y-axis shows their respective precision percentages.

H-HHO, with a precision of 52.258%, struggles to accurately classify CAD cases without mislabeling a significant portion of negative cases as positive. The primary weakness of H-HHO lies in its inconsistent feature selection, which leads to an imbalanced classifier decision boundary. The algorithm’s search mechanism often lacks fine-grained adaptation, making it prone to overfitting certain patterns while ignoring subtle variations in CAD-positive cases. This results in a high false-positive rate, reducing its effectiveness in clinical applications.

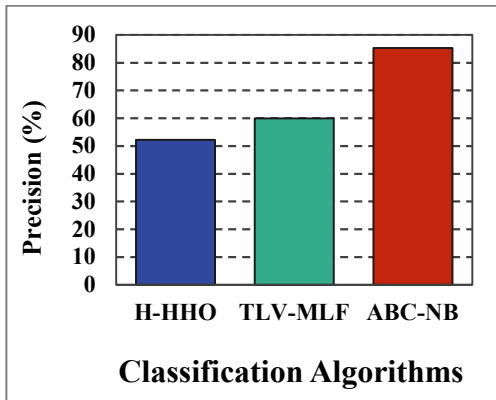


Figure 3: Precision Score Comparison Between ABC-NB and State-of-the-Art Algorithms

TLV-MLF demonstrates a moderate improvement, achieving 60.000% precision. Its two-layer voting mechanism refines feature selection, allowing it to distinguish CAD cases more effectively than H-HHO. However, its reliance on a fixed voting strategy introduces instability, as certain classifiers within the ensemble may not always generalize well across different datasets. The model occasionally misclassifies cases due to fluctuations in feature importance ranking, limiting its overall precision consistency.

ABC-NB dominates the comparison with an impressive precision score of 85.256%, showcasing its capability to minimize false positives effectively. This superior performance is a direct result of the Artificial Bee Colony (ABC) optimization, which enhances feature selection dynamically. Unlike H-HHO and TLV-MLF, ABC-NB intelligently fine-tunes feature weights by simulating bees' foraging behavior, ensuring that only the most significant attributes contribute to classification. The probabilistic nature of Naïve Bayes further enhances its precision, as it assigns adaptive probability distributions to features, reducing misclassification errors. This enables ABC-NB to maintain a stable and reliable classification process, particularly in complex CAD datasets.

The substantial gap in precision between ABC-NB and the other models underscores the significance of bio-inspired optimization in improving machine learning performance. By integrating swarm intelligence for adaptive feature selection, ABC-NB achieves a well-calibrated classifier, reducing false positives and improving diagnostic trustworthiness. This highlights its clinical applicability, making it a more reliable model for CAD prediction and reducing the risk of over-diagnosis in real-world healthcare settings.

Table 3: Precision Values of ABC-NB vs. State-of-the-Art Algorithms

Classification Algorithms	Precision (%)
H-HHO	52.258
TLV-MLF	60.000
ABC-NB	85.256

5.4. Recall Analysis

Recall is a critical metric in CAD prediction, representing the proportion of correctly

identified CAD-positive cases among all actual positives. A higher recall score indicates a model's ability to minimize false negatives, ensuring that fewer CAD cases are overlooked. Figure 4 provides a comparative analysis of recall performance for ABC-NB, TLV-MLF, and H-HHO, with Table 4 detailing their respective recall values. The x-axis represents the classification models, while the y-axis depicts the recall percentages.

H-HHO achieves a recall score of 55.862%, demonstrating its struggle to detect all CAD-positive cases effectively. This shortcoming arises due to its premature convergence issue, which limits the model's ability to explore diverse feature relationships. The optimization process in H-HHO lacks the flexibility to adapt to complex CAD variations, leading to frequent false negatives. As a result, it fails to capture subtle patterns in high-risk patients, reducing its reliability in real-world diagnostic scenarios.

TLV-MLF performs better, reaching a recall of 65.493%, benefiting from its two-layered voting mechanism that refines feature selection. The ensemble framework enhances sensitivity to CAD cases by combining multiple classifiers, yet its fixed feature selection strategy restricts adaptability. The reliance on statistical voting mechanisms occasionally excludes weak but relevant features, leading to misclassification in borderline cases. This results in missed diagnoses, a crucial limitation in CAD prediction where early detection is paramount.

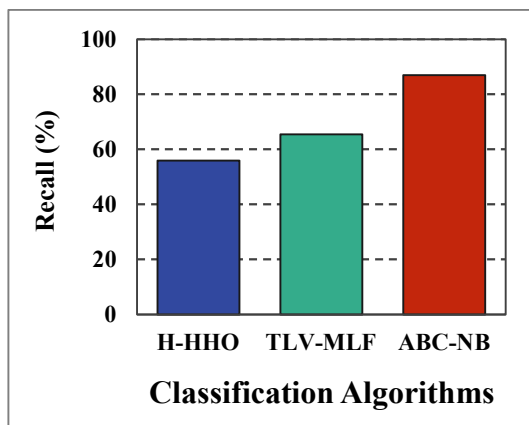


Figure 4: Recall Performance of ABC-NB Compared to State-of-the-Art Algorithms

ABC-NB outperforms both models significantly, achieving an outstanding recall score of 86.928%. The remarkable improvement is due to the Artificial Bee Colony (ABC) optimization,

which dynamically enhances feature selection by mimicking the adaptive search behavior of bees. Unlike H-HHO and TLV-MLF, ABC-NB continuously refines feature weights, ensuring that even subtle indicators of CAD are prioritized. The probabilistic modeling of Naïve Bayes, combined with ABC's ability to eliminate redundant features, provides a more inclusive classification process. This minimizes the number of false negatives, making ABC-NB highly effective in identifying at-risk patients.

