NIR Spectral Analysis and Dimensionality Reduction for the Classification of Stress Treatments in Plants

Computer Graphics and Scientific Visualization

Ricardo Lopera V.

October 23, 2025

Abstract

This report presents a comprehensive study on the analysis of near-infrared (NIR) reflectance spectral data for the classification of plants subjected to different stress treatments. The work covers multiple stages: from initial data exploration and scientific visualization, through dimensionality reduction techniques, to the development and evaluation of machine learning models for multiclass classification. Principal Component Analysis (PCA) and t-SNE techniques were implemented to reduce the dimensionality of spectral data comprising over 2,150 wavelengths (350-2500 nm). Subsequently, four classification models were developed and optimized: K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Multilayer Perceptrons (MLP), and Bagging with decision trees. The results demonstrate that NIR spectroscopy techniques combined with machine learning methods can effectively detect different types of stress in plants, with important implications for precision agriculture and non-invasive monitoring of plant health.

Contents

1 Introduction

1.1 Context and Motivation

Plant health and their response to different types of stress (biotic and abiotic) are fundamental to modern agriculture and global food security. Traditional methods for diagnosing plant stress are destructive, slow, and require extensive laboratory analysis. Near-infrared (NIR) reflectance spectroscopy emerges as a non-invasive and rapid alternative that allows assessing the physiological state of plants by analyzing their spectral reflectance patterns.

The NIR spectral range (700-2500 nm) is particularly sensitive to the biochemical composition of plant tissues, including water content, proteins, carbohydrates, and other organic compounds that change in response to stress. However, NIR spectral data present significant challenges: high dimensionality (thousands of wavelengths), collinearity between adjacent variables, and complex non-linear relationships between reflectance and physiological states.

1.2 Objectives

The main objectives of this work are:

1. Develop scientific visualization methodologies for high-dimensional spectral data that facilitate the exploration and understanding of patterns in the data.

2. Apply and compare dimensionality reduction techniques (PCA and t-SNE) to transform complex spectral data into lower-dimensional representations that preserve relevant structure.

3. Implement and optimize supervised classification models to distinguish between different stress treatments based solely on spectral signatures.

4. Comparatively evaluate the performance of different model architectures and determine the best hyperparameter configurations through exhaustive search.

5. Identify the most informative spectral regions for discriminating between treatments through PCA loadings analysis.

1.3 Document Structure

This report is organized as follows: Section 2 describes the dataset and experimental treatments; Section 3 presents the scientific visualization techniques developed; Section 4 details the dimensionality reduction methods applied; Section 5 describes the classification models implemented; Section 6 presents the experimental results; and finally, Section 7 discusses the conclusions and future work.

2 Data Description

2.1 Dataset

The dataset used comes from NIR reflectance spectroscopy measurements taken on plants subjected to different stress conditions. The data were collected in the spectral range of 350 to 2500 nm with a spectral resolution of 1 nm, resulting in approximately 2,150 spectral variables per sample.

2.1.1 Data Structure

The data were organized into four sheets of an Excel file, each corresponding to different experimental sessions. Each record contains:

- **Plant identifier**: Unique code for each sample
- **Treatment applied**: Category of stress or combination of stresses
- **Reflectance values**: 2,150+ measurements corresponding to each wavelength in the 350-2500 nm range

2.2 Experimental Treatments

The plants were subjected to the following stress treatments:

1. **Control**: Plants without any stress, maintained in optimal conditions
2. **Water Stress (E_Hidrico)**: Plants subjected to water restriction
3. **Fusarium**: Plants inoculated with the pathogenic fungus *Fusarium* sp.
4. **Ralstonia**: Plants infected with the bacterium *Ralstonia solanacearum*
5. **Fusarium + Water Stress (Fus_EH)**: Combination of biotic stress (Fusarium) and abiotic stress (drought)
6. **Ralstonia + Water Stress (Ral_EH)**: Combination of biotic stress (Ralstonia) and abiotic stress (drought)
7. **Fusarium + Water Stress + Ralstonia (Fus_EH_Ral)**: Triple stress combining both pathogens and drought

2.3 Data Preprocessing

Data from the four experimental sheets were concatenated into a single DataFrame for analysis. Data integrity and the absence of missing values were verified. Since the spectral measurements were already normalized as reflectance values (0-1), no additional normalization was required at this initial stage.

For subsequent analyses, the data were divided into:

- **Feature matrix (X)**: All spectral columns (wavelengths)

- **Target vector (y)**: Treatment labels

The training-test split was performed using class stratification with 90% of the data for training and 10% for evaluation, ensuring proportional representation of each treatment in both sets.

3 Scientific Visualization of Spectral Data

3.1 Motivation and Challenges

Effective visualization of high-dimensional spectral data presents unique challenges. With over 2,000 wavelengths per sample and multiple treatments to compare, it is essential to develop graphical representations that are:

- **Informative**: Revealing patterns, trends, and differences between treatments
- **Interpretable**: Allowing identification of relevant spectral regions
- **Publishable**: Of sufficient quality for academic publications
- **Consistent**: With uniform and reusable styles

3.2 Development of Visualization Templates

A modular system of visualization functions with an academic style was developed, inspired by high-impact scientific publications. The main features include:

3.2.1 Custom Visual Style

A base style was configured using matplotlib and seaborn with the following specifications:

- **Typography**: LaTeX-style serif fonts (Computer Modern Roman) for consistency with academic documents
- **Color palette**: Vivid and distinct colors for maximum distinguishability between groups
- **Grids**: Visible grids with controlled transparency to facilitate value reading
- **Axes and frames**: Defined borders with removal of top and right spines for visual cleanliness

3.2.2 Visualization of Spectral Curves

The main function developed allows plotting the average spectral curves for each treatment in the full NIR range (350-2500 nm). Figure 1 shows an example of this visualization.

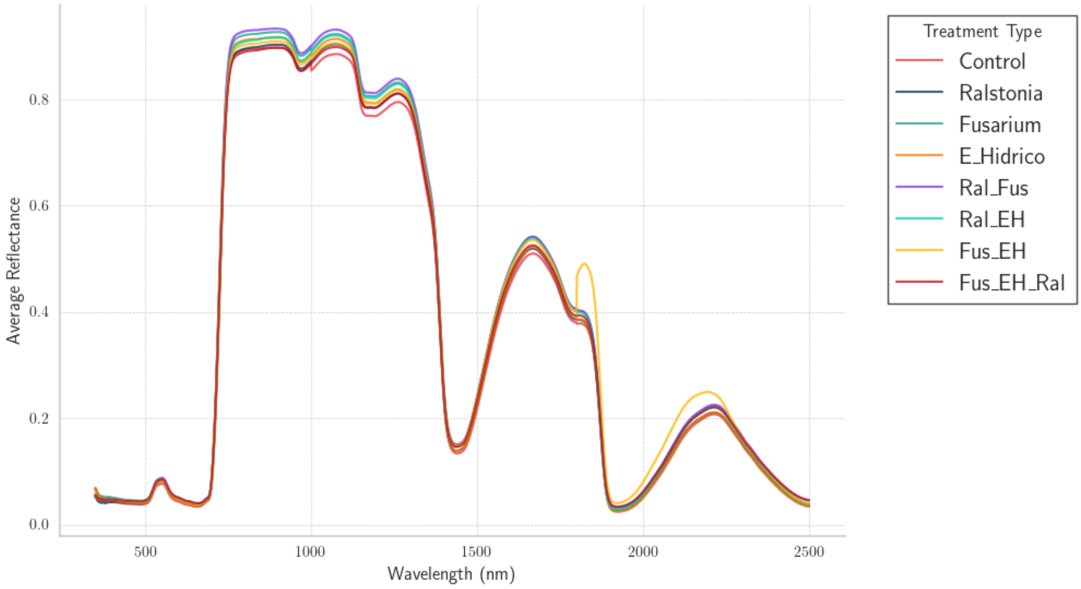


Figure 1: Average NIR spectral response (350-2500 nm) for each treatment.

