

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | |
|  | | Comparative Analysis of Multi-Layer Perceptron, Decision Trees and Genetic Programming for Financial Forecasting | | |  | | |
|  |  | | | | |  | |
|  | | |  |  | | | |
|  | | | **Ratjatji Malatji - U25267869**  **Nolwazi Mabaso - U22505511**  **Tlangelani Khosa - U22913808**  **Thomas Murashidzi - U25717376**  **—**  **May 24th, 2025**  **—**  **COS 314 – Artificial Intelligence**  **Assignment 3** |  | | |

Table of Contents

[Financial Stock Prediction Report 3](#_Toc199012427)

[Abstract 3](#_Toc199012428)

[Introduction 3](#_Toc199012429)

[**Objective** 3](#_Toc199012430)

[Methodology 3](#_Toc199012431)

[Dataset 3](#_Toc199012432)

[Genetic Programming (GP) 4](#_Toc199012433)

[Initial solution generation 4](#_Toc199012434)

[Fitness evaluation 4](#_Toc199012435)

[Selection criteria 4](#_Toc199012436)

[Genetic operators 4](#_Toc199012437)

[Generating a new population 5](#_Toc199012438)

[Design specifications: 5](#_Toc199012439)

[Multi-Layer Perceptron (MLP) 5](#_Toc199012440)

[**MLP Architecture** 6](#_Toc199012441)

[Data Preparation 6](#_Toc199012442)

[**Weight Initialization** 6](#_Toc199012443)

[**Loss Calculation** 6](#_Toc199012444)

[**Weight Updating** 7](#_Toc199012445)

[**Evaluation Metrics** 7](#_Toc199012446)

[Design specifications: 7](#_Toc199012447)

[Optimizer & Hyperparameters 7](#_Toc199012448)

[J48 Decision Tree the from Weka Package 8](#_Toc199012449)

[Design specifications 8](#_Toc199012450)

[Experimental Setup 8](#_Toc199012451)

[Implementation Details 8](#_Toc199012452)

[Results 9](#_Toc199012453)

[Performance Table 9](#_Toc199012454)

[Statistical Analysis 9](#_Toc199012455)

[Wilcoxon signed-rank test 10](#_Toc199012456)

[Wilcoxon signed-rank test comparing the accuracy of GP and MLP 11](#_Toc199012457)

[Wilcoxon signed-rank test comparing the accuracy of GP and MLP 12](#_Toc199012458)

[Conclusion 13](#_Toc199012459)

# Financial Stock Prediction Report

## Abstract

This report investigates the application of machine learning classification algorithms for predicting financial stock purchase decisions based on historical market data. Three models, namely genetic programming (GP), multi-layer perceptron’s (MLP), and the J48 decision tree (DT) algorithm were implemented and evaluated. Using the provided datasets, each model's performance was assessed based on the accuracy and F1-score on both training and testing datasets. Furthermore, a Wilcoxon signed-rank test was conducted to statistically compare the differences in the performance of the genetic program and multi-layer perceptron models. The findings provide insights into the effectiveness of these models in financial forecasting, highlighting their respective strengths and limitations in identifying patterns indicative of stock purchase opportunities.

# Introduction

## **Objective**

The objective of this assignment is to implement and rigorously evaluate three distinct machine learning classification models: GP, MLP and DT’s. These models will be applied to historical financial data with the specific aim of predicting whether or not a given financial stock should be purchased.

# Methodology

## Dataset

Two data sets were used to train and test our models. Both data sets contain the same features but differ in the amount of data they have. BTC\_train contains 998 rows of data while BTC\_test contains 262 rows. The features that are provided in both data sets are open, high, low, close adjusted close and output.