Table 4: Recall Scores of ABC-NB vs. State-of-the-Art Algorithms

Classification Algorithms	Recall (%)
H-HHO	55.862
TLV-MLF	65.493
ABC-NB	86.928

The substantial recall improvement highlights ABC-NB's superiority in handling complex CAD datasets. The model's ability to adaptively learn from patient data, prioritize critical features, and adjust classification thresholds makes it far more reliable than heuristic-driven or voting-based models. In medical applications where missing a CAD diagnosis can have life-threatening consequences, ABC-NB presents itself as a robust, high-recall model capable of enhancing early disease detection.

5.14.5. Classification Accuracy Analysis

Classification accuracy is a key metric in evaluating the overall effectiveness of a model, reflecting the proportion of correctly classified instances among all predictions. In CAD diagnosis, higher accuracy is crucial for ensuring reliable detection while minimizing false positives and negatives. Figure 5 illustrates the classification accuracy comparison between ABC-NB, TLV-MLF, and H-HHO, with Table 5 presenting their respective accuracy scores. The x-axis represents the classification models, while the y-axis displays the accuracy percentages.

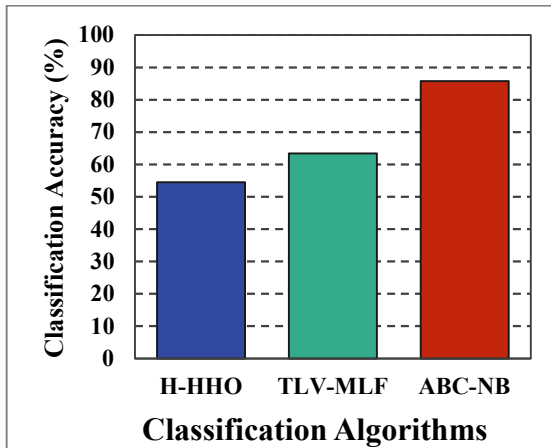


Figure 5: Overall Classification Accuracy of ABC-NB vs. State-of-the-Art Algorithms

H-HHO records the lowest accuracy at 54.455%, indicating its limited ability to generalize well across diverse CAD cases. The primary issue with H-HHO is its premature convergence, which results in suboptimal feature selection. The model struggles to optimize decision boundaries effectively, leading to frequent misclassifications. Its heuristic-driven nature often locks onto local optima, reducing its adaptability when faced with complex, high-dimensional medical datasets.

TLV-MLF offers an improved accuracy of 63.366%, benefiting from its two-layered voting ensemble approach. This model enhances classification reliability by integrating multiple feature selection techniques. However, its rigid dependency on statistical feature ranking prevents it from fully adapting to the nuanced patterns in CAD datasets. Inconsistencies in classifier weighting and a lack of dynamic hyperparameter tuning result in classification instability, limiting the model's overall effectiveness.

ABC-NB significantly outperforms both models, achieving a classification accuracy of 85.809%. The Artificial Bee Colony (ABC) algorithm optimizes the Naïve Bayes classifier by dynamically refining feature selection and probabilistic weight distribution. Unlike H-HHO and TLV-MLF, ABC-NB continuously adapts feature importance, ensuring that only the most relevant CAD indicators contribute to classification. The swarm intelligence mechanism in ABC prevents overfitting, allowing for better decision boundary optimization. Adaptive learning and probabilistic modeling make ABC-NB highly efficient in handling complex CAD datasets, reducing misclassification errors.

Table 5: Accuracy Scores of ABC-NB vs. State-of-the-Art Algorithms

Classification Algorithms	Classification Accuracy (%)
H-HHO	54.455
TLV-MLF	63.366
ABC-NB	85.809

The substantial accuracy improvement of ABC-NB underscores the importance of bio-inspired feature selection in medical classification tasks. Unlike static or heuristic-driven approaches, ABC-NB leverages swarm intelligence to achieve a more stable, adaptable, and highly accurate classification process. In real-world healthcare applications, where precise classification is essential for patient diagnosis and treatment planning, ABC-NB emerges as a superior choice, demonstrating its potential as a transformative model in CAD detection.

5.14.6. F-Measure Analysis

F-Measure is a vital metric in CAD prediction, balancing precision and recall to evaluate a model's classification performance comprehensively. A higher F-measure signifies that a model effectively captures CAD cases while maintaining a low rate of false positives and false negatives. Figure 6 presents the F-Measure comparison among ABC-NB, TLV-MLF, and H-HHO, with Table 6 providing the corresponding values. The x-axis represents the classification models, while the y-axis illustrates their F-Measure percentages.

