Shiza Butt

u21451631

Assignment 3: Structure-Based Genetic Programming Report

Introduction

This report details the implementation of Structure-Based Genetic Programming (SBGP) with Iterative Subtree-Based Adaptation (ISBA) for classifying hepatitis patients. Using the hepatitis dataset from the EpistasisLab repository, the compares the performance of Regular Genetic Programming (GP) with Structure-Based GP utilizing the ISBA approach, focusing on accuracy, stability, and applicability in medical diagnostics. The report covers dataset preprocessing, algorithm implementation, parameter settings, reflections, and result analysis.

Dataset Description and Preprocessing

The dataset used was the Hepatitis dataset, available in tab-separated format (hepatitis.tsv). The primary target variable represents patient survival outcomes, encoded as 1 for hepatitis negative and 2 for hepatitis positive. The dataset exhibited class imbalance, with significantly more instances for the (2) class, so I applied manual oversampling to the minority (1) class to ensure equal representation. After removing duplicates and re-mapping the target variable to binary (2→1, 1→0), I split the dataset into an 80/20 training-test split using random shuffling and indexing.

Structure-Based GP Algorithm: ISBA Implementation

Individual Representation

Each individual is represented as a syntax tree. The internal nodes are functions (mathematical operators), and the leaves are terminals (feature references or constants). This representation allows for flexible and interpretable model structures while maintaining a recursive evaluation pattern.

Initialisation Method

To initialise the population, I used the ramped half-and-half method, which generates trees of varying depths using two strategies. The full method grows trees to the maximum depth with only function nodes at internal levels, while the grow method randomly chooses functions or terminals, allowing uneven, irregular shapes. This hybrid approach ensures structural diversity from the outset and reduces the risk of premature convergence or overfitting. It gives the algorithm a balanced starting point, enabling both deep and shallow trees to be explored in early generations.

Function Set

I selected basic arithmetic operations to capture non-linear interactions without overcomplicating the search space, specifically add, sub, mul, and div. These were chosen for their generality and expressiveness and to allow the GP to form both linear and non-linear expressions. Division was protected against divide-by-zero issues to keep numerical stability throughout evolution.

Terminal Set

The terminal set consists of feature variables [X0, X1, ..., X18] and ephemeral random constants in the range [-5, 5] to introduce numeric flexibility. This allowed the GP trees to represent both feature-based and scalar logic which allowed for fine-grained thresholding and tenable numeric behaviour within evolved expressions.

Selection Method

I used tournament selection with size 3. This approach provides a balance between randomness and pressure—it increases the likelihood of selecting fitter individuals while avoiding domination by a single candidate. Its simplicity and tunability made it a strong fit for both the regular and structure-based GP settings.

Genetic Operators

To evolve the population, I used several operators. Crossover swaps subtrees between two parent individuals and was chosen to allow mixing of useful substructures and preserve partially successful building blocks. Mutation replaces a random subtree with a newly generated one, introducing novelty and helping escape local optima. Pruning enforced a max depth of 4, keeping models concise and preventing overfitting via overly complex trees. These operators collectively allowed me to explore new parts of the search space, refine good ideas, and prevent structural bloat which is important for interpretability and efficiency.

Fitness Function and Evolution Metrics

I defined the fitness function as Fitness = 1 - F1 Score. This inversion (since lower fitness is better in GP) made the objective compatible with standard minimisation routines. I selected the F1-score because it balances precision and recall, which is especially important in imbalanced classification problems like this one. To evaluate evolutionary performance, I tracked several metrics over generations. Best Fitness per Generation showed whether evolution was progressing and how quickly, while Average Fitness per Generation reflected population-level learning. The Stagnation Counter triggered early stopping if no improvement occurred for 10 generations. These metrics helped me monitor convergence, identify stagnation, and ensure that structural evolution was actually yielding generalisable models.

Termination Criterion

I used early stopping with a patience of 10 generations without fitness improvement. This saved time and prevented unnecessary computation after convergence, while also mitigating overfitting to the training data.