The **open** feature refers to the price of a stock at the beginning of a specific time period.   
The **high** feature refers to the highest price that stock traded during a specific time period.   
The **low** feature refers to the lowest price that stock traded during a specific time period.   
The **close** feature refers to the final stock price at the end of a trading period.   
**Adjusted close** refers to the closing price of the stock after accounting for any corporate actions such as stock splits, dividends, or new stock offerings. This feature provides a true representation of the stock's value over time, ensuring that past prices are comparable to current prices.  
The **output** is the target feature that the models are trying to predict, a binary representation shows whether a stock was purchased based on the other five features. Machine Learning Models

## Genetic Programming (GP)

*GP’s perform classification by evolving a population of computer programs to find the program that predicts the target feature the best based on an instance of data (*Mei, Chen, Lensen, Xue, Zhang, 2022)*.*

The following specific configurations and pre-processing steps were applied as detailed in the ImprovedGeneticProgrammingClassifier.java code:

### Initial solution generation

Our initial solutions were generated by a random procedure, typically a combination of the grow and full methods to ensure that the tree structures would be diverse (Chen & Xue, 2022). The genetic program were generated recursively by applying the function set to elements of the terminal set until a classifier is constructed. The classifiers were then evaluated on the respective datasets.

### Fitness evaluation

The F1-score of the generated program would determine its fitness. In the occurrence that a program exceeds the maximum length, a penalty value would be subtracted from the F1-score. This ensures that if two programs have a similar fitness score, the simpler program would have increased chances of being selected.

### Selection criteria

Tournament selection was used to select the programs that would become parents, four programs were selected and compared, the program with the highest fitness would be selected to reproduce. Additionally, elitism was implemented to ensure the programs with the highest fitness would always be selected to be part of the new generation.

### Genetic operators

Genetic operators drive the evolutionary process by introducing variation and guiding the search towards better solutions (Alhijawi & Awajan, 2024):

#### Cross over

Two parents were selected and randomly exchanged chosen subtrees to create two new offspring. This was done to increase the exploration of the program space.

#### Mutation

Random changes were applied to the genetic programs and the two types of mutation used are grow and shrink mutation. Mutation was used to increase the exploitation of the program space and helped escape local optima. The two types of mutation that were implemented were grow and shrink mutation.

### Generating a new population

After applying the selection and genetic operators, the offspring and the elite individuals from the previous program become the population of the new generation.

This process continued until a predetermined number of generations were created or an early stopping criterion is met.

### Design specifications:

* Population size: 100
* Maximum number of generations: 50
* Crossover rate: 0.7
* Mutation rate: 0.15
* Tournament size: 4
* Early stop patience = 10
* Elite size: 2
* Maximum depth of program tree: 5
* Fitness function: F-1 score of the program.
* Terminal sets: Open (F0), high (F1), low (F2), adjusted close (F3) and close (F4).
* Function sets: Addition, subtraction, multiplication, safe division and an if-then-else statement.

## Multi-Layer Perceptron (MLP)

MLP is a class of feedforward artificial neural network widely used for supervised learning tasks, including classification and regression (Touvron, Bojanowski, Caron, Cord, El-Nouby, Grave, Izacard, Joulin, Synnaeve, Verbeek & Jégou, 2022). MLP’s consist of an input layer, one or more hidden layers and an output layer (Naskath, Sivakamasundari, & Begum, 2023). MLPs are particularly effective for non-linear classification problems where traditional models like logistic regression or decision trees underperform (Das, Sachindra &Chanda, 2022).

### **MLP Architecture**

The following structure was used for MLP model:

1. **Input Layer:** Received features from the dataset and ensured that they were well-scale and could be used as input. The features that were used from the datasets were open, high, low, close and adjusted close.
2. **Hidden Layers:** This single layer was comprised of 16 neurons which were fully connected to the layers that preceded and succeeded it. The weights and bias’s that connected the neurons in the different layers were responsible for identifying patterns in the data and to create accurate predictions. If the model makes an incorrect prediction the weights and biases were iteratively adjusted to rectify the misclassification.
3. **Output Layer:** A sigmoid activation function was applied to the one neuron in the output layer. The results from the activation function were responsible for the model’s prediction.

The following specific configurations and pre-processing steps were applied as detailed in the MLP.py code:

### Data Preparation

The input features (x) and the target features (y) were separated from both the training and testing data set.