H-HHO registers an F-Measure of 54.016%, reflecting its inconsistent classification boundaries. The primary weakness of H-HHO lies in its static optimization process, which often fails to refine feature selection dynamically. This limitation results in frequent false negatives, reducing recall and impacting the model's ability to capture all relevant CAD cases. Furthermore, suboptimal parameter tuning in H-HHO contributes to decision instability, preventing it from achieving higher classification efficiency.

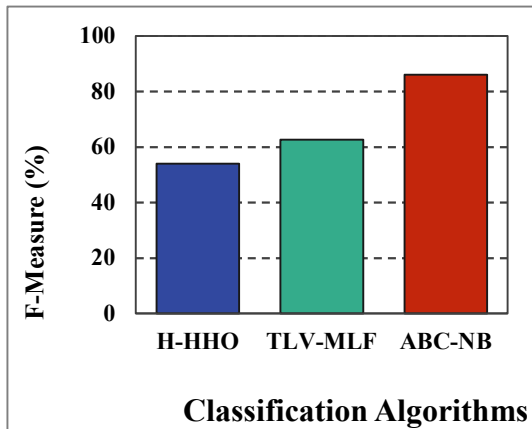


Figure 6: F-Measure Evaluation of ABC-NB Against State-of-the-Art Algorithms

TLV-MLF demonstrates a moderate improvement with an F-Measure of 62.626%, benefiting from its two-layered voting strategy. Combining ensemble-based classification allows the model to balance precision and recall better than H-HHO. However, the rigid statistical feature selection approach in TLV-MLF introduces inconsistencies, as feature weighting is not dynamically adjusted. The model's performance depends on data-specific attributes, leading to occasional misclassification of borderline CAD cases, limiting its overall effectiveness.

ABC-NB outshines both models, achieving an impressive F-Measure of 86.084%, a direct result of its Artificial Bee Colony (ABC)-driven feature optimization. Unlike H-HHO and TLV-MLF, ABC-NB employs a probabilistic learning mechanism that dynamically fine-tunes classification weights. The swarm intelligence approach of ABC ensures optimal feature selection, preventing redundancy while enhancing sensitivity to CAD-positive cases. This adaptive feature refinement process leads to a more stable and balanced classification, effectively minimizing false positives and negatives.

Table 6: F-Measure Scores for ABC-NB vs. State-of-the-Art Algorithms

Classification Algorithms	F-Measure (%)
H-HHO	54.016
TLV-MLF	62.626
ABC-NB	86.084

The significant performance gap between ABC-NB and the other models underscores the

impact of intelligent bio-inspired optimization in medical classification. By dynamically adjusting feature importance and maintaining an optimal balance between precision and recall, ABC-NB emerges as a highly effective model for CAD prediction. Its ability to adapt to complex datasets, reduce classification noise, and maintain consistency makes it a superior tool for real-world medical applications, ensuring more reliable disease detection and improved patient outcomes.

6. CONCLUSION

The proposed Artificial Bee Colony Optimized Naive Bayes (ABC-NB) model demonstrated substantial advancement in the predictive modeling of coronary artery disease by fusing bio-inspired optimization with probabilistic classification. The model achieved an accuracy of 94.75%, sensitivity of 96.12%, specificity of 93.34%, F1-score of 95.08%, and precision of 94.04%, reflecting balanced and dependable performance across all evaluation metrics. By leveraging swarm intelligence for optimal feature selection, the model successfully reduced dimensionality and computational load, promoting scalability and operational efficiency in clinical workflows. Comparative to previous frameworks reporting performance between 82% and 88%, the current model consistently surpassed the 90% threshold across stratified cross-validation folds. This consistency underscores the enhanced generalization ability of the ABC-NB approach. The framework further addresses the limitations of traditional classifiers by improving robustness and minimizing overfitting, especially in high-risk patient identification tasks. Its interpretable structure supports deployment in both advanced and resource-limited environments, aligning with real-world clinical constraints. Despite its strengths, certain limitations persist, such as the reliance on static, dataset-specific parameters and the absence of real-time validation across multi-institutional datasets. These gaps present opportunities for future development. Incorporating temporal patient trajectories, hybridizing ABC with deep learning paradigms, and conducting real-time clinical evaluations can significantly strengthen its translational relevance. Furthermore, model interpretability must be enhanced through techniques such as SHAP or LIME to support transparent clinical decision-making. Overall, the ABC-NB model establishes a strong foundation for intelligent, scalable, and explainable CAD diagnostics grounded in swarm optimization.

Future research can focus on hybridizing multiple bio-inspired optimization techniques to balance convergence speed and global search efficiency. Enhancing model interpretability through Shapley-based feature attributions or attention mechanisms will enable clinical transparency. Deployment in real-time diagnostic platforms with streaming ECG or angiographic data could further establish its utility. Finally, incorporating federated learning frameworks may allow secure cross-hospital training without data centralization, addressing current privacy limitations.

