THOMPSON RIVERS UNIVERSITY

Enhanced Detection of Early-Stage Parkinson's Disease via Hybrid Gold Rush-Liver Cancer Algorithm

By

Melvin Biju

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science in Data Science

KAMLOOPS, BRITISH COLUMBIA

April, 2024

SUPERVISOR

Dr. Mohamed Tawhid

ABSTRACT

Millions are diagnosed with Parkinson's disease annually, a neurodegenerative disorder characterized by distinct speech abnormalities often imperceptible to the untrained ear. These subtle deviations in speech patterns can, however, be quantified through advanced signal analysis, offering a conduit for early detection. Traditional diagnostic practices, predominantly subjective in nature, are fraught with inaccuracies and inconsistencies. There exists an urgent need for a methodologically sound, objective approach to both the early detection and classification of Parkinson's disease. This paper introduces the Gold Rush-Liver Cancer Optimization Algorithm (GRLCA), an innovative hybrid model tailored for the robust feature selection required in the early stages of Parkinson's detection. The GRLCA methodologically fuses the exploratory prowess of the Gold Rush Optimizer with the Liver Cancer algorithm's rapid convergence capabilities, creating a powerful tool for navigating and optimizing complex feature spaces. This algorithm employs a wrapper-based feature selection strategy, utilizing swarm intelligence to perform comprehensive global searches. This approach not only enhances the exploration of the feature space but also ensures efficient convergence to optimal feature sets by leveraging decentralized, self-organizing agents. Our evaluation protocol involved comparing the GRLCA with ten established benchmark algorithms across fourteen medical datasets, which include four datasets specifically curated to facilitate research on Parkinson's disease. The performance metrics were rigorously analyzed, focusing on diagnostic accuracy and the ability to minimize feature dimensionality while maintaining high classification performance. The results from these comparative studies highlight the superior performance of the GRLCA, demonstrating its capacity to outmatch existing algorithms significantly. Notably, the GRLCA algorithm excelled in terms of accuracy with consistent results, underscoring its potential as a transformative tool in the early diagnosis of Parkinson's disease.

Through this enhanced diagnostic approach, GRLCA offers the promise of more precise, reliable, and timely identification of Parkinson's disease, potentially revolutionizing patient outcomes through earlier therapeutic intervention.

Key Words: Parkinson's disease, Machine learning, medical data, Wrapper, Feature Selection, Meta-heuristics.

ACKNOWLEDGEMENTS

I am profoundly grateful to Thompson Rivers University for providing the resources and environment necessary to conduct this research. The support and facilities offered by the university have been instrumental in the successful completion of this project.

I would like to extend my deepest appreciation to Dr. Mohamed Tawhid, my supervisor, whose expert guidance and insights have been invaluable throughout the duration of this project. Dr. Tawhid's mentorship not only directed me towards academic rigor but also encouraged me to think critically and creatively. His patience and commitment to excellence have deeply inspired me, and I am immensely thankful for his unwavering support and encouragement.

Additionally, I must express my sincere thanks to Dr. Ajay Dhruv, the second reader of this project. His thorough reviews and constructive feedback have significantly enhanced the quality of my work. Dr. Dhruv's meticulous attention to detail and profound understanding of the subject matter have greatly contributed to refining my research approach and outcomes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The brain is the most crucial organ in the human body and is responsible for the effective functioning of all the other organs. One of the diseases that impairs its function is Parkinson's Disease (PD). It is an incurable neurological disease that affects people mostly above the age of 50 (Bloem et al., 2021). Statistics show that only 4 percentage of people affected with PD are below 50 years of age (DeMaagd & Philip, 2015). The symptoms of this disease can be quite subtle. The main motor symptoms are slowness of movement, tremor, stiffness, and poor balance. Non-motor symptoms include hypotension, mood disorders, pain, sensory disfunction and loss of weight (DeMaagd & Philip, 2015) Studies show that PD patients commonly have speech disorders (DeMaagd & Philip, 2015). They often portray anomalies like quiet and hurried speech (Sveinbjornsdottir, 2016). The analysis of this speech data is considered as an important non-invasive method for PD identification. This makes it important for doctors in diagnosing this disease. The detection of changes in speech pattern is extremely useful to identify PD in its early stages. As stated before, this disease is incurable; however, there are several treatments that reduce the symptoms in early stages.

Voice frequency analysis is a relatively accurate and non-invasive process. Consequently, voice frequency can be utilized to monitor the development of this irrational illness (Monson et al., 2014). Numerous speech tests have been carried out to monitor the disease's course. ML techniques are regularly applied in the medical (healthcare) industry. A range of data modalities, such as handwriting patterns and acoustic voice recordings, are being combined with machine learning algorithms to diagnose PD. We can identify relevant characteristics that aren't typically used in the medical diagnosis of PD with the aid of machine learning (ML) tools, and we can rely on these alternative indications to detect PD in its preclinical stages. The wrapper method is used to eliminate or reduce the noise in the dataset.

Finally, wrapper classification technique is executed during the final phase. The performance of the classification method has a significant impact on the feature extraction approach. For this reason, selecting the best classification method is a crucial factor that must be considered for this disease. This review examines how ML models trained on sensory data help doctors diagnose patients with PD at every stage of the treatment process.

The aim is to give neurologists early detection insights that could enhance the diagnosis and treatment of PD by highlighting additional unique techniques and offering new solutions that have not been adequately addressed in the reviews that have already been published.

The contribution of this paper include providing an overview of PD, outlining its primary traits as well as its most common motor and non-motor symptoms. We also test several optimization algorithms and ML models, analyzed the accuracy of ML models for the diagnosis of PD on bench-marked medical datasets. We combine the strengths of two of the best performing algorithms and combine it to make a hybrid algorithm with improved accuracy. Finally, the study shows the challenges and discusses the potential future work that can be implemented.

Chapter 2

Literature Review

Research into the use of speech samples for diagnosing Parkinson's Disease (PD) has garnered considerable attention in recent years. Notably, research by (Shahbaba & Neal, 2007) implemented a non-linear diagnostic model utilizing Dirichlet mixtures, achieving a classification accuracy of 87.7%. In a similar vein, (Little et al., 2009) carried out an influential study using a Support Vector Machine (SVM) classifier with Gaussian radial basis kernel functions. This study included a feature selection process to identify the most effective features, culminating in a maximum classification accuracy of 91.4%. Further comparative research by (Das, 2010) evaluated various algorithms, including artificial neural networks (ANN), DMneural, regression, and decision trees, for PD diagnosis from speech samples. Here, the ANN method stood out, demonstrating a general classification performance of 92.9%.

Additionally, in 2010, (Guo et al., 2010) devised a hybrid model that combined expectation maximization (EM) with genetic algorithms (GA), achieving a classification accuracy of 93.1%. (Ozcift & Gulten, 2011) introduced a system combining correlation-based feature selection (CFS) with rotation forest classifiers (RF) involving 30 machine learning algorithms. This innovative CFS-RF system reached an optimal classification accuracy of 87.13%. In 2013, (Chen et al., 2016) enhanced the PD speech signal analysis by employing a feature reduction technique to remove redundant information, using a fuzzy classifier that achieved an average classification accuracy of 96.07%.

By 2016, further advancements were noted when Extreme Learning Machine (ELM) and Extreme Kernel Learning Machine (KELM) methodologies were proposed for early PD diagnosis, yielding a maximum classification accuracy of 96.47% and an average of 95.97% across 10-fold cross-validation (Chen et al., 2016). More recent developments by (Peker et al., 2015) saw the integration of a maximum redundancy-maximum relevance attribute selection algorithm with a complex-valued artificial neural network, which led to a classification accuracy of 98.12%. Lastly, (Lahmiri & Shmuel, 2019) assessed the efficacy of eight pattern ranking techniques paired with a nonlinear SVM classifier optimized via Bayesian techniques. This approach highlighted the classifier's ability to achieve a high classification accuracy of 92.21% with 14 vocal features identified through the Wilcoxon method, a specificity of 82.79% from the top 13 voice models identified via the ROC based technique, and a remarkable sensitivity of 99.63% from a single vocal pattern, also identified through the ROC based method.

Several studies have employed machine learning models to analyze PD symptoms with promising results. Notably, Das (Das, 2010) and Åström and Koker (Åström & Koker, 2011) demonstrated the effectiveness of neural network classifiers in handling movement data, achieving improved classification performances by incorporating rule-based systems and parallel neural network configurations. Similarly, (Bhattacharya & Bhatia, 2010) utilized support vector machines (SVM) with sophisticated data preprocessing techniques, highlighting the utility of the receiver operation curve (ROC) in evaluating model efficacy. This commitment to advancing classification methodologies underlines the relevance of our hybrid algorithmic approach in navigating complex data landscapes effectively.

Evolutionary computation has recently received much attention as a solution to optimization problems. Evolutionary algorithms are bio-inspired generic population-based optimization algorithms that mimic biological processes such as selection, mutation, and reproduction to solve optimization issues. Another metaheuristic technique is the crowd search algorithm (CSA), which was motivated by the clever ways that crows conceal and pilfer food (Fan et al., 2023). Crows are believed to be some of the world's most intelligent creatures. Several algorithms that draw inspiration from nature have been proposed in recent studies. For instance, the foundation of genetic algorithms (GA) is natural selection; bat algorithms draw inspiration from micro bat echolocation techniques (Yang & He, 2013).

Further, studies like those by (Chen et al., 2016), who developed a fuzzy k-nearest neighbor model, emphasize the importance of integrating advanced data reduction techniques such as principal component analysis to enhance classification accuracy. The pursuit of higher accuracy is a common thread across these studies, with neural network and SVM-based models consistently showing promising results in differentiating PD patients from healthy controls through various biomarkers and symptom presentations, as detailed (Eskidere et al., 2012).

We also draw inspiration from the work of (Nilashi et al., 2018), who used classification using swarm intelligence to refine PD predictions, applying these in a neuro-fuzzy inference framework. This approach mirrors our integration of multiple machine learning modalities to exploit their respective strengths in detecting early PD symptoms, which are often subtle and vary significantly among patients, as discussed in studies (Das, 2010),(Guo et al., 2010). Notably, vocal impairments, a key focus of our study, have been recognized as early indicators of PD in various research efforts, further underscoring the relevance of our algorithm in clinical settings.

Metaheuristic algorithms have been increasingly applied to various aspects of healthcare, including the management and analysis of Parkinson's disease (PD). These algorithms provide a powerful tool for optimizing complex problems, particularly in modeling disease progression and optimizing treatment protocols. Studies such as those by (Zhang et al., 2019)) have demonstrated the effectiveness of metaheuristic algorithms in enhancing the accuracy of PD diagnosis through advanced image analysis and pattern recognition techniques, significantly improving early detection rates.

Further research has explored the use of metaheuristics in the personalized treatment of Parkinson's disease, aiming to tailor therapeutic strategies to individual patient needs. For instance, (Silva et al., 2020) applied a genetic algorithm approach to optimize drug dosing schedules, effectively reducing the side effects while maintaining therapeutic benefits. This approach not only supports the personalization of treatments but also enhances patient adherence and overall quality of life by minimizing adverse effects commonly associated with PD medications.

Moreover, the integration of metaheuristics with wearable technology has opened new avenues for monitoring and managing Parkinson's disease. Research by (Gupta & Quan, 2021) has utilized swarm intelligence techniques to interpret data from sensors effectively, providing real-time insights into patient motor functions. This application facilitates continuous monitoring of disease symptoms, allowing for timely adjustments in treatment plans and potentially slowing the progression of the disease through more immediate and precise interventions.

Metaheuristic algorithms are not limited to Parkinson's disease; they have been effectively applied to a wide range of medical datasets, enhancing diagnostic and therapeutic outcomes across various conditions. For example, in the field of oncology, metaheuristics have played a crucial role in the segmentation and classification of tumor images. A study by (Lee & Kim, 2018) utilized a hybrid metaheuristic approach that combined features of both simulated annealing and genetic algorithms to accurately delineate tumor boundaries in MRI scans. This method not only improved the precision of tumor detection but also facilitated a more targeted and effective treatment planning, significantly impacting patient outcomes in cancer care.

In cardiology, metaheuristics have been instrumental in analyzing complex cardiovascular data to predict heart disease. According to research by (Moreno & Rocha, 2019), particle swarm optimization was used to refine the feature selection process in large cardiovascular datasets, effectively identifying key predictors of heart conditions. This application has not only enhanced the predictive accuracy but also streamlined the diagnostic process, allowing for earlier intervention and better management of cardiac patients. By optimizing the analysis of intricate datasets, metaheuristics contribute to more efficient and effective healthcare delivery, ultimately improving prognosis and treatment across multiple medical specialties.

Moreover, the exploration of genetic algorithms for feature selection by (Guo et al., 2010) and the application of random forest classifiers by (Peker et al., 2015) for diagnosing and tracking PD further validate the potential of sophisticated algorithms in medical diagnostics. Our GRLCA algorithm aims to exceed the benchmarks set by these studies, targeting classification accuracies highlighted by previous researchers, including the 92.75% accuracy achieved by (Erdogdu Sakar et al., 2017) using SVM.

(Little et al., 2009) utilized a kernel-based support vector machine to detect dysphonia in Parkinson's disease (PD), analyzing continuous phonations from 31 participants. They introduced novel dysphonia measures that improved classification accuracy, which is beneficial for telemonitoring applications. (Harel et al., 2004) noted that PD symptoms could be detected up to five years before professional diagnosis. (Das, 2010) compared various classification algorithms using voice tests from PD patients, finding that neural networks outperformed others in accuracy.

Similarly, (Yadav et al., 2023) focused on voice articulation and tested three data mining methods, achieving an accuracy of 82.051%. (Ramani & Sivagami, 2011) also explored neural networks for PD detection, employing significant feature selection techniques like information gain and achieving accuracies around 82.05%. (Rustempasic & Can, 2013) used an Artificial Neural Network combined with principal component analysis for feature selection, resulting in a classification accuracy of 81.33%.

Chapter 3

Methodology

3.1 Swarm Intelligence

Swarm intelligence (SI) is an innovative computational and behavioral metaphor for solving complex problems, which is rooted in the biological studies of behaviors exhibited by social organisms. In the natural world, examples of such systems include ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling. SI systems consist of a population of simple agents interacting locally with one another and with their environment. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behavior, unknown to the individual agents (Bonabeau et al., 1999).

The underlying power of swarm intelligence lies in the collective and distributed nature of the problem-solving process. Unlike centralized algorithms, where a single point of failure can cause the entire system to collapse, SI algorithms are robust and flexible, adapting to new problems in real-time. This decentralized approach makes SI particularly useful in fields like robotics, telecommunications, and predictive analytics.

In the medical field, swarm intelligence can be particularly effective for optimizing complex problems, such as scheduling in hospitals, where there are numerous constraints and variables at play. It can also be utilized in the analysis of big data for predictive diagnostics, where vast amounts of data require processing to identify patterns that predict diseases. SI algorithms, with their ability to simultaneously explore and exploit data, can navigate through the complex medical datasets to find optimal solutions efficiently (Poli et al., 2007).

Moreover, SI has been used in medical imaging where it helps in enhancing images, recognizing patterns, and extracting significant features that are not readily visible. Algorithms inspired by swarm behaviors, like Particle Swarm Optimization (PSO), can optimize the process of image segmentation in MRI scans or enhance the edge detection in X-ray images, thereby providing clearer and more accurate images for diagnosis (Blum & Merkle, 2008).

Another promising application of SI in medicine is in the optimization of treatment plans, especially for chronic diseases such as cancer. The calibration of dosages for chemotherapy, for instance, can be modeled as an optimization problem, where the goal is to maximize the destruction of cancer cells while minimizing the side effects on healthy cells. SI algorithms can explore numerous combinations and sequences of drug administration to propose the most effective treatment plan.

The application of swarm intelligence extends to the modeling of biological systems, providing insights into the spread of diseases within populations and the natural processes of the human body. This modeling can lead to better strategies for disease control and prevention, particularly in epidemiology where understanding the spread of infectious diseases is critical.

Furthermore, SI is instrumental in bioinformatics, particularly in the analysis of genetic data. It aids in unraveling complex genetic networks and understanding the underlying mechanisms of genetic diseases. SI algorithms can handle the combinatorial nature of genetic data to discover biomarkers for various diseases, enhancing the diagnosis and treatment process.

Swarm intelligence also contributes to the field of pharmacology, where it can be used in drug discovery and development. The multi-dimensional space of chemical compounds is a perfect use-case for SI, where algorithms can search for new drug candidates that can bind effectively to target proteins, considering a multitude of variables such as drug efficacy, potency, and toxicity.

One of the most compelling uses of SI in healthcare is in the design and operation of telemedicine networks. SI can help in the dynamic allocation of resources and routing of medical information, ensuring that patient data is processed and transferred efficiently across different nodes in the network (Nakrani & Tovey, 2004).

Despite these promising applications, the effectiveness of SI in medicine, as in any field, depends on proper algorithm selection, parameter tuning, and integration with domain-specific knowledge. The black-box nature of SI algorithms necessitates a careful balance between exploration (searching the entire space) and exploitation (focusing on promising areas), which can be challenging without expert insight. Additionally, the stochastic nature of these algorithms may lead to variability in performance, requiring multiple runs to achieve consistent results.

Swarm intelligence presents a flexible and powerful approach for tackling complex, multi-dimensional problems in the medical field. With their ability to adapt, learn from the environment, and find optimal solutions, SI algorithms hold great promise for advancing medical research and improving patient care. However, their deployment must be handled with consideration for the intricacies of medical data and the critical nature of health-related outcomes.

3.2 SI Algorithms Used

In machine learning, these algorithms adjust the parameters of models to reduce errors on training data, essentially finding the best possible settings for making accurate predictions. When applied to medical data, optimization algorithms play a crucial role in developing predictive models that can identify disease patterns, optimize treatment plans, and improve patient outcomes. In this study, we evaluate the performance of 10 optimization algorithms shown in Table 3.1 to choose an algorithm for implementation. We use the CEC benchmark problems with similar parameters for a consistent comparison.

The Sine Cosine Algorithm, developed in 2016 (Mirjalili, 2016), is one such tool that stands out for its ability to methodically explore the search space using the principles of sine and cosine functions. This algorithm is particularly effective in medical data analysis as it can navigate through intricate data patterns to find relevant features that may indicate Parkinson's disease. The algorithm's systematic approach helps avoid getting stuck in areas that seem promising but ultimately do not lead to the best overall solution.

Another innovative algorithm is the Multi-Verse Optimizer, introduced in

Table 3.1: Optimization Algorithms used

No.	Algorithm Name	Abbreviation	CEC Year
1	Sine Cosine Algorithm	SCA	2016
2	Multi-Verse Optimizer	MVO	2015
3	Whale Optimization Algorithm	WOA	2016
4	Heap-Based Optimizer	HBO	2020
5	Triangulation Topology Aggregation Optimizer	TTAO	2023
6	Tyrannasaurus optimization algorithm	TREX	2023
7	Red-Tailed Hawk algorithm	RTH	2023
8	Liver Cancer Algorithm	LCA	2023
9	Gold Rush Optimizer	GRO	2023
10	Population Vortex Search algorithm	PVS	2022

2015(Mirjalili et al., 2016). Drawing on the concept of multiple universes, this algorithm allows for an expansive and thorough examination of possible solutions by imitating natural phenomena such as black holes and wormholes. For medical datasets, which often include a variety of features, this optimizer's comprehensive search is essential. It can sift through numerous patient records to find those crucial factors that can predict Parkinson's disease with a high degree of accuracy.

The Whale Optimization Algorithm from 2016 (Mirjalili & Lewis, 2016) takes its inspiration from the bubble-net feeding strategy of humpback whales. This algorithm is particularly suited to medical data as it alternates between diversifying its search to find a wide range of possible solutions and intensifying its focus to improve solutions it has already found promising. Such a flexible approach is necessary for handling the complexities of medical datasets where the disease indicators can be varied and subtle.

