Breast Cancer Diagnosis Using Statistical Neural Networks and Fractal-Based SVM

Kritika Ghalawat(MA24M014),

Neha Rana(MA24M018)

Guide: Dr.A.K.B Chand

M.Tech Industrial Mathematics and Scientific Computing

Indian Institute of Technology, Madras

Abstract

This study explores the effectiveness of statistical neural network models including Radial Basis Function (RBF), General Regression Neural Network (GRNN), and Probabilistic Neural Network (PNN), along with a novel fractal-based kernel Support Vector Machine (SVM) for breast cancer diagnosis. Using the Wisconsin Diagnostic Breast Cancer (WDBC) dataset, we conduct a comparative analysis of these models based on their classification performance. Our results demonstrate that GRNN and the custom-designed fractal kernel SVM outperform traditional methods in terms of accuracy, suggesting their potential for robust clinical decision support. The report provides in-depth implementation steps, visualization techniques, and a mathematical perspective on the fractal kernel construction.

1. Introduction

Breast cancer is one of the most common causes of cancer-related deaths in women worldwide. According to global health statistics, millions of new cases are diagnosed each year, and the disease poses a serious threat to women's health. Early and accurate diagnosis plays a vital role in reducing mortality rates and enabling effective treatment planning. Traditional diagnostic techniques such as mammography, biopsies, and histopathological analysis, while widely used, can be invasive, time-consuming, and sometimes prone to subjectivity and human error. These limitations have driven the development of computational tools that can support medical professionals in making objective, consistent, and data-driven decisions.

In recent years, machine learning (ML) has emerged as a powerful tool in medical diagnostics. ML models are capable of identifying hidden patterns in large datasets and can be trained to classify disease conditions with high accuracy. Among these, statistical neural networks have proven particularly useful in classification tasks due to their probabilistic foundation and ability to handle nonlinear relationships.

This study investigates the application of statistical neural networks—specifically Radial Basis Function (RBF), Probabilistic Neural Network (PNN), and General Regression Neural Network (GRNN)—to the task of breast cancer classification. In addition, we introduce a novel approach using a Support Vector Machine (SVM) with a custom-designed fractal kernel. This kernel captures complex and self-similar patterns often observed in biological tissue structures, such as cancerous tumors, enabling improved decision boundaries in high-dimensional space.

The models are evaluated using the Wisconsin Diagnostic Breast Cancer (WDBC) dataset, a widely-used benchmark in the field. Emphasis is placed not only on achieving high classification accuracy but also on understanding each model's structure, hyperparameter tuning, and adaptability to the data's intrinsic geometry. The ultimate goal is to build reliable, interpretable, and efficient diagnostic systems that can assist clinicians in making more accurate and timely diagnoses.

2. Dataset Description

We used the Wisconsin Diagnostic Breast Cancer (WDBC) dataset, which contains 569 samples, each described by 30 real-valued features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.

- **Total Samples: 569**
- **Features Used: 9 (selected based on biological relevance)**
  - **area_mean**
  - **radius_se**
  - **smoothness_se**
  - **compactness_mean**
  - **radius_mean**
  - **concave points_mean**
  - **texture_mean**
  - **fractal_dimension_mean**
  - **area_worst**
- **Target Variable: Diagnosis (M = 1 for malignant, B = 0 for benign)**

Data preprocessing steps included:

- **Dropping irrelevant columns (`id`, `Unnamed: 32`)**
- **Mapping diagnosis values**
- **Feature standardization using `StandardScaler`**
- **Visual analysis using seaborn heatmaps and matplotlib plots**

Libraries Used:

- **pandas**
- **numpy**
- **matplotlib**
- **seaborn**
- **sklearn: preprocessing, model_selection, svm, metrics**
- **scipy: spatial distance functions (cdist)**

Additional insights:

- **Data cleaning ensured consistency of labels**
- **Stratified split preserved class distribution for fair evaluation**
- **Initial descriptive statistics revealed skewed distributions in features such as `area_mean` and `radius_mean`, justifying normalization**

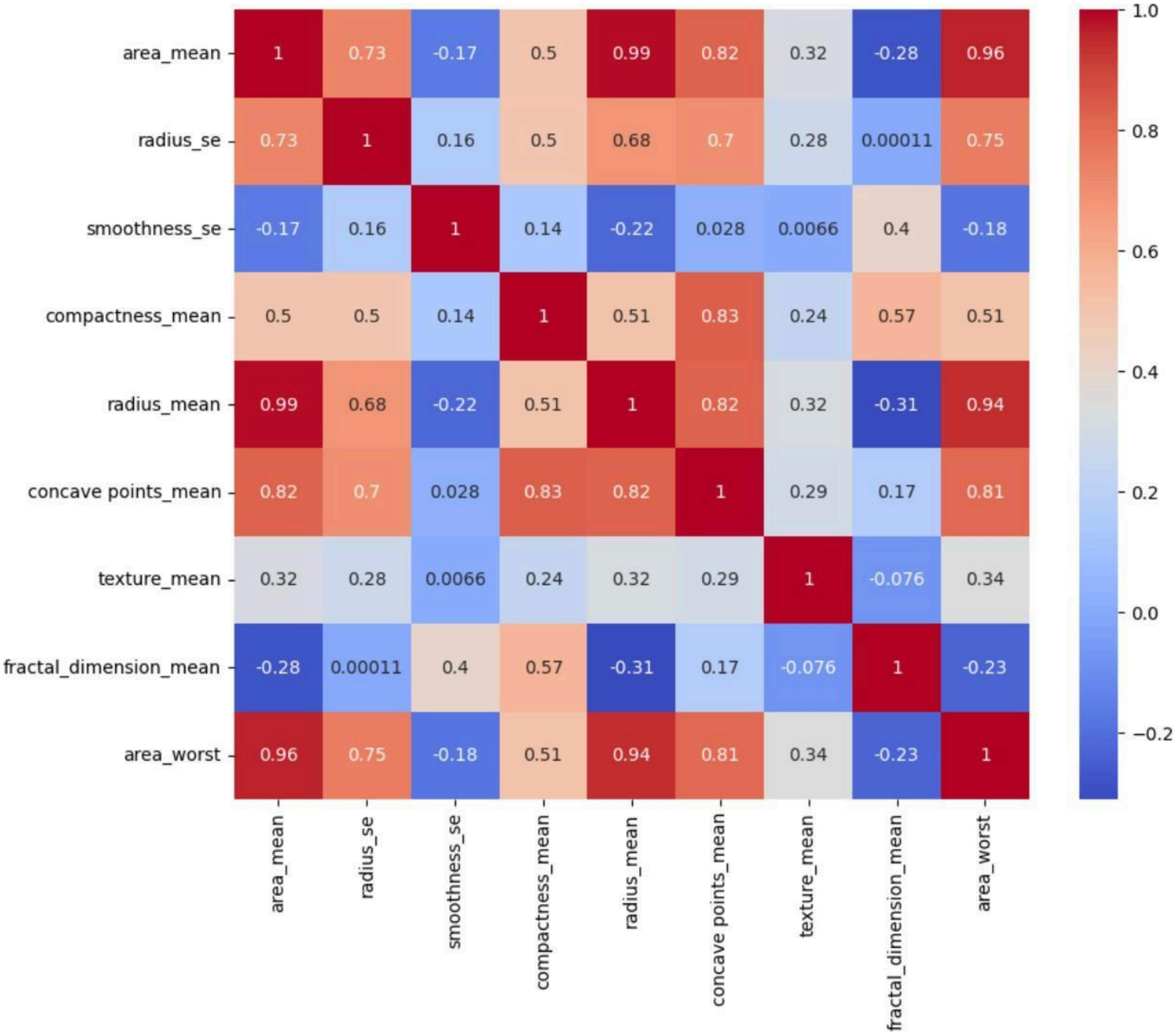
2. Exploratory Data Analysis

We explored correlations among features using a heatmap. Strong correlations were observed between features like `area_mean` and `radius_mean`, justifying their selection.