* X\_train and X\_test represent input features, namely, open, high, low, close and adj. close.
* y\_train and y\_test represent the binary target feature, 1 signaled that a stock was bought and 0 signaled that a stock was not bought.

Standard scalar were used to normalize the input features.

### **Weight Initialization**

Kaiming initialization was used to set the initial weights of the network. It ensures that the weights start at an appropriate scale, which helps to mitigate issues like the vanishing or exploding gradient problem during the initial phases of training (Narkhede, Bartakke, & Sutaone, 2022). Implementing Kaiming initialization leads to a faster and more stable convergence.

**Loss Calculation**

For quantifying the error between the model's predictions and the actual target values, weighted binary cross-entropy loss was employed. This loss function is particularly suitable for binary classification problems and accounts for potential class imbalance by assigning different weights to positive and negative classes (Hurtik, Tomasiello, Hula, & Hynar, 2022). To prevent overfitting and encourage a more generalized model, an L2 regularization term was incorporated into the loss function, penalizing large weight values.

### **Weight Updating**

Weights and bias were updated using mini-batch gradient descent with a fixed learning rate. Adam optimizer was used to adapt the learning rates dynamically

### **Evaluation Metrics**

The performance of the MLP model was comprehensively evaluated using the following standard classification metrics:

* Precision: Measures prediction accuracy for positive cases.
* Recall: Evaluates how many actual positives were predicted correctly.
* F1-score: Balances precision and recall for overall performance (Diallo, Edalo & Awe, 2024).
* Accuracy: Computes correct predictions over total samples.

### Design specifications:

* Activation functions: Rectified Linear Unit (ReLU) for the hidden layer and the sigmoid function for the output layer.
* Maximum number of generations: 50
* Crossover rate: 0.7

### Optimizer & Hyperparameters

* Input Size: 5
* Hidden Layer Size: 16 neurons in the single hidden layer
* Output Size: 1 neuron
* Optimizer: Adam Optimizer
* Learning Rate: 0.03
* Batch Size: 128
* Epochs: 2000

## J48 Decision Tree the from Weka Package

As described by Mienye and Jere (2024), decision Trees are a fundamental supervised learning algorithm that operates by recursively partitioning the dataset into smaller subsets based on feature values, ultimately creating a tree-like model of decisions. The decision tree model made use of the J48 algorithm, an open-source Java implementation of the C4.5 algorithm available in the Weka machine learning workbench. Key characteristics of J48 include its use of information gain ratio to select the splitting attribute at each node, and its built-in pruning mechanism to reduce overfitting (Chiang & Chi, 2022).

### Design specifications

The following specific configurations and pre-processing steps were applied as detailed in the J48DecisionTree.java code:

* **Feature Discretization**: All continuous numerical features, F0 to F4, were converted into nominal attributes by applying the Discretize filter with **10.** This ensured that J48 could process the continuous features by splitting on defined intervals.
* **Training Data Resampling**: The training data was resampled using the Resample filter. This involved setting setNoReplacement(false), setBiasToUniformClass(1.0), and setSampleSizePercent(100), with the seed value provided by the user.
* **J48 Classifier Parameters**:
* Confidence Factor: The setConfidenceFactor was set to 0.25f. This parameter controlled the error pruning, with lower value resulting in more pruning.
* Minimum Number of Objects (Instances) per Leaf: The setMinNumObj parameter was set to 5. This means that for a node to be split, each resulting leaf must contain at least 5 instances, helping to prevent the creation of overly specific or noisy branches.

# Experimental Setup

## Implementation Details

* The programming languages that were used to create the three models were Java and Python.
* Download and open the project folder and select the folder with the model you would like to run. Once within the folder, compile then run the java or python file. The program will be executed in the command line and can be executed without being linked to an IDE.
* A seed value is requested after executing the code, if one is not provided, one will be automatically generated for the user. Inputting a seed allows for reproducibility with the models results.
* Depending on the model the user selected, they may be prompted to enter parameters specific to that model.
* The program prompts the user to enter a data set to train the models and another to test how well the model performs with unseen data.