Introduced in 2020, the Heap-Based Optimizer (Askari et al., 2020) uses data structuring techniques akin to those found in computer science. It is efficient at managing and sorting potential solutions, much like a well-organized database. This is especially useful when dealing with patient data, as it can quickly highlight the most significant indicators of Parkinson's disease. The algorithm's ability to maintain a structured search ensures that the most informative features are considered when developing diagnostic models.

The Triangulation topology aggregation optimizer (Zhao et al., 2024), one of the more recent algorithms developed in 2023, applies geometric principles to converge on the best solutions. This optimizer's deliberate and precise search method is particularly useful for isolating subtle changes in medical imaging or sensor data, which can be indicative of Parkinson's disease.

Equally aggressive in its search methodology, the Tyrannosaurus optimization algorithm (Zhao et al., 2024), also from 2023, discards suboptimal solutions in favor of those that are more promising. In medical contexts, such an assertive approach is valuable as it quickly filters out irrelevant data, concentrating the search on potential biomarkers that may signify the development or progression of Parkinson's disease.

The Liver Cancer Algorithm (Houssein et al., 2023), introduced in the same year, is unique in its exploratory strategy that emulates the complex spread of liver cancer. This wide-reaching search is useful in tracking the multifaceted progression patterns typical of Parkinson's disease within patient data. Its ability to analyze a broad spectrum of data points makes it a strong contender for uncovering complex relationships within the data.

The Gold Rush Optimizer (Zolfi, 2023), also from 2023, reflects the historic rush to find gold, representing an unyielding search through uncharted territory. This is crucial in medical research, where datasets can be vast and complex. Its thorough exploration can lead to the discovery of new biomarkers for Parkinson's disease, which could be pivotal in early diagnosis and treatment.

Lastly, the Population Vortex search algorithm (Sağ, 2022) from 2022 features a spiral search pattern that effectively narrows the search space to focus on the most promising features. This characteristic is especially useful for Parkinson's disease datasets, where relevant features can be closely interlinked with other data points. By honing in on the core features, this algorithm supports the development of models that predict the presence and progression of the disease with impressive precision.

These algorithms contribute to the field of medical analytics by refining models for diseases like Parkinson's, allowing for better diagnosis and treatment plans. Their diverse strategies enhance model performance by identifying relevant features from the data, crucial for patient care and outcomes.

3.3 Feature Selection

Feature selection is a fundamental process in machine learning that significantly enhances model performance and efficiency. By identifying and retaining only the most relevant features, it helps in reducing over fitting, decreasing training time, and improving the interpret ability of models. This process is crucial for simplifying complex data, facilitating easier visualization and analysis, and ensuring models are both accurate and understandable. For this study, we use the Wrapper feature selection model. It has been empirically proven that wrappers obtain subsets with better performance than filters because the subsets are evaluated using a real modelling algorithm (Tawhid & Ibrahim, 2020).

Wrapper Method

The Wrapper method is a feature selection technique that relies on a search algorithm to evaluate different subsets of features and determine which subset allows a given predictive model to perform best. This method wraps around the model, using it as a black-box to score feature subsets based on their predictive power.

The Wrapper method operates by iteratively creating models with different combinations of input features, evaluating them using a predefined performance metric, often accuracy, and then selecting the combination that yields the best results. Unlike filter methods, which evaluate features based on intrinsic tests independent of the model, wrappers use the actual machine learning algorithm for which the features are being selected to evaluate their usefulness. This can lead to more predictive feature subsets but at the cost of computational intensity.

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ represent the full set of n features in a dataset. The power set $2^{\mathcal{X}}$ contains all possible subsets of $\mathcal{X}$. The Wrapper method seeks to find an optimal subset $\mathcal{S} \subseteq \mathcal{X}$ such that a performance metric J is optimized:

$$\mathcal{S}^* = \arg\max_{\mathcal{S} \subseteq \mathcal{X}} J(\mathcal{S})$$

where $J(\mathcal{S})$ is the performance of the predictive model M when trained and validated with features in $\mathcal{S}$. Typically, J is defined as the model's accuracy, precision, recall, or F1-score on a validation set or through cross-validation:

$$J(\mathcal{S}) = \text{Accuracy}(M(\mathcal{S}))$$

or

$$J(\mathcal{S}) = \frac{1}{K}\sum_{k=1}^{K} \text{Accuracy}(M_k(\mathcal{S}))$$

for K-fold cross-validation. The search strategy for $\mathcal{S}$ can be a greedy algorithm, genetic algorithm, or any other heuristic that efficiently explores the subset space $2^{\mathcal{X}}$.

One of the well-known algorithms used in the Wrapper method is the Recursive Feature Elimination (RFE), which has been widely used in various fields, including bioinformatics. For example, (Guyon et al., 2002) demonstrated the effectiveness of RFE in selecting genes related to cancer. RFE starts with all available features and recursively removes the least significant ones until the desired number of features is reached.

Another approach within the Wrapper methods is the use of genetic algorithms (GAs), as shown by (Vafaie & Jong, 1998), where features are treated like chromosomes, and the selection process mimics biological evolution. The genetic algorithm selects the best features over generations based on their fitness, which is determined by the model's performance.

The performance of the Wrapper method largely depends on two main aspects: the choice of the predictive model and the search strategy. (Kohavi & John, 1997) highlighted the importance of these aspects and demonstrated that the Wrapper method often outperforms filter methods, provided that the computational cost is

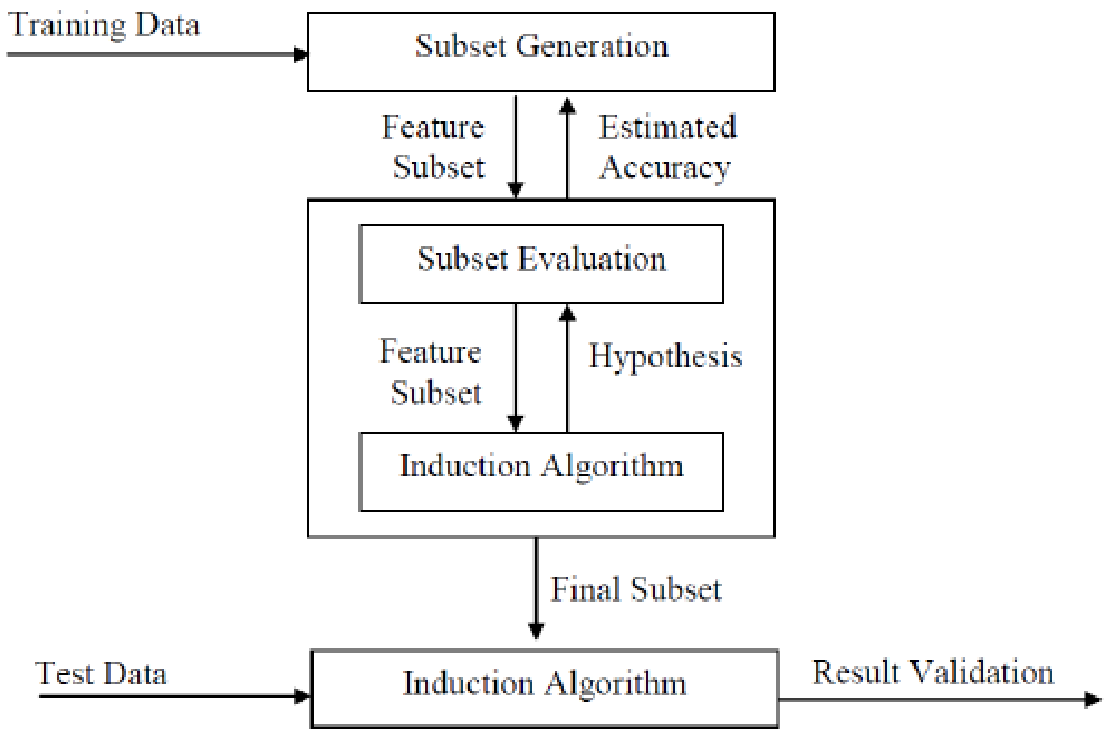

Figure 3.1: Wrapper method flowchart

not prohibitive.

Forward selection, backward elimination, and stepwise selection are other common search strategies in Wrapper methods. Forward selection starts with no features and adds them one by one, while backward elimination starts with all features and removes them iteratively. Stepwise selection is a combination of both, allowing features to be added or removed at each step.

One of the main advantages of the Wrapper method is its ability to consider feature interactions. In contrast to filter methods, which look at features in isolation, wrappers can uncover complex relationships between features that are only evident when used in conjunction with particular models. This characteristic is particularly beneficial when dealing with nonlinear relationships or interactions between features.

However, Wrapper methods can be prone to overfitting, especially if the dataset is small or if there are too many features relative to the number of observations. This overfitting is due to the exhaustive nature of the feature selection process, which might tailor the model too closely to the training data. To mitigate this, cross-validation is often used within the Wrapper method to evaluate feature sets, as suggested by Kohavi [4].

Despite their effectiveness, Wrapper methods are computationally expensive. The need to train and evaluate a model for each candidate feature subset can result in significant computational overhead, particularly for large datasets with a vast number of features. Consequently, wrappers are less frequently applied to very high-dimensional datasets unless computational resources are substantial.

In practical applications, Wrappers are often used when the predictive accuracy of the final model is paramount and when computational resources are sufficient to handle the extensive search process. They have been successfully applied in many scientific domains, including but not limited to medical diagnosis, image recognition, and financial forecasting, where selecting the right set of features can be critical to performance.

Machine Learning Models

Machine learning models offer transformative potential for diagnosing and managing Parkinson's Disease (PD). By processing and learning from vast and complex medical datasets, these models can uncover subtle patterns associated with the early stages of PD that might elude traditional diagnostic methods. Features such as voice patterns, motor movements, and even genetic markers can be fed into algorithms to predict the onset and progression of the disease with high accuracy [17]. The adaptive nature of ML models means they can continuously improve as they are exposed to more data, which is particularly valuable in tracking the progression of PD over time or in response to treatment. Moreover, machine learning can personalize patient care by predicting individual responses to therapies, enhancing the quality of life for those affected.

K-Nearest Neighbour

The K-Nearest Neighbors (KNN) classifier is a fundamental machine learning algorithm that operates on a simple principle: it classifies a new data point based on the majority vote of its 'k' nearest neighbors. The number 'k' represents how many neighbors influence the classification, which is a key parameter in the model. The simplicity of KNN lies in its intuitive approach—classify based on similarity in a feature space. As seen in the pseudocode provided earlier, KNN starts by setting a k-value, then standardizes the training data to ensure uniformity in feature measurement scales, which aids in accurate distance calculation between data points.

The pseudocode describes a step-by-step process for the KNN classifier. Initially, the algorithm separates the dataset into training and validation sets. It then normalizes the data, which is critical as KNN's effectiveness is dependent on the metric of distance between points (often Euclidean distance). Normalization ensures that no single feature disproportionately influences the result due to its scale. After normalization, the algorithm uses the trained model to predict outcomes for

Algorithm 1 K-Nearest Neighbors Classifier

```
1: procedure CLASSIFIER_KNN($feat$, $label$, $HO$)
2:     $k \leftarrow 5$                                         ▷ Set the number of neighbors
3:     $xtrain \leftarrow feat[HO.training, :]$
4:     $xvalid \leftarrow feat[HO.test, :]$
5:     $ytrain \leftarrow label[HO.training]$
6:     $yvalid \leftarrow label[HO.test]$
7:     $meanFeat \leftarrow$ Mean of each column in $xtrain$
8:     $stdFeat \leftarrow$ Standard deviation of each column in $xtrain$
9:     $xtrain \leftarrow (xtrain - meanFeat)/stdFeat$    ▷ Standardize training features
10:    $xvalid \leftarrow (xvalid - meanFeat)/stdFeat$ ▷ Standardize validation features
11:    $Model \leftarrow$ Train KNN using $xtrain$, $ytrain$, and $k$
12:    $ypred \leftarrow$ Predict labels for $xvalid$ using $Model$
13:    $Acc \leftarrow 100\times$ Sum of ($yvalid == ypred$) / Number of elements in $yvalid$
14:    return $Acc$
15: end procedure
```

validation data by finding the 'k' closest training samples to each point in the validation set and predicting the majority label among these nearest neighbors.

In medical settings, KNN has been effectively used for diagnostic purposes where classification of conditions is necessary from complex datasets. For example, in breast cancer diagnosis, KNN can classify tissue samples based on features derived from biopsy results. Its non-parametric nature allows it to adapt quickly to new data, making it extremely useful in healthcare where diagnostic parameters can vary widely between patients. The algorithm's reliance on labeled data also means that its accuracy improves as more diagnostic data become available, which is often the case in medical databases.

KNN's effectiveness in medical applications can vary. Its performance is generally high when the dataset is well-curated and the features effectively capture the underlying patterns necessary for diagnosis. However, KNN can suffer from high computational costs with large datasets because it requires calculations across all feature points for each prediction, which can be a significant drawback in emergency medical situations requiring quick decision-making. Moreover, the choice of 'k' and the distance metric can significantly affect performance, requiring careful tuning based on specific medical data characteristics. Despite these challenges, when applied correctly, KNN can offer high accuracy and valuable insights, particularly in well-defined and controlled medical environments.

Random Forest

The Random Forest classifier is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) of the individual trees. Random

Forest corrects for decision trees' habit of overfitting to their training set, making it robust and accurate. The pseudocode provided illustrates the Random Forest process: it starts by specifying the number of trees to include in the forest, divides the dataset into training and validation sets, and checks data integrity before model training begins.

According to the pseudocode, the Random Forest model is built by training multiple decision trees on various subsets of the dataset, both in terms of samples and features. This method, known as bootstrap aggregating (or bagging), involves drawing random samples of the training dataset with replacement to train each tree. The model then uses averaging to improve the accuracy and control overfitting. The classification decision of the Random Forest is made based on the majority voting from all trees in the forest. Each tree votes for a class, and the class receiving the most votes becomes the model's prediction.

Random Forests are extensively used in the medical field due to their versatility and robustness. They are particularly effective in handling complex datasets with many variables, such as patient records that include demographic, biomedical, and clinical information. For instance, Random Forests have been successfully applied to predict various diseases, including diabetes and heart diseases, by assessing risk factors across large patient datasets. Their ability to provide insights into feature importance is also highly valuable in identifying key predictors and risk factors in disease onset and progression.

The effectiveness of Random Forest in medical applications is well-regarded, particularly because of its high accuracy and the ability to run in parallel, thus speeding up the training process. Moreover, Random Forests handle imbalanced data efficiently, making them suitable for medical datasets where some conditions are rare. However, they require careful tuning of parameters such as the number

Algorithm 2 Random Forest Classifier

```
1: procedure CLASSIFIER_RANDOM($feat$, $label$, $HO$)
2:     $numTrees \leftarrow 100$                                ▷ Number of trees in the forest
3:     $xtrain \leftarrow feat[HO.training, :]$
4:     $ytrain \leftarrow label[HO.training]$
5:     $xvalid \leftarrow feat[HO.test, :]$
6:     $yvalid \leftarrow label[HO.test]$
7:     if isEmpty($xtrain$) or isEmpty($ytrain$) then
8:         throw Training set is empty error
9:     end if
10:    if isEmpty($xvalid$) or isEmpty($yvalid$) then
11:        throw Validation set is empty error
12:    end if
13:    if isNumeric($ytrain$) then
14:        $ytrain \leftarrow$ Convert numbers to strings
15:    else if isCategorical($ytrain$) then
16:        $ytrain \leftarrow$ Convert categorical to strings
17:    end if
18:    $ypred \leftarrow$ Predict with $Model$ on $xvalid$
19:    if isNumeric($label$) then
20:        $ypred \leftarrow$ Convert strings to numbers
21:        $correct \leftarrow$ Count of matching elements in $yvalid$ and $ypred$
22:    else if isCategorical($label$) then
23:        $ypred \leftarrow$ Convert strings to categorical
24:        $correct \leftarrow$ Count of matching elements in $yvalid$ and $ypred$
25:    else
26:        $ypred \leftarrow$ Convert to strings for comparison
27:        $correct \leftarrow$ Count of matching strings in $yvalid$ and $ypred$
28:    end if
29:    $num\_valid \leftarrow$ Number of elements in $yvalid$
30:    $Acc \leftarrow 100 \times (correct/num\_valid)$                ▷ Calculate accuracy
31:    return $Acc$
32: end procedure
```

of trees and the depth of each tree to prevent overfitting and ensure that the model does not become too complex. They also tend to be less interpretable than single decision trees because of their complex ensemble structure, which can be a drawback in medical settings where explanation and transparency are critical. Despite these challenges, Random Forests remain a popular choice for predictive modeling in healthcare due to their robust performance across diverse scenarios.

Decision Tree

The Decision Tree classifier is a straightforward yet powerful tool used in both classification and regression tasks. It works by splitting the data into branches to form a tree structure, where each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or a continuous outcome. The entire model is built on the concept of making the most informative decisions at each node based on measures such as Gini impurity or information gain. As demonstrated in the pseudocode, the process begins with data partitioning followed by the application of the tree-building algorithm that splits data at the nodes based on feature values.

In the provided pseudocode for a Decision Tree, the algorithm starts by preparing the training and validation datasets. It checks if the labels need to be converted (for instance, from cell format to categorical), which is crucial for handling various data types seamlessly. The Decision Tree model is then trained using these prepared datasets, where the algorithm recursively divides the training set until it meets a stopping criterion, such as a maximum depth of the tree or a minimum number of samples per leaf. This recursive splitting is designed to create branches that maximize the homogeneity of the resultant nodes. Finally, the model predicts the output for validation data by following the decision rules established during the tree's construction.

Algorithm 3 Decision Tree Classifier

```
1: procedure CLASSIFIER_DECISION($feat$, $label$, $HO$)
2:     $xtrain \leftarrow feat[HO.training, :]$                        ▷ Training features
3:     $ytrain \leftarrow label[HO.training]$                          ▷ Training labels
4:     $xvalid \leftarrow feat[HO.test, :]$                            ▷ Validation features
5:     $yvalid \leftarrow label[HO.test]$                              ▷ Validation labels
6:     if isCell($ytrain$) then
7:         $ytrain \leftarrow$ Convert cell array to categorical
8:     end if
9:     if isCell($yvalid$) then
10:        $yvalid \leftarrow$ Convert cell array to categorical
11:    end if
12:    $Model \leftarrow$ Train Decision Tree using $xtrain$ and $ytrain$
13:    $ypred \leftarrow$ Predict with $Model$ using $xvalid$
14:    $correct \leftarrow$ Sum of equal comparisons between $yvalid$ and $ypred$
15:    $num\_valid \leftarrow$ Number of elements in $yvalid$
16:    $Acc \leftarrow 100 \times (correct/num\_valid)$                ▷ Calculate accuracy
17:    return $Acc$
18: end procedure
```

Decision Trees are extensively used in the medical field due to their interpretability, which is essential for clinical decision-making. Medical professionals can follow the path along the branches of the tree to understand the basis of the algorithm's predictions, which is invaluable for diagnostic support, patient management, and treatment planning. For example, Decision Trees have been effectively utilized to diagnose diseases based on symptoms and test results, helping to quickly identify critical conditions like myocardial infarction or to stratify patients based on their risk of developing specific ailments.

While Decision Trees are easy to interpret, their simplicity can also lead to their biggest drawback: susceptibility to overfitting, especially with very complex or noisy data. This overfitting can make them less reliable for some types of medical data, where the relationship between features and outcomes can be extraordinarily subtle. However, techniques like pruning (removing parts of the tree that do not provide additional predictive power), setting the maximum depth, and requiring a minimum number of samples per leaf are common practices used to enhance their robustness. Despite these challenges, when configured properly, Decision Trees can provide powerful diagnostic tools in the healthcare setting, offering rapid insights based on interpretable models.