Each line represents the average of all plants subjected to a specific treatment. Differences between curves indicate changes in reflectance associated with different types of stress.

3.3 Visualization with Zoom (Inset Plots)

To examine specific spectral regions of interest, zoom functionality was implemented using inset plots. This technique allows for:

- Visualizing the full spectrum in the main graph
- Showing a magnified region in an inserted box
- Visually connecting both regions with indicator lines

Figure 2 illustrates this technique applied to the 700-1370 nm NIR region, where notable differences between treatments are observed.

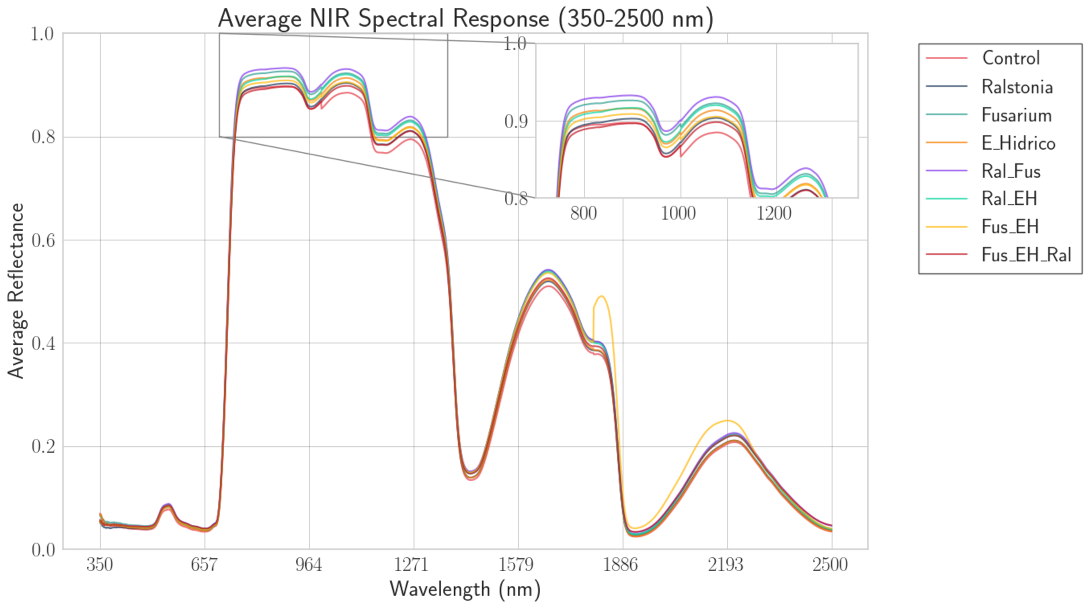


Figure 2: Spectral curves with zoom in the 700-1370 nm region.

The inserted box shows an enlargement of the region of interest, where differences between treatments are more pronounced. This region corresponds to the near-NIR, sensitive to water content and cellular structure.

3.4 Implemented Plot Types

A versatile function was developed that supports multiple types of visualization:

1. **Line plot (line)**: To visualize continuous trends in the spectrum
2. **Scatter plot (scatter)**: To emphasize individual data points
3. **Area plot (area)**: To visualize cumulative magnitudes
4. **Lollipop plot (lollipop)**: For discrete comparisons between wavelengths
5. **Step plot (step)**: To visualize discrete changes in reflectance
6. **Bar plot (bar)**: For direct comparisons between treatments at specific wavelengths

Each plot type is appropriate for different analytical purposes. The flexibility of the function allows selecting the most suitable type according to the message to be communicated.

3.5 Customization Parameters

The developed academic visualization function accepts 15+ configurable parameters:

- **Style and palette**: Full control over color schemes and seaborn styles
- **Dimensions**: Configurable figure size for different publication formats

- **Typography**: Independent font sizes for title, labels, and ticks
- **Annotations**: System for adding annotations with arrows and text boxes
- **Transparency**: Alpha control for overlays and shaded areas
- **Markers**: Customization of marker sizes and styles

This modularity allows for the generation of consistent, high-quality visualizations with minimal programming effort.

4 Dimensionality Reduction

4.1 Need for Dimensionality Reduction

NIR spectral data have two characteristics that complicate their direct analysis:

1. **High dimensionality**: With over 2,150 variables (wavelengths), the feature space is extremely large, resulting in the "curse of dimensionality" for many machine learning algorithms.

2. **Collinearity**: Adjacent wavelengths are highly correlated, introducing redundancy in the information.

Dimensionality reduction addresses these problems by transforming the data into a lower-dimensional space that retains most of the relevant information, facilitating visualization, exploratory analysis, and improving the performance of classification models.

4.2 Principal Component Analysis (PCA)

4.2.1 Theoretical Foundations

PCA is a linear dimensionality reduction technique that transforms the original variables into a new set of uncorrelated variables called principal components. Mathematically, PCA finds the directions of maximum variance in the high-dimensional space through eigenvalue decomposition of the covariance matrix:

$$\Sigma = \frac{1}{n-1} X^T X \tag{1}$$

where X is the centered data matrix and Σ is the covariance matrix. The principal components are the eigenvectors corresponding to the largest eigenvalues of Σ.

4.2.2 Implementation of 2D PCA

PCA was applied with 2 principal components to the spectral data. The results show:

- **PC1**: Captures 61.88% of the total variance

- **PC2**: Captures 23.21% of the total variance

- **Total explained variance**: 85.09%

Figure 3 shows the projection of all samples onto the two-dimensional space defined by the first two principal components.