# Results

## Performance Table

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Run** | | **Model** | **Seed value** | | **Training data** | | | **Testing data** | | | |
| Accuracy | F1 Score | | Accuracy | F1 Score | | |
| 1. | Genetic programming | | 1 | 68.34 | | | 77.31 | 96.20 | | 96.09 \* |
| 1. | Multi-layer perceptron | | 1 | 89.58 | | | 89.53 \* | 88.59 | | 88.46 |
| 1. | Decision tree | | 1 | 60.12 | | | 60.10 | 57.04 | | 54.70 |
| 2. | Genetic programming | | 314 | 53.34 | | | 67.93 | 51.33 | | 67.01 |
| 2. | Multi-layer perceptron | | 314 | 87.88 | | | 87.87 \* | 96.58 | | 96.57 \* |
| 2. | Decision tree | | 314 | 67.138 | | | 67.1 | 57.04 | | 53.00 |
| 3. | Genetic programming | | 999 | 39.42 | | | 41.69 | 65.4 | | 73.62 |
| 3. | Multi-layer perceptron | | 999 | 82.87 | | | 82.69 \* | 96.58 | | 96.57 \* |
| 3. | Decision tree | | 999 | 61.828 | | | 61.2 | 54.378 | | 49.1 |

\* Indicates the best F-1 score for the training and testing dataset.

The results show MLP achieved the best results for 5/6 instances, 3/3 instances where the data was being trained and 2/3 instances where the training data was used to test. The MLP model achieved an accuracy of above 96% for all test instances, proving to be the best and most consistent of the three models when performing financial forecasting.

By using optimization techniques like back propagation and gradient decent alongside non-linear activation functions, MLP was able to learn complex, non-linear relationships in the data. Additionally, feature learning gave MLP an advantage over the other models.

## Statistical Analysis

The Wilcoxon signed-rank test is employed to determine if two dependent groups, GP’s and MLP’s, differ in their ability to make good predictions. The Wilcoxon signed-rank test is a non-parametric test, it does not assume a normal distribution of the performance differences, which is often the case with machine learning model results (Li, Wu, Wei, Guo, Yu, Wang, Li, & Fan, 2021.). Additionally, this test is ideal for paired comparisons, as both models are evaluated on the same training and testing datasets, allowing for a direct and robust assessment of the statistical significance of any observed performance disparities.