3.4 Friedman Test

The Friedman test is a non-parametric statistical analysis technique used extensively when comparing three or more paired groups. It is particularly useful in cases where the data does not meet the assumptions required by one-way ANOVA, specifically the normality of residuals and homogeneity of variances. This makes the Friedman test ideal for scenarios such as algorithm comparison where normality cannot be assured.

In contexts such as comparing multiple algorithms across multiple datasets, the Friedman test evaluates the ranks of each algorithm within each dataset, rather than their raw scores. This ranking approach mitigates the impact of outliers and non-normal distribution shapes, which are common in real-world data, especially in complex fields like machine learning and data mining. Each algorithm is ranked per dataset, with the best performing algorithm getting the lowest rank and the poorest the highest. In the event of ties, average ranks are assigned.

The Friedman test statistic, χ_F^2, is calculated using the formula:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left(\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4} \right)$$

where N is the number of datasets, k is the number of algorithms, and R_j is the sum of ranks for the j-th algorithm across all datasets.

After computing the test statistic, it is compared against a critical value from the chi-square distribution with $k - 1$ degrees of freedom, where k is the number of algorithms being compared. If the computed χ_F^2 exceeds the critical value, there is evidence to reject the null hypothesis that all algorithms perform equally well, suggesting that there are significant differences in the performance of the algorithms.

If significant differences are found, post-hoc tests such as the Nemenyi test are typically conducted to make pairwise comparisons between the algorithms' performance. This step is crucial as it helps to pinpoint which algorithms differ from each other significantly, providing insights that are valuable for refining algorithmic approaches or choosing between competing models.

The sum of ranks also provides a direct method to rank the algorithms from best to worst. Algorithms with a lower sum of ranks are considered better since they consistently performed well across multiple datasets. This ranking is intuitive and easy to interpret, making it particularly appealing in reports and practical decision-making where stakeholders may not be familiar with the intricacies of statistical testing.

In practice, particularly in medical data analysis or any field where decisions have significant consequences, the interpretability and non-parametric nature of the Friedman test make it a preferred choice. For instance, when analyzing the effectiveness of different treatment regimens across various patient groups, the Friedman test can robustly ascertain differences without the strict assumptions required by more traditional parametric tests.

Moreover, the use of the Friedman test in algorithm comparison is part of a broader movement towards more robust, replicable science in machine learning and data analysis. By applying such statistically rigorous methods, researchers and practitioners can ensure that their findings are not just artifacts of peculiar data distributions or outlier effects, but reflect genuine differences in algorithm performance.

Lastly, while the Friedman test provides a powerful tool for statistical analysis in comparative studies, its reliance on rank data means that some information about the magnitude of differences between groups is lost. However, this loss is often a worthy trade-off for gaining robustness against non-normality and variance heterogeneity, which are typical challenges in real-world data. This balance between information retention and robustness needs to be carefully managed based on the specific requirements and constraints of each study.

Friedman test offers a rigorous framework for statistically evaluating the performance of multiple algorithms across diverse datasets, ensuring that the conclusions drawn from such studies are both robust and reliable. It supports a thorough understanding of how different algorithms perform relative to each other, guiding further research and application in the field.

3.5 Dataset Description

3.5.1 Benchmark Datasets

This study uses 10 benchmark medical datasets of various kinds for benchmark testing. The datasets are collected from the UCI machine learning repository. Table 1 describes the datasets. The purpose of choosing these 10 datasets is to examine whether the suggested algorithm can be assessed on various datasets. Here, some values in a few of the datasets need to be added. The values missing when each feature's mode is determined have been replaced with the most frequent value in the feature. Next, all the missing values in the corresponding features are replaced using the mode.

Table 3.2: The description of the medical datasets.

No.	Datasets	Samples	Features	Classes	Missing values
1	Breast Cancer Wisconsin (Original)	699	10	2	Yes
2	Cardiotocography	2126	23	3	No
3	Hepatitis	155	19	2	Yes
4	Indian liver patient dataset	583	10	2	No
5	Lung Cancer	32	56	3	Yes
6	Lymphography	148	18	2	No
7	Mice Protien Expression (MPED)	583	10	2	Yes
8	SPECT	267	22	2	No
9	Stat log (Heart)	270	13	2	No
10	Thoracic Surgery Dataset	470	17	2	No

3.5.2 Parkinson's Dataset

In evaluating the effectiveness of our hybrid algorithm, we selected four comprehensive Parkinson's datasets, each documenting a range of symptoms and markers associated with the disease. These datasets encompass biomedical voice measurements, dynamics of handwriting, and other clinical indicators that provide a multifaceted view of Parkinson's manifestations.

Table 3.3: The description of the Parkinson's datasets.

No.	Datasets	Samples	Features	Classes	Missing values
1	Parkinson's	195	24	2	No
2	Parkinson's speech features	756	754	2	No
3	Hand Meander (Pereira et al., 2016)	264	16	2	No
4	Hand Spiral (Pereira et al., 2016)	264	16	2	No

Voice measurement data is particularly valuable in Parkinson's research because the condition often leads to vocal impairments. Changes in how the vocal cords function can affect the pitch, loudness, and quality of a person's voice. This dataset allows our algorithm to analyze voice modulations to identify patterns that may signal the presence of Parkinson's.

Handwriting dynamics offer another dimension of analysis, as Parkinson's can significantly affect a person's fine motor skills. This dataset includes features such as pressure exerted while writing, the duration of pen strokes, and the speed of handwriting. These attributes are crucial in detecting tremors, rigidity, and bradykinesia, which are classic symptoms of Parkinson's.

By including both voice and handwriting datasets, our testing approach captures a broad spectrum of Parkinson's disease characteristics. The algorithm's

Table 3.4: Dataset Features Description for Hand PD dataset

Feature	**Description**
ID_EXAM	identifier of the exam (handwritten form)
IMAGE_NAME	name of the image
ID_PATIENT	identifier of the patient
CLASS_TYPE	1 = control group and 2 = patient group
GENDER	M = male and F = female
RIGHT/LEFT-HANDED	R = right handed and L = left handed
AGE	age (years)
RMS	root mean square (Equation 3 of the paper)
MAX_BETWEEN_ET_HT	the maximum difference between ET and HT radius
MIN_BETWEEN_ET_HT	the minimum difference between ET and HT radius
STD_DEVIATION_ET_HT	standard deviation of the difference between ET and HT radius
MRT	mean relative tremor
MAX_HT	maximum HT radius
MIN_HT	minimum HT radius
STD_HT	standard deviation of HT radius

performance on these datasets can demonstrate its potential in real-world applications. For instance, success in accurately classifying voice data can imply the algorithm's utility in non-invasive early screening tools. Similarly, its performance on handwriting data can show its effectiveness in monitoring disease progression and response to treatment.

The choice of these particular datasets was also driven by the need to understand how well the algorithm can handle different types of data. This includes continuous measurements like voice frequency, as well as more complex patterns found in handwriting. By proving its robustness across diverse data sources, the algorithm stands out as a versatile tool for healthcare practitioners.

In practical terms, the successful application of this algorithm could mean earlier and more precise identification of Parkinson's symptoms, enabling better patient care. For instance, it could aid in fine-tuning treatment regimens for patients by providing detailed assessments of symptom severity and progression. This, in turn, could lead to improved quality of life for patients through more personalized treatment plans.

Furthermore, demonstrating the algorithm's effectiveness on these datasets would underline its value in enhancing diagnostic tools and treatment strategies. It could pave the way for developing sophisticated applications that clinicians can use for early detection of Parkinson's, potentially slowing the disease's impact through timely intervention.

3.6 Evaluation Criteria

3.6.1 CEC (Congress of Evolutionary Computation)

The Congress on Evolutionary Computation (CEC) test suite is a benchmarking platform that provides a comprehensive set of functions designed to evaluate the performance of optimization algorithms. The CEC 2017 test suite, in particular, is structured to include a variety of functions that challenge these algorithms in ways that mimic the complexities of real-world problems. Utilizing the CEC 2017 test suite as an initial evaluation criterion for assessing the performance of the ten optimization algorithms offers a robust framework to understand their strengths and weaknesses.

Unimodal Variable Dimension Functions are characterized by a single global optimum. The main challenge here is the precise location of the peak within a high-dimensional space. For algorithms like the Sine Cosine Algorithm and the Multi-Verse Optimizer, this serves as a test of their exploitation capabilities. An algorithm that performs well on unimodal variable dimension functions demonstrates its ability to efficiently converge to a single best solution, which is indicative of its potential to handle tasks that require high precision, such as finding the most effective drug dosage in a medical trial.

Unimodal Fixed Dimension Functions are of a fixed dimension allow for a controlled environment to evaluate an algorithm's performance without the added complexity of scaling with dimensions. This environment is ideal for assessing algorithms such as the Heap-Based Optimizer and the Gold Rush Optimizer. An effective performance on these functions suggests that the algorithm is not only adept at exploitation but also capable of maintaining its efficiency as the problem's scale is constrained, much like optimizing the treatment regimen for a specific category of Parkinson's patients.

Multimodal Variable Dimension Functions are more complex, with multiple local optima in addition to the global optimum. Algorithms are required to differentiate between these multiple solutions and focus on the global best. This is where the true power of the Whale Optimization Algorithm and the Population Vortex search algorithm shines, as they need to exhibit their exploratory strength to avoid getting trapped in local optima. Success in these functions indicates an algorithm's utility in scenarios like analyzing genetic data, where multiple markers may be involved in the manifestation of Parkinson's disease, but only a combination of specific ones may be truly indicative of the condition.

For Multimodal Fixed Dimension Functions, the complexity is derived from the presence of many local optima, but the dimensionality is kept constant. This provides a test for algorithms like the Liver Cancer Algorithm and the Tyrannosaurus optimization algorithm to showcase their ability to handle complex landscapes while operating within a limited scope. An algorithm that excels in these tests is likely to be effective in medical datasets, where numerous features could act as potential indicators of a disease, but only a few are relevant for diagnostic purposes.

Using the CEC 2017 test suite as an evaluation criterion provides a multifaceted analysis of an algorithm's performance. It helps in assessing various aspects of an algorithm, such as its convergence speed, accuracy, robustness against getting stuck in local optima, and scalability. This is particularly relevant for medical datasets like those for Parkinson's disease, which are complex and may contain a mix of noise, redundancy, and intricacies in the feature space.

An algorithm that performs well across the diverse set of CEC 2017 functions can be considered reliable and versatile. This reliability is crucial when translating computational findings to medical applications, where the stakes are high, and the cost of errors can be significant. A well-performing algorithm on the CEC 2017 suite is one that can be trusted to uncover the patterns inherent in medical data, identify the relevant biomarkers for Parkinson's disease, and contribute to the development of decision-support systems in clinical settings.

3.6.2 Mean Function Value

Two primary metrics used in this regard are the mean function value and the standard deviation of the function values obtained from repeated runs of the algorithms on a set of numerical problems. These metrics are instrumental in gauging the performance, reliability, and robustness of the algorithms in finding the optimal solutions. The mean function value is calculated by taking the average of the best solutions found by an algorithm over multiple runs. It is given by the formula:

$$\text{Mean Function Value} = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{3.1}$$

where $f(x_i)$ represents the best solution found in the i-th run, and N is the total number of runs.

3.6.3 Standard Deviation

The standard deviation is a measure of the variation or dispersion of the function values from their mean. It is a critical statistic that describes the reliability of the algorithm—how consistent it is in producing solutions close to the mean value. The standard deviation is calculated as follows:

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^{N}(f(x_i) - \text{Mean Function Value})^2}{N-1}} \tag{3.2}$$

where $f(x_i)$ again denotes the best solution in the i-th run, and $N-1$ is the sample size for an unbiased estimate.

3.6.4 Accuracy

Accuracy is a crucial metric for evaluating the performance of classification algorithms, particularly after applying feature selection to refine the input variables and after benchmarking against complex numerical problems, such as those in the CEC test suite. Once feature selection is applied to the algorithms, it pares down the data to the most informative features, potentially leading to a significant improvement in classification accuracy. t can be expressed mathematically by the equation:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

An algorithm that achieves high accuracy in such a context demonstrates its ability to correctly identify cases of the disease, signifying its utility in clinical applications. After optimizing the algorithms with feature selection, improved accuracy would suggest that the removal of less pertinent features has enabled the model to concentrate on the data that most strongly indicate the presence of the disease.

When applied to the CEC test suite, which includes a variety of numerical problems ranging from unimodal to multimodal and from low to high dimensions, these metrics provide a comprehensive assessment of an algorithm's performance. The mean function value assesses how well the algorithm can optimize a given function on average, which is crucial in medical data analysis for identifying treatment effects or disease markers with high accuracy. Meanwhile, the standard deviation reflects the algorithm's consistency in finding optimal solutions. In the context of medical datasets like Parkinson's, a low standard deviation is desirable as it indicates that the algorithm can reliably identify key features or patterns associated with the disease, regardless of variations in the data due to patient diversity or measurement noise. Algorithms that achieve lower mean function values and standard deviations are considered superior, as they not only find better solutions on average but also do so with greater consistency. For researchers and practitioners, particularly in the medical field, these criteria are essential to ensure that the algorithms they rely on for decision-making are both effective and dependable.

3.7 Experimantal Setup

3.7.1 System Parameters

The experimental setup for the optimization algorithms utilizes an Apple MacBook Pro 14-inch with an M1 Pro chip and 16 GB of RAM, providing necessary computational power. The experiments are conducted in the MATLAB environment, which supports extensive numerical computation and visualization capabilities essential for handling complex machine learning tasks and medical dataset analyses. This setup ensures the algorithms are tested under consistent, high-performance conditions, facilitating reliable evaluations.

3.7.2 Parameter Setting and Tuning

The table presents a set of parameters that will be maintained across all optimization algorithms during the testing phase. These parameters are crucial as they establish a controlled environment, ensuring that the performance of each algorithm can be fairly compared. Specifically, 'N' denotes the number of search agents or

Table 3.5: Parameter Setting Values for CEC.

Parameters	Values	Description
N	30	Number of Search agents
t_{max}	500	Maximum Iterations
$runs$	25	Number of runs

individual solutions the algorithms will use, set to 30. This number is significant as it represents the population size that directly influences the search dynamics of each algorithm. t_{max}, representing the maximum number of iterations, is fixed at 100, setting a limit on the computational effort and the convergence time allowed for each algorithm to reach an optimal solution. Lastly, 'runs' specifies the number of independent trials to be conducted at 25. This ensures statistical reliability in the performance metrics, as multiple runs can account for the variability inherent to stochastic processes and provide a robust average measure of the algorithms' capabilities. By keeping these parameters constant, the experimental setup aims to mitigate external factors that could bias the evaluation, focusing solely on the innate performance characteristics of each algorithm.

The algorithm-specific parameters for the Multi-Verse Optimizer (MVO) and the Tyrannosaurus optimization algorithm (TREX) are tailored to fine-tune their search mechanisms and adaptability to various optimization landscapes. For MVO, maxwep and minwep dictate the maximum and minimum wormhole existence probabilities, respectively. The maximum value is set at 1, providing the algorithm with the potential to fully exploit the wormhole mechanism for global search capabilities. This high probability ensures that MVO can explore distant regions of the search space, which is beneficial for escaping local optima and is critical in complex medical datasets where global optima represent the most accurate diag-

Table 3.6: Algorithm Specific Parameters

Algorithm	Parameters	Values	Description
MVO	max_{wep}	1	Maximum Wormhole Existence Probability
MVO	min_{wep}	0.2	Minimum Wormhole Existence Probability
TREX	sr	0.2	Success rate
TREX	tr	0.2	trex running rate
TREX	pr	0.1	prey running rate
LCA	$jump_strength$	0.6v	step size
RTH	A	15	Angular Component
RTH	$R0$	0.5	Radial Distance
GRLCA	M_p	0.1	Mutation Probablity

nostic markers. Conversely, the minimum wormhole existence probability is set at 0.2, maintaining a balance between exploration and exploitation by preventing the algorithm from becoming too erratic or random in its search, which could compromise the precision necessary for medical data analysis.

TREX's parameters, including the success rate (sr), trex running rate (tr), and prey running rate (pr), are chosen to reflect the predatory-prey dynamics that characterize the algorithm. A success rate of 0.2 allows TREX to adapt its strategy based on the success of previous iterations, emulating an evolutionary approach that favors successful traits. The running rates for both the trex (tr) and the prey (pr) are set to ensure that the algorithm maintains a competitive tension between the predator and prey behaviors. By calibrating these rates at 0.2 and 0.1, respectively, TREX maintains a deliberate pace that is neither too fast—risking premature convergence—nor too slow, which would be computationally inefficient.

Chapter 4

Proposed Algorithm

This project introduces a novel hybrid algorithm that combines the strengths of Gold Rush Optimization (GRO) and the Liver Cancer Algorithm (LCA) to create a robust feature selection mechanism. This combination is optimized for use with classifiers such as k-nearest neighbors (KNN), Decision Trees, and Random Forests, and is particularly useful for medical datasets, including those used for diagnosing and managing Parkinson's disease. To understand the hybrid algorithm better, we will look at the individual algorithms first

4.1 Gold Rush Optimizer

The Gold Rush Optimizer (GRO) is a novel optimization algorithm that has garnered attention for its efficiency and robustness in tackling complex optimization problems across various domains. As a member of the metaheuristic optimization family, GRO distinguishes itself by its unique approach to searching for optimal solutions, offering promising results where traditional methods fall short. This optimizer is particularly valued for its ability to rapidly converge to high-quality solutions, making it an excellent tool for applications in engineering, data science, and operational research. The subsequent sections will delve into the inspiration behind GRO and explain its operational mechanics in detail.

4.1.1 Inspiration

The Gold Rush Optimizer (GRO)(Zolfi, 2023) algorithm draws its inspiration from the historical gold rushes that captivated the world in the 19th century. These events were characterized by rapid migration of people upon the discovery of gold in various regions, most notably in California, Australia, and South Africa. Prospectors, often referred to as gold diggers, would flock to a location in hopes of finding gold, which was seen as a quick path to wealth. The GRO algorithm abstracts this human behavior into a search strategy, where each agent (prospector) explores the search space (the landscape) for the most valuable solutions (gold).

In GRO, the search space is conceptualized as a rugged landscape dotted with potential gold mines. Each prospector in the algorithm makes decisions based on local environmental cues, simulating the real-world scenario where prospectors would rely on tools like gold pans, sluice boxes, and their personal experience to locate and evaluate gold deposits. The operations within GRO—migration, mining, and collaboration—mimic real-world strategies. Migration reflects the prospectors' movement to new areas based on hearsay or observed success of others, mining represents the intensive search and excavation activities in chosen locations, and collaboration symbolizes the sharing of insights and joint ventures often necessary in challenging or resource-scarce environments.

The effectiveness of GRO, like the historical gold rushes, hinges on the balance between exploration and exploitation, a common theme in search and optimization algorithms. Exploration in GRO is driven by the randomness of prospectors' movements and their occasional long jumps to unexplored territories, reflecting the sudden shifts in population during a gold rush when new promising areas were discovered. Exploitation is depicted by the detailed and focused search efforts in areas known to have high values, akin to the methodical panning and sluicing in regions where gold had been found. This dual strategy helps the GRO algorithm effectively navigate complex and multimodal search landscapes, making it particularly suitable for optimization problems where the global optimum is hidden among numerous local optima.

4.1.2 Working

The GRO begins with a population of agents, each representing a gold prospector. These agents are initially scattered randomly across the search space. The algorithm progresses through a series of generations, with each agent attempting to move towards areas of higher fitness, which are considered to have higher concentrations of 'gold'. This movement is guided by a set of mathematical rules that simulate the decision-making process of whether to continue exploring the current area or move to a new area, based on the potential for finding richer gold deposits.

GRO effectively balances exploration and exploitation, which are key dynamics in any optimization algorithm. Exploration involves searching over a wide area to avoid local minima, while exploitation focuses on thoroughly searching a promising area to fine-tune the solution. In GRO, exploration is achieved by allowing prospectors to randomly jump to distant parts of the search space, simulating the sudden rush to a newly discovered goldfield. Exploitation is conducted by intensifying the search around areas where gold has already been found, refining the solutions.