ISBA Integration and Structural Influence

The Structure-based GP, more specifically - the ISBA method extended standard GP by incorporating structural guidance and diversity control, using two phases. In Global Exploration (m = 5), I performed 5 independent GP runs, each seeded differently, to gather structurally diverse high-performing individuals from across the search space. In Local Refinement (n = 3), for each top individual, I ran 3 refinement cycles where a subtree of depth d = 2 was locked (preserved), and the rest of the individual was evolved, helping to improve solutions while retaining structurally useful sub-patterns. Additionally, I implemented cosine similarity (GSim) filtering during crossover: if two parent trees produced highly similar outputs (similarity > 0.7), crossover was skipped to maintain structural novelty and prevent population stagnation. This added structural diversity and enforced novelty—key components of any structure-based GP system. While elitism preserved high-performing individuals unchanged, ISBA went further by reusing, adapting, and structurally refining individuals. ISBA is structurally driven because it explicitly monitored and shaped the internal form of solutions, using subtree locking and similarity metrics as tools.

Parameter Settings and Rationale

To determine the appropriate configuration for both Regular and Structure-Based GP, I conducted several pilot runs varying key parameters (e.g., population size, mutation rate, and tree depth) while monitoring F1-score convergence, overfitting behaviour, and runtime. The final parameter settings represent a trade-off between evolutionary effectiveness, computational efficiency, and structural control

Parameter Settings and Rationale

Parameter	Value	Rationale & Selection Justification
Population Size	200	Empirically, smaller populations (<100) tended to converge too quickly with reduced diversity. A size of 200 was chosen to maintain a broad search space, supporting generalisation and effective crossover. It also remained computationally feasible for multiple seed runs.
Max Tree Depth	4	Pilot runs with depths >5 led to bloated trees with little generalisation benefit. Depth 4 strikes a balance by capturing essential feature interactions while constraining model complexity and interpretability. This also aligns with prior GP studies on classification tasks.
Max Generations	50	Early experiments showed that most performance gains occurred within 40–50 generations. 50 ensures enough time for convergence without excessive computation, especially under early stopping. This range is standard in medium-scale GP applications.
Crossover Probability	0.7	High enough to promote recombination of useful substructures,key for structural evolution,but not so dominant as to overshadow mutation’s role. Value was validated by observing population stagnation when crossover was reduced.
Mutation Probability	0.3	Complementary to crossover: too little mutation led to premature convergence, while >0.4 disrupted promising individuals. Mutation at 0.3 was found to maintain novelty while preserving selective gains.
Tournament Size	3	Larger tournaments (>5) increased elitism and reduced diversity; smaller ones (<2) slowed convergence. A size of 3 provided a sweet spot, maintaining pressure while allowing exploratory paths to survive. Based on common best practices in symbolic GP.

Early Stopping	10	Set based on empirical observation: if no improvement in validation performance occurs over 10 generations, the population has often stagnated. Prevents unnecessary computation in already-converged runs.
ISBA Global Runs (m)	5	Selected after testing m ∈ {3, 5, 7}. Five global runs ensured sufficient exploration across diverse starting points without excessive runtime. This parameter determines how broadly structural patterns are searched across the fitness landscape.
ISBA Local Runs (n)	3	During ISBA, local fine-tuning of individuals showed diminishing returns beyond 3 inner runs. More than 3 led to marginal fitness gains while significantly increasing computational overhead.
Subtree Lock Depth (d)	2	Depth-2 subtrees were identified as the most stable functional patterns during earlier evolutionary snapshots. Locking them reduced destructive edits in core logic while allowing peripheral evolution to continue freely.