## Wilcoxon signed-rank test

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Run** | | **Model** | **Seed value** | **Results from testing data** | | | | | | |
| Accuracy | F1 Score | Precision | Recall | | |
| 1. | GP | | 1 | 97.99 | 97.85 | 100.00 | | 95.79 |
| 1. | MLP | | 1 | 90.11 | 90.03 | 91.76 | | 90.11 |
| 2. | GP | | 314 | 56.78 | 66.57 | 52.76 | | 90.53 |
| 2. | MLP | | 314 | 87.83 | 87.67 | 90.24 | | 87.83 |
| 3. | GP | | 999 | 67.34 | 71.62 | 61.19 | | 86.32 |
| 3. | MLP | | 999 | 82.13 | 81.58 | 86.87 | | 82.12 |
| 4. | GP | | 444 | 78.33 | 81.31 | 72.09 | | 93.23 |
| 4. | MLP | | 444 | 88.21 | 88.06 | 90.48 | | 88.21 |
| 5. | GP | | 555 | 83.65 | 80.72 | 100.00 | | 67.67 |
| 5. | MLP | | 555 | 94.30 | 94.28 | 94.89 | | 94.29 |
| 6. | GP | | 13 | 98.48 | 98.47 | 100.00 | | 96.99 |
| 6. | MLP | | 13 | 92.40 | 92.36 | 93.41 | | 92.40 |
| 7. | GP | | 779876543217 | 97.72 | 97.79 | 100 | | 95.68 |
| 7. | MLP | | 779876543217 | 81.74 | 81.12 | 86.22 | | 81.75 |
| 8. | GP | | 881234567898 | 83.65 | 80.72 | 100 | | 67.67 |
| 8. | MLP | | 881234567898 | 84.03 | 83.64 | 87.93 | | 84.03 |
| 9. | GP | | 52545856 | 98.48 | 98.47 | 100 | | 96.99 |
| 9. | MLP | | 52545856 | 83.65 | 83.23 | 87.77 | | 83.65 |
| 10. | GP | | 2 | 58.17 | 80.72 | 100 | | 67.67 |
| 10. | MLP | | 2 | 85.55 | 85.3 | 88.46 | | 85.56 |
| 11. | GP | | 32 | 66.16 | 51.89 | 92.31 | | 36.09 |
| 11. | MLP | | 32 | 88.21 | 88.06 | 90.48 | | 88.21 |
| 12. | GP | | 4 | 83.65 | 80.72 | 100 | | 67.67 |
| 12. | MLP | | 4 | 81.36 | 80.74 | 86.47 | | 81.37 |
| 13. | GP | | 13069854 | 89.95 | 88.24 | 100 | | 78.95 |
| 13. | MLP | | 13069854 | 83.27 | 882.82 | 87.50 | | 83.27 |
| 14. | GP | | 14004 | 98.10 | 98.15 | 96.38 | | 100 |
| 14. | MLP | | 14004 | 86.69 | 86.47 | 89.51 | | 86.69 |
| 15. | GP | | 15005199 | 83.65 | 80.72 | 100 | | 67.67 |
| 15. | MLP | | 15005199 | 87.83 | 87.69 | 89.93 | | 87.83 |
| 16. | GP | | 6 | 83.65 | 80.72 | 100 | | 67.67 |
| 16. | MLP | | 6 | 88.97 | 88.85 | 90.98 | | 88.97 |
| 17. | GP | | 52765 | 93.62 | 93.60 | 100 | | 87.97 |
| 17. | MLP | | 52765 | 83.27 | 82.82 | 87.50 | | 83.27 |
| 18. | GP | | 188193599 | 65.02 | 50.54 | 88.68 | | 35.34 |
| 18. | MLP | | 188193599 | 85.17 | 84.86 | 88.59 | | 85.17 |
| 19. | GP | | 31 | 49.43 | 0 | 0 | | 0 |
| 19. | MLP | | 31 | 87.45 | 87.27 | 89.99 | | 87.45 |
| 20. | GP | | 12365478 | 38.58 | 37.89 | 40.74 | | 35.42 |
| 20. | MLP | | 12365478 | 96.96 | 96.96 | 97.13 | | 96.96 |

The results for the Wilcoxon test were generated using the above results.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **run** | **F1 for GP** | **F1 for MLP** | **Difference between MLP & GP F1** | **Unsigned difference** | **rank of diff.** | **Signed rank** |  |  |
| 1 | 90.03 | 97.85 | -7.82 | 7.82 | 8 | -8 | The sum of all the negative signed ranks | W- |
| 2 | 87.67 | 66.57 | 21.1 | 21.1 | 16 | 16 | -88 |
| 3 | 81.58 | 71.62 | 9.96 | 9.96 | 10 | 10 |  |
| 4 | 88.06 | 81.31 | 6.75 | 6.75 | 6 | 6 |  |
| 5 | 94.28 | 80.72 | 13.56 | 13.56 | 14 | 14 |  |  |
| 6 | 92.36 | 98.47 | -6.11 | 6.11 | 5 | -5 |  |  |
| 7 | 81.12 | 97.79 | -16.67 | 16.67 | 17 | -17 | The sum of all the positive signed ranks | W+ |
| 8 | 83.64 | 80.72 | 2.92 | 2.92 | 2 | 2 | 200 |
| 9 | 83.23 | 98.47 | -15.24 | 15.24 | 18 | -18 |  |
| 10 | 85.3 | 80.72 | 4.58 | 4.58 | 3 | 3 |  |
| 11 | 88.06 | 51.89 | 36.17 | 36.17 | 25 | 25 |  |  |
| 12 | 80.74 | 80.72 | 0.02 | 0.02 | 1 | 1 |  |  |
| 13 | 82.82 | 88.24 | -5.42 | 5.42 | 5 | -5 |  |  |
| 14 | 86.47 | 98.15 | -11.68 | 11.68 | 17 | -17 | The min of the 2 sums | -88 |
| 15 | 87.69 | 80.72 | 6.97 | 6.97 | 9 | 9 |  |
| 16 | 88.85 | 80.72 | 8.13 | 8.13 | 13 | 13 |  |  |
| 17 | 82.82 | 93.6 | -10.78 | 10.78 | 18 | -18 | **Statistically significant** | -88<56 |
| 18 | 84.86 | 50.54 | 34.32 | 34.32 | 31 | 31 |  |
| 19 | 87.27 | 0 | 87.27 | 87.27 | 35 | 35 |  |  |
| 20 | 96.96 | 37.89 | 59.07 | 59.07 | 35 | 35 |  |  |

