Genetic Programming for Breast Cancer Classification

Nick Aksamit
Department of Computer Science
Brock University
St. Catharines, Canada
na16dg@brocku.ca

I. Introduction

Throughout the billions of years that life has existed on Earth, those that are successful in the environment survive and reproduce at higher rates than those less fortunate. This leads to biological structures that endure due to their strength, or in other words, high fitness. Genetic Algorithms (GAs) take reference to Darwinian natural selection and DNA replication, meshing the two concepts into an algorithm that performs adequately in many areas where others do not [1].

Genetic operators such as selection, crossover, and mutation are implemented in GAs, each collaborating to find the best possible fitness. Solutions are named "chromosomes" and can take the form of various schemas, but are ultimately static and frequently numerical. The Genetic Program (GP) is an advancement over GAs. Similar genetic operators are utilized, but it is the chromosome, or lack thereof, that remains a primary difference between the two algorithms.

GPs have been applied to a wide variety of areas, such as art, hydraulic engineering, drug design, and many more [2]. In this work, a GP is implemented for the classification of breast cancer using features extracted from breast mass imaging. Afterwards, comparisons are made between experiments in which the breast cancer data set attributes differ. Breast cancer has overtaken lung cancer as the most frequently diagnosed cancer, and caused nearly 700,000 deaths in 2020 alone [3]. Early detection reduces the mortality rate of this disease, and results express whether GP is a valid classifier for breast cancer detection, along with providing the necessary features for classification.

The rest of this work is divided into sections as follows: Section II explains the information used to construct the problem. This includes GP parameters, and the data set. Following is Section III, where results from the experimentation are gathered, illustrated, and discussed. Lastly, in Section IV results are generalized, with discourse on where future work may head for improvement of results found in Section III.

II. Experimental Setup

In this section, details on experimentation are addressed. All GP parameters, and the data set are listed in subsequent subsections. All results gathered for this work was with use of the DEAP system [4].

	Malignant	Benign
Count	212	357
Distribution	37%	63%

TABLE I
Data Classifications

A. Data Set

The breast cancer Wisconsin data set is a popular binary classification task found in the UCI machine learning repository [5]. All entries illustrate features extracted from an image of a tumour, where designations are either benign or malignant. As is illustrated in Table I, there is a large discrepancy between the distribution of the two classification options, with more benign entries than malignant. Therefore, a stratified split was performed in order to maintain similar distributions between training and testing sets. It is the hope that this reduces any bias due to the uneven classification data entries.

Table II expresses the various attributes present in the breast cancer data set. Attribute 1 is the identification number, which has no relation to the classification of either benign or malignant. For this purpose, it is excluded from experimentation. Each of the remaining attributes have three numerical values attached to them, being the mean, standard deviation (st. dev.), and worst. For example attribute 2, the radius, has three values: mean radius, standard deviation of radii, and worst radius. When illustrating the union of two sets, a hyphenated form will be used, such as mean-worst.

The experimentation within this work involves comparisons between all seven combinations of the three attribute values. For example, one experiment may utilize the mean, standard deviation, and worst value, while another uses only the worst. In this way, it will be easier to determine if all features are necessary for classification using a GP system, or if a subset is sufficient.

B. GP Parameters

The parameters of a GP system have a large effect on the results obtained. For example, a larger population size can lead to a better outcome, but at the expense of having to compute additional individuals. The crossover method used for all experimentations is one point crossover, where one point

Number	Attribute
1	ID Number
2	Radius
3	Texture
4	Perimeter
5	Area
6	Smoothness
7	Compactness
8	Concavity
9	Concave Points
10	Symmetry
11	Fractal Dimension

TABLE II
DATA SET ATTRIBUTES

Parameter	Value
Population Size	300
Crossover Probability	90%
Mutation Probability	10%
Generations	100
Number of Elites	1
Min Tree Size (Init.)	2
Max Tree Size (Init.)	4
Min Tree Size (Mut.)	1
Max Tree Size (Mut.)	2
Tournament Size	3

TABLE III
DEFAULT GP PARAMETERS

is selected on two parent trees and the subtree at the point is swapped. For mutation, a random point is selected and a new subtree is generated within a specific size limit. When selecting individuals for the new generation, a tournament approach was utilized [6]. Lastly, tree initialization follows the half and half approach, where there is a 50% chance of either full or grow generation.

