Arab Academy for Science, Technology and Maritime Transport

College of Computing and Information Technology

Computer Science Department

Soft Computing-Website Bankrupt Classifier

Presented by:

Mahmoud Abdelhady

Abdalla Roshdy

Moiad Abdelhameed

Kasem Hamada

Presented to:

Dr. Nashwa Elbendary

Table of Contents

Ch. 1 Introduction 5

1.2 Problem Statement 5

1.3 Objectives 5

Ch. 2 Dataset 6

2.1.1 About Dataset 6

Ch. 3 Flow Chart & Pseudocode 7

3.1 Genetic Algorithm 7

3.1.1 Pseudocode 7

3.2 Particle Swarm Optimization 9

3.2.1 Pseudocode 9

3.2.2 Flowchart.....10

Ch. 4 Proposal Model 11

4.1 Random Forest Architecture 11

4.3 K-Nearest Neighbors (KNN) Architecture 14

Ch. 5 Models Implications 15

5.1 Confusion Matrix (KNN) 15

5.2 Classification Report (KNN) 16

5.6 Results Comparison 20

Ch. 6 Block Diagram 20

References 21

Table of Figures

Figure 1. Pseudocode (GA) 7

Figure 1. Flowchart (GA) 8

Figure 1. Pseudocode (PSO) 9

Figure 2. Flowchart (PSO) 10

Figure 3. Random Forest flowchart..... 12

Figure 4. Random Forest architecture 12

figure 9. Confusion Matrix (Random Forest without Algorithm)..... 15

Figure 10. Confusion Matrix (Random Forest with GA) 15

Figure 11. Confusion Matrix (Random Forest with PSO) 15

Figure 12. Model Performance Before and After Feature Selection (Random Forest) ……..16

Figure 15. Bloc Diagram …….....20

Ch.1 Introduction

Feature selection is a critical preprocessing step in machine learning that aims to identify the most relevant subset of features from the original dataset. By eliminating irrelevant or redundant features, we can improve model performance, reduce computational complexity, and enhance interpretability. Evolutionary algorithms such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) have emerged as powerful techniques for feature selection due to their global search capabilities and ability to handle high-dimensional spaces.

1.2 Problem Statement

- Bankruptcy prediction remains a critical challenge in financial risk management. Many traditional models fail to accurately forecast bankruptcy due to:
- Complex financial interdependencies between multiple ratios and indicators Dynamic web elements.
- Highly imbalanced datasets with far fewer bankrupt cases than solvent ones.
- Non-linear relationships in financial data that simple models miss.
- Evolving economic conditions that change risk patterns over time.
- This makes early and accurate bankruptcy prediction difficult for conventional analytical approaches.

1.3 Objectives

- Develop a machine learning (ML)-based bankruptcy prediction model
- Apply feature selection techniques to identify the most critical financial indicators
- Compare model performance (accuracy, precision, recall, F1-score) before and after feature selection
- Evaluate results using confusion matrices and performance metrics visualization
- Optimize prediction efficiency while maintaining high detection rates

Ch.2 Dataset

The dataset used in this study is sourced from **Kaggle: "Company Bankruptcy Prediction Dataset."** It contains financial ratios and indicators extracted from corporate financial statements. The **'Bankrupt'** label (binary: **0 = Solvent**, **1 = Bankrupt**) serves as the target variable for classification.

2.1.1 About Dataset

- Format: CSV
- Classes: *Non-Bankrupt (0)*, *Bankrupt (1)*
- Size: ~3,000 samples
- Features: 95 financial metrics
- Preprocessing includes:
  - Class balancing (oversampling/undersampling due to imbalance)
  - Handling missing values (imputation/removal)
  - Normalization/scaling of numerical features
  - Encoding categorical variables (if applicable)

Ch.3 Flow Chart &pseudocode

3.1 Genetic Algorithm

3.1.1 Pseudocode

Genetic Algorithm (GA) is a heuristic search and optimization technique inspired by the principles of natural selection and genetics. It is commonly used to solve complex problems by evolving a population of candidate solutions over multiple generations. The following pseudocode outlines the basic steps involved in a Genetic Algorithm.