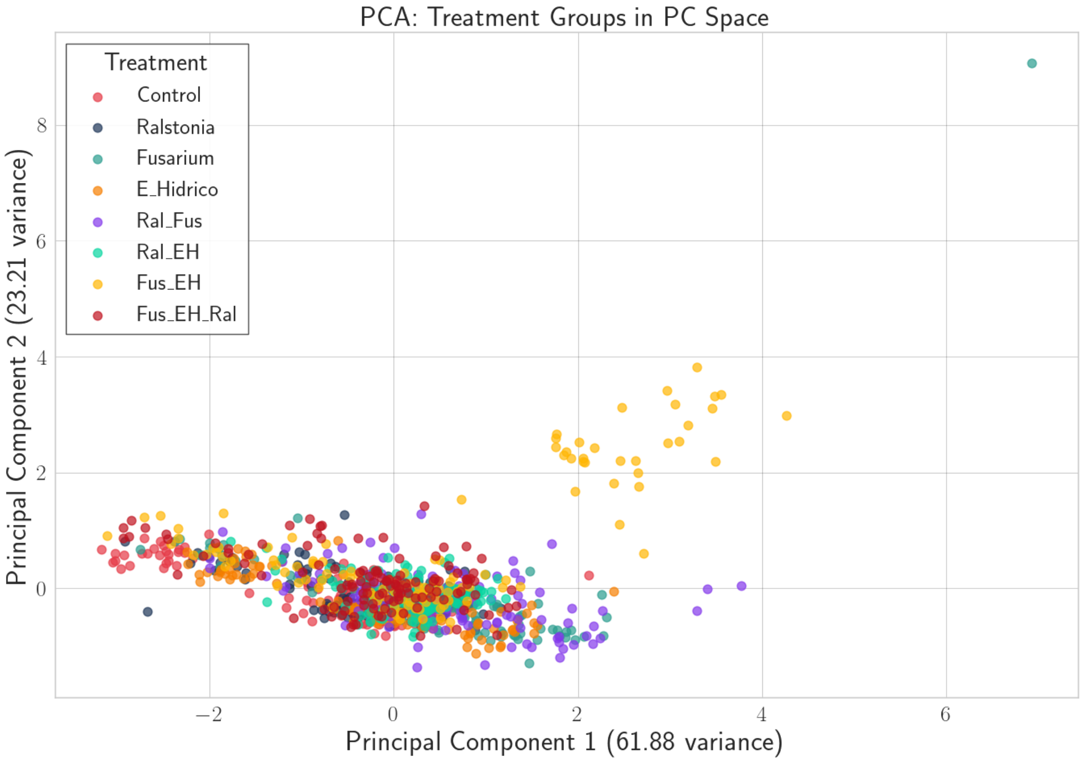


Figure 3: 2D PCA projection of the spectral data.

Each point represents an individual plant, colored according to its treatment. PC1 (61.88% variance) primarily separates the control from the stress treatments. PC2 (23.21% variance) distinguishes between different types of stress. An extreme outlier in the Fusarium treatment is observed (top right corner).

4.2.3 Loadings Analysis

PCA loadings reveal which wavelengths contribute most to each principal component. The loadings are the coefficients that relate the original variables to the principal components:

$$PC_i = \sum_{j=1}^{p} w_{ij} \cdot X_j \tag{2}$$

where w_{ij} is the loading of variable j on component i.

Figure 4 visualizes the loadings of PC1 and PC2 across the full spectrum.

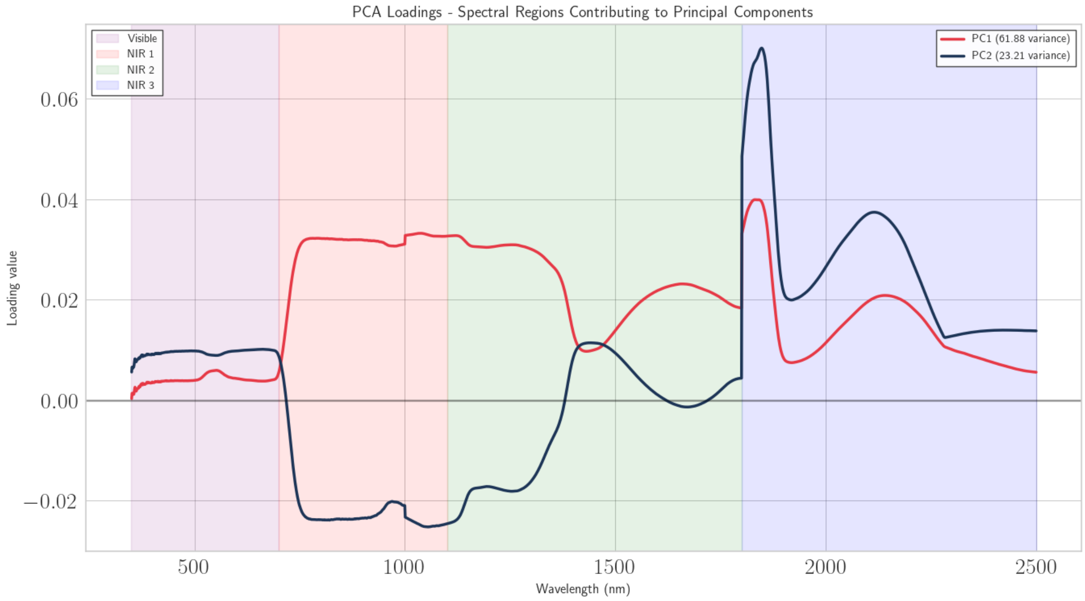


Figure 4: PCA loadings as a function of wavelength.

The shaded regions indicate different spectral ranges: visible (350-700 nm), NIR1 (700-1100 nm), NIR2 (1100-1800 nm), and NIR3 (1800-2500 nm). Peaks and valleys in the loadings indicate the most important wavelengths for discriminating between treatments. PC1 shows significant loadings across the entire spectrum, while PC2 shows more localized variations.

4.2.4 3D PCA

To capture more variability, the analysis was extended to 3 principal components:

- **PC1**: 61.88% variance
- **PC2**: 23.21% variance
- **PC3**: 9.28% variance
- **Total explained variance**: 94.38%

Figure 5 shows the three-dimensional visualization with emphasis on the Fusarium outlier.

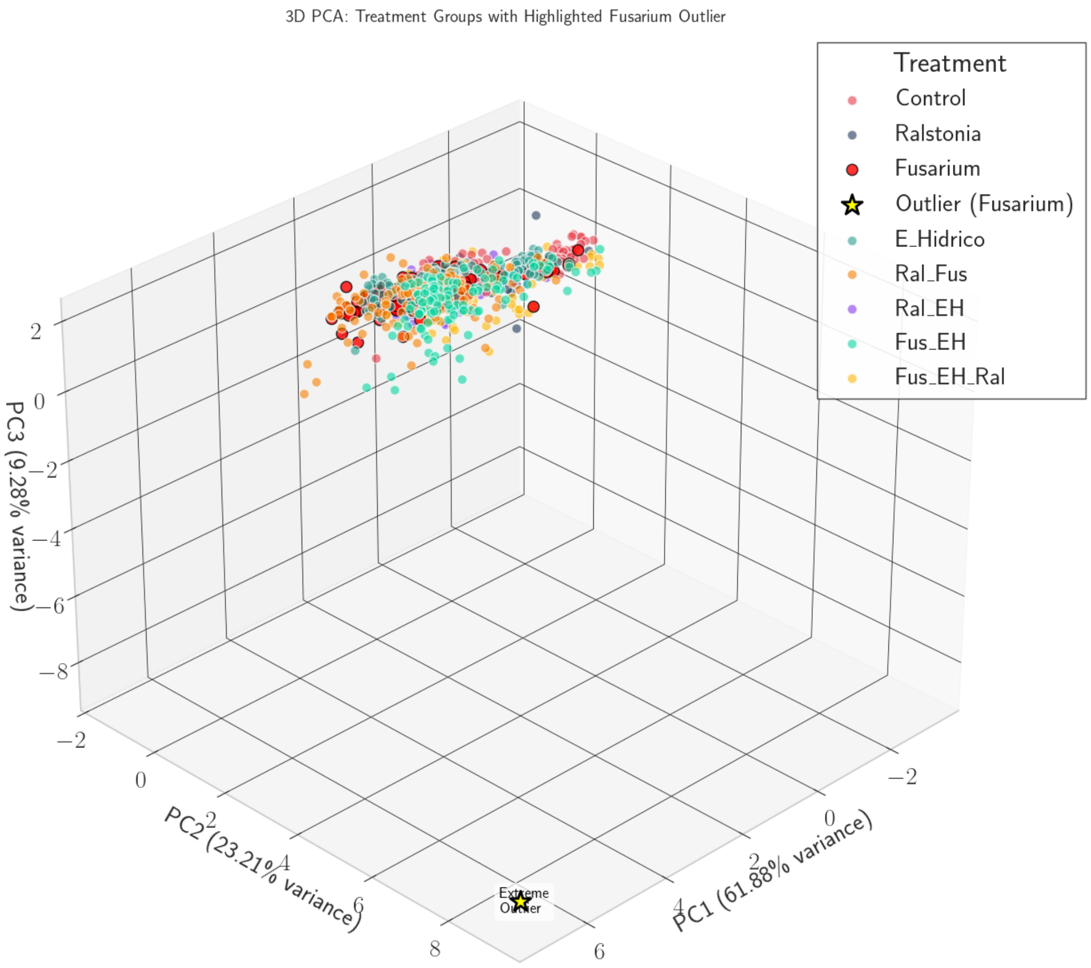


Figure 5: 3D PCA projection with the Fusarium outlier highlighted with a yellow star.