The default parameter set is exhibited in Table III, where all values remain static along the experimentation process. Preliminary studies found that a much higher crossover rate than mutation lead to better overall results, and so 90% and 10% were exercised, respectively. Similarly, a tournament size of 3 found good results, along with only one elite individual.

C. GP Language

In GP, a tree structure is utilized by every individual in the population. The tree includes various function symbols and terminals, combining together to make up a program. Each function symbol has a certain arity, return type, and types for arguments. A strongly-typed GP system is employed since the function set uses both booleans and decimal values. Table VII, found in Appendix A illustrates the exercised function set. Arithmetic functions return decimal values, while relational and boolean functions return values of type boolean. The ITE function represents an if, then, and else statement, and returns only a decimal value. All of the terminals are expressed in Table VIII. T and F represent simple boolean values, while the ephemeral constant is generated only once, taking a random decimal between -5 and 5, and remains constant throughout existence.

D. Fitness Function

The fitness of an individual defines its quality. Usually, it takes the form of a function f: $x \rightarrow \mathbb{R}$, where x is an individual of the population. Since f returns a real number and takes an individual as input, the quality of different individuals can be compared by relational operators.

For the purposes of classification, the fitness is defined as the number of correct predictions an individual gets on the training set. The following Algorithm 1 illustrates how calculations were performed. Values that were at least 0 or higher are classified as malignant, and the rest benign. The number of hits are calculated, which signify the number of correct classifications that were made by an individual. Thus, a higher fitness is better.

Algorithm 1 Fitness

Require: x be a compiled program
  hits $\leftarrow$ 0
  for each training example t **do**
    if x(t.arguments) $\geq$ 0 and t.classification = M **then**
      hits $\leftarrow$ hits + 1
    else if x(t.arguments) $<$ 0 and t.classification = B **then**
      hits $\leftarrow$ hits + 1
    end if
  end for
  return hits

E. Objective and Additional Information

The intention of this work is to determine which features are necessary for a GP system to accurately classify breast tumour types. In the data set provided, a total of 11 attributes are presented. The first is an identification number, which is not employed due to lack of correspondence with classification. The remaining 10 attributes each have three values attached to them, namely a mean, standard deviation, and worst. In this work, it will be determined if it is necessary to apply all, or if merely a subset is sufficient. A total of 7 experiments are completed, using all possible combinations of the mean, standard deviation, and worst values. The data set is split into an 80-20% training and testing set, with a stratified approach to ensure the distribution of both classifications are relatively the same. All hypothesis testing used in this work is at the 95% confidence level, and each experiment is executed 10 times.

III. RESULTS AND DISCUSSION

Results are separated into two sections, namely by training and testing. The training section is further segregated into examination between the fitness, and size of individuals per generation. It forms topics that relate to the training of the classification model. Afterwards, discussion is raised on how the best models generalize to fit the testing set.

A. Training

1) Fitness: All seven experiments were carried out as specified in Section II. The average fitness of all populations

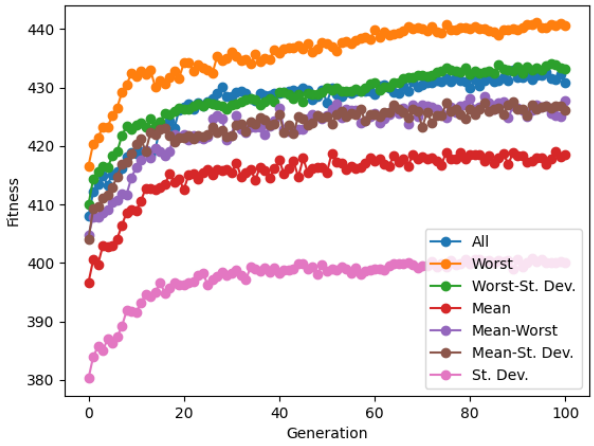

Fig. 1. Average Generational Fitness

each generation is illustrated in Figure 1. From first glance, it appears that utilizing the worst values of each attribute gives the best fitness, with standard deviation producing the worst. Right after initialization, the fitness of worst values is the highest, and consistently rises above all other data configurations. This gives the presupposition that using a subset of values can generate higher fitness than all of them. It should be noted that the three highest plotted lines all include these worst data points.