In-Depth Reflections and ISBA Strategy

Initial Approach to Structure-Based GP

When I first explored structure-based genetic programming (SBGP), I mistakenly equated “structure-based” with “simplicity-focused.” My earliest implementations enforced structure through explicit penalties and constraints aimed at reducing complexity. These included a tree size penalty added to the fitness function to discourage large or deep trees, a node similarity penalty that penalised individuals with repeating or redundant subtrees to encourage diversity, a structural complexity multiplier applied as a weight to the fitness score to bias selection toward simpler models, and aggressive post-operation pruning enforced after every mutation and crossover, often discarding potentially useful structural patterns. I expected these constraints to yield more generalisable and interpretable models, especially since traditional GP tends to evolve bloated, overly complex trees. But these attempts consistently underperformed, especially when compared to the flexibility of standard GP.

Challenges and Realisations

The core reason became clear through iterative testing: the hepatitis dataset contains complex, multi-variable, non-linear relationships. Simplistic models—no matter how interpretable—were incapable of learning meaningful patterns in such a space. In fact, my structure-focused constraints were actively suppressing useful expressivity, leading to models that underfit and failed to distinguish between classes effectively.

Transition to ISBA and Key Realisations

Implementing ISBA (Iterative Subtree-Based Adaptation) was a turning point. It helped me move beyond naive simplification and embrace structure in a more intelligent, guided way. Key Innovations in ISBA include a Two-Phase Evolution Strategy: Global exploration (m=5) allowed five independent GP runs to explore diverse regions of the search space, raising the chance of finding fundamentally different high-performing structures, while Local refinement (n=3) reused the best global individual but selectively locked subtrees of depth d=2, enabling focused structural enhancement while preserving previously evolved building blocks. Subtree Locking meant that instead of pruning complexity blindly, I preserved specific meaningful subtrees and allowed only the surrounding context to evolve, avoiding "restarting" the structure in each generation, unlike earlier methods that would disrupt useful segments. The Cosine Similarity Filter (GSim) incorporated a behavioural similarity metric during crossover: if two individuals had high output similarity (>0.7 cosine similarity), crossover was skipped, maintaining diversity in behaviour, not just syntax—a smarter way of guiding evolution. These changes aligned with the spirit of structure-based evolution: not to simplify for its own sake, but to control and refine structural expressiveness in a principled way.