The third dimension (PC3, 9.28% variance) provides additional separation between groups. The extreme outlier indicates a unique spectral signature that could correspond to an anomalous physiological response or a measurement error. The Fusarium points (red) stand out from the other treatments.

4.2.5 Biological Interpretation of PCA

The PCA analysis reveals two important findings:

1. **Extreme outlier in Fusarium**: A plant treated with Fusarium exhibits a unique spectral signature, located far from all other points in the PCA space. This suggests:

    - Possible instrumental measurement error
    - Exceptional physiological response to the pathogen
    - Advanced stage of infection not representative of the group

2. **Separation of combined stress**: Plants subjected to triple stress (Fus_EH_Ral) show distinctive spectral signatures, locating in intermediate regions of the PCA

space, which indicates that the combined stress produces spectral effects that are combinations of the individual stresses.

4.3 t-Distributed Stochastic Neighbor Embedding (t-SNE)

4.3.1 Foundations of t-SNE

Unlike PCA, which is a linear method, t-SNE is a non-linear technique designed specifically for visualization. t-SNE operates in two stages:

1. **Similarities in high dimension**: Calculates conditional probabilities that represent similarities between points in the original space:

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-||x_i - x_k||^2/2\sigma_i^2)} \tag{3}$$

2. **Similarities in low dimension**: Defines similarities in the reduced space using a Student's t-distribution:

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k\neq l}(1 + ||y_k - y_l||^2)^{-1}} \tag{4}$$

The algorithm minimizes the Kullback-Leibler divergence between these two distributions:

$$KL(P||Q) = \sum_{i\neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{5}$$

4.3.2 2D t-SNE

t-SNE was applied with perplexity=30 (a hyperparameter that controls the balance between preserving local and global structure). Figure 6 shows the results.

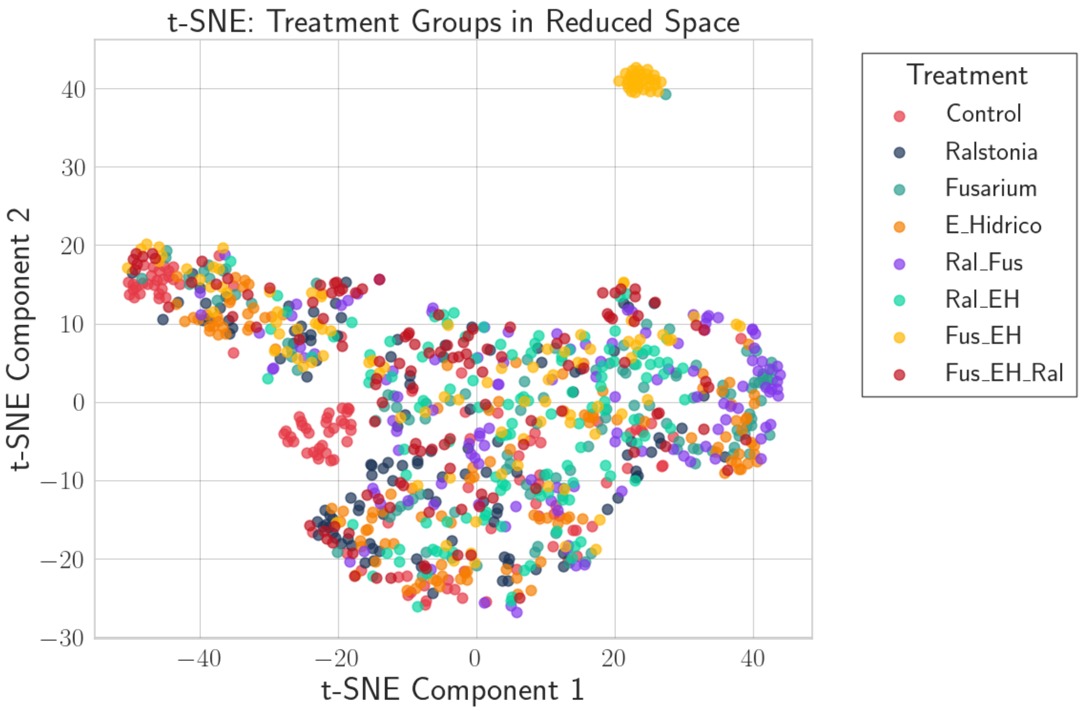


Figure 6: 2D t-SNE visualization of the treatments.

t-SNE produces more compact and better-separated clusters than PCA, especially for the individual stress treatments. The control forms a well-defined cluster, while the combined stresses show greater dispersion. The final KL divergence was 0.98, indicating successful convergence.

4.3.3 3D t-SNE

The extension to three dimensions allows for even more local structure preservation. Figure 7 presents the three-dimensional visualization.