The plot alludes that within the data subsets, which are separated by standard deviation, mean, and worst, it is mean and standard deviation that struggle to compete. The latter can be found at the bottom by a large average margin (17.6 with mean, and 38.5 with worst). Thus when only interpreting the visualization of average training results, using the worst values of all attributes is best, and standard deviation is worst.

Statistical testing is complete on the final generation of each experiment, using the average fitnesses. A QQ-plot, along with the Shapiro-Wilk test are initially applied to examine normality of the data points, and found that only mean, mean-worst, and standard deviation are normal. Thus, when comparisons are drawn between these subset pairings, the Student's t-test will be used, with Mann-Whitney U employed for the rest. With a preliminary Kruskal-Wallis test, it is discovered that there is a statistical difference in median between the seven experiments. This grants further exploration into determining where the anomalies lie.

Further statistical testing was applied to each pairing of data subsets. The results are expressed in Table IV, in the form of ranking from first (best) to last. Ranking was done by scoring experiments that were found statistically better than others. Similarly to what was expressed by the plot in Figure 1, using the worst subset gives the best average performance, while solely relying on standard deviation is inferior. It should be noted that application of all subsets is ranked as third out of five, expressing that it may contain too much information. In

Rank	Data Subset
1	Worst
2	Worst-St. Dev.
3	All
4	Mean
4	Mean-Worst
4	Mean-St. Dev.
5	St. Dev.

TABLE IV
RANKING OF FINAL AVERAGE FITNESS

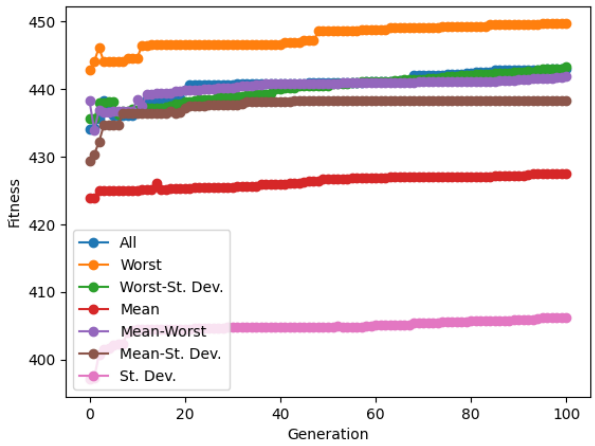

Fig. 2. Elite Generational Fitness

addition, the pairing of standard deviation and worst performs adequately as rank two. However, it is unknown whether this comes directly from using the worst values, or if the pairing itself is favourable.

Similar to comparisons of the average, the fitness of elites are also displayed in Figure 2. The plotted values resemble that of the averages (Figure 1), but with an increased fitness per generation due to averaging the elite individual in each ten executions. Patterns express parallelism with what is expressed in averages, except with the mean-worst subset, which is much closer to both worst-st. dev. and all. Once more, standard deviation is surpassed by every subset, while worst is the best.

Moreover, another hierarchy is presented in Table V, but using only elite individuals of the final generation. Statistical testing followed the same process as for the previous rankings (Table IV), with normality, Kruskal-Wallis, and then the Student's t or Mann-Whitney U test. Worst is given the highest placement, with standard deviation the lowest. This follows the same pattern expressed in the former ranking of average performance. Notably, some of the intermediate subsets have exchanged orders of ranking. Mean is now found near the bottom, while all and mean-worst are adjacent to the top.

2) Size: The average sizing of individuals throughout experimentation is visualized in Figure 3. Various contrasts in positioning can be seen in comparison with the average fitnesses (Figure 1). Particularly, the subset containing worst

Rank	Data Subset
1	Worst
2	All
2	Worst-St. Dev.
2	Mean-Worst
3	Mean-St. Dev.
4	Mean
5	St. Dev.