Results

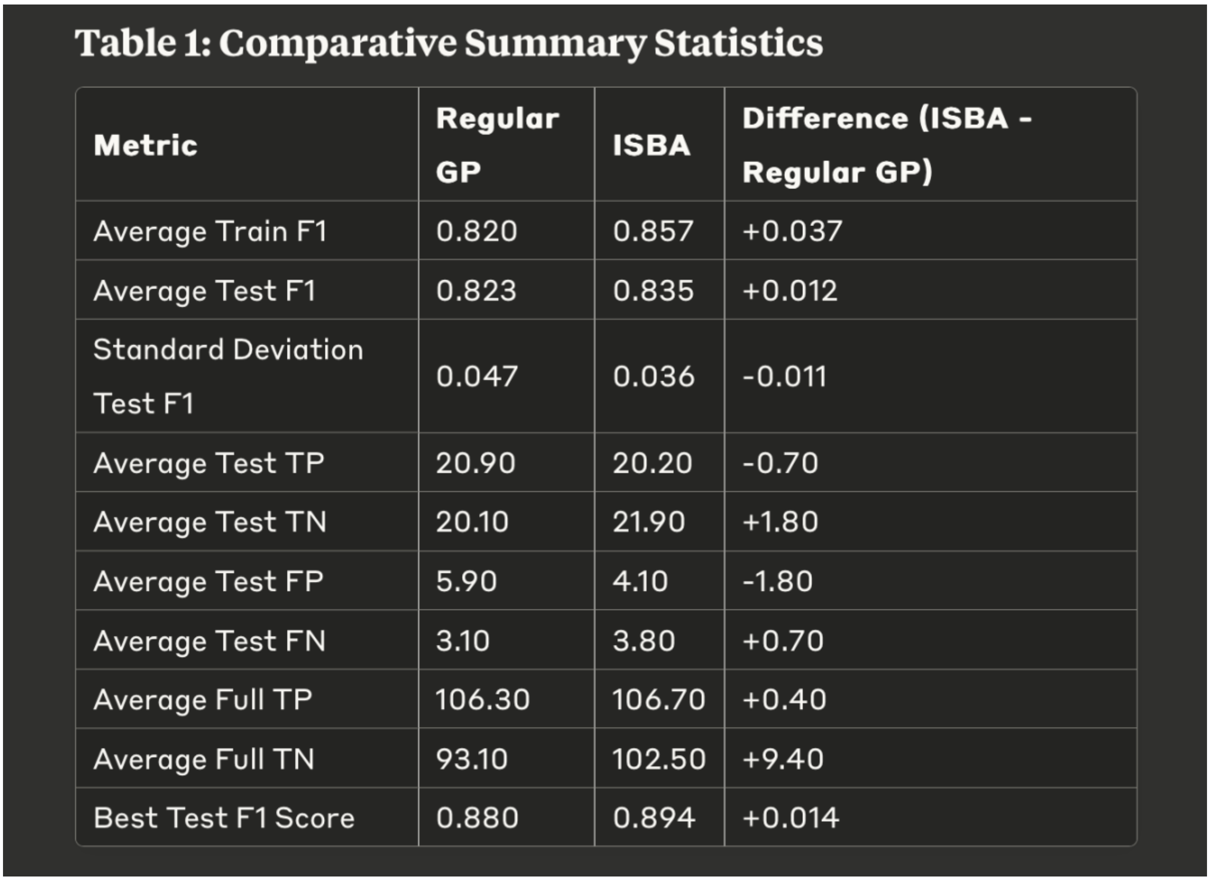

Table 1: Comparative Summary Statistics

Metric	Regular GP	ISBA	Difference (ISBA - Regular GP)
Average Train F1	0.820	0.857	+0.037
Average Test F1	0.823	0.835	+0.012
Standard Deviation Test F1	0.047	0.036	-0.011
Average Test TP	20.90	20.20	-0.70
Average Test TN	20.10	21.90	+1.80
Average Test FP	5.90	4.10	-1.80
Average Test FN	3.10	3.80	+0.70
Average Full TP	106.30	106.70	+0.40
Average Full TN	93.10	102.50	+9.40
Best Test F1 Score	0.880	0.894	+0.014