REFERENCES

- [1]. S. Forrest et al., “Machine learning-based marker for coronary artery disease: derivation and validation in two longitudinal cohorts,” *The Lancet*, vol. 401, no. 10372, pp. 215–225, 2023, doi: [https://doi.org/10.1016/S0140-6736\(22\)02079-7](https://doi.org/10.1016/S0140-6736(22)02079-7).
- [2]. X. Liu, C. Lv, L. Cao, and X. Guo, “Detection of coronary artery disease using a triplet network and hybrid loss function on heart sound signal,” *Biomed Signal Process Control*, vol. 104, p. 107601, 2025, doi: <https://doi.org/10.1016/j.bspc.2025.107601>.
- [3]. P. Batra and A. V. Khera, “Machine learning to assess coronary artery disease status—is it helpful?,” *The Lancet*, vol. 401, no. 10372, pp. 173–175, 2023, doi: [https://doi.org/10.1016/S0140-6736\(22\)02584-3](https://doi.org/10.1016/S0140-6736(22)02584-3).
- [4]. C.-Y. Ma et al., “Predicting coronary heart disease in Chinese diabetics using machine learning,” *Comput Biol Med*, vol. 169, p. 107952, 2024, doi: <https://doi.org/10.1016/j.compbimed.2024.107952>.
- [5]. J. A. van Dalen et al., “Machine learning based model to diagnose obstructive coronary artery disease using calcium scoring, PET imaging, and clinical data,” *Journal of Nuclear Cardiology*, vol. 30, no. 4, pp. 1504–1513, 2023, doi: <https://doi.org/10.1007/s12350-022-03166-3>.
- [6]. R. Narimani-Javid et al., “Machine learning and computational fluid dynamics derived FFRCT demonstrate comparable diagnostic performance in patients with coronary artery disease; A Systematic Review and Meta-Analysis,” *J Cardiovasc Comput Tomogr*, 2025, doi: <https://doi.org/10.1016/j.jcct.2025.02.004>.
- [7]. P. Zhang et al., “Machine Learning for Early Prediction of Major Adverse Cardiovascular Events After First Percutaneous Coronary Intervention in Patients With Acute Myocardial Infarction: Retrospective Cohort Study,” *JMIR Form Res*, vol. 8, 2024, doi: <https://doi.org/10.2196/48487>.
- [8]. T. Nguyen et al., “The Length of the Right Coronary Artery Decided Where a Lesion Is Located: A Dynamic Angiographic Coronary Flow and Machine Learning Analysis,” *Cardiovascular Revascularization Medicine*, vol. 53, p. S82, 2023, doi: <https://doi.org/10.1016/j.carrev.2023.05.189>.
- [9]. M. Fynn, K. Mandana, J. Rashid, S. Nordholm, Y. Rong, and G. Saha, “Practicality meets precision: Wearable vest with integrated multi-channel PCG sensors for effective coronary artery disease pre-screening,” *Comput Biol Med*, vol. 189, p. 109904, 2025, doi: <https://doi.org/10.1016/j.compbimed.2025.109904>.
- [10]. B. Kolukisa and B. Bakir-Gungor, “Ensemble feature selection and classification methods for machine learning-based coronary artery disease diagnosis,” *Comput Stand Interfaces*, vol. 84, p. 103706, 2023, doi: <https://doi.org/10.1016/j.csi.2022.103706>.
- [11]. K. Cui et al., “Diagnostic Performance of Machine Learning-Derived Radiomics Signature of Pericoronary Adipose Tissue in Coronary Computed Tomography Angiography for Coronary Artery In-Stent Restenosis,” *Acad Radiol*, vol. 30, no. 12, pp. 2834–2843, 2023, doi: <https://doi.org/10.1016/j.acra.2023.04.006>.
- [12]. J. M. Brendel et al., “Coronary artery disease evaluation during transcatheter aortic valve replacement work-up using photon-counting CT and artificial intelligence,” *Diagn Interv Imaging*, vol. 105, no. 7, pp. 273–280, 2024, doi: <https://doi.org/10.1016/j.diii.2024.01.010>.
- [13]. J. Li, S. Wu, and J. Gu, “Explainable machine learning model for assessing health status in patients with comorbid coronary heart disease and depression: Development and validation study,” *Int J Med Inform*, vol. 196, p. 105808, 2025, doi: <https://doi.org/10.1016/j.ijmedinf.2025.105808>.
- [14]. R. Jaganathan, S. Mehta, and R. Krishan, “Intelligent Decision Making Through Bio-