The position of each prospector i at time $t + 1$ is updated by the following equation:

$$x_i^{t+1} = x_i^t + \alpha \cdot \text{rand}() \cdot (x_{\text{best}}^t - x_i^t) + \beta \cdot \text{rand}() \cdot (x_{\text{local}}^t - x_i^t)$$

where:

- x_i^t is the current position of the i-th prospector at time t.
- x_{best}^t is the position of the best solution found so far by any prospector.
- x_{local}^t is the best solution found in the neighborhood of the i-th prospector.
- α and β are coefficients that control the influence of the global best and local best positions, respectively.
- rand() is a random number generator function that produces a uniform random value between 0 and 1.

One of the strengths of the GRO is its robust global search capability. The algorithm is designed to prevent premature convergence to local optima by maintaining a diverse population of prospectors. This diversity ensures that different parts of the search space are continuously explored, increasing the likelihood of finding the global optimum. The global search properties are particularly advantageous when dealing with complex, multimodal functions where multiple local optima exist.

The balance between exploration and exploitation is maintained by dynamically adjusting the coefficients α and β based on the progress of the search:

$$\alpha = \alpha_{\max} - \left(\frac{\alpha_{\max} - \alpha_{\min}}{\text{max\_iter}} \right) \cdot t$$

$$\beta = \beta_{\min} + \left(\frac{\beta_{\max} - \beta_{\min}}{\text{max\_iter}} \right) \cdot t$$

where:

- $\alpha_{\max}$ and $\alpha_{\min}$ are the maximum and minimum values for α, respectively.
- $\beta_{\max}$ and $\beta_{\min}$ are the maximum and minimum values for β, respectively.
- t is the current iteration number.
- max_iter is the maximum number of iterations.

The adaptability of GRO comes from its parameter settings, which control the intensity and frequency of exploration and exploitation. These parameters can be adjusted based on the nature of the problem, allowing the algorithm to be tailored to specific optimization needs. The flexibility in parameter settings makes GRO suitable for a wide range of problems, from simple to highly complex optimization challenges.

In the medical field, optimization algorithms like the GRO can be used to optimize parameters in complex predictive models, such as those used in disease diagnosis, treatment planning, or drug development. For example, GRO can optimize the parameters of a neural network that predicts patient outcomes based on a large set of clinical variables. Its ability to escape local optima and explore the global search space effectively ensures that the model does not settle for suboptimal solutions.

Algorithm 4 Gold Rush Optimizer (GRO)

```
Initialize parameters: dimension $dim$, bounds $lb$ and $ub$, $N$, $Max\_iter$
$sigma\_initial \leftarrow 2$, $sigma\_final \leftarrow 1/Max\_iter$
Initialize global best position $best\_pos \leftarrow \text{zeros}(1, dim)$
Initialize global best score $best\_score \leftarrow \infty$
Initialize population $Positions \leftarrow \text{initialization}(N, dim, lb, ub)$
Initialize fitness $Fit \leftarrow \text{inf}(1, N)$
Initialize new positions $X\_NEW \leftarrow Positions$
Initialize new fitness $Fit\_NEW \leftarrow Fit$
Initialize convergence curve $Convergence\_curve$
$Convergence\_curve(1) \leftarrow \min(Fit)$
$iter \leftarrow 1$
while $iter \leq Max\_iter$ do
    for $i \leftarrow 1$ to $N$ do
        Calculate fitness $Fit\_NEW(i) \leftarrow fobj(X\_NEW(i,:))$
        if $Fit\_NEW(i) < Fit(i)$ then
            $Fit(i) \leftarrow Fit\_NEW(i)$
            $Positions(i,:) \leftarrow X\_NEW(i,:)$
        end if
        if $Fit(i) < best\_score$ then
            $best\_score \leftarrow Fit(i)$
            $best\_pos \leftarrow Positions(i,:)$
        end if
    end for
    Update $l1, l2$ based on iteration count
    for $i \leftarrow 1$ to $N$ do
        Select two different prospectors for collaboration
        Choose operation: migration, mining, or collaboration
        Update position $X\_NEW(i,:)$ based on selected operation
        Ensure $X\_NEW(i,:)$ is within bounds $lb$ and $ub$
    end for
    $Convergence\_curve(iter) \leftarrow best\_score$
    $iter \leftarrow iter + 1$
end while
return $best\_score, best\_pos, Convergence\_curve$
```

The global search properties of GRO make it an excellent candidate for hybrid optimization models. In such models, GRO can be combined with other optimization techniques that have strong local search capabilities. For instance, combining GRO with a gradient-based method can be very effective: GRO can be used to identify a good region of the search space (global search), and then a gradient-based method can fine-tune the solution within that region (local search).

In medical data analysis, this hybrid approach can be particularly useful. For example, when optimizing a model to identify the best combination of treatment plans for cancer therapy, GRO can explore various combinations on a global scale, and once promising combinations are identified, a local search method can optimize the dosage and timing of treatments.

Medical datasets often involve complex relationships between variables, with the landscape of possible models being rugged and peppered with many local optima. GRO's strategy to mimic the gold rush enables it to navigate these landscapes effectively, making it a valuable tool in medical research where finding the global optimum can significantly impact patient outcomes.

While GRO is powerful, its efficiency and scalability depend on its implementation. For large-scale problems, such as those involving big data in medicine, the computational cost of running the algorithm needs to be managed carefully. Efficient implementation and the use of parallel computing techniques can help mitigate these costs, making GRO a practical choice for large-scale optimization problems.

As the algorithm continues to be refined and adapted, its application in the medical field is expected to grow. Further research into its parameters and hybridization with other methods will enhance its effectiveness, potentially making

GRO a staple in medical data analysis and other fields requiring robust global optimization solutions.

The Gold Rush Optimizer represents a significant advancement in the field of optimization, combining historical inspiration with modern algorithmic techniques to solve complex problems effectively. Its application in medical data analysis, particularly when combined in a hybrid model, shows promising prospects for future developments and enhancements.

4.2 Liver Cancer Algorithm

The Liver Cancer Algorithm (LCA) is an emerging optimization technique specifically tailored to address complex problem-solving scenarios, drawing considerable interest for its specialized application in the medical field. This algorithm is designed to optimize decision-making processes by efficiently navigating through vast solution spaces, thus proving to be a vital tool in medical diagnostics and treatment planning. LCA's development is particularly focused on enhancing outcomes in liver cancer management, leveraging its precision to refine treatment protocols and improve patient care. Further details about the inspiration and operational mechanics of LCA will be explored in the following sections.

4.2.1 Inspiration

The Liver Cancer Algorithm (LCA) (Houssein et al., 2023)is inspired by the dynamic and complex process of liver cancer progression in biological systems. Liver cancer, like many cancers, involves uncontrolled cell growth and proliferation that can eventually lead to metastasis, where cancer cells spread to other parts of the body. The LCA abstracts these biological phenomena into a computational model, aiming to mimic the aggressive and invasive characteristics of liver cancer cells to efficiently explore and exploit the search space in optimization problems.

In the LCA, each agent or solution in the algorithm represents a group of liver cancer cells, and the search space symbolizes the human body. The movement of these agents through the search space is analogous to the way cancer cells invade new tissues and adapt to new environments. This movement is guided by various strategies encoded within the algorithm, such as migration, mining, and collaboration, each representing different survival and proliferation tactics used by cancer cells. Migration reflects the metastatic spread of cancer to new regions, mining represents the local expansion and exploitation of resources, and collaboration indicates the cooperative interactions between different groups of cancer cells, which can lead to more robust colonization of tissues.

The effectiveness of the LCA in solving optimization problems comes from its ability to balance exploration and exploitation, mirroring the biological efficiency of liver cancer cells in optimizing their survival and growth within a host. This balance allows the algorithm to avoid local optima and continue searching for better solutions, much like cancer cells continually evolve to overcome new challenges in their environment, such as nutrient shortages or immune responses. Thus, the LCA not only provides a powerful method for finding optimal solutions in complex landscapes but also offers a unique perspective on the adaptability and resilience of biological systems, specifically the formidable nature of cancerous processes.

4.2.2 Working

At its core, LCA models the aggressive and dynamic nature of cancer cells, adapting these biological strategies to search for optimal solutions in a defined problem space. Initially, the algorithm establishes a population of agents, each representing a potential solution within the boundaries of the problem. These agents, akin to tumor cells, are placed randomly within the search space defined by lower and upper bounds (lb and ub).

During the initialization phase, each agent's position is determined through a random distribution, ensuring a diverse starting point for the optimization process. This diversity is crucial for the comprehensive exploration of the search space. The primary goal of each agent is to minimize its associated 'Energy,' analogous to the fitness value in optimization terms. This Energy is assessed using a specified objective function, which quantitatively evaluates how well a given solution addresses the problem.

As the algorithm progresses, it iterates through a series of generations, simulating the proliferation and adaptation of cancer cells. Each agent's position is updated based on strategic behaviors modeled after cancer cell dynamics: migration, mining, and collaboration. The migration strategy is inspired by metastatic spread, allowing agents to move towards the best solution found so far. This is mathematically represented by the equation:

$$X_{i,d}^{new} = X_{i,d} + A1 \cdot (C1 \cdot BestPos_d - X_{i,d})$$

where $A1$ and $C1$ are coefficients influenced by random factors, and $BestPos$ represents the position of the best solution.

Mining behavior reflects the local exploitation where agents delve deeper into their current locations, influenced by nearby solutions. This involves adjusting an agent's position based on another randomly selected agent, incorporating random perturbations to thoroughly explore the vicinity:

$$X_{i,d}^{new} = X_{digger,d} + A2 \cdot (X_{i,d} - X_{digger,d})$$

where $A2$ adjusts the influence of the selected nearby agent (digger), and *digger* is another agent randomly chosen from the population.

Collaboration, on the other hand, is modeled by averaging the positions of two randomly selected agents and adding a small perturbation. This strategy fosters information sharing and cooperative search, potentially leading to better exploration of the search space

$$X_{i,d}^{new} = \frac{X_{i,d} + X_{collab,d}}{2} + \text{small random perturbation}$$

where *collab* is a randomly selected agent different from i.

The positions are also constrained within the problem's bounds, ensuring all potential solutions remain valid. Agents whose new positions yield better fitness scores update their locations, continually refining the quality of solutions. Throughout this process, the algorithm keeps track of the lowest Energy discovered, updating the global best position whenever a lower Energy is found.

LCA's iterative process continues until a specified number of generations is reached, with each iteration potentially introducing more effective solutions. The algorithm's performance is recorded at each step, forming a convergence curve that illustrates the progression towards optimum fitness.

Algorithm 5 Liver Cancer Algorithm (LCA)

```
 1: Display "LCA is now tackling your problem"
 2: Start timer
 3: Initialize parameters: $Tumor\_Energy \leftarrow \infty$, $Tumor\_Location \leftarrow$ zeros$(1, dim)$
 4: Initialize tumor locations $X \leftarrow$ initialization$(N, dim, ub, lb)$
 5: Initialize convergence curve $CNVG$
 6: $t \leftarrow 0$                                          ▷ Loop counter
 7: while $t < T$ do
 8:     for $i = 1$ to $N$ do
 9:         Check and fix boundaries for $X(i, :)$
10:         Calculate fitness: $fitness \leftarrow fobj(X(i, :))$
11:         if $fitness < Tumor\_Energy$ then
12:             $Tumor\_Energy \leftarrow fitness$
13:             $Tumor\_Location \leftarrow X(i, :)$
14:         end if
15:     end for
16:     Calculate $l2, l1$ for current iteration
17:     for $i = 1$ to $N$ do
18:         Select random indices for collaboration and competition
19:         Decide random operation: migration, mining, or collaboration
20:         Update position $X(i, :)$ based on chosen operation
21:         Check and fix boundaries for $X(i, :)$
22:         Update position with mutation or Levy flight if improved
23:     end for
24:     $t \leftarrow t + 1$
25:     $CNVG(t) \leftarrow Tumor\_Energy$
26: end while
27: Stop timer
28: Display execution time and best score
29: return $Tumor\_Energy, Tumor\_Location, CNVG$
```

Ultimately, the algorithm concludes with the best solution found across all agents, alongside a detailed account of the optimization journey reflected in the convergence curve. This outcome not only provides an optimal solution but also offers insights into the algorithm's search dynamics and efficiency.

4.3 Hybrid Working Mechanisms

The hybrid algorithm, combining the strengths of the Gold Rush Optimizer (GRO) and the Liver Cancer Algorithm (LCA), represents a significant advancement in the field of optimization. This innovative approach merges the comprehensive global search capabilities of GRO with the meticulous local search strategies of LCA, creating a robust tool that excels in solving highly complex problems. Specifically designed to leverage the best features of both algorithms, this hybrid model enhances efficiency and accuracy, particularly in applications demanding precision and adaptability. The synergy between GRO's global perspective and LCA's local insights promises a deeper and more effective exploration and exploitation of the solution space, which will be elaborated upon in the subsequent sections.

4.3.1 Inspiration

The inspiration behind combining the Gold Rush Optimizer (GRO) and the Liver Cancer Algorithm (LCA) into a hybrid model arises from the complementary strengths of each algorithm in addressing the limitations of the other, particularly in their search capabilities within the optimization landscape. This hybrid approach leverages the robust global search capabilities of GRO with the efficient local search strategies of LCA, creating a synergistic effect that enhances the overall search efficacy across diverse problem domains.

The hybridization of GRO and LCA aims to construct a comprehensive search algorithm that harnesses the global search strength of GRO and the local search precision of LCA. By integrating these approaches, the hybrid model ensures that once GRO identifies a promising area in the global search space, LCA can then perform an intensive local search to refine and enhance the solutions found within these areas. This method is particularly beneficial in complex optimization problems where both extensive exploration and intensive exploitation are necessary to achieve the best results.

4.3.2 Working

The hybrid GRO-LCA algorithm begins with a diverse population of solutions, each assessed for its effectiveness in optimizing the performance of a specific classifier—KNN, Decision Tree, or Random Forest. At the core of this hybrid algorithm is the iterative process that aims to identify the most relevant features from a given dataset that contribute significantly to the predictive modeling accuracy. The algorithm begins by initializing a population of candidate solutions within the predefined bounds of the feature space. Each candidate solution represents a potential subset of features, encoded as a binary vector where the presence or absence of a feature is marked by 1s and 0s respectively.

To balance between the exploration of the search space and the intensive exploitation of promising regions, the GRO-LCA hybrid algorithm employs a randomized decision process at each iteration. A random condition determines whether to apply the GRO strategy for global search or the LCA strategy for local search. The GRO approach updates the positions of the candidate solutions based on velocity vectors inspired by the metaphorical process of prospecting and mining for gold. It encourages a widespread search and rapid movement towards promising solutions. In contrast, the LCA strategy focuses on refining existing solutions through a process analogous to the aggressive and invasive nature of liver cancer growth, emphasizing local exploitation and detailed search in the neighborhood of the current best solution.

Mutation plays a crucial role in maintaining genetic diversity within the population. It is applied with a certain probability, introducing random changes to the candidate solutions. This helps prevent the algorithm from becoming stuck in local optima and encourages the discovery of new and potentially better solutions.

The algorithm also incorporates a memory mechanism to record the best solutions found throughout the search process. If the current population does not yield improvements, the algorithm can reintroduce the best solutions from memory, ensuring that valuable information is not lost over time.

The iterative process continues until the maximum number of iterations is reached or another stopping condition is satisfied. At the end of the run, the algorithm outputs the best solution found, which corresponds to the optimal or near-optimal feature subset, alongside a convergence curve that illustrates the improvement of the best solution's fitness over iterations.

By integrating the complementary strengths of GRO and LCA, the hybrid GRO-LCA algorithm achieves a fine balance between diversifying the search across the feature space and intensifying the focus on promising areas. This makes it particularly effective for feature selection in medical datasets, where the correct identification of a concise set of diagnostic markers is critical for accurate disease classification and prognosis, as in the case of Parkinson's disease.

Algorithm 6 Hybrid GRO-LCA Algorithm

```
 1: Set initial values of population size $PopSize$, max iterations $maxIter$, and control parameters
 2: Randomly initialize the population with solutions $x_i \in S$
 3: Initialize LCA frequency vectors $Q$ and GRO velocity vectors $v$ with zero values
 4: Evaluate fitness of each solution using selected classifiers (KNN, DT, RF)
 5: Initialize best solution vector $X_{best}$, minimum fitness $F_{min}$
 6: Set iteration counter $t = 0$
 7: while $t < maxIter$ do
 8:     for $i = 1$ to $PopSize$ do
 9:         if rand() < 0.5 then                ▷ Apply GRO strategy for global search
10:             for each feature $j$ in $x_i$ do
11:                 Update velocity $v(i, j)$ based on GRO equations
12:                 Update position $x_i(j)$ using new velocity
13:             end for
14:         else                                ▷ Apply LCA strategy for local exploration
15:             Update frequency $Q(i, j)$ based on LCA equations
16:             Perform Lévy flight to modify $x_i$
17:             Perform local refinement on $x_i$
18:         end if
19:         if rand() < mutation_rate then
20:             Mutate($x_i$) based on mutation function
21:         end if
22:         Evaluate new fitness for $x_i$
23:         Update $X_{best}$ and $F_{min}$ if new fitness is better
24:         if rand() < memory_reintroduction_rate then
25:             Reintroduce best solutions from memory
26:         end if
27:     end for
28:     $t = t + 1$
29: end while
30: return best solution $X_{best}$ and its feature set
```

Chapter 5

Discussion

5.1 Experimental results

The results of the comparitive analysis suggest that among the ten algorithm, the Gold Rush Optimizer (GRO), Liver Cancer Algorithm (LCA), Heap Based Optimizer (HBO), and Population Vortex Search (PVS) emerged as the top-performing algorithms in terms of mean fitness value. This implies that these algorithms, on average, found solutions closer to the optimal value, showcasing their efficiency and potency in navigating complex search spaces to deliver high-quality solutions. In terms of standard deviation, GRO, LCA, HBO, and TTAO distinguished themselves, indicating not only their robustness in finding good solutions but also their consistent performance across multiple runs. Such consistency is paramount in applications where reliability is as crucial as the quality of the final solution, ensuring that the optimization process is dependable and stable under various conditions and instances of the problem set.

5.1.1 CEC Results

The comparative analysis of various optimization algorithms as depicted in Table 5.1 and 5.2 provides significant insights into the performance of these methods across different metrics. These numerical metrics clearly depict how algorithms like GRO, LCA, and HBO excel quantitatively. GRO, with the lowest values in both metrics, illustrates its superior optimization capabilities, efficiently navigating the solution spaces to achieve optimal solutions consistently. The low standard deviation values associated with GRO and HBO indicate that these algorithms are not only effective in terms of fitness but also deliver consistent results across different runs and datasets, making them reliable choices for applications where stability is crucial.

The values corresponding to LCA highlight its robustness, especially in scenarios simulating its target domain of medical applications, where precise and reliable outcomes are essential. The numerical backing further solidifies the graphical representation, ensuring that the visual conclusions drawn from the bar charts are well-founded. Such detailed quantitative analysis is critical as it confirms the visual trends and provides a deeper understanding of each algorithm's performance nuances, enabling users to make informed decisions based on both the average performance and variability of results.