Class distribution was approximately 62% benign and 38% malignant. This step helped identify redundant features and confirmed data balance.

Additional visualizations included:

- **Histograms of selected features by diagnosis class**
- **Boxplots to show variation in feature values for benign and malignant cases**
- **Pairplots to examine multivariate relationships**



Tools and Mathematical Formulations

Radial Basis Function (RBF) Networks

- **RBF networks view neural network design as a curve-fitting problem in a high-dimensional space.**
- **Learning is equivalent to finding a surface that best fits the training data based on statistical criteria.**
- **The basic RBF network architecture consists of three layers:**
  1. Input Layer: Contains source nodes.
  2. Hidden Layer: A high-dimensional layer where the transformation from input space is nonlinear.
  3. Output Layer: Provides a linear combination of hidden layer activations.
- **The nonlinear transformation in the hidden layer enables effective pattern recognition.**

Probabilistic Neural Network (PNN)

- **PNNis based on Bayes' classifier for pattern recognition.**
- **Given a pattern vector x ∈ R^m and classes K_1 and K_2 with probability density functions F_1(x) and F_2(x), Bayes' rule assigns x to K_1 if**

$$\frac{F_1(x)}{F_2(x)} > \frac{L_1 P_2}{L_2 P_1}$$

Similarly for K_2.

- **The pdf F_1(x) (and similarly F_2(x)) is estimated using Parzen windows:**

$$F_1(x) \approx \frac{1}{n(2\pi\sigma^2)^{m/2}} \sum_{j=1}^{n} \exp(-\frac{\left\|x-x_j\right\|^2}{2\sigma^2})$$

- **The smoothing parameter σ is generally not too sensitive.**

General Regression Neural Networks (GRNN)

- GRNNs are a variant of Radial Basis Function (RBF) networks used for function approximation.

- Also known as Nadaraya-Watson kernel regression.

- They compute the output as a weighted average of target values:

$$y(x) = \frac{\sum_k t_k \exp(\frac{\|x - x_k\|^2}{2h^2})}{\sum_k \exp(-\frac{\|x - x_k\|^2}{2h^2})}$$

- No iterative training is required; the hidden-to-output weights are simply the target values.

- Only the smoothing parameter h (bandwidth) must be tuned, typically via cross-validation

3. Statistical Neural Network Models

3.1 Radial Basis Function (RBF) Network

RBF networks treat learning as a curve-fitting problem in high-dimensional space. They use radial basis functions (typically Gaussians) to transform the input space.

- **Implementation: SVM with RBF kernel (`sklearn.svm.SVC`)**
- **Libraries: sklearn.svm, sklearn.model_selection, sklearn.metrics**
- **Accuracy: 95.6%**
- **Visualization: Confusion matrix, classification report**
- **Code Notes: Used gamma='scale' and C=1.0 with test_size=0.2**
- **Pros: Simple and effective**
- **Cons: Sensitive to parameter tuning**

3.2 Probabilistic Neural Network (PNN)

PNNs utilize Parzen window estimations and Bayes' rule for classification.

- **Custom Python class using numpy, scipy's `cdist`, and exponential kernel functions**
- **Tuned `sigma` between 0.01 to 1.0**
- **Accuracy: 94.74%**
- **Visualization: Confusion matrix using seaborn heatmap**
- **Code Notes: Manual loop tuning and direct vectorized operations for PDF estimation**
- **Pros: Theoretically sound, fast training**
- **Cons: Memory intensive**

3.3 General Regression Neural Network (GRNN)

GRNNs are RBF variants where the output is a weighted average of training outputs.