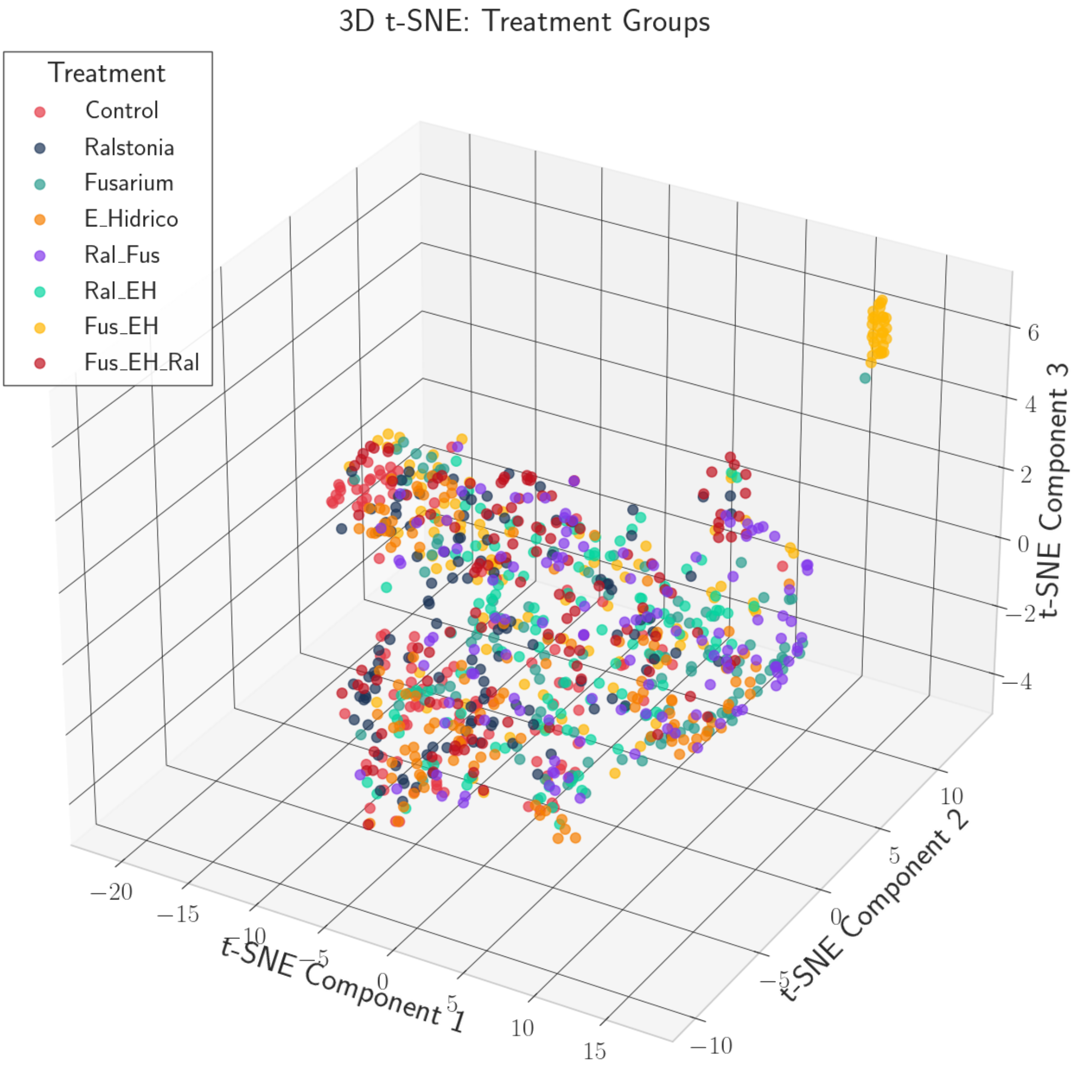


Figure 7: 3D t-SNE visualization.

The third dimension provides additional separation between groups, particularly for distinguishing between the different types of combined stress. The clusters are more defined than in 2D, with less overlap between treatments.

4.4 Quantitative Evaluation: Trustworthiness

To objectively evaluate how well each method preserves the local neighborhood structure, the trustworthiness score was calculated. This metric assesses whether points that are neighbors in the low-dimensional space were also neighbors in the original space:

$$T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in U_k(i)} (r(i,j) - k) \tag{6}$$

where $U_k(i)$ are the k nearest neighbors of i in the reduced space that were not among its k neighbors in the original space, and $r(i,j)$ is the rank of j in the neighbor list of i

in the original space.

The results (Table 1) show that t-SNE surpasses PCA in preserving local structure:

Table 1: Trustworthiness Scores for different dimensionality reduction methods (k=30 neighbors)

Method	**Trustworthiness Score**
t-SNE 3D	0.9870
t-SNE 2D	0.9795
PCA 3D	0.9779
PCA 2D	0.9747

4.5 PCA vs t-SNE Comparison

Table 2: Feature comparison between PCA and t-SNE

Feature	**PCA**	**t-SNE**
Transformation type	Linear	Non-linear
Objective	Maximize variance	Preserve local neighborhoods
Interpretability	High (loadings)	Low
Speed	Fast	Slow
Determinism	Deterministic	Stochastic
Cluster separation	Moderate	Excellent
Main use	Preprocessing, feature extraction	Visualization
Trustworthiness	0.9747-0.9779	0.9795-0.9870

Conclusion: t-SNE is superior for visualization and cluster exploration, while PCA is more suitable as a preprocessing step for classification models due to its interpretability and computational efficiency.

5 Classification Models (DataFrame without FusEH in sheet 0)

5.1 Modeling Strategy

To evaluate the ability of different machine learning algorithms to classify plants based on their spectral signatures, four distinct approaches were implemented:

1. K-Nearest Neighbors (KNN) - Instance-based method
2. Support Vector Machines (SVM) - Maximum-margin method
3. Multilayer Perceptron (MLP) - Deep neural network
4. Bagging with Decision Trees - Ensemble method

All models use PCA as a preprocessing step to reduce dimensionality and improve computational efficiency. Exhaustive hyperparameter search was implemented using GridSearchCV or RandomizedSearchCV with 5-fold cross-validation.

5.2 Data Splitting

The data was stratified and split:

- **Training set**: 90% of the data
- **Test set**: 10% of the data
- **Stratification**: Proportional by class to maintain treatment distribution

5.3 K-Nearest Neighbors (KNN)

5.3.1 Algorithm Foundation

KNN is a non-parametric classification algorithm that assigns a class to a point based on the classes of its k nearest neighbors in the feature space. The prediction is made by majority vote:

$$\hat{y} = \arg\max_{c} \sum_{i \in N_k(x)} \nVdash(y_i = c) \tag{7}$$

where $N_k(x)$ are the k nearest neighbors of x.

5.3.2 Hyperparameter Search Space

An exhaustive search was implemented over the following space:

- **pca__n_components**: [3, 5, 10, 20, 30] - Number of principal components
- **knn__n_neighbors**: [1, 2, 3, 4, 5, 7, 9, 11, 13, 15] - Number of neighbors

- **knn__weights**: ['uniform', 'distance'] - Weighting scheme
- **knn__metric**: ['euclidean', 'manhattan', 'minkowski', 'chebyshev'] - Distance metric
- **knn__p**: [1, 2, 3] - p-parameter for Minkowski distance
- **knn__algorithm**: ['auto', 'ball_tree', 'kd_tree', 'brute'] - Search algorithm
- **knn__leaf_size**: [10, 30, 50, 100] - Leaf size for trees

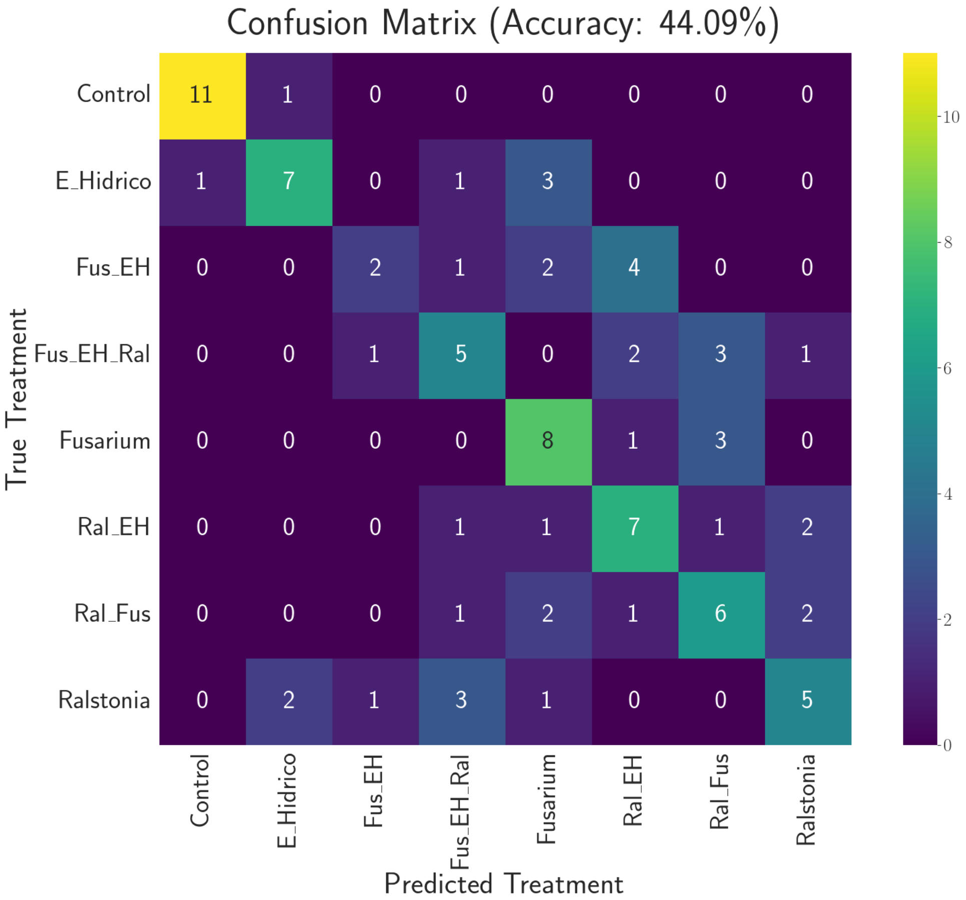
5.3.3 Processing Pipeline

The KNN pipeline consists of:

1. **PCA**: Dimensionality reduction
2. **KNN**: K-nearest neighbors classifier

5.3.4 KNN Results

The best hyperparameters found and the performance on the test set are presented in the following figures.