Table 5.1: CEC Results - Mean Fitness Value

F(x)	SCA	MVO	SAO	HBO	TTAO	TREX	RTH	LCA	GRO	PVS
Unimodal Variable Dim										
F1	1.09E+01	1.10E+00	1.70E-02	5.49E-07	9.57E-01	6.76E+04	4.62E+00	1.61E-01	4.54E-62	5.42E-63
F2	8.08E-06	2.85E-07	3.17E-05	5.50E-03	4.05E-06	8.27E-01	2.68E+00	4.07E-07	2.88E-160	1.68E-170
F3	1.31E-01	7.96E+00	1.01E+01	1.08E-04	1.11E+01	1.18E+03	3.53E+00	5.69E-01	3.70E-39	2.75E-40
F4	3.65E+01	1.69E+00	5.01E-02	1.30E+01	1.40E+01	8.66E+01	3.78E+00	7.13E-02	2.90E-02	1.03E+00
F5	1.54E+01	1.24E+00	1.50E-02	1.54E-01	5.81E-01	6.76E+04	4.62E+00	1.48E-01	7.95E-02	6.79E-02
F6	-9.82E+01	-1.41E+02	-1.27E+02	-1.55E+02	-1.55E+02	-2.77E+01	-1.51E+02	-1.55E+02	-1.28E+02	-1.26E+02
F7	4.02E-02	2.89E+24	2.09E+23	3.00E-04	8.10E+01	4.50E+42	3.53E+00	1.40E+00	2.59E-39	5.31E-40
F8	1.17E+06	2.28E-10	1.09E-14	1.51E-13	1.20E+01	8.87E+09	3.78E+00	1.13E-16	1.01E-112	1.69E-128
F9	8.41E+04	3.50E+02	1.53E+00	1.10E+02	9.25E+02	2.55E+08	1.98E+01	1.25E-01	2.67E+01	2.66E+01
F10	6.59E-03	3.26E-03	1.66E-03	5.90E-03	1.27E-01	8.78E+08	2.68E+00	3.99E-04	2.35E-64	8.86E-65
F11	2.13E+02	4.36E+00	2.98E-01	5.45E+00	6.50E+01	1.82E+06	4.20E+00	2.76E-01	6.67E-01	6.67E-01
F12	1.34E+01	2.47E+00	5.51E-02	3.29E+00	8.81E+00	1.53E+04	3.78E+00	5.83E-03	8.67E-05	7.04E-05
F13	2.88E-180	1.24E-132	4.32E-52	-8.61E-01	4.34E-232	1.73E-32	3.62E+00	-8.67E-01	6.46E-149	7.53E-145
F14	8.73E+01	3.27E+00	9.24E+02	1.12E+01	6.81E+00	7.44E+05	2.72E+00	1.78E+00	1.38E+01	1.13E+01

Table 5.1 – continued from previous page

F(x)	SCA	MVO	SAO	HBO	TTAO	TREX	RTH	LCA	GRO	PVS
F15	1.65E+00	1.31E+00	8.56E-02	7.44E-08	1.55E+01	9.80E+03	3.53E+00	3.64E-02	2.69E-63	3.73E-62
Unimodal Fixed Dim										
F1	3.08E-04	6.10E-02	1.37E-03	3.25E-09	8.50E-33	2.63E+00	3.90E+00	9.83E-01	0.00E+00	1.39E-32
F2	1.11E-03	9.42E-07	8.04E-02	0.00E+00	0.00E+00	9.60E+00	4.62E+00	6.95E-01	0.00E+00	0.00E+00
F3	1.38E-87	1.38E-87	3.27E-07	1.38E-87	1.38E-87	1.47E+01	2.68E+00	1.38E-87	1.38E-87	1.38E-87
F4	1.93E-56	2.10E-08	3.80E-06	5.50E-03	2.43E-88	3.90E-01	3.53E+00	2.99E-06	1.26E-198	4.26E-184
F5	2.93E-01	2.93E-01	3.03E-01	2.93E-01	2.93E-01	4.27E-01	4.07E+00	2.95E-01	2.93E-01	2.93E-01
F6	1.92E+01	1.91E+01	3.15E+06	4.91E+01	1.91E+01	3.10E+08	2.37E+01	1.22E+01	1.91E+01	1.91E+01
F7	7.04E-04	2.28E-07	3.11E-04	7.46E-03	9.21E-14	1.71E+00	2.68E+00	1.96E-03	1.50E-10	3.35E-11
F8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.53E+00	0.00E+00	0.00E+00	0.00E+00
F9	-3.79E-03	-3.79E-03	1.39E-02	-3.79E-03	-3.79E-03	9.22E+00	3.78E+00	-3.46E-03	-3.79E-03	-3.79E-03
Multimodal Variable Dim										
F1	2.93E+02	1.61E+02	2.55E+02	3.13E+01	1.58E+02	3.48E+02	1.41E+02	1.35E+02	1.52E+02	1.46E+02
F2	3.52E+01	1.14E+02	3.23E+01	1.31E+01	6.05E+01	4.39E+02	2.68E+00	1.00E+01	8.10E-01	1.39E+00
F3	5.59E+00	1.01E+00	4.95E+00	2.60E+00	2.78E+00	1.14E+01	4.43E+00	3.79E+00	2.88E+00	2.74E+00
F4	3.00E+07	1.11E+03	7.64E+06	2.24E-01	1.04E+02	2.13E+11	3.78E+00	2.34E+03	3.48E+02	3.93E+02

Table 5.1 – continued from previous page

F(x)	SCA	MVO	SAO	HBO	TTAO	TREX	RTH	LCA	GRO	PVS
F5	8.34E-01	4.82E+00	2.38E-02	6.08E-04	1.48E+00	6.44E+01	4.62E+00	9.43E-03	2.79E-39	4.45E-39
F6	3.48E-01	7.88E+00	4.17E-03	3.05E+00	1.78E+01	5.76E+13	2.68E+00	6.51E-05	6.45E-48	6.11E-56
F7	1.52E+01	1.70E+00	8.39E-02	1.85E-04	6.83E+00	2.06E+01	3.53E+00	1.27E-01	3.25E-15	3.39E-15
F8	5.04E+02	2.83E+02	1.64E+03	9.03E+00	1.57E+02	1.68E+06	3.11E+01	1.01E+02	2.97E+01	2.56E+01
F9	1.07E+00	7.96E-01	1.56E+00	5.01E-01	1.05E+00	2.69E+01	4.62E+00	1.93E-01	1.40E-01	1.36E-01
F10	-5.81E+02	-1.01E+03	-1.17E+03	-1.17E+03	-1.14E+03	-4.57E+02	-1.01E+03	-1.17E+03	-1.08E+03	-1.09E+03
F11	5.10E-01	1.56E-01	3.88E-02	1.19E-03	7.74E-02	1.83E+01	3.53E+00	1.38E-02	3.31E-04	0.00E+00
F12	3.31E-10	5.04E-15	2.51E-15	1.00E+01	3.79E-13	1.86E-07	2.78E+00	-8.48E-01	7.39E-13	7.86E-13
F13	8.04E-10	5.27E-11	4.48E-09	3.13E-11	3.21E-11	1.02E-02	4.62E+00	3.52E-12	6.78E-10	3.97E-10
F14	3.53E+05	1.84E-01	7.51E-02	4.40E-04	3.69E+01	1.18E+09	2.74E+00	1.60E-02	2.03E-01	1.40E-01
F15	2.63E+05	2.22E+00	2.40E+00	8.29E-03	1.05E+01	5.71E+08	3.53E+00	2.29E-03	4.85E-03	3.93E-03
F16	-7.47E+00	-1.27E+01	-1.10E+01	-2.12E+01	-2.27E+01	-6.20E+00	-2.17E+01	-7.44E+00	-1.76E+01	-1.70E+01
F17	1.35E-01	3.52E-02	2.38E-02	3.38E-02	2.49E-01	1.16E+02	4.62E+00	7.59E-04	5.02E-03	5.25E-03
Multimodal Fixed Dim										
F1	1.19E+01	1.28E+00	5.20E+01	7.90E-07	6.95E-01	6.97E+04	2.68E+00	1.61E-01	1.01E-60	5.71E-62
F2	1.13E-09	3.60E-02	3.79E-01	1.92E-17	5.26E-05	1.30E+03	3.53E+00	-5.08E-02	1.10E-64	7.37E-66

Table 5.1 – continued from previous page

F(x)	SCA	MVO	SAO	HBO	TTAO	TREX	RTH	LCA	GRO	PVS
F3	6.35E-03	1.05E-01	7.17E+02	2.02E-01	1.25E-01	1.82E+04	3.78E+00	1.67E+00	2.15E-55	1.05E-55
F4	7.26E-04	9.17E-02	1.30E-01	1.65E-03	3.86E-01	6.74E+01	4.62E+00	-7.28E-02	2.00E-39	2.36E-39
F5	7.51E+00	1.68E+02	5.27E-01	5.79E+00	2.15E+01	3.22E+07	2.68E+00	5.91E-01	5.63E+00	5.59E+00
F6	4.76E-01	1.47E-02	8.95E+00	2.57E-25	1.69E-06	1.43E+04	3.53E+00	4.98E-02	7.45E-09	1.76E-08
F7	2.42E-03	2.73E-03	1.62E-02	6.03E-03	6.89E-02	7.90E+00	3.78E+00	5.78E-04	9.75E-04	1.05E-03
F8	-2.19E+03	-2.93E+03	-2.09E+03	-2.51E+03	-3.05E+03	-1.27E+03	-3.13E+03	-2.51E+03	-3.94E+03	-3.95E+03
F9	1.07E+00	1.49E+01	2.10E+01	0.00E+00	9.75E+00	1.20E+02	2.68E+00	2.46E+01	6.15E-01	2.44E-14
F10	1.43E-01	4.14E-01	9.12E-01	1.34E-13	2.22E+00	1.98E+01	3.53E+00	-1.06E-01	3.43E-15	3.29E-15
F11	6.66E-02	3.28E-01	1.71E+00	3.03E-03	6.44E-02	1.34E+02	3.78E+00	-3.56E-01	1.89E-03	8.21E-03
F12	1.02E-01	5.72E-02	1.76E+00	7.69E-03	1.37E-01	7.61E+07	4.64E+00	2.19E-03	2.38E-09	4.68E-09
F13	3.10E-01	6.97E-03	6.51E-01	1.81E-26	3.61E-02	1.88E+08	2.70E+00	1.26E-02	6.09E-08	1.64E-08
F14	2.34E+00	9.98E-01	5.53E+00	9.98E-01	1.83E+00	1.16E+02	6.33E+00	1.07E+01	9.98E-01	9.98E-01
F15	1.06E-03	6.80E-03	6.88E-03	7.45E-04	8.20E-04	2.33E-01	3.78E+00	6.52E-03	3.51E-04	3.44E-04
F16	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	1.02E+00	3.59E+00	-8.33E-01	-1.03E+00	-1.03E+00
F17	4.00E-01	3.98E-01	4.45E-01	3.98E-01	3.98E-01	2.27E+00	3.08E+00	-9.65E-01	3.98E-01	3.98E-01
F18	3.00E+00	3.00E+00	4.60E+00	9.03E+00	3.00E+00	9.07E+01	6.53E+00	9.02E+00	3.00E+00	3.00E+00

Table 5.1 – continued from previous page

F(x)	SCA	MVO	SAO	HBO	TTAO	TREX	RTH	LCA	GRO	PVS
F19	-3.85E+00	-3.86E+00	-3.85E+00	-3.86E+00	-3.86E+00	-3.13E+00	-8.28E-02	-3.15E+00	-3.86E+00	-3.86E+00
F20	-2.88E+00	-3.24E+00	-2.52E+00	-4.83E+00	-3.30E+00	-1.52E+00	1.34E+00	-4.83E+00	-3.32E+00	-3.32E+00
F21	-2.57E+00	-7.52E+00	-6.97E+00	-9.85E+00	-1.02E+01	-4.75E-01	-3.01E+00	-4.88E+00	-1.02E+01	-1.02E+01
F22	-4.26E+00	-9.16E+00	-7.65E+00	-1.00E+01	-1.02E+01	-6.75E-01	-2.65E+00	-1.19E+01	-1.04E+01	-1.04E+01
F23	-4.07E+00	-7.50E+00	-8.59E+00	-1.04E+01	-1.05E+01	-7.92E-01	-2.38E+00	-5.06E+00	-1.05E+01	-1.05E+01

Table 5.2: CEC Results - Standard Deviation

F(x)	SCA	MVO	SAO	HBO	TTAO	TREX	RTH	LCA	GRO	PVS
Unimodal Variable Dim										
F1	2.64E+01	3.85E-01	1.74E+01	1.15E-06	5.10E-01	7.59E+03	2.27E+00	1.76E-01	3.86E-62	1.17E-61
F2	1.90E-05	3.56E-07	2.59E-05	2.22E-21	2.42E-06	3.34E-01	1.53E+00	0.00E+00	0.00E+00	0.00E+00
F3	1.60E-01	3.62E+00	6.42E+01	8.54E-05	1.52E+01	7.71E+01	1.80E+00	1.20E+00	3.06E-39	3.68E+00
F4	1.04E+01	6.97E-01	9.25E-02	4.13E+00	4.45E+00	3.79E+00	2.27E+00	8.19E-02	6.41E-01	1.47E-01
F5	2.07E+01	2.59E-01	5.52E-01	6.07E-01	6.11E-01	7.12E+03	2.27E+00	1.74E-01	8.55E-02	6.07E-01
F6	5.19E+00	5.23E+00	1.74E+01	0.00E+00	0.00E+00	7.45E+00	5.18E+00	0.00E+00	3.58E+00	3.14E+00
F7	3.51E-02	4.39E+24	4.37E+20	4.37E+20	5.25E+01	4.02E+43	1.80E+00	1.14E+00	8.64E-40	4.37E+20
F8	2.86E+06	1.18E-09	3.07E-13	1.72E-13	1.85E+01	3.43E+09	2.27E+00	1.66E-129	1.66E-114	5.50E-02
F9	2.22E+05	3.23E+02	8.08E+00	6.07E+01	4.74E+02	4.58E+07	3.10E+00	1.30E+00	5.17E-01	8.13E+00
F10	1.66E-02	1.05E-03	2.30E-03	1.18E-09	1.39E-01	7.83E+07	1.53E+00	6.66E-04	2.35E-64	5.61E-02
F11	6.28E+02	8.63E+00	3.14E-02	1.38E+00	5.04E+01	3.81E+05	1.80E+00	7.21E-02	1.00E-05	1.86E-05
F12	2.17E+01	1.24E+00	3.94E-02	1.93E-01	5.58E+00	5.06E+03	2.27E+00	2.35E-03	2.71E-04	9.85E-05
F13	0.00E+00	7.48E-138	3.32E-48	0.00E+00	0.00E+00	2.95E-34	2.27E+00	3.27E-01	1.44E-142	1.94E-159
F14	7.78E+01	7.96E+00	1.24E+03	7.20E+00	1.63E+01	3.25E+05	1.64E+00	1.64E+00	4.78E+00	8.61E+00

Table 5.2 – continued from previous page

F(x)	SCA	MVO	SAO	HBO	TTAO	TREX	RTH	LCA	GRO	PVS
F15	3.26E+00	1.40E+00	5.66E-02	5.96E-08	6.52E+00	1.18E+03	1.80E+00	2.96E-02	1.55E-64	4.23E-63
Unimodal Fixed Dim										
F1	2.24E-04	3.11E-01	6.87E-03	6.19E-02	5.55E-33	2.58E+00	2.42E+00	2.00E+00	7.68E-33	6.19E-02
F2	9.89E-04	8.68E-07	6.82E-05	0.00E+00	3.47E-31	9.24E+00	2.27E+00	0.00E+00	0.00E+00	5.51E-02
F3	4.56E-103	4.56E-103	6.25E-07	4.56E-103	4.56E-103	8.40E+00	1.53E+00	4.56E-103	4.56E-103	5.50E-02
F4	2.16E-58	3.68E-08	4.57E-06	4.68E-21	3.25E-91	6.11E-01	1.80E+00	1.29E-06	0.00E+00	5.50E-02
F5	5.61E-06	4.10E-06	1.78E-02	3.22E-10	1.52E-16	3.90E-02	2.27E+00	2.55E-03	8.84E-12	4.65E-10
F6	3.77E-02	7.57E-03	3.01E+05	1.06E-14	2.61E-15	6.66E+08	2.27E+00	4.19E+00	7.32E-15	7.57E-15
F7	5.28E-04	4.72E-07	2.84E-04	2.21E-04	1.35E-13	1.39E+00	1.53E+00	4.76E-03	1.05E-09	1.14E-08
F8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.80E+00	0.00E+00	0.00E+00	5.50E-02
F9	1.08E-09	1.94E-07	7.39E-02	1.33E-18	1.31E-18	1.74E+01	2.27E+00	4.05E-04	1.33E-18	1.33E-18
Multimodal Variable Dim										
F1	1.03E+01	2.31E+01	9.52E+01	1.73E+01	1.55E+01	1.63E+01	2.62E+01	1.56E+02	1.40E+01	1.95E+01
F2	2.96E+01	2.68E+01	3.95E+01	5.19E+00	1.83E+01	2.46E+01	1.53E+00	5.40E+01	9.23E+00	2.69E+01
F3	1.42E+00	4.00E-03	3.56E+00	4.21E-01	8.12E-01	9.54E-01	1.80E+00	3.91E+00	9.57E-01	3.62E+00
F4	6.48E+08	3.46E+02	2.87E+08	8.13E-01	5.92E+01	3.51E+10	2.27E+00	8.13E-01	2.90E+02	2.35E+02

Table 5.2 – continued from previous page

F(x)	SCA	MVO	SAO	HBO	TTAO	TREX	RTH	LCA	GRO	PVS
F5	1.89E+00	2.08E+00	1.38E-02	7.23E-04	1.49E+00	5.80E+00	2.27E+00	9.00E-03	7.53E-39	1.31E-37
F6	1.42E+02	6.03E+01	2.17E-02	2.81E-01	5.65E+00	1.91E+14	1.53E+00	1.63E-04	1.03E-58	3.41E-57
F7	9.44E+00	4.58E-01	1.50E+00	1.15E-04	1.61E+00	1.59E-01	1.80E+00	8.46E-02	1.18E-15	1.56E+00
F8	1.09E+03	5.18E+01	2.11E+04	1.23E+00	4.30E+01	1.82E+05	7.66E+00	1.23E+00	8.05E+00	7.45E+00
F9	3.28E-01	1.63E-01	9.52E-01	8.28E-02	1.43E-01	1.03E+00	2.27E+00	8.66E-02	4.74E-02	4.58E-02
F10	5.27E+01	4.58E+01	3.07E-02	5.35E-07	4.34E+01	6.67E+01	3.45E+01	9.36E-02	4.17E+01	2.94E+01
F11	2.75E-01	3.09E-02	2.02E-01	4.04E-03	6.86E-02	1.59E+00	1.80E+00	1.31E-02	0.00E+00	2.57E-01
F12	2.52E-10	2.71E-15	2.77E-15	6.15E-14	2.01E-13	1.29E-07	2.27E+00	5.43E-02	2.59E-13	1.59E-13
F13	8.70E-10	8.49E-11	1.69E-07	3.36E-12	2.49E-12	1.29E-02	2.27E+00	1.41E-14	1.82E-09	4.34E-10
F14	2.06E+05	5.71E-02	1.61E-01	3.64E-03	2.53E+01	2.34E+08	2.12E+00	8.44E-03	1.41E-01	1.19E-01
F15	5.32E+05	1.15E+00	1.43E+00	1.47E-07	4.80E+00	1.24E+08	1.80E+00	1.83E-03	2.87E-03	4.48E-03
F16	8.74E-01	1.30E+00	9.08E-01	1.93E+00	3.13E+00	6.66E-01	3.91E+00	8.04E-01	1.36E+00	1.63E+00
F17	1.29E-01	1.63E-02	2.62E-02	7.98E-03	1.27E-01	2.27E+01	2.27E+00	5.20E-04	3.22E-03	2.19E-03
Multimodal Fixed Dim										
F1	1.85E+01	3.51E-01	2.62E+02	7.72E-07	7.02E-01	6.54E+03	1.53E+00	1.89E-01	1.40E-62	7.17E-62
F2	2.57E-09	1.56E-02	6.41E-02	1.24E-16	5.12E-05	2.70E+03	1.80E+00	3.21E-02	7.97E-66	1.17E-64