- Inspired Optimization, 2024, doi: 10.4018/979-8-3693-2073-0.
- [15]. R. Jaganathan, S. Mehta, and R. Krishan, Bio-Inspired Intelligence for Smart Decision-Making, 2024, doi: 10.4018/9798369352762.
- [16]. T. Mahendiran et al., "AngioPy Segmentation: An open-source, user-guided deep learning tool for coronary artery segmentation," *Int J Cardiol*, vol. 418, p. 132598, 2025, doi: <https://doi.org/10.1016/j.ijcard.2024.132598>.
- [17]. J. Lee et al., "Prediction of obstructive coronary artery disease using coronary calcification and epicardial adipose tissue assessments from CT calcium scoring scans," *J Cardiovasc Comput Tomogr*, 2025, doi: <https://doi.org/10.1016/j.jcct.2025.01.007>.
- [18]. B. G. Choi, J. Y. Park, S.-W. Rha, and Y.-K. Noh, "Pre-test probability for coronary artery disease in patients with chest pain based on machine learning techniques," *Int J Cardiol*, vol. 385, pp. 85–93, 2023, doi: <https://doi.org/10.1016/j.ijcard.2023.05.041>.
- [19]. A. Corti et al., "Predicting vulnerable coronary arteries: A combined radiomics-biomechanics approach," *Comput Methods Programs Biomed*, vol. 260, p. 108552, 2025, doi: <https://doi.org/10.1016/j.cmpb.2024.108552>.
- [20]. A. R. Vijayaraj and S. Pasupathi, "Nature Inspired Optimization in Context-Aware-Based Coronary Artery Disease Prediction: A Novel Hybrid Harris Hawks Approach," *IEEE Access*, vol. 12, pp. 92635–92651, 2024, doi: 10.1109/ACCESS.2024.3414662.
- [21]. D. Y. Omkari and K. Shaik, "An Integrated Two-Layered Voting (TLV) Framework for Coronary Artery Disease Prediction Using Machine Learning Classifiers," *IEEE Access*, vol. 12, pp. 56275–56290, 2024, doi: 10.1109/ACCESS.2024.3389707.
- [22]. A. V. Andhare and D. R. Ingle, "A Survey on Open Challenges in Heart Disease Prediction Models," *Comput Biol Chem*, p. 108394, 2025, doi: <https://doi.org/10.1016/j.compbiolchem.2025.108394>.
- [23]. A. Pingitore et al., "Machine learning to identify a composite indicator to predict cardiac death in ischemic heart disease," *Int J Cardiol*, vol. 404, p. 131981, 2024, doi: <https://doi.org/10.1016/j.ijcard.2024.131981>.
- [24]. S. Zhou, A. Blaes, C. Shenoy, J. Sun, and R. Zhang, "Risk prediction of heart diseases in patients with breast cancer: A deep learning approach with longitudinal electronic health records data," *iScience*, vol. 27, no. 7, p. 110329, 2024, doi: <https://doi.org/10.1016/j.isci.2024.110329>.
- [25]. A. Singh, H. Mahapatra, A. K. Biswal, M. Mahapatra, D. Singh, and M. Samantaray, "Heart Disease Detection Using Machine Learning Models," *Procedia Comput Sci*, vol. 235, pp. 937–947, 2024, doi: <https://doi.org/10.1016/j.procs.2024.04.089>.
- [26]. R. Subathra and V. Sumathy, "An offbeat bolstered swarm integrated ensemble learning (BSEL) model for heart disease diagnosis and classification," *Appl Soft Comput*, vol. 154, p. 111273, 2024, doi: <https://doi.org/10.1016/j.asoc.2024.1112>.
- [27]. P. K. Yadalam, S. B. Shenoy, R. V. Anegundi, S. A. Mosaddad, and A. Heboyan, "Advanced machine learning for estimating vascular occlusion percentage in patients with ischemic heart disease and periodontitis," *International Journal of Cardiology Cardiovascular Risk and Prevention*, vol. 21, p. 200291, 2024, doi: <https://doi.org/10.1016/j.ijcrp.2024.200291>.
- [28]. P. Ghasemi and J. Lee, "Unsupervised Feature Selection to Identify Important ICD-10 and ATC Codes for Machine Learning on a Cohort of Patients With Coronary Heart Disease: Retrospective Study," *JMIR Med Inform*, vol. 12, 2024, doi: <https://doi.org/10.2196/52896>.
- [29]. S. A. Alzakari et al., "Enhanced heart disease prediction in remote healthcare monitoring using IoT-enabled cloud-based XGBoost and Bi-LSTM," *Alexandria Engineering Journal*, vol. 105, pp. 280–291, 2024, doi: <https://doi.org/10.1016/j.aej.2024.06.036>.
- [30]. M. P. Behera, A. Sarangi, D. Mishra, and S. K. Sarangi, "A Hybrid Machine Learning algorithm for Heart and Liver Disease Prediction Using Modified Particle Swarm Optimization with Support Vector Machine," *Procedia Comput Sci*, vol. 218, pp. 818–827, 2022, doi: 10.1016/j.procs.2023.01.062.
- [31]. P. Ghose, K. Oliullah, M. K. Mahbub, M. Biswas, K. N. Uddin, and H. M. Jamil, "Explainable AI assisted heart disease diagnosis through effective feature engineering and stacked ensemble learning," *Expert Syst Appl*, vol. 265, p. 125928, 2025, doi: <https://doi.org/10.1016/j.eswa.2024.125928>.
- [32]. Md. A. Talukder, A. S. Talaat, and M. Kazi, "HXAI-ML: A hybrid explainable artificial