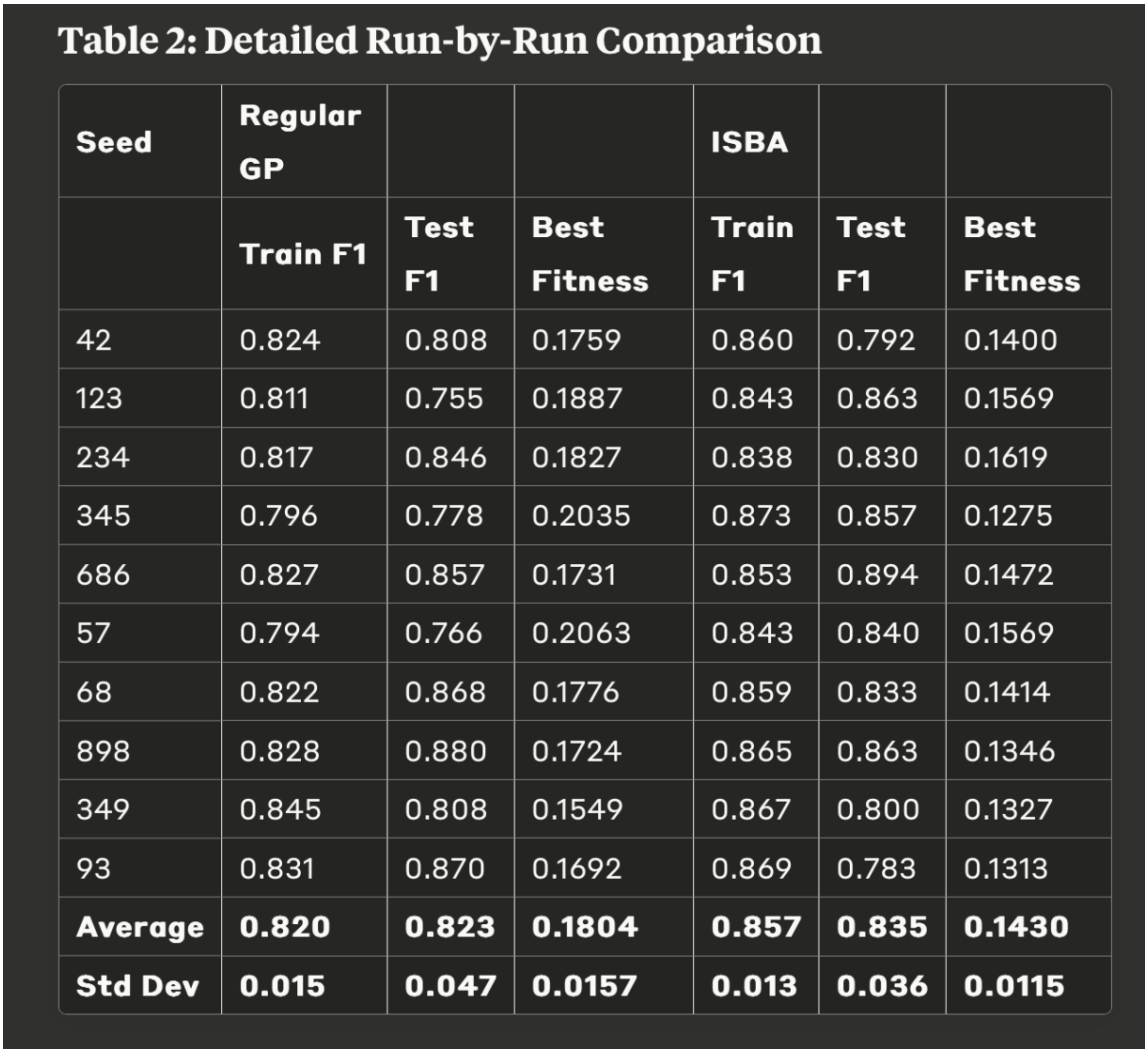

Table 2: Detailed Run-by-Run Comparison

Seed	Regular GP			ISBA		
	Train F1	Test F1	Best Fitness	Train F1	Test F1	Best Fitness
42	0.824	0.808	0.1759	0.860	0.792	0.1400
123	0.811	0.755	0.1887	0.843	0.863	0.1569
234	0.817	0.846	0.1827	0.838	0.830	0.1619
345	0.796	0.778	0.2035	0.873	0.857	0.1275
686	0.827	0.857	0.1731	0.853	0.894	0.1472
57	0.794	0.766	0.2063	0.843	0.840	0.1569
68	0.822	0.868	0.1776	0.859	0.833	0.1414
898	0.828	0.880	0.1724	0.865	0.863	0.1346
349	0.845	0.808	0.1549	0.867	0.800	0.1327
93	0.831	0.870	0.1692	0.869	0.783	0.1313
Average	**0.820**	**0.823**	**0.1804**	**0.857**	**0.835**	**0.1430**
Std Dev	**0.015**	**0.047**	**0.0157**	**0.013**	**0.036**	**0.0115**