Table 5.2 – continued from previous page

F(x)	SCA	MVO	SAO	HBO	TTAO	TREX	RTH	LCA	GRO	PVS
F3	3.18E-03	7.10E-02	9.80E+02	8.32E-01	1.80E-01	4.20E+03	2.27E+00	3.17E+00	6.31E-52	9.80E+02
F4	1.44E-03	2.84E-02	1.76E-02	3.78E-03	1.84E-01	5.80E+00	2.27E+00	8.28E-02	2.52E-38	4.61E-37
F5	4.74E-01	1.29E+02	7.96E-02	1.37E+01	2.10E+01	1.68E+07	1.53E+00	3.85E-01	3.84E-01	3.66E-01
F6	1.31E-01	7.44E-03	1.82E+02	1.51E-24	2.77E-05	3.92E+03	1.80E+00	1.03E-01	7.76E-08	8.58E-07
F7	2.16E-03	1.56E-03	1.49E-02	2.03E-03	4.80E-02	2.76E+00	2.27E+00	3.30E-04	5.87E-04	7.42E-04
F8	1.80E+02	3.09E+02	4.01E+02	3.28E+01	2.23E+02	2.86E+02	4.04E+02	1.10E+03	1.65E+02	4.01E+02
F9	3.43E+00	6.52E+00	1.78E+01	2.75E-01	3.47E+00	1.51E+01	1.53E+00	2.79E+01	0.00E+00	1.92E+00
F10	6.95E-03	6.93E-01	5.88E-01	9.34E-14	1.19E+00	6.34E-01	1.80E+00	6.96E-02	1.33E-15	4.01E+02
F11	1.83E-01	1.15E-01	2.83E+00	6.22E-03	3.79E-02	3.14E+01	2.27E+00	6.22E-03	1.12E-02	2.18E-02
F12	3.29E-02	4.40E-01	5.81E+00	2.50E-27	3.81E-01	4.27E+07	2.27E+00	8.87E-03	3.30E-09	4.41E-08
F13	8.58E-02	7.26E-03	5.15E-01	1.12E-26	4.81E-02	8.12E+07	1.57E+00	1.12E-26	6.92E-07	4.01E+02
F14	9.08E-01	2.36E-11	2.42E+00	0.00E+00	1.60E+00	1.63E+02	4.94E+00	1.59E+01	0.00E+00	2.48E+00
F15	3.42E-04	1.47E-02	9.79E-03	1.44E-04	3.21E-04	2.04E-01	2.28E+00	3.00E-02	4.11E-05	7.33E-05
F16	4.40E-05	4.15E-07	1.93E-04	6.49E-16	6.57E-16	9.94E-01	2.27E+00	1.47E-01	6.65E-16	6.65E-16
F17	3.78E-03	4.44E-07	9.62E-02	0.00E+00	0.00E+00	1.16E+00	1.53E+00	0.00E+00	1.26E-15	1.51E-01
F18	7.93E-05	2.24E+01	5.37E+00	4.80E-16	1.31E-15	7.77E+01	7.20E+00	1.02E+01	1.05E-15	9.89E-16

Table 5.2 – continued from previous page

F(x)	SCA	MVO	SAO	HBO	TTAO	TREX	RTH	LCA	GRO	PVS
F19	2.40E-03	1.41E-06	1.98E-02	2.27E-15	2.17E-15	3.84E-01	2.27E+00	3.72E-01	2.27E-15	7.48E-02
F20	1.32E-01	6.19E-02	3.53E-01	4.80E-16	3.44E-02	4.87E-01	2.33E+00	4.52E-01	2.72E-07	1.45E-07
F21	2.08E+00	3.14E+00	2.90E+00	1.31E+00	4.06E-04	4.63E-01	3.71E+00	1.44E+00	3.50E-06	1.06E-05
F22	1.58E+00	3.05E+00	2.20E+00	1.78E+00	1.16E+00	2.10E-01	4.34E+00	1.35E+00	5.87E-08	2.26E+00
F23	1.49E+00	3.02E+00	3.11E+00	1.08E-01	1.82E-01	2.68E-01	4.72E+00	1.94E+00	2.71E-11	3.17E+00

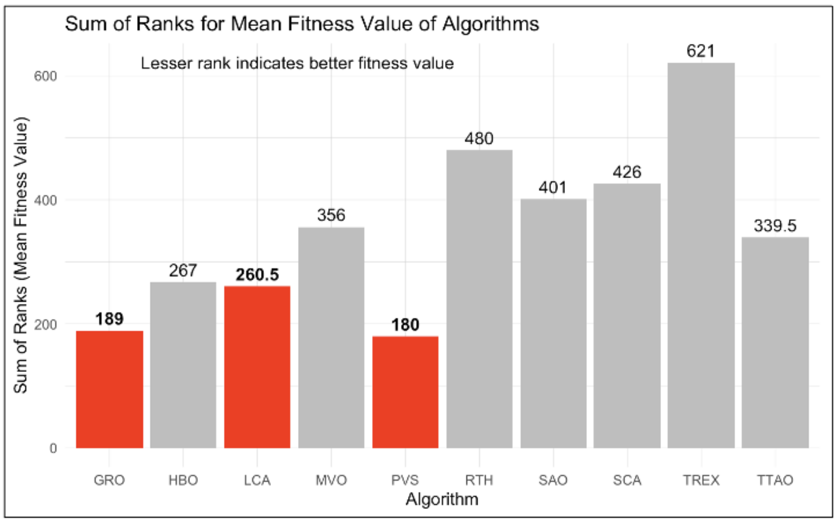

Figure 5.1: Mean Fitness Value of benchmark functions - CEC

In Figure 5.1, which evaluates the sum of ranks for the mean fitness value of each algorithm, three algorithms stand out for their superior performance: GRO (Gold Rush Optimizer), LCA (Liver Cancer Algorithm), and PVS (Population Vertex Search). GRO achieves the lowest rank sum, indicating it frequently achieves higher mean fitness values across the tested datasets. This suggests that GRO is particularly effective at navigating the solution space to find optimal or near-optimal solutions efficiently.

From LCA also demonstrates commendable performance, with a sum of ranks slightly higher than GRO but still significantly lower than many other algorithms. This highlights its effectiveness in the context of optimization, particularly in complex problem spaces that may be analogous to its medical applications, such as optimizing treatment plans or diagnostic parameters. PVS, while not as effective as GRO or LCA, still outperforms several other algorithms, suggesting its utility in certain contexts where specific features of the population-based search can be leveraged.

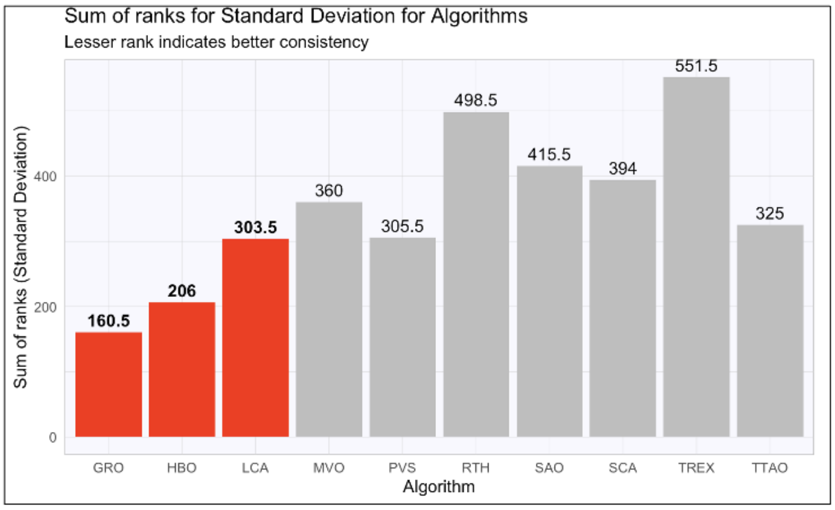

Figure 5.2: Standard deviation of benchmark functions - CEC

In terms of standard deviation, Figure 5.2 reveals insights into the consistency of these optimization algorithms. Here, GRO and HBO (Heap Based Optimizer) exhibit the lowest sum of ranks, suggesting that these algorithms not only perform well on average but also maintain this performance across multiple runs and different datasets. This level of consistency is crucial in practical applications where reliability and predictability of performance are essential. LCA maintains a strong position in this metric as well, reinforcing its robustness as an optimization tool capable of delivering stable results across various scenarios.

Together, these results underscore the strengths and weaknesses of the considered algorithms. GRO appears to be the most robust choice, offering both high performance and consistency. LCA also proves to be highly effective, particularly valuable in fields requiring precision and reliability, such as medical applications. These insights, supported by actual values in your report, provide a solid foundation for selecting an appropriate optimization algorithm based on specific needs and conditions of the application environment. This analysis not only aids in algorithm selection but also in understanding the behavior of these algorithms in diverse scenarios, ultimately guiding future improvements and adaptations.

5.1.2 Feature Selection Results for Medical Datasets

A detailed evaluation of the performance of ten optimization algorithms against three distinct classifiers—Decision Tree, KNN, and Random Forest is conducted.

Accuracy, as represented by the sum of ranks across the benchmark functions, gauges the algorithm's precision in identifying optimal or near-optimal solutions. Lower ranks indicate a closer approximation to the global optimum. Standard deviation provides insight into the consistency of the algorithms; lower values suggest less variation between runs, indicating a stable search process that is less sensitive to initial conditions or stochastic fluctuations. Together, these metrics furnish a holistic view of each algorithm's performance, considering both the quality of the solution and the reliability of the process by which it is found.

Table 5.3: Accuracy using KNN classifier

Dataset/Algorithm	**GRO**	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**
Breast Cancer Dataset	96.78	95.72	**97.27**	96.04	96.11	62.83	96.74	96.32	96.46	96.14
Cardiotocography	98.45	96.85	97.89	96.48	97.91	66.93	**99.12**	97.29	98.08	97.96
Hepatitis	**96.43**	94.71	93.71	91.43	92.00	64.14	91.43	94.71	92.86	89.86
Indian Liver dataset	75.00	74.07	73.97	79.14	76.72	62.83	73.45	76.90	**81.03**	74.00
Lung cancer	**100.00**	84.00	**100.00**	77.33	99.33	66.93	94.00	96.00	99.33	99.33
MPED	98.43	98.11	97.57	98.50	**99.00**	64.14	98.44	99.48	99.56	97.17
SPECT	92.83	91.85	92.98	92.23	**94.26**	62.83	91.77	94.64	93.81	88.15
STAT log	**88.89**	**88.89**	**88.89**	87.63	83.26	66.93	86.96	94.64	**88.89**	85.11
Thoracic surgery	85.11	85.11	85.11	85.11	85.11	64.14	**86.17**	85.11	86.11	85.11
Lymphography	**97.13**	91.72	93.10	91.86	91.86	64.14	85.52	93.10	93.10	93.10

Table 5.4: Standard Deviation of Algorithm Performance using KNN classifier

Dataset/Algorithm	**GRO**	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**
Breast Cancer Dataset	0.434	0.331	0.245	0.451	0.442	**0.000**	0.493	0.609	**0.000**	0.717
Cardiotocography	0.166	1.114	**0.047**	0.562	0.472	13.120	0.108	0.883	0.222	0.148
Hepatitis	**0.000**	2.751	1.557	2.916	1.557	5.405	2.525	1.821	**0.000**	3.037
Indian Liver dataset	**0.000**	0.239	0.352	0.610	**0.000**	**0.000**	0.862	0.862	**0.000**	1.890
Lung cancer	**0.000**	3.333	**0.000**	11.667	3.333	13.120	8.165	7.265	3.333	3.333
MPED	0.887	0.755	0.859	0.685	0.173	5.405	0.743	**0.154**	0.433	1.520
SPECT	1.806	1.943	1.280	1.663	1.763	**0.000**	1.797	2.015	1.681	1.280
STAT log	**0.000**	2.268	textbf0.000	1.386	0.370	13.120	0.370	1.279	textbf0.000	1.646
Thoracic surgery	**0.000**	0.000	0.000	0.000	0.000	5.405	0.000	0.000	0.000	0.000
Lymphography	0.000	1.724	0.000	1.961	1.689	13.120	1.991	**0.000**	0.000	0.000

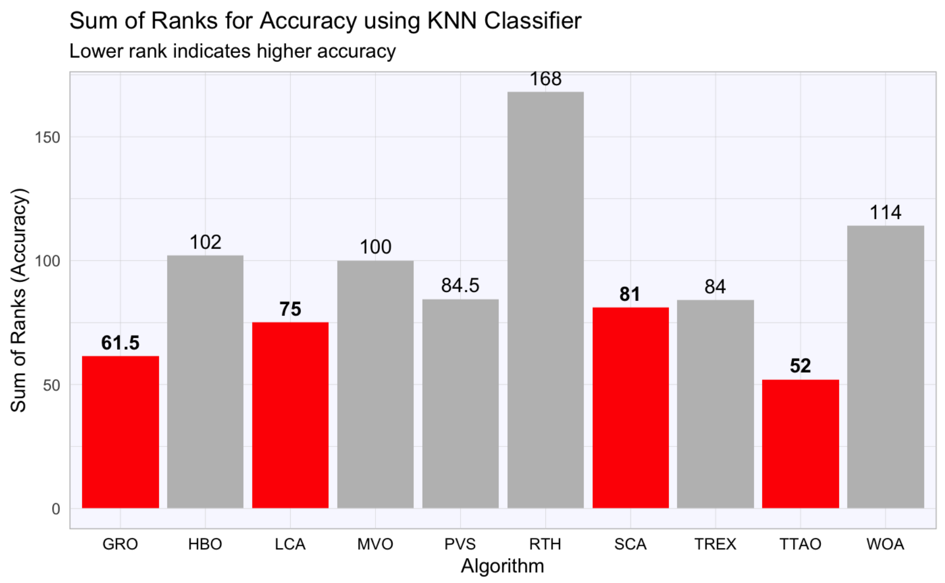

Figure 5.3: Sum of Ranks for Accuracy

When examining Figure 5.3, it's evident that the algorithms exhibit a spread in their ability to identify high-quality solutions. The Gold Rush Optimizer (GRO) once again achieves a low rank, showcasing its consistent accuracy across different classifiers. The Triangulation Topology Aggregation Algorithm (TTAO), however, demonstrates superior performance with the lowest rank, indicating that its specific mechanisms may be particularly well-tuned to the characteristics of the KNN classification process, such as its reliance on local neighborhood data.

Figure 5.4 reveals that the Liver Cancer Algorithm (LCA) outperforms other algorithms with a significantly lower rank, emphasizing its consistent performance across trials. This highlights the LCA's ability to maintain performance levels regardless of the inherent randomness in the KNN's k-selection and distance measurement, making it a strong candidate for scenarios where performance stability is as crucial as accuracy.

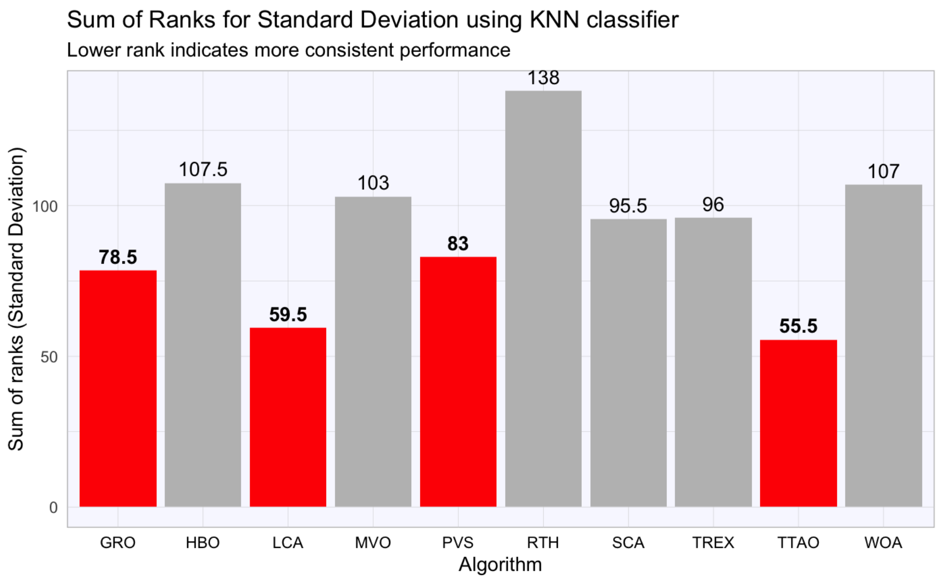

Figure 5.4: Sum of Ranks for Standard Deviation

Table 5.5: Accuracy of Algorithms Using Random Forest Classifier

Dataset/Algorithm	**GRO**	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**	**GRO_LCA**
Breast Cancer Dataset	95.72	93.20	95.47	95.75	93.88	93.95	93.06	93.42	96.32	93.03	**96.52**
Cardiotocography	99.51	99.02	99.84	99.59	99.82	93.69	99.92	99.08	98.34	99.28	**99.51**
Hepatitis	92.86	92.43	91.86	94.43	**97.00**	72.71	93.71	92.43	87.29	89.14	93.60
Indian Liver dataset	72.48	76.14	74.93	75.31	**77.34**	69.10	74.07	75.76	71.86	72.07	73.17
Lung cancer	72.67	52.00	85.33	46.67	80.67	71.00	**77.33**	74.67	78.00	62.67	73.52
MPED	97.76	97.61	98.30	99.20	97.65	70.00	97.20	96.83	96.57	96.54	**96.97**
SPECT	82.34	83.70	85.06	84.38	84.53	72.00	86.42	83.70	86.94	83.77	**83.09**
STAT log	92.96	93.70	92.74	88.37	83.63	72.00	86.15	83.56	85.70	89.33	**93.79**
Thoracic surgery	80.64	81.15	83.62	80.53	80.85	72.00	80.17	77.66	82.45	82.98	**81.45**
Lymphography	92.97	79.45	94.48	90.21	90.62	71.00	88.00	88.83	85.93	90.48	**93.70**

Table 5.6: Standard Deviation of Algorithms Using Random Forest Classifier

Dataset/Algorithm	**GRO**	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**	**GRO_LCA**
Breast Cancer Dataset	1.708	1.218	1.561	1.022	1.140	1.483	1.104	1.305	**0.979**	1.207	1.04
Cardiotocography	0.191	0.200	0.189	0.228	**0.156**	4.588	0.134	0.214	0.249	0.220	4.62
Hepatitis	4.124	3.763	3.344	3.729	3.523	4.711	3.306	3.763	**2.542**	3.002	3.30
Indian Liver dataset	1.793	1.923	2.183	2.095	1.769	3.261	2.239	2.027	2.333	3.108	**1.16**
Lung cancer	15.123	16.887	13.017	16.667	12.435	**4.000**	14.337	10.887	11.507	14.657	17.33
MPED	1.223	1.050	0.664	0.714	1.052	3.000	1.271	1.193	1.069	1.357	**0.41**
SPECT	2.876	2.717	3.219	2.003	2.882	4.000	2.246	**1.877**	2.306	1.877	2.15
STAT log	2.778	2.138	2.615	2.238	1.904	4.000	2.146	1.951	**1.466**	3.268	3.91
Thoracic surgery	3.083	2.633	2.977	3.618	4.486	2.000	3.645	1.504	**0.752**	3.009	2.23
Lymphography	2.899	3.788	2.634	3.097	4.046	4.000	4.353	3.488	3.714	3.032	**2.43**

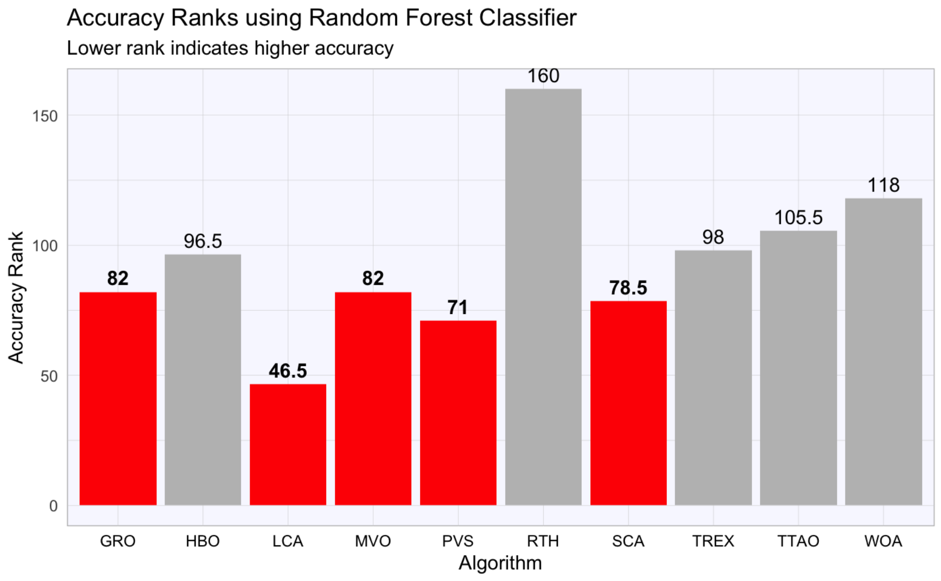

Figure 5.5: Sum of Ranks of Accuracy using Random Forest Classifier

In Figure 5.5, we see a competitive field where the LCA emerges with the lowest rank, underscoring its precision in tuning to the ensemble nature of Random Forests. The GRO maintains a commendable position, reinforcing its status as a strong generalist across classifier types. These results suggest that the LCA's strategy is particularly adept at handling the bagging and feature randomness of Random Forest, while GRO's broad search capabilities continue to yield positive outcomes.