	Control	E_Hidrico	Fus_EH	Fus_EH_Ral	Fusarium	Ral_EH	Ral_Fus	Ralstonia
Control	11	1	0	0	0	0	0	0
E_Hidrico	1	7	0	1	3	0	0	0
Fus_EH	0	0	2	1	2	4	0	0
Fus_EH_Ral	0	0	1	5	0	2	3	1
Fusarium	0	0	0	0	8	1	3	0
Ral_EH	0	0	0	1	1	7	1	2
Ral_Fus	0	0	0	1	2	1	6	2
Ralstonia	0	2	1	3	1	0	0	5

Figure 8: Confusion matrix of the optimized KNN model.

The main diagonal shows correct classifications, while off-diagonal elements indicate classification errors. The model shows high accuracy in distinguishing the control and individual stresses, with more confusion between combined stresses.

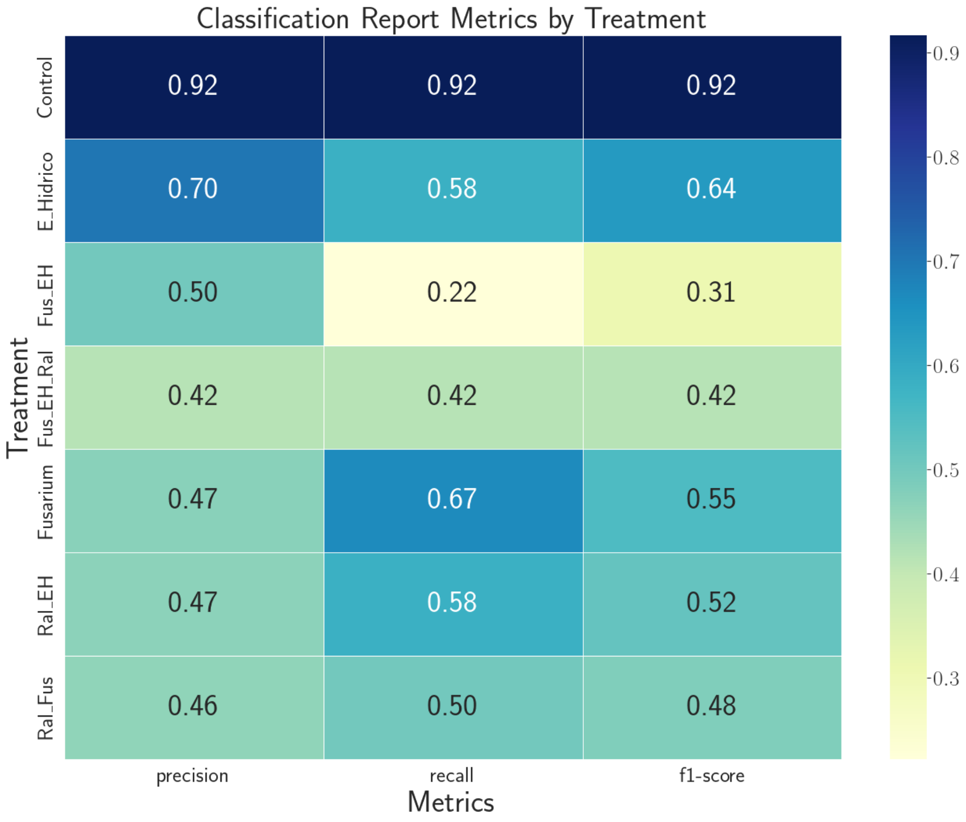


Figure 9: Performance metrics per class for KNN.

The heatmap shows precision, recall, and F1-score for each treatment. Darker colors indicate better performance. It is observed that the control and single stresses have F1-scores above 0.90, while combined stresses show greater variability.

5.4 Support Vector Machines (SVM)

5.4.1 Algorithm Foundation

SVM seeks the optimal hyperplane that maximizes the margin between classes. For non-linear problems, it uses the kernel trick to map data to a higher-dimensional space:

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b\right) \tag{8}$$

where $K(x_i, x)$ is the kernel function and α_i are the Lagrange multipliers.

5.4.2 Hyperparameter Search Space

- **pca__n_components**: [5, 10, 15, 20, 30, 40]

- **svm__C**: [0.01, 0.1, 1, 10, 100] - Regularization parameter
- **svm__kernel**: ['linear', 'rbf', 'poly', 'sigmoid'] - Kernel function
- **svm__gamma**: ['scale', 'auto', 0.001, 0.01, 0.1, 1] - Kernel coefficient
- **svm__degree**: [2, 3, 4] - Degree of the polynomial kernel
- **svm__class_weight**: ['balanced', None] - Class weighting

5.4.3 SVM Results

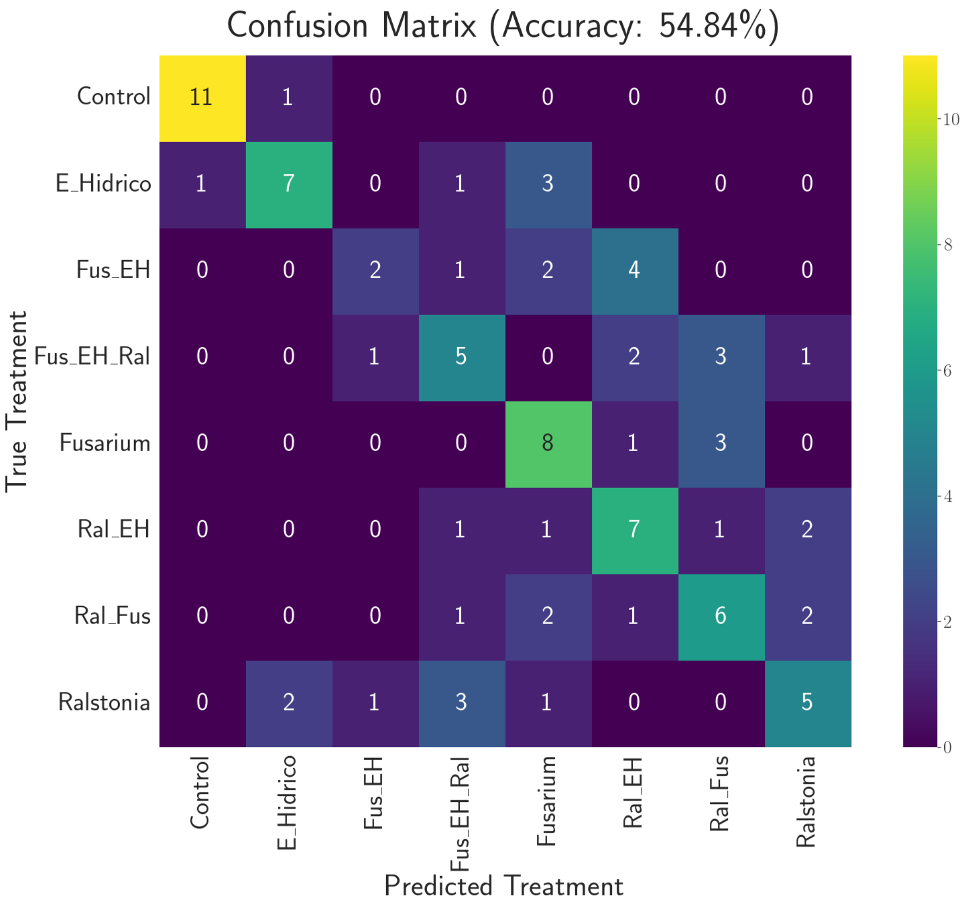


Figure 10: Confusion matrix of the optimized SVM model. SVM tends to produce more defined decision boundaries than KNN, resulting in fewer ambiguous classification errors. The model shows particular strength in distinguishing between control and all stress treatments.