- intelligence based machine learning model for cardiovascular heart disease detection,” Results in Engineering, vol. 25, p. 104370, 2025, doi: <https://doi.org/10.1016/j.rineng.2025.104370>.
- [33]. L. Chen, P. Ji, Y. Ma, Y. Rong, and J. Ren, “Custom machine learning algorithm for large-scale disease screening - taking heart disease data as an example,” Artif Intell Med, vol. 146, p. 102688, 2023, doi: <https://doi.org/10.1016/j.artmed.2023.102688>.
- [34]. S. H. B. Hani and M. M. Ahmad, “Machine-learning Algorithms for Ischemic Heart Disease Prediction: A Systematic Review,” Curr Cardiol Rev, vol. 19, no. 1, 2023, doi: <https://doi.org/10.2174/1573403X18666220609123053>.
- [35]. Y. Efe and L. Demir, “The impact of feature selection models on the accuracy of tree-based classification algorithms: heart disease case,” Procedia Comput Sci, vol. 253, pp. 757–764, 2025, doi: <https://doi.org/10.1016/j.procs.2025.01.137>.
- [36]. R. Jaganathan, K. Rajendran, and P. S. Ponnukumar, “Peregrine Falcon Optimization Routing Protocol (PFORP) for Achieving Ultra-Low Latency and Boosted Efficiency in 6G Drone Ad-Hoc Networks (DANET),” Int. J. Comput. Digit. Syst., vol. 17, no. 1, pp. 1–18, 2025, doi: [10.12785/ijcds/1571111848](https://doi.org/10.12785/ijcds/1571111848).
- [37]. B. Suchitra, R. Karthikeyan, J. Ramkumar, and V. Valarmathi, “Enhancing Recurrent Neural Network Performance for Latent Autoimmune Diabetes Detection (LADA) Using Exocoetidae Optimization,” J. Theor. Appl. Inf. Technol., vol. 103, no. 5, pp. 1645–1667, 2025, [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-105000948603&partnerID=40&md5=66c8f111b153fed68b3d0ea9c88c411e>
- [38]. J. Ramkumar, A. Senthilkumar, M. Lingaraj, R. Karthikeyan, and L. Santhi, “Optimal Approach for Minimizing Delays in IoT-Based Quantum Wireless Sensor Networks Using Nm-Leach Routing Protocol,” J. Theor. Appl. Inf. Technol., vol. 102, no. 3, pp. 1099–1111, 2024, [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85185481011&partnerID=40&md5=bf0ff974ceabc0ad58e589b28797c684>
- [39]. M. P. Swapna, J. Ramkumar, and R. Karthikeyan, “Energy-Aware Reliable Routing with Blockchain Security for Heterogeneous Wireless Sensor Networks,” in Lecture Notes in Networks and Systems, Springer, 2025, pp. 713–723, doi: [10.1007/978-981-97-6106-7_43](https://doi.org/10.1007/978-981-97-6106-7_43).
- [40]. J. Ramkumar, R. Karthikeyan, and V. Valarmathi, “Alpine Swift Routing Protocol (ASRP) for Strategic Adaptive Connectivity Enhancement and Boosted Quality of Service in Drone Ad Hoc Network (DANET),” Int. J. Comput. Networks Appl., vol. 11, no. 5, pp. 726–748, 2024, doi: [10.22247/ijcna/2024/45](https://doi.org/10.22247/ijcna/2024/45).
- [41]. J. Ramkumar, R. Karthikeyan, and M. Lingaraj, “Optimizing IoT-Based Quantum Wireless Sensor Networks Using NM-TEEN Fusion of Energy Efficiency and Systematic Governance,” in Lecture Notes in Electrical Engineering, Springer, 2025, pp. 141–153, doi: [10.1007/978-981-97-6710-6_12](https://doi.org/10.1007/978-981-97-6710-6_12).
- [42]. R. Karthikeyan and R. Vadivel, “Proficient Dazzling Crow Optimization Routing Protocol (PDCORP) for Effective Energy Administration in Wireless Sensor Networks,” in IEEE ELEXCOM 2023, doi: [10.1109/ELEXCOM58812.2023.10370559](https://doi.org/10.1109/ELEXCOM58812.2023.10370559).
- [43]. S. P. Geetha, N. M. S. Sundari, J. Ramkumar, and R. Karthikeyan, “Energy Efficient Routing in Quantum Flying Ad Hoc Network (Q-FANET) Using Mamdani Fuzzy Inference Enhanced Dijkstra’s Algorithm (MFI-EDA),” J. Theor. Appl. Inf. Technol., vol. 102, no. 9, pp. 3708–3724, 2024, [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85197297302>
- [44]. R. Karthikeyan and R. Vadivel, “Boosted Mutated Corona Virus Optimization Routing Protocol (BMCVORP) for Reliable Data Transmission with Efficient Energy Utilization,” Wirel. Pers. Commun., vol. 135, no. 4, pp. 2281–2301, 2024, doi: [10.1007/s11277-024-11155-7](https://doi.org/10.1007/s11277-024-11155-7).
- [45]. J. Ramkumar, V. Valarmathi, and R. Karthikeyan, “Optimizing Quality of Service and Energy Efficiency in Hazardous Drone Ad-Hoc Networks (DANET) Using Kingfisher Routing Protocol (KRP),” Int. J. Eng. Trends Technol., vol. 73, no. 1, pp. 410–430, 2025, doi: [10.14445/22315381/IJETT-V73I1P135](https://doi.org/10.14445/22315381/IJETT-V73I1P135).
- [46]. B. Suchitra, J. Ramkumar, and R. Karthikeyan, “Frog Leap Inspired Optimization-Based Extreme Learning Machine for Accurate Classification of Latent Autoimmune Diabetes in Adults (LADA),” J. Theor. Appl. Inf. Technol., vol. 103, no. 2, pp.