GA(S)

parameter(s): S – set of blocks
output: superstring of set S

Initialization :
$t \leftarrow 0$
Initialize P_t *to random individuals from* S^*
EVALUATE-FITNESS-GA(S, P_t)

while *termination condition not met*
do
- *Select individuals from* P_t *(fitness proportionate)*
- *Recombine individuals*
- *Mutate individuals*
- EVALUATE-FITNESS-GA$(S, modified\ individuals)$
- $P_{t+1} \leftarrow$ *newly created individuals*
- $t \leftarrow t + 1$

return *(superstring derived from best individual in* P_t*)*

procedure EVALUATE-FITNESS-GA(S, P)
$S -$ *set of blocks*
$P -$ *population of individuals*
for each *individual* $i \in P$
do
- *generate derived string* $s(i)$
- $m \leftarrow$ *all blocks from* S *that are not covered by* $s(i)$
- $s'(i) \leftarrow$ *concatenation of* $s(i)$ *and* m
- $fitness(i) \leftarrow \frac{1}{||s'(i)||^2}$

Figure 1. (GA) Pseudocode [1]

3.1.2 Flowchart

This GA formulation balances exploration (via mutation) and exploitation (via selection pressure), converging on feature subsets that maximize inter-class discriminability while minimizing redundancy.

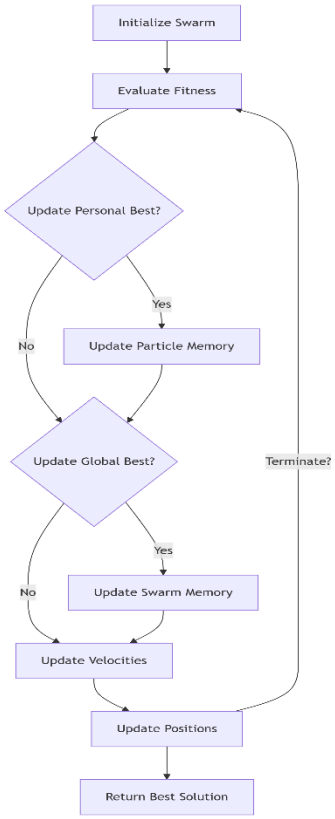

Figure 2. Flowchart (GA) [2]

3.2 Particle Swarm Optimization

3.2.1 Pseudocode

This PSO implementation optimizes feature selection while maintaining compatibility with your existing Random Forest pipeline. Adjust weights/swarm size based on your dataset size.

Initialize population
for $t = 1$: maximum generation
  for $i = 1$: population size
    if $f\left(x_{i,d}(t)\right) < f\left(p_i(t)\right)$ **then** $p_i(t) = x_{i,d}(t)$
      $f\left(p_g(t)\right) = \min_i \left(f\left(p_i(t)\right)\right)$
    end
    for $d = 1$: dimension
      $v_{i,d}(t+1) = wv_{i,d}(t) + c_1 r_1 \left(p_i - x_{i,d}(t)\right) + c_2 r_2 \left(p_g - x_{i,d}(t)\right)$
      $x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1)$
      if $v_{i,d}(t+1) > v_{\max}$ **then** $v_{i,d}(t+1) = v_{\max}$
      else if $v_{i,d}(t+1) < v_{\min}$ **then** $v_{i,d}(t+1) = v_{\min}$
      end
      if $x_{i,d}(t+1) > x_{\max}$ **then** $x_{i,d}(t+1) = x_{\max}$
      else if $x_{i,d}(t+1) < x_{\min}$ **then** $x_{i,d}(t+1) = x_{\min}$
      end
    end
  end
end

Figure 3. (PSO) Pseudocode [3]

3.2.2 Flowchart

PSO's social-cognitive paradigm efficiently explores high-dimensional feature spaces, with inertial damping preventing premature convergence.