## Wilcoxon signed-rank test comparing the accuracy of GP and MLP

## Wilcoxon signed-rank test comparing the accuracy of GP and MLP

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **run** | **Acc. for GP** | **Acc. for MLP** | **Difference between MLP & GP Acc's** | **Unsigned difference** | **rank of diff.** | **Signed rank** |  |  |
| 1 | 97.99 | 90.11 | -7.88 | 7.88 | 7 | -7 | The sum of all the negative signed ranks | **w-** |
| 2 | 56.78 | 87.83 | 31.05 | 31.05 | 18 | 18 | -67 |
| 3 | 67.34 | 82.13 | 14.79 | 14.79 | 12 | 12 |  |
| 4 | 78.33 | 88.21 | 9.88 | 9.88 | 8 | 8 |  |
| 5 | 83.65 | 94.3 | 10.65 | 10.65 | 10 | 10 |  |  |
| 6 | 98.48 | 92.4 | -6.08 | 6.08 | 5 | -5 |  |  |
| 7 | 97.72 | 81.74 | -15.98 | 15.98 | 14 | -14 | The sum of all the positive signed ranks | **w+** |
| 8 | 83.65 | 84.03 | 0.38 | 0.38 | 1 | 1 | 143 |
| 9 | 98.48 | 83.65 | -14.83 | 14.83 | 13 | -13 |  |
| 10 | 58.17 | 85.55 | 27.38 | 27.38 | 17 | 17 |  |
| 11 | 66.16 | 88.21 | 22.05 | 22.05 | 16 | 16 |  |  |
| 12 | 83.65 | 81.36 | -2.29 | 2.29 | 2 | -2 |  |  |
| 13 | 89.95 | 83.27 | -6.68 | 6.68 | 6 | -6 |  |  |
| 14 | 98.1 | 86.69 | -11.41 | 11.41 | 11 | -11 | The min of the 2 sums | -67 |
| 15 | 83.65 | 87.83 | 4.18 | 4.18 | 3 | 3 |  |
| 16 | 83.65 | 88.97 | 5.32 | 5.32 | 4 | 4 |  |  |
| 17 | 93.62 | 83.27 | -10.35 | 10.35 | 9 | -9 | **Statistically significant** | -67<52 |
| 18 | 65.02 | 85.17 | 20.15 | 20.15 | 15 | 15 |  |
| 19 | 49.43 | 87.45 | 38.02 | 38.02 | 19 | 19 |  |  |
| 20 | 38.58 | 96.96 | 58.38 | 58.38 | 20 | 20 |  |  |

To assess whether the difference in model performance between GP and MLP is statistically significant, a Wilcoxon Signed-Rank Test was conducted. The test yielded a W⁺ value of 141 and a W⁻ value of -69 when comparing the F-1scores and yielded a W⁺ value of 143 and a W⁻ value of -67 when comparing the model’s accuracies. Since the test indicated a significant difference between the model’s performance, we reject the null hypothesis that there is no median difference between the two models' performance. The results suggest that GP consistently outperformed MLP across the folds, validating the practical observations made during the cross-validation process.