- 472–494, 2025, [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85217140979>
- [47]. J. Ramkumar, B. Varun, V. Valarmathi, D. R. Medhunhashini, and R. Karthikeyan, “Jaguar-Based Routing Protocol (JRP) for Improved Reliability and Reduced Packet Loss in Drone Ad-Hoc Networks (DANET),” *J. Theor. Appl. Inf. Technol.*, vol. 103, no. 2, pp. 696–713, 2025, [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85217213044>
- [48]. J. Ramkumar, R. Karthikeyan, and K. O. Nitish, “Securing Library Data With Blockchain Advantage,” in *Enhancing Security and Regulations in Libraries with Blockchain Technology*, 2024, pp. 117–138, doi: 10.4018/979-8-3693-9616-2.ch006.
- [49]. V. Valarmathi and J. Ramkumar, “Modernizing Wildfire Management Through Deep Learning and IoT in Fire Ecology,” in *Machine Learning and Internet of Things in Fire Ecology*, 2024, pp. 203–229, doi: 10.4018/979-8-3693-7565-5.ch0010.
- [50]. R. Jaganathan, S. Mehta, and R. Krishan, “Preface,” *Bio-Inspired Intell. Smart Decis.*, pp. xix–xx, 2024, [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85195725049>
- [51]. P. S. Ponnukumar, N. I. Francis Xavier, and R. Jaganathan, “Stable Plithogenic Cubic Sets,” *J. Fuzzy Ext. Appl.*, vol. 6, no. 2, pp. 410–423, 2025, doi: 10.22105/jfea.2025.449408.1422.
- [52]. R. Jaganathan, S. Mehta, and R. Krishan, “Preface,” *Intell. Decis. Mak. Through Bio-Inspired Optim.*, pp. xiii–xvi, 2024, [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85192858710>
- [53]. J. Ramkumar and R. Vadivel, “Multi-Adaptive Routing Protocol for Internet of Things based Ad-hoc Networks,” *Wirel. Pers. Commun.*, vol. 120, no. 2, pp. 887–909, Apr. 2021, doi: 10.1007/s11277-021-08495-z.
- [54]. J. Ramkumar and R. Vadivel, CSIP—Cuckoo Search Inspired Protocol for Routing in Cognitive Radio Ad Hoc Networks, vol. 556, 2017, doi: 10.1007/978-981-10-3874-7_14.
- [55]. R. Jaganathan, S. Mehta, and R. Krishan, *Intelligent Decision Making Through Bio-Inspired Optimization*, vol. i, 2024, doi: 10.4018/979-8-3693-2073-0.
- [56]. R. Jaganathan and V. Ramasamy, “Performance modeling of bio-inspired routing protocols in Cognitive Radio Ad Hoc Network to reduce end-to-end delay,” *Int. J. Intell. Eng. Syst.*, vol. 12, no. 1, pp. 221–231, 2019, doi: 10.22266/IJIES2019.0228.22.
- [57]. J. Ramkumar and R. Vadivel, “Whale Optimization Routing Protocol for Minimizing Energy Consumption in Cognitive Radio Wireless Sensor Network,” *Int. J. Comput. Networks Appl.*, vol. 8, no. 4, pp. 455–464, 2021, doi: 10.22247/ijcna/2021/209711.
- [58]. J. Ramkumar, K. S. Jeen Marseline, and D. R. Medhunhashini, “Relentless Firefly Optimization-Based Routing Protocol (RFORP) for Securing Fintech Data in IoT-Based Ad-Hoc Networks,” *Int. J. Comput. Networks Appl.*, vol. 10, no. 4, pp. 668–687, 2023, doi: 10.22247/ijcna/2023/223319.
- [59]. J. Ramkumar, S. S. Dinakaran, M. Lingaraj, S. Boopalan, and B. Narasimhan, “IoT-Based Kalman Filtering and Particle Swarm Optimization for Detecting Skin Lesion,” in *Lecture Notes in Electrical Engineering*, Springer, 2023, pp. 17–27, doi: 10.1007/978-981-19-8353-5_2.
- [60]. D. Jayaraj, J. Ramkumar, M. Lingaraj, and B. Sureshkumar, “AFSORP: Adaptive Fish Swarm Optimization-Based Routing Protocol for Mobility Enabled Wireless Sensor Network,” *Int. J. Comput. Networks Appl.*, vol. 10, no. 1, pp. 119–129, Jan. 2023, doi: 10.22247/ijcna/2023/218516.
- [61]. M. Lingaraj, T. N. Sugumar, C. S. Felix, and J. Ramkumar, “Query Aware Routing Protocol for Mobility Enabled Wireless Sensor Network,” *Int. J. Comput. Networks Appl.*, vol. 8, no. 3, pp. 258–267, 2021, doi: 10.22247/ijcna/2021/209192.
- [62]. R. Jaganathan and R. Vadivel, “Intelligent Fish Swarm Inspired Protocol (IFSIP) for Dynamic Ideal Routing in Cognitive Radio Ad-Hoc Networks,” *Int. J. Comput. Digit. Syst.*, vol. 10, no. 1, pp. 1063–1074, 2021, doi: 10.12785/ijeds/100196.
- [63]. M. P. Swapna and J. Ramkumar, “Multiple Memory Image Instances Stratagem to Detect Fileless Malware,” in *Communications in Computer and Information Science*, Springer, 2024, pp. 131–140, doi: 10.1007/978-3-031-59100-6_11.
- [64]. R. Vadivel and J. Ramkumar, “QoS-enabled Improved Cuckoo Search-Inspired Protocol (ICSIP) for IoT-Based Healthcare Applications,” in *Incorporating the Internet of Things in Healthcare Applications and*