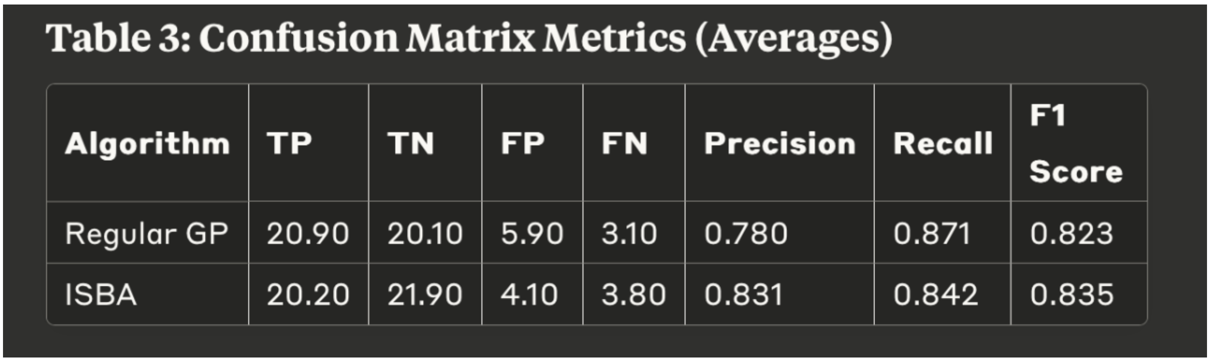

Table 3: Confusion Matrix Metrics (Averages)

Algorithm	TP	TN	FP	FN	Precision	Recall	F1 Score
Regular GP	20.90	20.10	5.90	3.10	0.780	0.871	0.823
ISBA	20.20	21.90	4.10	3.80	0.831	0.842	0.835