## Conclusion

This assignment explored the application of three machine learning models, Genetic Programming (GP), Multi-Layer Perceptron (MLP), and Decision Tree, to address the problem of financial stock classification. The goal was to predict whether a financial stock should be purchased based on historical data. Each model was implemented and evaluated using standardized training and test datasets.

The Genetic Programming model was developed from first principles, the MLP utilized existing Python libraries to leverage deep learning capabilities, and the Decision Tree was implemented using the Weka framework. Model performance was assessed using metrics such as accuracy and F1-score, and comparative evaluation was carried out using the Wilcoxon signed-rank test to determine statistical significance between GP and MLP results. The results indicated that MLP was the best of the 3 models for financial forecasting.

Overall, the assignment demonstrated the practical application of machine learning techniques in financial forecasting and highlighted the strengths and trade-offs of different model architectures in solving classification problems.

**BIBLIOGRAPHY**

Alhijawi, B. and Awajan, A., 04. Genetic algorithms: Theory, genetic operators, solutions, and applications. Evolutionary Intelligence, 17(), pp.14-1.

Chen, Q. and Xue, B., 0. Generalisation in genetic programming for symbolic regression: challenges and future directions. In Women in Computational Intelligence: Key Advances and Perspectives on Emerging Topics (pp. 81-0). Cham: Springer International Publishing.

Chiang, J.K. and Chi, R., 0, May. Comparison of Decision Tree J48 and CART in Liver Cancer Symptom with CARNEGIE-MELLON UNIVERSITY Data. In 0 IEEE 4th Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS) (pp. 8-1). IEEE.

Das, P., Sachindra, D.A. and Chanda, K., 0. Machine learning-based rainfall forecasting with multiple non-linear feature selection algorithms. Water Resources Management, (1), pp.04-071.

Diallo, R., Edalo, C. and Awe, O.O., 04. Machine Learning Evaluation of Imbalanced Health Data: A Comparative Analysis of Balanced Accuracy, MCC, and F1 Score. In Practical Statistical Learning and Data Science Methods: Case Studies from LISA 00 Global Network, USA (pp. 8-1). Cham: Springer Nature Switzerland.

Hurtik, P., Tomasiello, S., Hula, J. and Hynar, D., 0. Binary cross-entropy with dynamical clipping. Neural Computing and Applications, 4(14), pp.109-1041.

Li, X., Wu, Y., Wei, M., Guo, Y., Yu, Z., Wang, H., Li, Z. and Fan, H., 01. A novel index of functional connectivity: phase lag based on Wilcoxon signed rank test. Cognitive Neurodynamics, 1, pp.1-.

Mei, Y., Chen, Q., Lensen, A., Xue, B. and Zhang, M., 0. Explainable artificial intelligence by genetic programming: A survey. IEEE Transactions on Evolutionary Computation, 7(), pp.1-41.

Mienye, I.D. and Jere, N., 04. A survey of decision trees: Concepts, algorithms, and applications. IEEE access.

Narkhede, M.V., Bartakke, P.P. and Sutaone, M.S., 0. A review on weight initialization strategies for neural networks. Artificial intelligence review, (1), pp.91-.

Naskath, J., Sivakamasundari, G. and Begum, A.A.S., 0. A study on different deep learning algorithms used in deep neural nets: MLP SOM and DBN. Wireless personal communications, 18(4), pp.91-9.

Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., Grave, E., Izacard, G., Joulin, A., Synnaeve, G., Verbeek, J. and Jégou, H., 0. Resmlp: Feedforward networks for image classification with data-efficient training. IEEE transactions on pattern analysis and machine intelligence, 4(4), pp.14-1.

Villavicencio, C.N., Macrohon, J.J.E., Inbaraj, X.A., Jeng, J.H. and Hsieh, J.G., 01. Covid-19 prediction applying supervised machine learning algorithms with comparative analysis using weka. Algorithms, 14(7), p.01.