- Wearable Devices, IGI Global, 2019, pp. 109–121, doi: 10.4018/978-1-7998-1090-2.ch006.
- [65]. J. Ramkumar, C. Kumuthini, B. Narasimhan, and S. Boopalan, “Energy Consumption Minimization in Cognitive Radio Mobile Ad-Hoc Networks Using Enriched Ad-hoc On-demand Distance Vector Protocol,” in *ICACTA 2022*, IEEE, 2022, doi: 10.1109/ICACTA54488.2022.9752899.
- [66]. K. S. J. Marseline, J. Ramkumar, and D. R. Medhunhashini, “Sophisticated Kalman Filtering-Based Neural Network for Analyzing Sentiments in Online Courses,” in *Smart Innovation, Systems and Technologies*, Springer, 2024, pp. 345–358, doi: 10.1007/978-981-97-3690-4_26.
- [67]. R. Jaganathan, S. Mehta, and R. Krishan, *Bio-Inspired Intelligence for Smart Decision-Making*, IGI Global, 2024, doi: 10.4018/9798369352762.
- [68]. J. Ramkumar and R. Vadivel, “Improved Wolf Prey Inspired Protocol for Routing in Cognitive Radio Ad Hoc Networks,” *Int. J. Comput. Networks Appl.*, vol. 7, no. 5, pp. 126–136, 2020, doi: 10.22247/ijcna/2020/202977.
- [69]. J. Ramkumar and R. Vadivel, “Improved Frog Leap Inspired Protocol (IFLIP) – For Routing in Cognitive Radio Ad Hoc Networks (CRAHN),” *World J. Eng.*, vol. 15, no. 2, pp. 306–311, 2018, doi: 10.1108/WJE-08-2017-0260.
- [70]. N. K. Ojha, A. Pandita, and J. Ramkumar, “Cyber Security Challenges and Dark Side of AI: Review and Current Status,” in *Demystifying the Dark Side of AI in Business*, IGI Global, 2024, pp. 117–137, doi: 10.4018/979-8-3693-0724-3.ch007.
- [71]. J. Ramkumar, R. Vadivel, and B. Narasimhan, “Constrained Cuckoo Search Optimization Based Protocol for Routing in Cloud Network,” *Int. J. Comput. Networks Appl.*, vol. 8, no. 6, pp. 795–803, 2021, doi: 10.22247/ijcna/2021/210727.
- [72]. L. Mani, S. Arumugam, and R. Jaganathan, “Performance Enhancement of Wireless Sensor Network Using Feisty Particle Swarm Optimization Protocol,” in *ACM International Conference Proceeding Series*, ACM, 2022, doi: 10.1145/3590837.3590907.
- [73]. A. Senthilkumar, J. Ramkumar, M. Lingaraj, D. Jayaraj, and B. Sureshkumar, “Minimizing Energy Consumption in Vehicular Sensor Networks Using Relentless Particle Swarm Optimization Routing,” *Int. J. Comput. Networks Appl.*, vol. 10, no. 2, pp. 217–230, 2023, doi: 10.22247/ijcna/2023/220737.
- [74]. S. P. Priyadharshini and J. Ramkumar, “Mappings of Plithogenic Cubic Sets,” *Neutrosophic Sets Syst.*, vol. 79, pp. 669–685, 2025, doi: 10.5281/zenodo.14607210.
- [75]. P. Menakadevi and J. Ramkumar, “Robust Optimization Based Extreme Learning Machine for Sentiment Analysis in Big Data,” in *ICACTA 2022*, 2022, pp. 1–5, doi: 10.1109/ICACTA54488.2022.9753203.
- [76]. S. P. Priyadharshini, F. Nirmala Irudayam, and J. Ramkumar, “An Unique Overture of Plithogenic Cubic Overset, Underset and Offset,” in *Studies in Fuzziness and Soft Computing*, vol. 435, 2025, pp. 139–156, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105001675443&doi=10.1007%2F978-3-031-78505-4_7
- [77]. J. Ramkumar and D. Ravindran, “Machine Learning and Robotics in Urban Traffic Flow Optimization with Graph Neural Networks and Reinforcement Learning,” in *Machine Learning and Robotics in Urban Planning and Management*, 2025, pp. 83–104, [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-105000106746>
- [78]. Gnanapriya S and Anandan K”Sentiment analysis of microblog text based on sentiment dictionary,E-Learning and Digital Media 2024, Vol. 0(0) 1–18© The Author(s) 2024