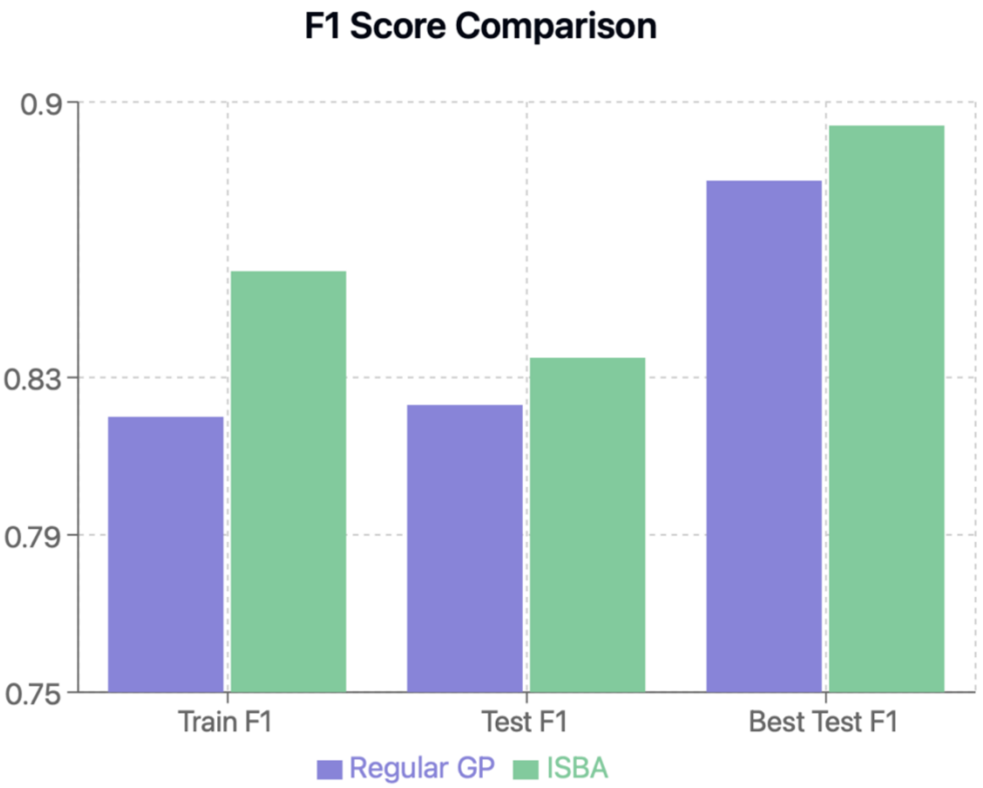

Chart 1: Comparison of F1 Scores between Regular GP and SBGP

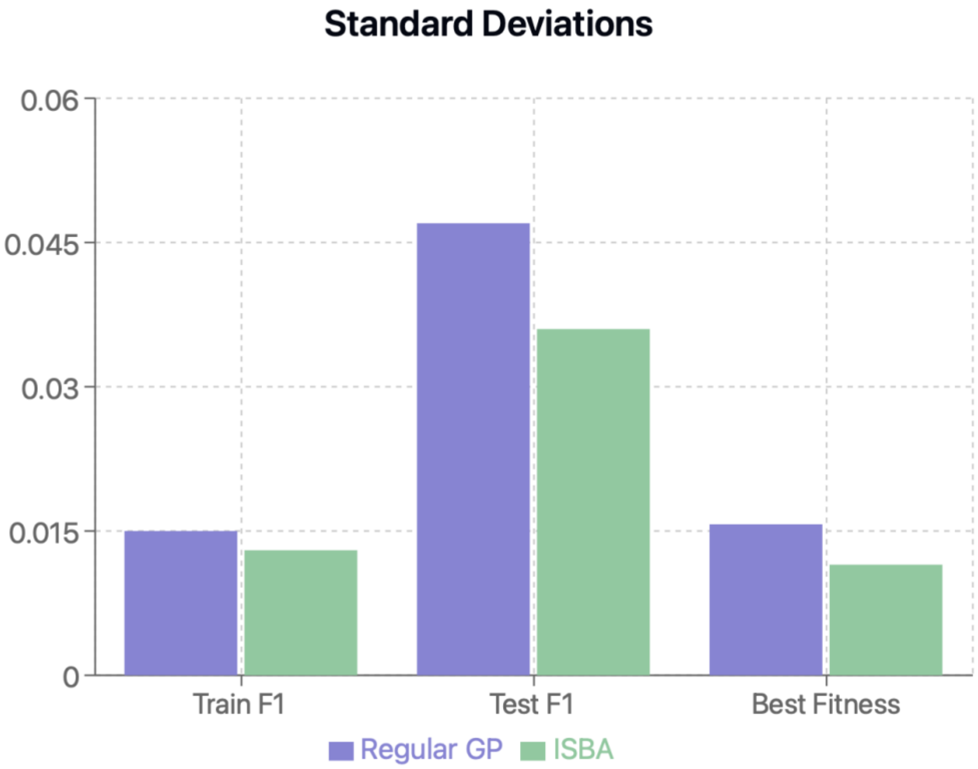

Chart 2: Standard Deviation of F1 Scores for Regular GP and SBGP

Discussion of Results

Comparative Analysis of Structural-Based GP and Regular GP

The comparison between the Structural-Based Genetic Programming (Structural-Based GP) and Regular Genetic Programming (GP) approaches reveals meaningful distinctions in performance, as evidenced by the data in Table 1 and the accompanying F1 score and standard deviation charts. Structural-Based GP demonstrates a clear advantage in training performance, achieving an average Train F1 score of 0.857 compared to Regular GP's 0.820, a difference of +0.037. This suggests that Structural-Based GP's structured optimization process, incorporating global and local phases, enables it to more effectively capture patterns within the hepatitis dataset during training. In terms of generalization, Structural-Based GP also outperforms Regular GP on the average Test F1 score, with 0.835 against 0.823, an improvement of +0.012. The Best Test F1 Score further highlights Structural-Based GP's edge, reaching 0.894 compared to Regular GP's 0.880, a difference of +0.014, indicating Structural-Based GP's capacity to produce a superior model in its best run.

Consistency across runs is another area where Structural-Based GP excels. The standard deviation of Test F1 scores for Structural-Based GP is 0.036, notably lower than Regular GP's 0.047, a reduction of -0.011, as illustrated in the standard deviations chart. This lower variability suggests that Structural-Based GP delivers more stable performance across the 10 runs, which is a critical consideration for reliability in practical applications. Examining the confusion matrix metrics in Table 3, Structural-Based GP reduces False Positives (4.10 vs. 5.90) and False Negatives (3.80 vs. 3.10), while increasing True Negatives (21.90 vs. 20.10) and slightly decreasing True Positives (20.20 vs. 20.90). These results indicate that Structural-Based GP achieves a more balanced classification, particularly in minimizing over-prediction errors, which is essential in medical diagnostics such as hepatitis prediction, where false positives can lead to unnecessary interventions.