TABLE V
RANKING OF FINAL ELITE FITNESS

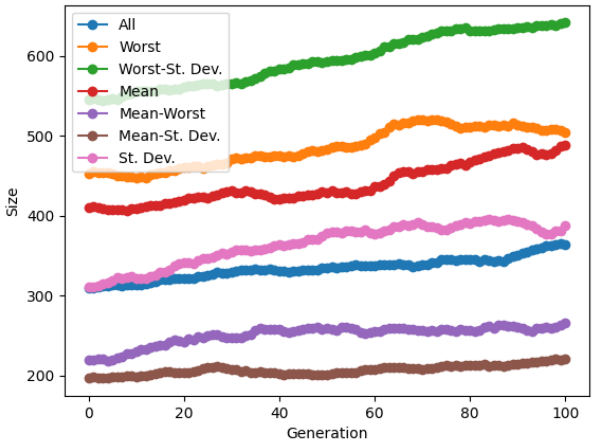

Fig. 3. Average Generational Size of Individuals

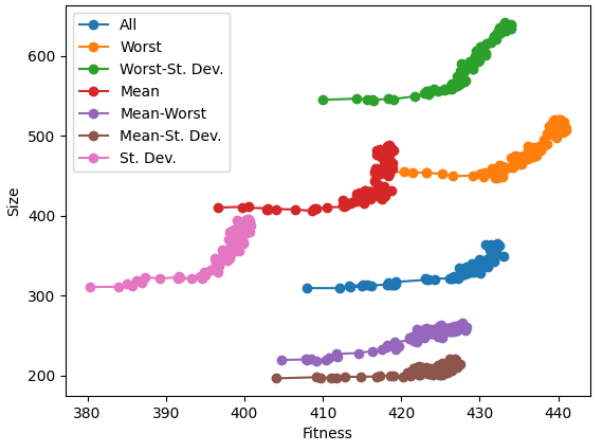

Fig. 4. Average Size and Fitness

Rank	Data Subset	Correct Classifications
1	Mean-Worst	111
2	Mean	108
3	All	107
4	Worst	106
5	Worst-St. Dev.	105
6	Mean-St. Dev.	101
7	St. Dev.	93

TABLE VI
CORRECT CLASSIFICATIONS ON TESTING SET

values is not directly succeeding all others, but has been surpassed by worst-st. dev. It also seems that out of all the experiments plotted, it is only the worst and standard deviation groups that have noticeably decreased in average size, specifically in the later generations. This is interesting, as the average fitness in those same generations rise in both groups of data sets. Perhaps this optimization in size could be due to successful mutations occurring later, where convoluted branches become largely compressed. Otherwise, the remaining experiments have their average size either increase, or increase slightly at the beginning and plateau afterwards.

Throughout execution of the GP system, there may be some correlation between average fitness, and average size of individuals in the populations. Figure 4 sheds light on these potential associations. Seemingly, as the size increases, the fitness does as well. However, some discrepancies also occur. In all subsets of the data, there are points where sizes are notably similar, but fitness values are not. On the plot it takes the shape of a flat line. This is favourable since it means individuals are becoming optimized. However, there are also circumstances where the size increases, but the fitness remains the same. This is most notable in mean, as there are many individuals of the same fitness, but have less size.

Having individuals of same size and differing fitness leads to the notion of some degeneration within the GP system. The decline in quality may be a result of redundant branches in the generated trees, as a result of unnecessary mutations. This may include events such as an ”if false” condition, which can easily be replaced by the ”else” subtree. No action was taken to reduce the redundancy, and so individuals may have accumulated dispensable subtrees throughout many generations.

B. Testing

Although a model may present adequate performance on the training set, there is no guarantee that it has generalized well. Sometimes overfitting may occur, where a model achieves high fitness in training, but comparably low in testing. This section aims to compare rankings between the training and testing, along with illustrating some of the found solutions. Correct classifications from the best models for each data subset are presented in Table VI.