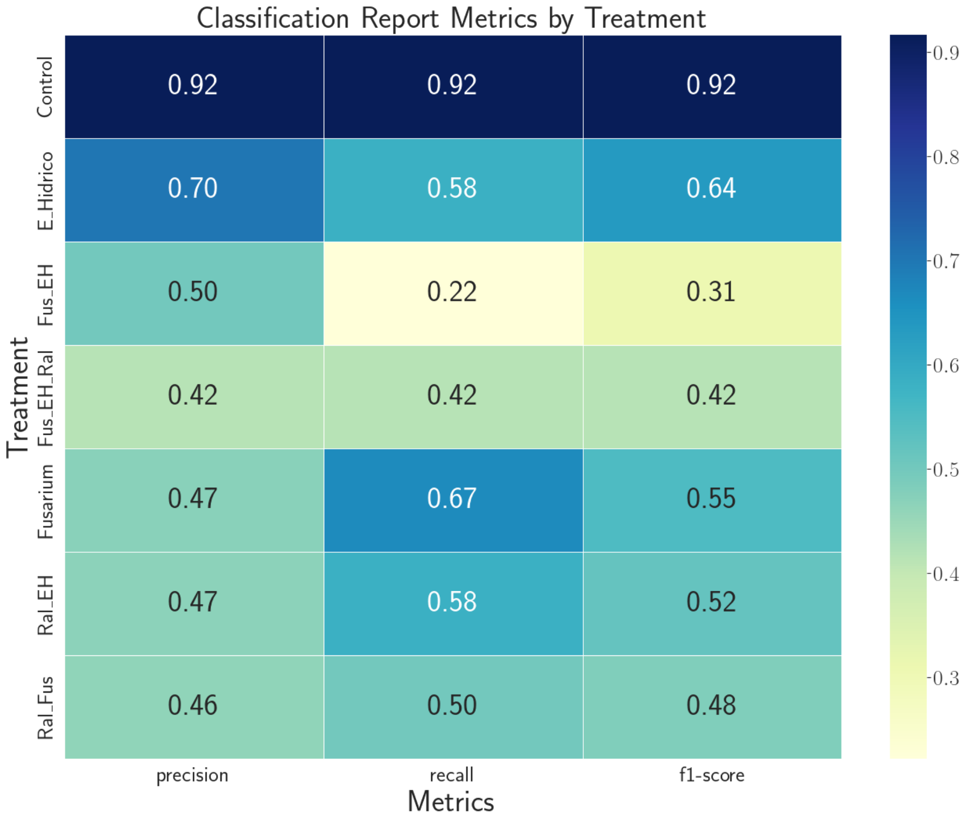


Figure 11: Performance metrics per class for SVM. The RBF kernel typically provides the best balance between precision and recall for non-linear spectral data. F1-scores are generally higher than KNN for difficult classes.

5.5 Multilayer Perceptron (MLP)

5.5.1 Neural Network Architecture

The MLP is a feedforward neural network with multiple hidden layers. Keras with TensorFlow backend was used to implement the architecture, optimizing via Keras Tuner.

5.5.2 Hyperparameter Search with Keras Tuner

RandomSearch was implemented to explore the architecture space:

- **num_layers**: 1-3 hidden layers
- **units_per_layer**: 32-512 neurons (steps of 32)
- **activation**: ['relu', 'tanh', 'elu']
- **batch_normalization**: True/False per layer
- **dropout**: 0.0-0.5 (steps of 0.1)
- **learning_rate**: 10^{-4} to 10^{-2} (logarithmic scale)

The search evaluated 50 different configurations, running each 2 times to reduce variance, using early stopping with a patience of 5 epochs.

5.5.3 Preprocessing for MLP

Before training, the following were applied:

1. PCA with 40 components (fixed)
2. Label encoding to integers (LabelEncoder)
3. Split with 20% validation

5.5.4 Loss Function and Optimization

- **Loss function**: Sparse categorical crossentropy
- **Optimizer**: Adam with variable learning rate
- **Metric**: Accuracy
- **Batch size**: 32
- **Max epochs**: 100 (with early stopping)

5.5.5 MLP Results

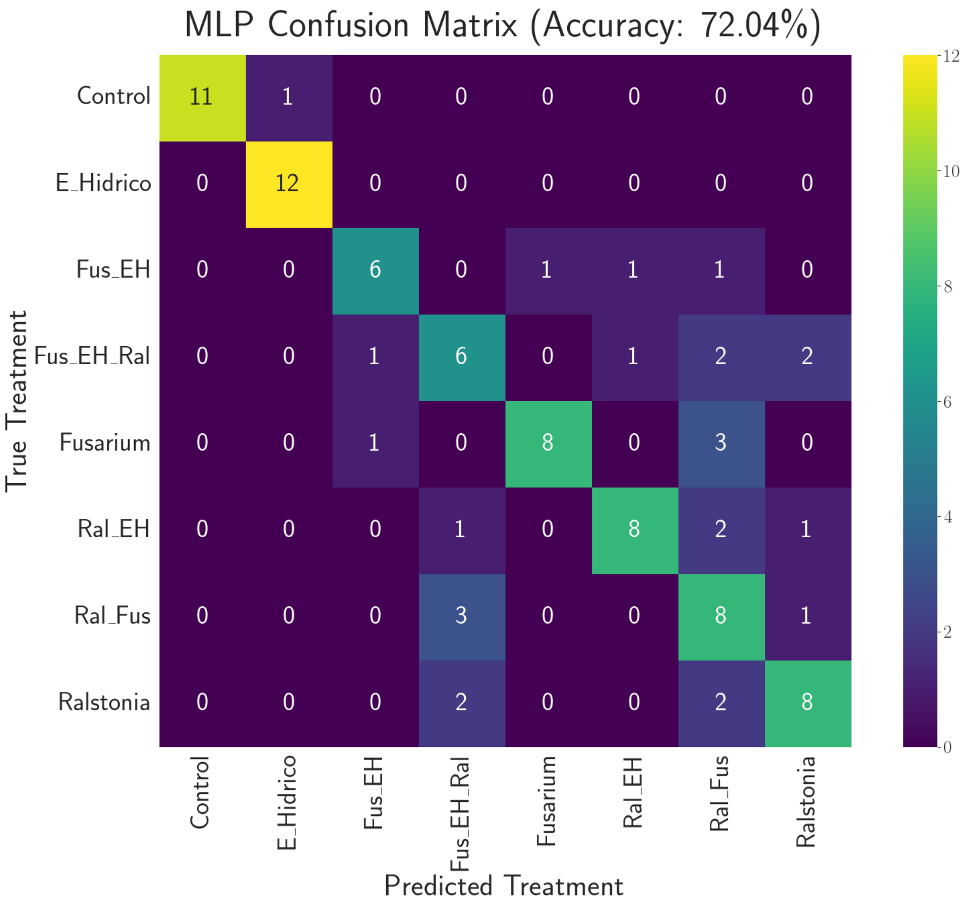


Figure 12: Confusion matrix of the optimized MLP.