The detailed run-by-run comparison in Table 2 further supports these findings. Structural-Based GP consistently achieves higher or comparable Train F1 scores across most seeds, for instance, 0.873 versus 0.796 for seed 345—and its best fitness values are generally lower (e.g., 0.1275 vs. 0.2035 for seed 345), reflecting more optimized solutions. However, the F1 Score Comparison chart indicates that Regular GP can occasionally outperform Structural-Based GP on Test F1 for specific seeds, such as seed 898 with a score of 0.880. This suggests that while Regular GP can achieve competitive results under certain conditions, Structural-Based GP's structured approach, with mechanisms like subtree fixing and similarity-based selection, generally provides enhanced accuracy and stability, making it a more robust choice for this dataset.

Analysis of Structural-Based GP Performance

When evaluated independently, the Structural-Based GP algorithm exhibits strong and consistent performance in the context of hepatitis classification. As reported in Table 1, Structural-Based GP achieves an average Train F1 score of 0.857, demonstrating its effectiveness in modeling the training data. Its average Test F1 score of 0.835 reflects solid generalization to unseen data, while the Best Test F1 Score of 0.894 (achieved with seed 686) underscores its potential to produce highly accurate models, as shown in the F1 Score Comparison chart, where Structural-Based GP consistently surpasses Regular GP across Train F1, Test F1, and Best Test F1 metrics.

The standard deviation of Test F1 scores for Structural-Based GP, at 0.036, as depicted in the standard deviations chart, indicates a high degree of stability across the 10 runs. This low variability can be attributed to Structural-Based GP's multi-phase optimization strategy, which includes five global runs and three local runs with fixed subtrees of depth two, effectively mitigating the stochastic nature of genetic programming. The confusion matrix metrics in Table 3 further illustrate this stability, with Structural-Based GP achieving an average True Positive rate of 20.20, True Negative rate of 21.90, False Positive rate of 4.10, and False Negative rate of 3.80. The reduction in False Positives and the balanced increase in True Negatives suggest that Structural-Based GP excels at minimizing misclassifications, a crucial attribute in medical applications where diagnostic accuracy directly impacts patient outcomes.

Table 2 provides a detailed run-by-run breakdown, revealing that Structural-Based GP's Train F1 scores range from 0.838 (seed 234) to 0.873 (seed 345), with Test F1 scores spanning from 0.783 (seed 93) to 0.894 (seed 686). The best fitness values, such as 0.1275 for seed 345 and 0.1313 for seed 93, indicate Structural-Based GP's ability to converge to highly optimized solutions. The F1 Score Comparison chart reinforces this consistency, with Structural-Based GP maintaining a higher average performance across all evaluated metrics. This reliability likely stems from Structural-Based GP's use of global and local similarity indices, which promote diversity while ensuring effective population evolution. Collectively, these results position Structural-Based GP as a robust and accurate tool for hepatitis classification, offering dependable performance with minimal variability across runs.

Conclusion

This report demonstrates the effectiveness of Structure-Based Genetic Programming with ISBA in addressing the challenges of hepatitis classification. By leveraging a two-phase evolution strategy, subtree locking, and similarity-based diversity control, Structural-Based GP outperforms Regular GP in both training and test performance, achieving an average Test F1 score of 0.835 compared to 0.823, and a Best Test F1 Score of 0.894 against 0.880. Its lower standard deviation (0.036 vs. 0.047) further highlights its consistency, making it a reliable choice for medical diagnostics where stability and accuracy are paramount. The implementation journey revealed the importance of balancing complexity and structure, with ISBA providing a principled approach to evolving expressive yet functional models. These findings underscore the potential of structure-based methods in enhancing genetic programming for complex classification tasks, offering a pathway for future improvements in parameter tuning and structural refinement.