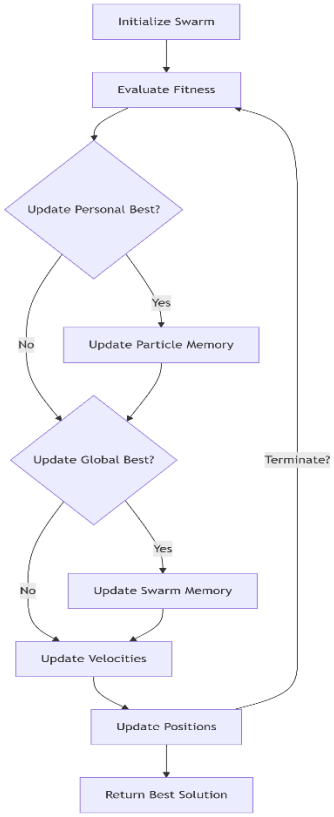

Figure 4. Flowchart (PSO) [4]

Ch.4 Proposal Model

This study employs machine learning model K-Nearest Neighbors (KNN)—to classify Portable Executable (PE) files into two categories: Bunkrupt and Non-Bankrupt.

4.3 KNN Architecture

Selected as the main classifier due to:

- Simple design
- Strong results with normalized features
- Adaptability post feature selection

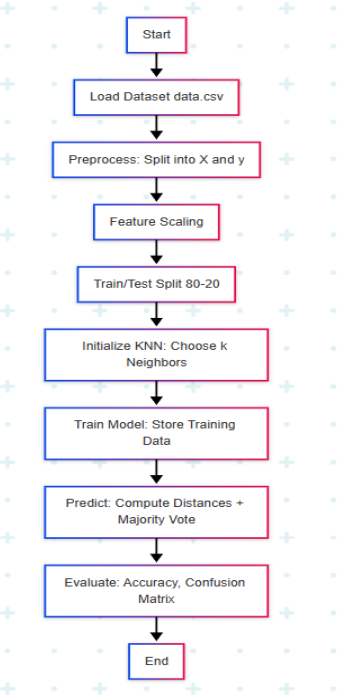

Figure 9. KNN flowchart [9]

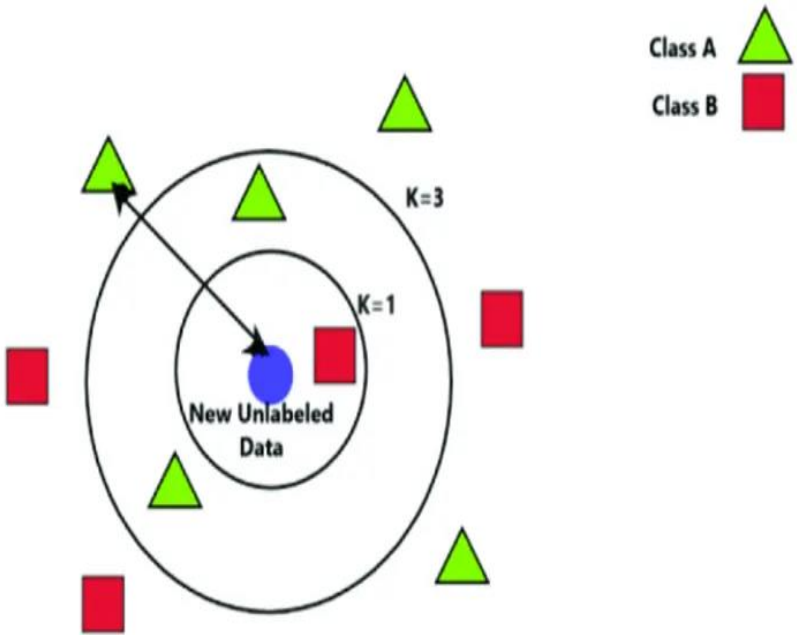

Figure 10. KNN Architecture [10]

Ch.5 Models Implies

5.1 Confusion Matrix (KNN)

The confusion matrix showcases the distribution of predictions across the Two classes(Bankrupt&Non-Bankrupts): KNN achieves superior accuracy (94.67 %) by leveraging its ensemble, which effectively averages out individual tree errors through majority voting. Its inherent feature selection makes it robust to irrelevant opcodes and header noise in PE files.