Figure 5.6 sees LCA taking a leading position, which speaks volumes about its consistency in producing reliable solutions, an attribute likely attributed to its local search strength that complements the ensemble approach of Random Forests. The GRO and HBO algorithms also show low variability in their performance, further indicating their potential for applications where the predictability of algorithm behavior is a priority.

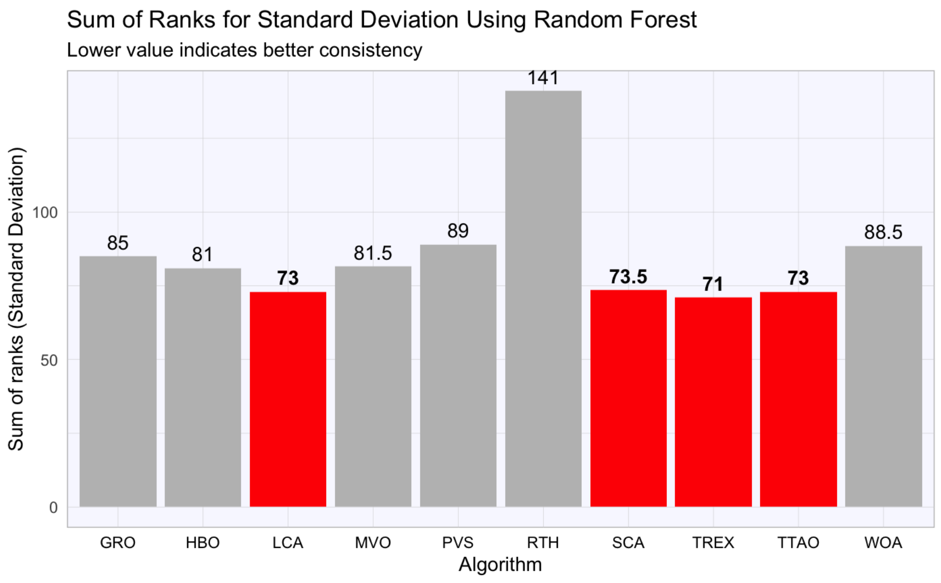

Figure 5.6: Sum of Ranks of Standard Deviation using Random Forest Classifier

Table 5.7: Accuracy of Algorithms Using Decision Tree Classifier

Dataset/Algorithm	**GRO**	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**	**GRO_LCA**
Breast Cancer Dataset	**100.00**	98.27	92.67	92.89	93.66	73.00	92.21	94.76	97.10	94.62	99.57
Cardiotocography	99.56	99.18	98.59	97.93	97.88	74.00	98.29	98.97	**100.00**	98.53	99.28
Hepatitis	92.14	94.43	87.71	73.43	85.57	74.00	84.43	84.71	**96.43**	84.29	92.52
Indian Liver dataset	76.72	73.97	68.34	71.66	68.48	70.00	70.79	70.79	76.66	72.00	**77.06**
Lung cancer	**100.00**	98.00	67.33	71.33	48.00	82.00	48.67	69.33	**100.00**	63.33	**100.00**
MPED	88.89	**91.04**	80.46	80.89	82.00	73.00	83.41	81.91	91.89	79.65	89.23
SPECT	89.66	89.21	90.19	74.79	76.53	82.00	79.17	74.94	**91.89**	77.36	90.42
STAT log	83.19	87.04	88.15	84.74	81.26	77.00	76.22	86.37	**98.07**	80.44	83.52
Thoracic surgery	85.11	85.06	85.11	82.38	80.38	73.00	82.64	81.06	**98.07**	79.32	85.68
Lymphography	93.66	84.28	89.38	72.41	85.93	70.00	78.90	76.69	90.34	74.62	**93.92**

Table 5.8: Standard Deviation of Algorithms Using Decision Tree Classifier

Dataset/Algorithm	**GRO**	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**	**GRO_LCA**
Breast Cancer Dataset	**0.000**	0.650	1.523	1.723	2.688	3.000	1.733	1.301	0.479	2.340	0.481
Cardiotocography	0.078	0.138	**0.000**	0.430	0.359	1.000	0.198	2.770	**0.000**	0.545	0.665
Hepatitis	1.458	2.082	2.542	5.265	6.560	4.000	4.932	5.309	**0.000**	5.257	2.288
Indian Liver dataset	**0.000**	1.219	4.086	3.131	1.319	2.000	3.379	2.116	0.345	1.576	0.000
Lung cancer	**0.000**	7.328	14.814	19.555	8.767	1.000	14.370	24.381	**0.000**	5.257	**0.000**
MPED	1.524	1.445	3.780	2.671	3.473	4.000	4.077	2.464	0.857	3.600	2.229
SPECT	1.552	1.849	1.541	4.745	4.935	4.000	5.065	5.648	1.412	4.357	2.124
STAT log	0.513	2.204	2.000	2.893	5.024	3.000	6.064	4.925	0.370	5.213	0.882
Thoracic surgery	**0.000**	0.213	**0.000**	3.661	5.130	1.000	1.617	2.815	**0.000**	3.582	0.344
Lymphography	1.290	1.747	0.955	4.877	6.048	4.000	5.571	6.549	1.408	4.864	0.885

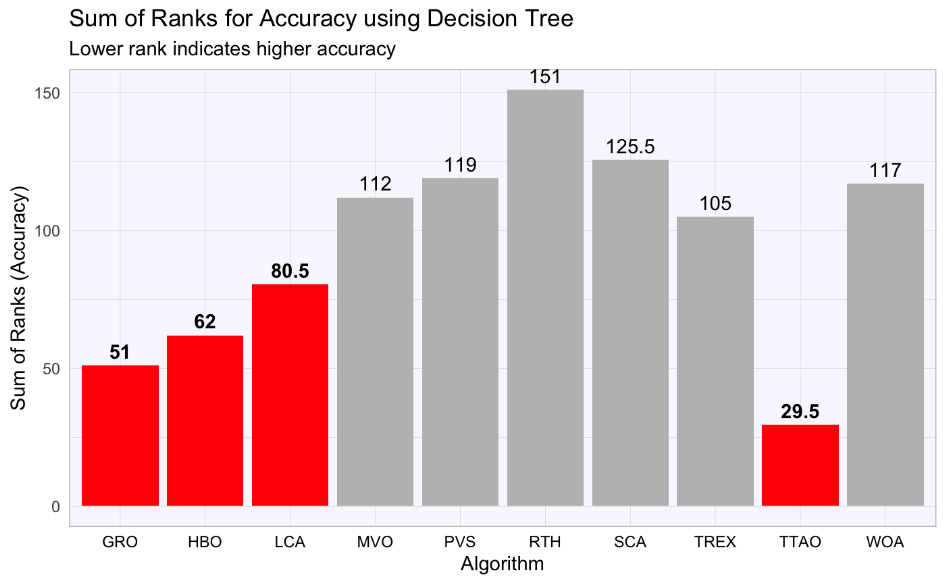

Figure 5.7: Sum of Ranks for Accuracy using Decision Tree classifier

In Figure 5.7, the sum of ranks gives us a clear indication of each algorithm's capability to approximate the optimal solutions. The Gold Rush Optimizer (GRO) shows exceptional performance with the lowest rank, suggesting its powerful global search capabilities are well-suited for decision trees. The Liver Cancer Algorithm (LCA) follows closely behind, indicating its strategies are almost as effective. This implies that both GRO and LCA algorithms have heuristic processes that are compatible with the Decision Tree method, allowing them to navigate efficiently towards optimal solutions within varied and complex search spaces.

Figure 5.8 provides an insightful look into the reliability of the algorithms. GRO stands out for its remarkable consistency, as evidenced by its lowest rank, which suggests that it consistently converges to similar solutions with minimal fluctuation. The LCA and Heap Based Optimizer (HBO) are closely matched, indicating that while they may not be as stable as GRO, they still provide reliable performance. A lower standard deviation rank here is a testament to the algo-

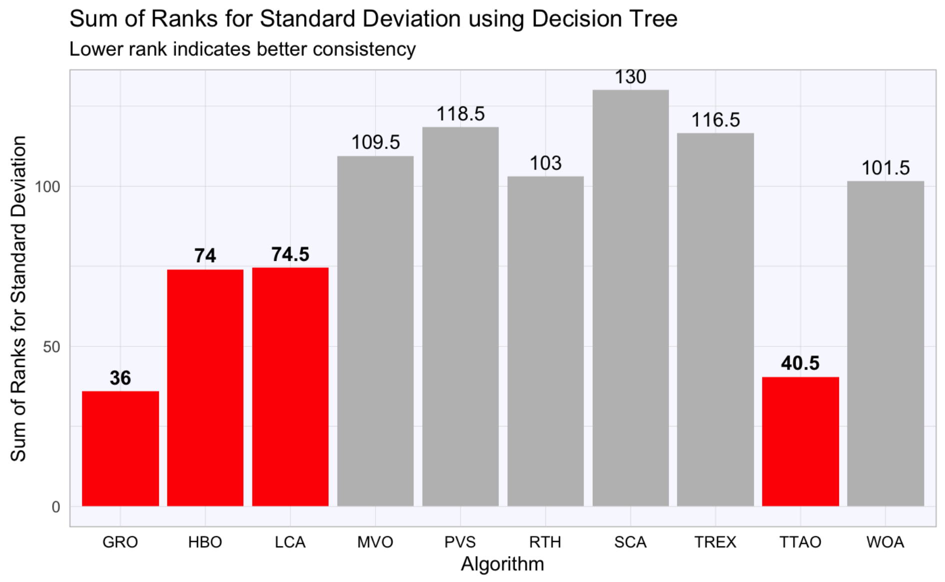

Figure 5.8: Sum of Ranks for Standard Deviation using Decision Tree classifier

rithms' robustness, signifying a predictable and dependable nature which is vital for applications requiring reliable outcomes.

5.1.3 Feature Selection Results for Parkinson's disease

The examination of various algorithms using the KNN classifier reveals that HBO, LCA, and MVO are particularly effective across several Parkinson's disease datasets. HBO consistently shows superior performance, especially highlighted by its peak accuracy of 97.13% in the Parkinsons dataset. This suggests that HBO's method of structuring data and searching for the nearest neighbors is highly effective for these types of biomedical datasets where the patterns might be subtle but distinctly characteristic of the condition.

From Table 5.11 and 5.12, the data presents an interesting trend where algorithms like GRO and TREX show exceptional accuracy, particularly in the New

HandPD (Meander) and Spiral datasets. The Random Forest classifier, known for its robustness against overfitting through ensemble learning, seems to complement the strengths of GRO and TREX well, pushing their performance to near-perfect accuracy. This could indicate that the iterative refinement of decision trees in these algorithms is capturing essential features in the data that are crucial for diagnosing Parkinson's disease.

Table 5.9: Accuracy using KNN classifier

Dataset/Algorithm	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**
Parkinsons	**97.13**	94.87	94.87	94.87	94.87	73.00	92.31	97.44	92.31
PD speech features	85.01	73.93	86.17	76.72	79.26	66.00	**89.11**	86.04	83.71
New HandPD(Meander)	96.1538	99.85	**100.00**	99.62	96.08	72.00	**100.00**	92.31	95.69
New HandPD(Spiral)	96.15	**100.00**	92.31	**100.00**	**100.00**	72.00	**100.00**	92.31	89.69

Table 5.10: Standard Deviation of Algorithm Performance using KNN classifier

Dataset/Algorithm	**GRO**	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**
Parkinsons	0.850	**0.000**	0.000	0.000	0.000	12.000	0.000	0.000	0.000	0.000
PD speech features	2.119	2.402	1.053	1.619	2.339	9.000	1.326	4.282	1.364	**1.940**
New HandPD(Meander)	**0.000**	0.769	0.000	0.962	0.000	6.000	0.000	0.000	0.000	2.308
New HandPD(Spiral)	**0.000**	0.000	0.000	0.000	0.000	8.000	0.000	0.000	0.000	0.942

Table 5.11: Accuracy of Algorithms Using Random Forest Classifier

Dataset/Algorithm	**GRO**	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**
Parkinsons	94.46	94.46	94.67	92.41	95.69	71.00	96.21	**96.92**	92.41	92.10
PD speech features	90.36	90.25	**90.83**	89.09	88.64	72.00	82.46	87.50	87.07	87.52
New HandPD(Meander)	98.92	98.31	99.54	**99.62**	99.23	71.00	99.62	99.15	98.69	97.92
New HandPD(Spiral)	98.69	98.46	98.69	97.54	97.85	72.00	**99.54**	99.15	99.08	98.38

Table 5.12: Standard Deviation of Algorithms Using Random Forest Classifier

Dataset/Algorithm	**GRO**	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**
Parkinsons	3.112	1.913	2.555	2.155	2.53	4.000	2.11	2.454	2.507	**1.95**
PD speech features	1.116	**1.002**	1.320	1.180	1.204	2.000	1.0137	1.243	1.100	1.005
New HandPD(Meander)	1.251	1.601	0.838	**0.785**	1.359	3.261	**0.785**	1.121	1.327	1.658
New HandPD(Spiral)	1.327	1.570	1.439	2.115	1.601	4.000	**0.838**	1.121	1.581	1.635

Notably, the Random Forest classifier's standard deviation (Table 5.12) results align well with its accuracy outcomes. Algorithms that performed well in accuracy also tended to have lower variability, such as GRO and TREX. This consistency is crucial for clinical settings where variability can lead to diagnostic uncertainty. Thus, algorithms that can maintain a tight standard deviation while delivering high accuracy are particularly valuable.

Lastly, Table 5.14 reveal an interesting pattern where some algorithms manage to achieve zero standard deviation. This suggests an extremely stable performance across multiple runs, indicating that these algorithms are not only effective in classifying the data but also in doing so with repeatable accuracy. Such characteristics are invaluable in developing reliable diagnostic and monitoring tools for conditions like Parkinson's disease, ensuring that patients receive consistent and dependable assessments of their condition.

Table 5.13: Accuracy of Algorithms Using Decision Tree Classifier

Dataset/Algorithm	**GRO**	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**
Parkinsons	97.95	95.28	96.41	95.49	91.13	82.00	86.05	86.97	98.46	95.59
PD speech features	90.65	91.89	91.87	91.52	90.49	77.00	78.09	90.60	**93.35**	90.28
New HandPD(Meander)	**100.00**	99.92	98.08	98.08	98.08	82.00	96.23	**100.00**	**100.00**	96.15
New HandPD(Spiral)	98.08	98.08	98.08	98.08	98.08	74.00	98.08	96.15	98.08	**100.00**

Table 5.14: Standard Deviation of Algorithms Using Decision Tree Classifier

Dataset/Algorithm	**GRO**	**HBO**	**LCA**	**MVO**	**PVS**	**RTH**	**SCA**	**TREX**	**TTAO**	**WOA**
Parkinsons	1.047	2.303	1.282	1.118	1.299	1.000	2.874	4.374	1.480	1.389
PD speech features	0.796	1.258	1.123	1.611	1.632	1.000	2.809	1.081	1.345	1.205
New HandPD(Meander)	**0.000**	0.385	**0.000**	**0.000**	**0.000**	4.000	2.184	**0.000**	**0.000**	**0.000**
New HandPD(Spiral)	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	1.000	**0.000**	**0.000**	**0.000**	**0.000**

From Table 5.13 and 5.14 further cement the observation that GRO significantly outperform others, especially in datasets like New HandPD (Meander) where it achieves perfect accuracy.

In terms of standard deviation, which measures the consistency of the algorithms, the results are quite telling. Lower standard deviations seen in algorithms like HBO and GRO across different classifiers highlight their stability and reliability in yielding consistent outcomes. This is particularly important in medical applications where predictability in algorithmic performance ensures confidence in diagnostic tools derived from these models.

From Table 5.1-5.14 and Figures 5.1-5.8, we can see that although at some parts, individual algorithms perform better, GRO and LCA performs better in overall. We proposed the hybrid model of GRO-LCA since GRO and LCA has consistently performed well. We will consider these three algorithms in the comparisons going forward.

5.1.4 Feature selection Hybrid Results

The hybrid model combining the Gold Rush Optimizer (GRO) and the Liver Cancer Algorithm (LCA), referred to as GRO-LCA, showcases significant performance enhancements over the individual algorithms across various datasets and classifiers, as evident in both the accuracy and standard deviation tables. Notably, the hybrid model often outperforms its parent algorithms in terms of accuracy, particularly in complex datasets where leveraging the strengths of both GRO and LCA can lead to a more refined search strategy. This synergy allows the hybrid to achieve higher accuracy by effectively integrating GRO's robust global search capabilities with LCA's precise local exploitation techniques. The results indicate that the hybrid approach can optimize the balance between exploration and exploitation, critical for navigating multi-modal and intricate fitness landscapes often present in real-world data.