The data subset utilizing worst values, which performed best on the training set, does not achieve comparable results in testing. Overall it takes fourth place. However, analogous to training is the performance of standard deviation. It has again been outclassed by all other subsets. Noticably, all data subsets that merge standard deviation with another subset are at the bottom of the rankings. With their subpar performance in training, and clear non-generalization in testing, it cannot be recommended that the standard deviation data values be used for classification of breast cancer type. Contradictory to training, mean itself, and its fusion with worst, overtakes all

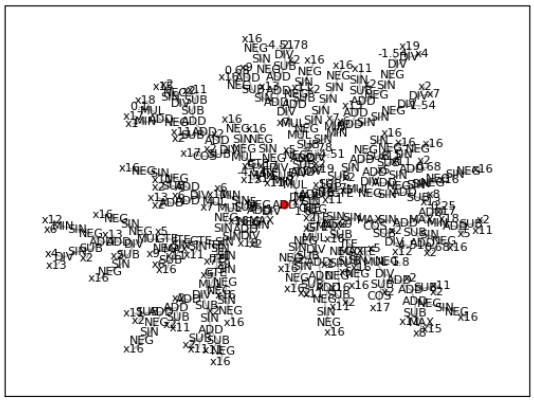

Fig. 5. Mean-Worst Superior Individual

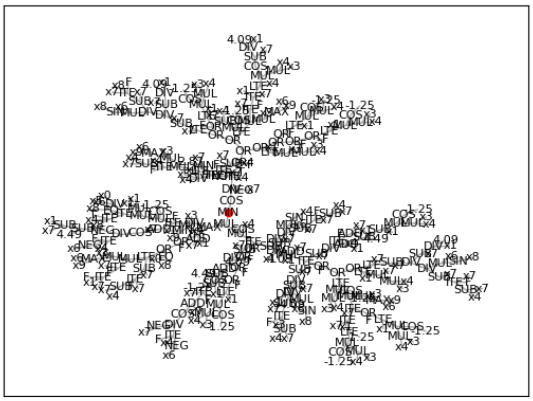

Fig. 7. Worst Superior Individual

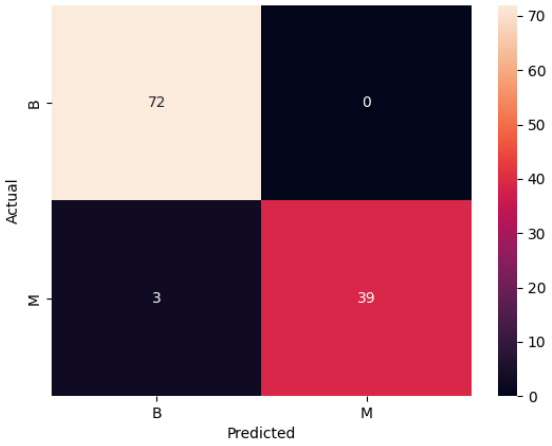

Fig. 6. Mean-Worst Testing Confusion Matrix

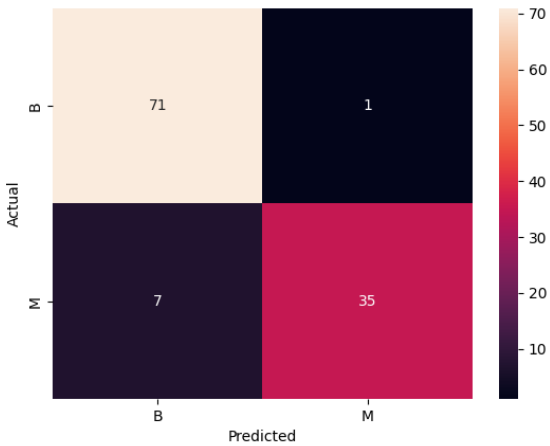

Fig. 8. Worst Testing Confusion Matrix

other experiments in testing classifications. It seems that worst by itself does not have the aspects for generality, but when amalgamated with mean, performs adequately.

Figure 5 illustrates the individual used in classification of the testing set for the mean-worst data. It is the solution with the highest fitness from training, and from the selected individuals between data sets, the one that is the best in testing as well. From first glance, the tree appears very convoluted. The red dot represents the root, and branches from many directions reach a large depth. Results from the superior are exhibited in Figure 6, where classifications of benign tumours are perfect, and 3 malignant were incorrectly projected as benign. From the gathered data, there are more overall samples of benign than malignant. Therefore when training, more cases of benign are observed. This may explain why the flawless performance comes from the benign category, rather than malignant.