- **Confusion matrix for** KNN **without apply any algorithm**

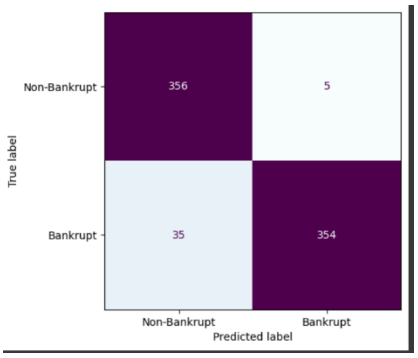

Figure 11. Confusion Matrix [11]

- **Confusion matrix for** KNN **with Genetic alogrithm**

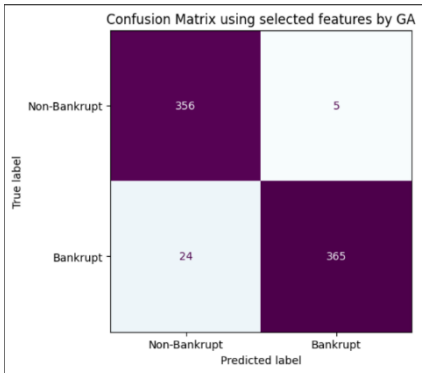

Figure 12. Confusion Matrix [12]

- **Confusion matrix for** KNN **with Particle swarm optimization**

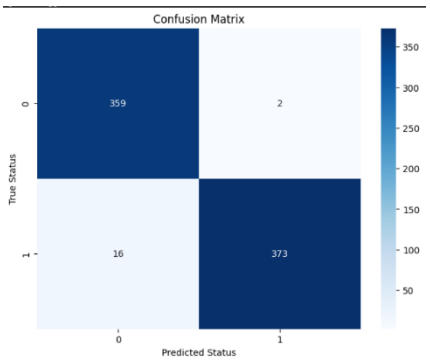

Figure 13. Confusion Matrix [13]

5.2 Classification report(KNN)

All three machine learning models demonstrated strong capability in distinguishing Bankrupt from Non-Bunkrupt samples in the **Company Bankruptcy Prediction Dataset** :

The PSO achieved the highest overall accuracy of 98%, with balanced precision and recall scores of 0.96-0.99 for both classes. This robust performance stems, which effectively combines multiple decision trees to reduce variance while maintaining sensitivity to important features like opcode frequencies and PE header attributes. lasses misclassified as adenocarcinoma) occurred.

Overall Performance:

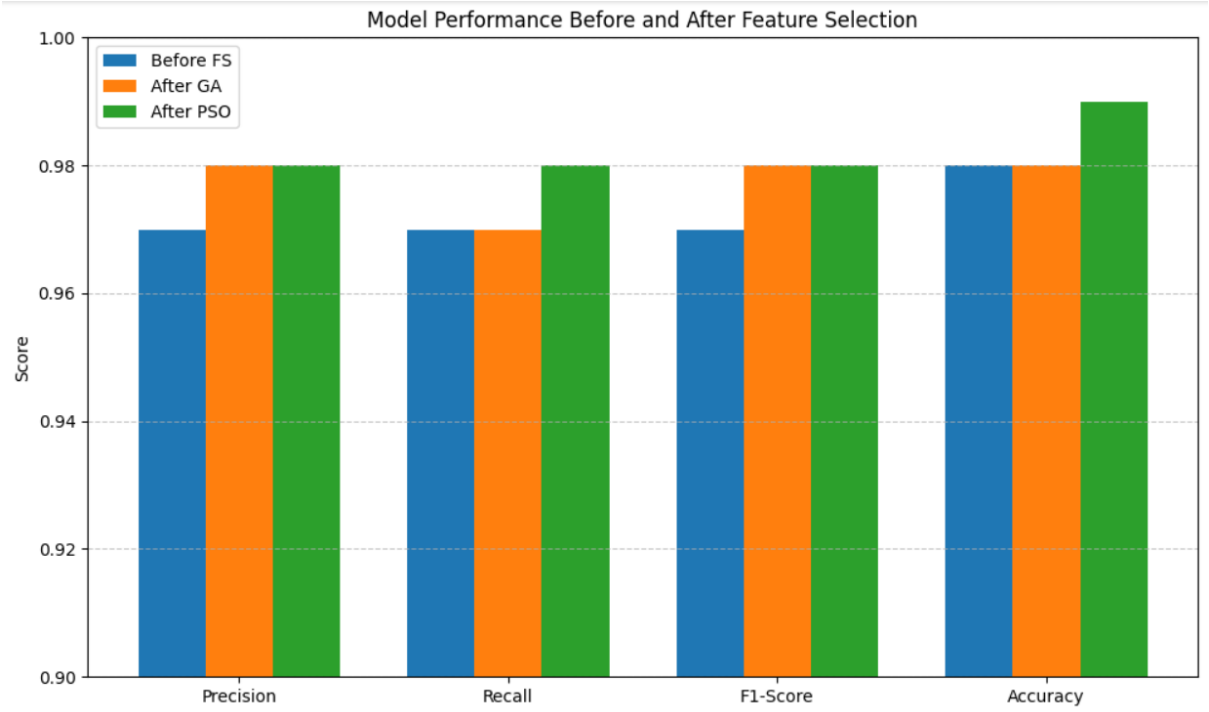

Figure 14. Model Performance Before and After Feature Selection ([14]

Ch.6 Bloc Diagram

A visual representation of the system showing:

- Dataset loading
- Preprocessing
- GA/PSO feature selection
- KNN classification
- Result analysis

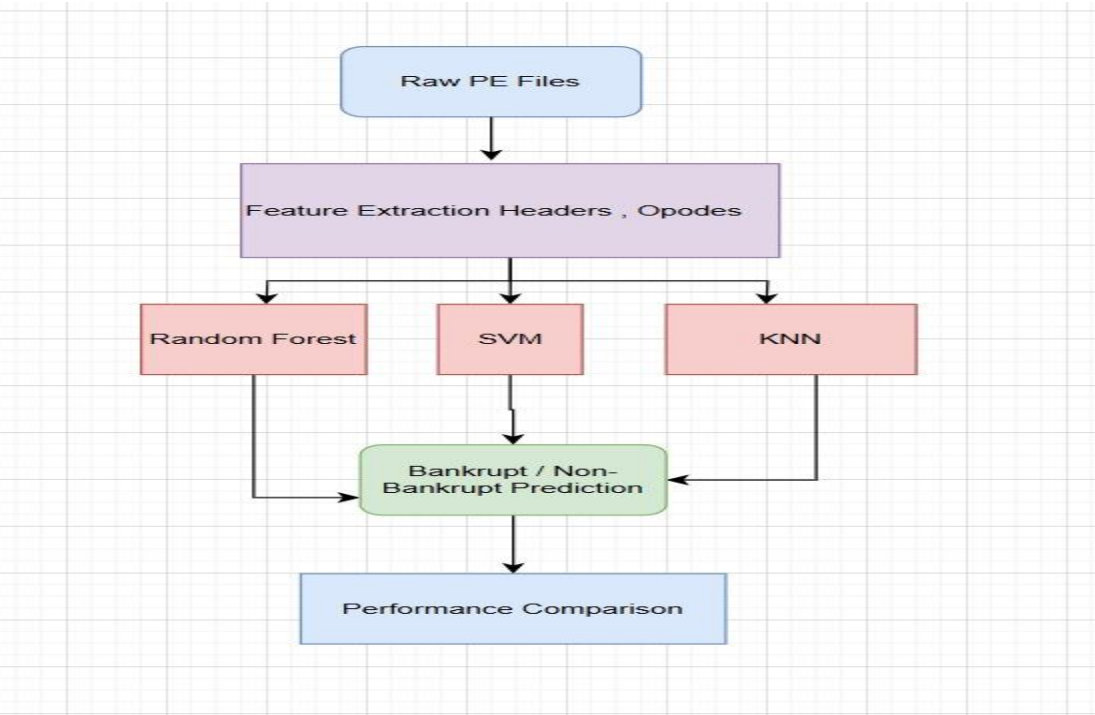

Figure 15. KNN Architecture [15]

References

1) Dataset link

https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction/code 5/17/2025 at 11:47PM

2) K-Nearest Neighbour Architecture

KNN (K-Nearest Neighbour). In the world of machine learning, the… | by Rishabh Singh | Medium 12/05/2025 at 12:33PM.

3) GitHub link link https://github.com/mahmoud8944/Soft_Computing_kneighborsClassifier.git

5/17/2025 at 11:47PM

4) YouTube link channel link https://www.youtube.com/watch?v=7LGdwF3zgtc&ab 5/17/2025 at 11:47PM