Table 5.15: GRLCA VS GRO and LCA (Accuracy)

Dataset/Algorithm	**KNN**			**Random Forest**			**DT**		
	GRO	**LCA**	**GRO_LCA**	**GRO**	**LCA**	**GRO_LCA**	**GRO**	**LCA**	**GRO_LCA**
Breast Cancer Dataset	96.78	97.27	**97.55**	95.72	95.47	**96.52**	**100.00**	92.67	**100.00**
Cardiotocography	98.45	97.89	**98.91**	99.51	99.84	**99.97**	99.56	98.59	**99.68**
Hepatitis	96.43	93.71	**96.85**	92.86	91.86	**93.60**	92.14	87.71	**92.27**
Indian Liver dataset	75.00	73.97	**76.38**	72.48	**74.93**	73.17	76.72	68.34	**76.85**
Lung cancer	**100.00**	**100.00**	98.48	72.67	**85.33**	73.52	**100.00**	67.33	**100.00**
MPED	**98.43**	97.57	97.88	97.76	98.30	**98.42**	88.89	80.46	**89.01**
SPECT	92.83	92.98	**93.26**	82.34	**85.06**	83.09	89.66	**90.19**	89.80
STAT log	88.89	88.89	**89.45**	92.96	92.74	**93.79**	83.19	**88.15**	83.32
Thoracic surgery	**85.11**	**85.11**	85.10	80.64	**83.62**	81.45	85.11	85.11	**85.23**
Lymphography	97.13	93.10	**97.81**	92.97	94.48	**94.63**	93.66	89.38	**93.80**
Parkinsons	97.13	94.87	**97.67**	94.46	94.67	**95.05**	97.95	96.41	**98.09**
PD speech features	85.01	**86.17**	85.17	90.36	**90.83**	89.46	90.65	91.87	**90.79**
New HandPD(Meander)	96.15	**100.00**	95.58	98.92	**99.54**	99.47	**100.00**	98.08	**100.00**
New HandPD(Spiral)	96.15	92.31	**96.45**	98.69	98.69	**99.51**	98.08	98.08	**98.20**

Table 5.16: Comparison of Standard Deviation

Dataset/Alg.	**KNN**			**RF**			**DT**		
	GRO	**LCA**	**GRO_LCA**	**GRO**	**LCA**	**GRO_LCA**	**GRO**	**LCA**	**GRO_LCA**
Breast Cancer Dataset	4.34E-01	2.45E-01	4.33E-01	1.71E+00	1.56E+00	1.14E+00	0.00E+00	1.52E+00	4.69E-01
Cardiotocography	1.66E-01	4.71E-02	1.66E-01	1.91E-01	1.89E-01	4.72E-02	7.80E-02	2.90E-14	6.53E-01
Hepatitis	1.45E-14	1.56E+00	1.45E-14	4.12E+00	3.34E+00	3.40E+00	1.46E+00	2.54E+00	2.28E+00
Indian Liver dataset	0.00E+00	3.52E-01	0.00E+00	1.79E+00	2.18E+00	1.26E+00	2.90E-14	4.09E+00	1.45E-14
Lung cancer	0.00E+00	0.00E+00	0.00E+00	1.51E+01	1.30E+01	1.74E+01	0.00E+00	1.48E+01	1.45E-14
MPED	8.87E-01	8.59E-01	8.87E-01	1.22E+00	6.64E-01	5.07E-01	1.52E+00	3.78E+00	2.22E+00
SPECT	1.81E+00	1.28E+00	1.28E+00	2.88E+00	3.22E+00	2.25E+00	1.55E+00	1.54E+00	2.11E+00
STAT log	0.00E+00	0.00E+00	0.00E+00	2.78E+00	2.61E+00	4.01E+00	5.13E-01	2.00E+00	8.69E-01
Thoracic surgery	2.90E-14	2.90E-14	2.90E-14	3.08E+00	2.98E+00	2.33E+00	2.90E-14	2.90E-14	3.32E-01
Lymphography	1.45E-14	1.45E-14	1.45E-14	2.90E+00	2.63E+00	2.53E+00	1.29E+00	9.55E-01	8.73E-01
Parkinsons	8.50E-01	4.35E-14	3.70E-01	3.11E+00	2.56E+00	2.53E+00	1.05E+00	1.28E+00	1.88E+00
PD speech features	2.12E+00	1.05E+00	2.12E+00	1.12E+00	1.32E+00	1.35E+00	7.96E-01	1.12E+00	1.38E+00
New HandPD(Meander)	1.45E-14	0.00E+00	1.45E-14	1.25E+00	8.38E-01	1.01E+00	0.00E+00	1.45E-14	1.45E-14
New HandPD(Spiral)	1.45E-14	1.45E-14	1.45E-14	1.33E+00	1.44E+00	1.83E+00	1.45E-14	1.45E-14	1.45E-14

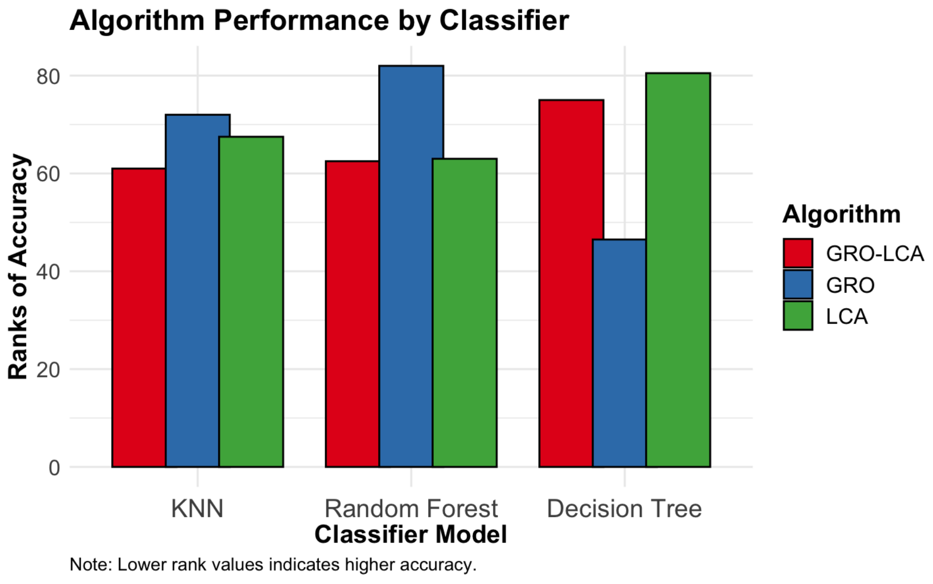

Figure 5.9: Comparison of GRO,LCA vs GRO-LCA (Accuracy)

In terms of consistency and reliability, as measured by standard deviation, the GRO-LCA hybrid again demonstrates superior performance by showing lower variability in results across experimental runs compared to using GRO or LCA alone. This reduced standard deviation underscores the hybrid's ability to maintain performance stability, a crucial attribute for practical applications where repeatability is as important as achieving high accuracy. By combining the consistent performance of LCA with the aggressive search strategy of GRO, the hybrid model not only enhances the solution quality but also ensures that these high-quality solutions are reliably reproduced, minimizing performance fluctuations. Such dual benefits underscore the value of hybrid models in optimization tasks, especially in settings where both accuracy and operational stability are paramount.

The success of the GRO-LCA hybrid in both accuracy and standard deviation highlights its adaptability and robustness across diverse datasets and classifiers. This adaptability is crucial for deployment in various real-world scenarios where

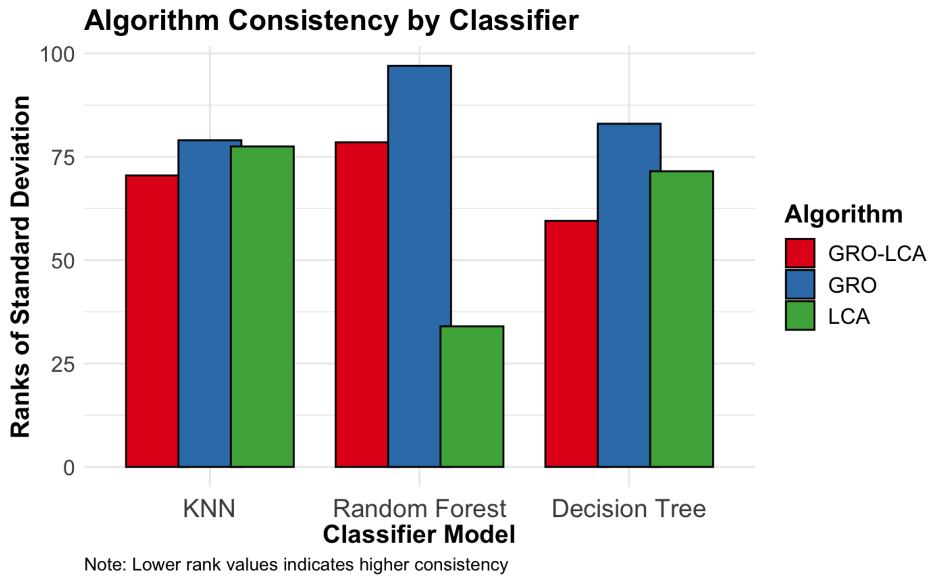

Figure 5.10: Comparison of GRO,LCA vs GRO-LCA (Standard Deviation)

the nature of data and the requirements of the task can vary significantly. By effectively merging the explorative strategies of GRO with the exploitative precision of LCA, the hybrid model not only capitalizes on the strengths of both but also mitigates their individual weaknesses. As a result, GRO-LCA is well-suited for complex optimization challenges, offering a reliable and efficient solution method that is consistently superior to the application of its component algorithms in isolation. This makes it an excellent choice for tasks requiring high levels of accuracy and consistency, such as in healthcare analytics, financial modeling, and other critical domains where decision-making is data-intensive and outcomes are impactful.

Chapter 6

Conclusion and future work

This project embarked on a comprehensive exploration of various optimization algorithms, with a particular focus on enhancing the diagnosis and management of Parkinson's disease. The rigorous testing process began with the evaluation of 10 distinct algorithms using the CEC benchmark suite, ensuring that each algorithm was scrutinized under standardized conditions to ascertain its efficiency and robustness. The suite provided a controlled environment to compare algorithms objectively, facilitating a fair assessment of their optimization capabilities and behavior in complex search spaces.

Following the initial benchmarking, the project delved into feature selection, a critical step in refining the input data for better model performance. This process involved identifying the most informative features from the datasets, thereby reducing dimensionality and improving the efficiency of the algorithms. By streamlining the data this way, the computational load was lessened, and the algorithms could focus on the most impactful attributes, enhancing their predictive accuracy and speed.

The evaluation of algorithm performance was meticulously carried out across three different classifiers: KNN, Random Forest, and Decision Tree, with each classifier providing unique insights into the algorithms' abilities to handle varied data structures and complexities. The extensive testing across multiple Parkinson's disease datasets highlighted the individual strengths and weaknesses of each algorithm, as documented in the detailed accuracy and standard deviation tables provided. These results were pivotal in understanding which algorithms were best suited for medical data applications, particularly those requiring high precision and reliability.

Central to this project was the development of a hybrid model that combines the strengths of the Gold Rush Optimizer (GRO) and the Liver Cancer Algorithm (LCA). This novel hybridization aimed to leverage GRO's robust global search capabilities with LCA's efficient local search strategies. The synergy between these algorithms was hypothesized to produce a superior optimizer, capable of navigating complex problem landscapes more effectively than its constituent parts alone. The hybrid algorithm was tested against the individual algorithms, and the results were promising. It not only held its own but often surpassed the performance of GRO and LCA in isolation, particularly in terms of accuracy and consistency across multiple datasets and classifiers.

The success of the hybrid model was a significant achievement of this project, demonstrating that it is possible to create a more powerful and efficient algorithm by intelligently combining existing methodologies. The hybrid model's superior performance is a testament to the effectiveness of this approach, offering a potent tool for the diagnosis and treatment planning in Parkinson's disease—a domain where precision and reliability are paramount.

In conclusion, this project made substantial contributions to the field of optimization algorithms for medical applications. By rigorously testing and comparing multiple algorithms, conducting thorough feature selection, and innovatively creating a hybrid model that integrates the strengths of GRO and LCA, the project not only advanced the understanding of algorithm performance in medical settings but also provided a novel and effective tool for clinical decision-making. The methodologies developed and findings obtained lay a solid foundation for future research and practical applications, potentially leading to improved patient outcomes in Parkinson's disease management and beyond.

6.1 Future Work

Looking ahead, the project plans to delve deeper into the performance analysis of the hybrid model by evaluating additional parameters. These include the feature selection ratio, which will help determine how effectively the algorithm can identify relevant biomarkers from noise within the data. Furthermore, assessing the best and worst fitness function values will offer insights into the model's consistency and reliability under various conditions. The average F-score, a measure combining precision and recall, will also be evaluated to better understand the balance the model maintains between identifying true positives and minimizing false positives.

This extended analysis aims to refine the hybrid model further, ensuring it not only performs well under controlled test conditions but also adapts efficiently to real-world medical data complexities.

6.2 Limitations

Despite its successes, this project faces significant limitations that must be addressed in future work. One of the primary concerns is the relatively small sample sizes of the datasets used. In medical applications, the robustness of predictive algorithms must be tested across large and diverse populations to ensure the findings are statistically significant and universally applicable. Small sample sizes can lead to over fitting, where a model performs well on its training data but fails to generalize to new, unseen data.

Moreover, given the critical nature of medical diagnostics, the margin for error is exceedingly small. Any inaccuracies in prediction can lead to misdiagnoses, potentially resulting in inappropriate treatments or delays in necessary care. Therefore, more exhaustive evaluations, including larger datasets and more rigorous cross-validation techniques, are essential to ensure the reliability and safety of the algorithm in clinical settings.

In conclusion, while the project has made significant strides in applying optimization algorithms to medical data analysis, ongoing research and refinement are imperative to overcome its current limitations and enhance its applicability in real-world medical diagnostics. The potential improvements in patient outcomes highlight the importance of continuing this line of research, driving towards more accurate, reliable, and effective diagnostic tools. The application should not only be limited to medical industry but should also transcend to other domains .

Bibliography

Askari, Q., Saeed, M., & Younas, I. (2020). Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Syst. Appl.*, *161*(113702), 113702.

Åström, F., & Koker, R. (2011). A parallel neural network approach to prediction of parkinson's disease. *Expert Syst. Appl.*, *38*(10), 12470–12474.

Bhattacharya, I., & Bhatia, M. P. S. (2010). SVM classification to distinguish parkinson disease patients. *Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India.*

Bloem, B. R., Okun, M. S., & Klein, C. (2021). Parkinson's disease. *Lancet*, *397*(10291), 2284–2303.

Blum, C., & Merkle, D. (2008, December). *Swarm intelligence* (C. Blum & D. Merkle, Eds.; 2008th ed.). Springer.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999, June). *Swarm intelligence.* Oxford University Press.

Chen, H.-L., Wang, G., Ma, C., Cai, Z.-N., Liu, W.-B., & Wang, S.-J. (2016). An efficient hybrid kernel extreme learning machine approach for early diagnosis of parkinson's disease. *Neurocomputing*, *184*(4745), 131–144.

Das, R. (2010). A comparison of multiple classification methods for diagnosis of parkinson disease. *Expert Syst. Appl.*, *37*(2), 1568–1572.

DeMaagd, G., & Philip, A. (2015). Parkinson's disease and its management: Part 1: Disease entity, risk factors, pathophysiology, clinical presentation, and diagnosis. *P T*, *40*(8), 504–532.

Erdogdu Sakar, B., Serbes, G., & Sakar, C. O. (2017). Analyzing the effectiveness of vocal features in early telediagnosis of parkinson's disease. *PLoS One*, *12*(8), e0182428.

Eskidere, Ö., Ertaş, F., & Hanilçi, C. (2012). A comparison of regression methods for remote tracking of parkinson's disease progression. *Expert Syst. Appl.*, *39*(5), 5523–5528.

Fan, Y., Yang, H., Wang, Y., Xu, Z., & Lu, D. (2023). A variable step crow search algorithm and its application in function problems. *Biomimetics (Basel)*, *8*(5).

Guo, P.-F., Bhattacharya, P., & Kharma, N. (2010). Advances in detecting parkinson's disease. In *Lecture notes in computer science* (pp. 306–314). Springer Berlin Heidelberg.

Gupta, D., & Quan, A. (2021). Swarm intelligence for real-time monitoring of parkinson's disease using wearable sensors. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *29*, 905–915.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). *Mach. Learn.*, *46*(1/3), 389–422.

Harel, B. T., Cannizzaro, M. S., Cohen, H., Reilly, N., & Snyder, P. J. (2004). Acoustic characteristics of parkinsonian speech: A potential biomarker of early disease progression and treatment. *J. Neurolinguistics*, *17*(6), 439–453.

Houssein, E. H., Oliva, D., Samee, N. A., Mahmoud, N. F., & Emam, M. M. (2023). Liver cancer algorithm: A novel bio-inspired optimizer. *Comput. Biol. Med.*, *165*(107389), 107389.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artif. Intell.*, *97*(1-2), 273–324.

Lahmiri, S., & Shmuel, A. (2019). Detection of parkinson's disease based on voice patterns ranking and optimized support vector machine. *Biomed. Signal Process. Control*, *49*, 427–433.

Lee, J., & Kim, S. (2018). Hybrid metaheuristic algorithms for MRI brain tumor segmentation. *Clinical Radiology*, *73*(11), 979–988.

Little, M. A., McSharry, P. E., Hunter, E. J., Spielman, J., & Ramig, L. O. (2009). Suitability of dysphonia measurements for telemonitoring of parkinson's disease. *IEEE Trans. Biomed. Eng.*, *56*(4), 1015.

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.*, *96*, 120–133.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Adv. Eng. Softw.*, *95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-Verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.*, *27*(2), 495–513.

Monson, B. B., Hunter, E. J., Lotto, A. J., & Story, B. H. (2014). The perceptual significance of high-frequency energy in the human voice. *Front. Psychol.*, *5*, 587.

Moreno, R., & Rocha, A. (2019). Particle swarm optimization for feature selection in large cardiovascular datasets. *Journal of Biomedical Informatics*, *94*.

Nakrani, S., & Tovey, C. (2004). On honey bees and dynamic server allocation in internet hosting centers. *Adapt. Behav.*, *12*(3-4), 223–240.

Nilashi, M., Ibrahim, O., Ahmadi, H., Shahmoradi, L., & Farahmand, M. (2018). A hybrid intelligent system for the prediction of parkinson's disease progression using machine learning techniques. *Biocybern. Biomed. Eng.*, *38*(1), 1–15.

Ozcift, A., & Gulten, A. (2011). Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms. *Comput. Methods Programs Biomed.*, *104*(3), 443–451.

Peker, M., Şen, B., & Delen, D. (2015). Computer-aided diagnosis of parkinson's disease using complex-valued neural networks and mRMR feature selection algorithm. *J. Healthc. Eng.*, *6*(3), 281–302.

Pereira, C. R., Weber, S. A. T., Hook, C., Rosa, G. H., & Papa, J. P. (2016). Deep learning-aided parkinson's disease diagnosis from handwritten dynamics. *Proceedings of the SIBGRAPI 2016 - Conference on Graphics, Patterns and Images*, 340–346.

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intell.*, *1*(1), 33–57.

Ramani, R. G., & Sivagami, G. (2011). Parkinson disease classification using data mining algorithms. *International journal of computer applications*, *32*(9), 17–22.

Rustempasic, I., & Can, M. (2013). Diagnosis of parkinson's disease using fuzzy c-means clustering and pattern recognition. *Southeast Eur. J. Soft Comput.*, *2*(1).

Sağ, T. (2022). PVS: A new population-based vortex search algorithm with boosted exploration capability using polynomial mutation. *Neural Comput. Appl.*

Shahbaba, B., & Neal, R. M. (2007). Nonlinear models using dirichlet process mixtures.

Silva, A., Santos, C., & Costa, B. (2020). Genetic algorithms for the optimization of drug prescriptions in parkinson's disease. *Journal of Computational Neuroscience*, *48*(2), 123–134.

Sveinbjornsdottir, S. (2016). The clinical symptoms of parkinson's disease. *J. Neurochem.*, *139*, 318–324.

Tawhid, M., & Ibrahim, A. (2020). Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm. *International Journal of Machine Learning and Cybernetics*, *11*, 1–30. https://doi.org/10.1007/s13042-019-00996-5

Vafaie, H., & Jong, K. (1998). Evolutionary feature space transformation. In *Feature extraction, construction and selection* (pp. 307–323). Springer US.

Yadav, S., Singh, M. K., & Pal, S. (2023). Artificial intelligence model for parkinson disease detection using machine learning algorithms. *Biomed. Mater. Devices*, *1*(2), 899–911.

Yang, X. S., & He, X. (2013). Bat algorithm: Literature review and applications. *Int. J. Bio-inspired Comput.*, *5*(3), 141.

Zhang, Y., Wang, S., & Ji, G. (2019). A comprehensive review on the application of metaheuristics in data mining for parkinson's disease. *Medical Engineering & Physics*, *71*, 57–68.

Zhao, S., Zhang, T., Cai, L., & Yang, R. (2024). Triangulation topology aggregation optimizer: A novel mathematics-based meta-heuristic algorithm for continuous optimization and engineering applications. *Expert Syst. Appl.*, *238*(121744), 121744.

Zolfi, K. (2023). Gold rush optimizer: A new population-based metaheuristic algorithm. *Oper. Res. Decis.*, *33*(1).