To determine performance of the data set that was dominant during training, the best individual from the worst data subset is portrayed in Figure 7. When observing, it appears sparse in comparison with Figure 5, however this may just be from how the graph is created. Nonetheless, there still exist branches that are long in size and contain various subtrees. The testing set classifications however, express where problems lie in performance. Figure 8 conveys a difficulty in categorization of malignant cancer, all the while having a nearly unflawed benign grouping. This identifies a similar matter found in the mean-worst subset, and possibly might be due to an uneven distribution of classifications in the overall data set. Although accuracy levels are not 100%, they are still high at levels such as 97% and 93%, respectively between mean-worst and worst. A notable find is that where subsets lack in testing accuracy, most of the incorrectly classified samples come from the malignant category.

IV. Conclusions and Future Work

Genetic programming has proven itself to be a valid classifier for breast cancer type. Given the data from [7], experiments were carried out to test how GP performs when subsets of the data are selected. This is because for all of the attributes included, three values are involved, namely the mean, standard deviation, and worst. Therefore, seven experiments were completed in total using all combinations of the three aforementioned values.

Initially, the breast cancer data was split into a training and testing set. Then, experimentation was performed with static parameters. The subset of data utilizing the worst values performed best during training, receiving approximately 97% accuracy on average in the later generations. However, although usage of the worst subset is best for training, this is not the case during testing. At this stage in experimentation, mean-worst data values performed the best, leading to the notion that although worst was superior during training, it did not generalize well to unseen data. This is surprising as in training mean-worst was on average statistically ranked fourth out of five, but achieved a near flawless accuracy in testing. It was noticed that during testing, malignant values were slightly more difficult to classify. This might likely be due to the greater amount of benign samples in the overall data.

Throughout all experimentations there exist individuals with inefficient trees, or a tree that has the same fitness as another, but a larger size. This is a well-known and common problem in GP systems known as bloat [8]. Bloat is notably exacerbated in the mean subset of data, as is seen by the near-vertical plots in Figure 4. Future work should encompass improving tree efficiency. Perhaps this will introduce additional competitiveness between individuals as there will be less bloat, which may in turn positively affect the genetic operators.

Acronyms

GP Genetic Program
GA Genetic Algorithm

References

[1] K.-F. Man, K. S. Tang, and S. Kwong, *Genetic algorithms: concepts and designs*. Springer Science & Business Media, 2001.

[2] A. H. Gandomi, A. H. Alavi, and C. Ryan, *Handbook of genetic programming applications*. Springer, 2015.

[3] S. Lei, R. Zheng, S. Zhang, S. Wang, R. Chen, K. Sun, H. Zeng, J. Zhou, and W. Wei, "Global patterns of breast cancer incidence and mortality: A population-based cancer registry data analysis from 2000 to 2020," *Cancer Communications*, vol. 41, no. 11, pp. 1183–1194, 2021.

[4] F.-A. Fortin, F.-M. De Rainville, M.-A. G. Gardner, M. Parizeau, and C. Gagné, "Deap: Evolutionary algorithms made easy," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2171–2175, 2012.

[5] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[6] B. L. Miller, D. E. Goldberg, *et al.*, "Genetic algorithms, tournament selection, and the effects of noise," *Complex systems*, vol. 9, no. 3, pp. 193–212, 1995.

[7] W. H. Wolberg, W. N. Street, and O. L. Mangasarian, "Breast cancer wisconsin (diagnostic) data set," 1995.

[8] S. Luke and L. Panait, "A comparison of bloat control methods for genetic programming," *Evolutionary Computation*, vol. 14, no. 3, pp. 309–344, 2006.

Appendix A
Function and Terminal Set

Function	Arity	Example
ADD	2	$x + y$
SUB	2	$x - y$
MUL	2	$x * y$
DIV	2	x/y (1 if $y = 0$)
NEG	2	$-x$
COS	1	$cos(x)$
SIN	1	$sin(x)$
MAX	1	$max(x)$
MIN	1	$min(x)$
ITE	3	if x, then y, else z
EQ	2	$x == y$
GT	2	$x > y$
GTE	2	$x >= y$
LTE	2	$x <= y$
LT	2	$x < y$
AND	2	x and y
OR	2	x or y
NOT	1	not x

TABLE VII
Function Set

Terminal	Value
F	False
T	True
Ephemeral Constant	$x = \text{rand}(-5, 5), x \in \mathbb{R}$

TABLE VIII
Terminal Set