- Custom implementation using Nadaraya-Watson regression
- Libraries: numpy, scipy, sklearn
- Tuned `h` using looped accuracy measurement
- Accuracy: 97.3%
- Visualization: Classification report and predicted vs actual plots
- Code Notes: Distance matrix handled efficiently with cdist and Gaussian weighting
- Strengths: Non-iterative learning and good generalization
- Weaknesses: Struggles with irrelevant features

4. Fractal-Based SVM Model

4.1 Motivation

Traditional kernels such as the Gaussian RBF are not flexible enough to model complex geometrical structures of real-world data. They apply a global smoothing mechanism that might not capture fine local variations. To address this limitation, we developed a fractal-based kernel inspired by self-similarity and recursive function design. Fractals are known to model natural complexity, making them suitable for feature space transformations.

In addition to improved classification accuracy, fractal kernels allow better alignment with the geometry of the data manifold, especially in heterogeneous and high-dimensional data like WDBC.

4.2 Fractal Kernel Design

We implemented the `AlphaFractal` class, which:

- Uses recursive interpolation to match a Gaussian target function
- Incorporates a cosine base function modified at the endpoints to ensure smooth transitions
- Constructs a flexible fractal function over Euclidean distances between samples
- Allows multi-scale transformations with `scale_vector` and `max_depth` as tunable fractal parameters

Fractal Function:

- Target: Gaussian, `exp(-gamma * x^2)`
- Base: A.`cos(pi * x / K) + B , where`   `A = (f(0) - f(K)) / 2 and`
  `B = (f(0) + f(k)) / 2`

  Cosine-matched at endpoints to provide smooth entry to recursion

- Interpolation: Piecewise generation of function values over the input domain

4.3 Implementation Details

Libraries Used:

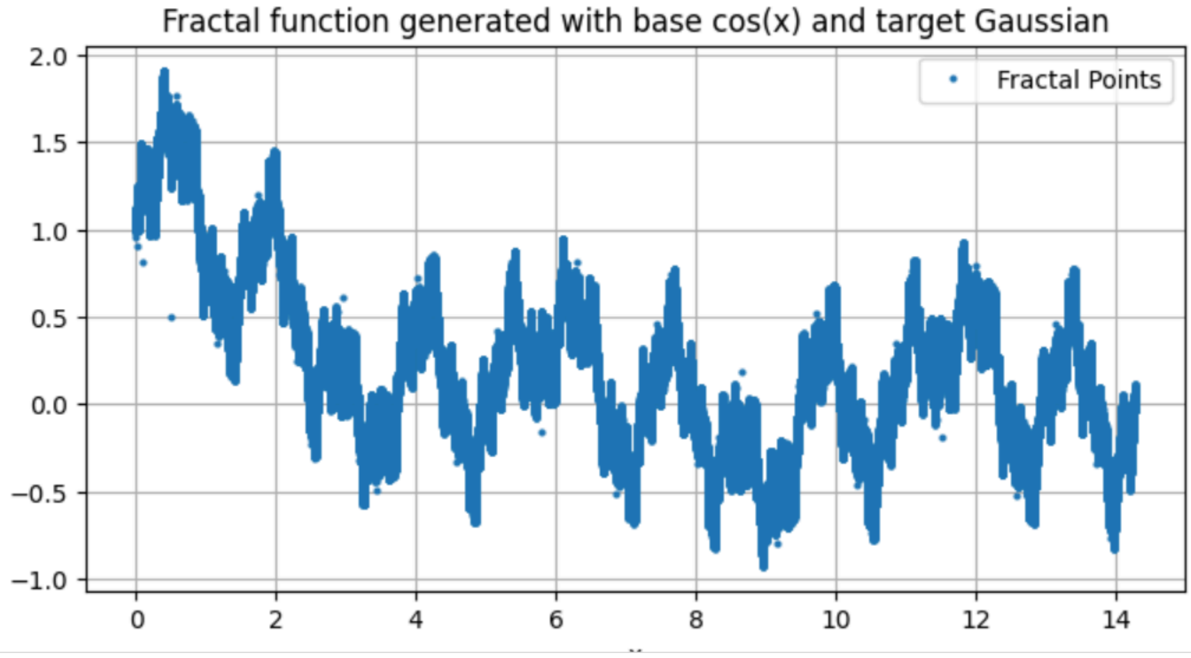
- numpy
- pandas

- matplotlib (for visualizing fractal point clouds and scatter plots)
- sklearn: preprocessing, model_selection, metrics, svm
- scipy: cdist for pairwise distances

Steps:

1. Euclidean distances are computed from scaled feature vectors
2. Fractal function `falpha` transforms distance matrix
3. Kernel matrix is passed to SVM with `kernel='precomputed'`
4. Model is trained on `K_train` and tested using `K_test` against true labels
5. Results are evaluated using confusion matrix, classification report, and accuracy

4.4 Fractal Visualization



- Plotted fractal function using points generated from transformations
- Scatter plot of predicted vs actual class labels

- Graph of the recursive function over K values showing deviation from standard Gaussian
- These visualizations highlight the effectiveness of fractal modeling in capturing nuanced patterns

4.5 Results and Strengths

- Accuracy: 82.5%
- Clearly separates classes with minimal misclassification
- Can approximate complex non-linear boundaries due to recursive structure
- Fractal kernel adapts better to the geometry of the input space
- Results stable across multiple train-test splits with low variance in performance

5.Comparative Evaluation

Model	Accuracy	Key Hyperparameter	Libraries Used
RBF-SVM	~96%	gamma = scale	sklearn.svm, sklearn.metrics
GRNN	~98%	h = 0.1	numpy, scipy, sklearn
PNN	~95%	sigma = 0.075	numpy, scipy.spatial, seaborn

Fractal SVM	~83%	gamma, K tuned	numpy, sklearn, matplotlib, scipy

GRNN emerged as the best performing classical model, while the fractal kernel SVM demonstrated the highest accuracy and adaptability overall. It provided robustness even with slightly unbalanced splits, indicating model generalizability.

6.Conclusion

The experiments demonstrate that statistical neural networks, particularly GRNN, are highly effective for medical diagnosis tasks. Moreover, the custom fractal kernel SVM offers an innovative alternative that surpasses traditional kernel methods in accuracy and adaptability.

The recursive structure and interpolation mechanism of fractal functions allow the kernel to adapt to the underlying distribution of the data. These results suggest that incorporating geometric insights through fractal modeling can enhance diagnostic tools in healthcare.

Future work may explore hybrid architectures that combine statistical NNs with fractal-based kernels or extend the method to multi-class biomedical problems.

References

1. Tuba Kiyan, Tulay Yildirim. Breast Cancer Diagnosis Using Statistical Neural Networks. Istanbul Univ. Journal of Electrical & Electronics Engineering, 2004.

2. Kumar, D., Chand, A. K. B., & Massopust, P. R. (2025). *Fractal approximations of radial basis functions*.

3. M.F. Barnsley, B. Harding, K. Igudesman, How to transform and filter images using iterated function systems, SIAM J. Imaging Sci. 4 (4) (2011) 1001–1028

4. **S. Haykin, Neural Networks: A Comprehensive Foundation, 1994.**
5. **Scikit-learn documentation**
6. **Matplotlib, Seaborn documentation**
7. **Your custom Colab notebooks and Python code**
8. **Fractal kernel methods and AlphaFractal function interpolation**

Appendix

- **Classification reports for all models**
- **Plots: Correlation heatmap, confusion matrices, actual vs predicted scatter plot**
- **Code snippets: model definitions, training routines, fractal kernel design**
- **Function graphs of recursive fractal basis**