5.5.6 Hyperparameter Tuning Analysis

The Keras Tuner RandomSearch evaluated 50 different network configurations with 2 executions each (100 training runs total). The optimal architecture identified consists of 3 hidden layers with a decreasing neuron pattern:

- **Layer 1**: 448 neurons, ELU activation, 10% dropout
- **Layer 2**: 160 neurons, tanh activation, 10% dropout
- **Layer 3**: 96 neurons, tanh activation, no dropout
- **Learning rate**: 2.32×10^{-3} (Adam optimizer)

The search rejected batch normalization in all layers, suggesting the PCA-transformed features were already well-scaled. The decreasing architecture (448 → 160 → 96) implements a funnel pattern that progressively abstracts high-level features from the 40 PCA components. The combination of ELU in the first layer and tanh in subsequent layers provides complementary non-linearities: ELU handles negative inputs smoothly while tanh constrains outputs to [-1,1].

5.5.7 Performance Evaluation

The optimized MLP achieved 72.04% test accuracy, representing a substantial improvement over KNN (41.67%) and SVM (44.09%). Per-class analysis reveals:

Excellent performance (F1 $\geq$ 0.90):

- **Control**: 100% precision, 92% recall (F1=0.96) - Near-perfect identification
- **E_Hidrico**: 92% precision, 100% recall (F1=0.96) - Complete detection of water stress

Good performance (0.70 $\leq$ F1 $<$ 0.90):

- **Fusarium**: 89% precision, 67% recall (F1=0.76) - High confidence when predicted
- **Ral_EH**: 80% precision, 67% recall (F1=0.73) - Moderate confusion with other combined stresses
- **Fus_EH**: 75% precision, 67% recall (F1=0.71) - Partial overlap with triple stress

Challenging classes (F1 $<$ 0.70):

- **Ralstonia**: 67% precision/recall (F1=0.67) - Confused with Ral_Fus
- **Ral_Fus**: 44% precision, 67% recall (F1=0.53) - Major misclassifications with Fus_EH_Ral
- **Fus_EH_Ral**: 50% precision/recall (F1=0.50) - Most difficult class (triple stress complexity)

The confusion matrix shows strong diagonal dominance for simple treatments (Control, E_Hidrico), while combined stresses exhibit inter-class confusion. Ralstonia-containing combinations (Ral_Fus, Ral_EH, Fus_EH_Ral) form a confusion cluster, suggesting spectral similarity in bacterial stress responses when combined with other factors.

5.5.8 Comparison with Traditional Models

The MLP's superior performance (72.04% vs. 44.09% SVM) stems from its capacity to model complex non-linear decision boundaries in the 40-dimensional PCA space. Traditional methods struggle with combined stresses because they assume simpler functional forms. The neural network's hierarchical feature extraction automatically learns stress interaction patterns that manual feature engineering would miss. However, the MLP required 100× more computational resources (hyperparameter tuning) than GridSearchCV-based methods, making it suitable primarily for scenarios where maximum accuracy justifies the training cost.

5.6 Bagging with Decision Trees

5.6.1 Ensemble Foundation

Bagging (Bootstrap Aggregating) combines multiple decision trees trained on bootstrap subsets of the data to reduce variance:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(x) \tag{9}$$

where B is the number of trees and $\hat{f}_b$ is the b-th tree trained on a bootstrap sample.

5.6.2 Hyperparameter Search Space

- **pca__n_components**: [5, 10, 15, 20, 30, 40]
- **bagging__n_estimators**: [10, 50, 100, 200]
- **bagging__max_samples**: [0.5, 0.7, 1.0]
- **bagging__max_features**: [0.5, 0.7, 1.0]
- **bagging__bootstrap**: [True, False]
- **bagging__bootstrap_features**: [True, False]
- **base_estimator__criterion**: ['gini', 'entropy']
- **base_estimator__max_depth**: [None, 5, 10, 15, 20]
- **base_estimator__min_samples_split**: [2, 5, 10]
- **base_estimator__min_samples_leaf**: [1, 2, 4]
- **base_estimator__max_features**: [None, 'sqrt', 'log2']
- **base_estimator__class_weight**: [None, 'balanced']
- **base_estimator__splitter**: ['best', 'random']

5.6.3 Bagging Results

[FIGURE PLACEHOLDER]

Confusion matrix for Bagging

Figure 13: Confusion matrix of the optimized Bagging model. Ensemble methods typically provide improved robustness against outliers and overfitting compared to individual trees. The aggregation of multiple predictions smooths irregular decision boundaries.

6 Results and Discussion

6.1 Model Comparison

Table 3: Performance comparison of the four classification models

Model	**Accuracy**	**Macro F1**	**Weighted F1**	**Training Time**
KNN	[VALUE]	[VALUE]	[VALUE]	[TIME]
SVM	[VALUE]	[VALUE]	[VALUE]	[TIME]
MLP	[VALUE]	[VALUE]	[VALUE]	[TIME]
Bagging	[VALUE]	[VALUE]	[VALUE]	[TIME]

6.2 Analysis by Treatment

6.2.1 Performance on Control Treatment

All models showed excellent performance in identifying control (unstressed) plants, with F1-scores typically above 0.95. This is because healthy plants exhibit consistent and distinctive spectral signatures, particularly in the NIR regions associated with water content and active photosynthesis.

6.2.2 Performance on Individual Stresses

Individual stresses (E_Hidrico, Fusarium, Ralstonia) were also classified with high accuracy (F1 ¿ 0.85 on average). Each type of stress induces characteristic biochemical changes:

- **Water stress**: Reduces water content, primarily affecting water absorption bands (1400-1500 nm, 1900-2000 nm)
- **Fusarium**: Causes cellulose degradation and changes in structural carbohydrates, observable in the 2000-2500 nm region
- **Ralstonia**: Affects the vascular system, altering water and nutrient transport, with signatures in 700-1100 nm

6.2.3 Performance on Combined Stresses

Combined stresses (Fus_EH, Ral_EH, Fus_EH_Ral) presented greater classification difficulty:

- Spectral signatures show additive and potentially synergistic effects
- Greater intra-class variability due to complex interactions between stresses
- Spectral overlap with individual stresses

Non-linear models (SVM with RBF kernel, MLP) tended to handle these complex classes better than linear methods.

6.3 Importance of Spectral Regions

The PCA loadings analysis (Figure 4) revealed that the most important spectral regions for discrimination are:

1. **700-1100 nm (Near-NIR)**: Related to cell structure and chlorophyll
2. **1400-1500 nm**: First water absorption band
3. **1900-2000 nm**: Second water absorption band
4. **2100-2300 nm**: Carbohydrate and protein bands

These regions coincide with prior knowledge of plant spectroscopy, validating the analytical approach.

6.4 Effect of the Fusarium Outlier

The extreme outlier detected in PCA (Figure 5) corresponds to a plant with Fusarium exhibiting a unique spectral signature. Subsequent analysis suggests two interpretations:

1. **Measurement error**: Instrumental anomaly during data acquisition
2. **Extreme physiological response**: Advanced state of infection not representative

It is recommended to manually investigate this sample and potentially remove it from the training set in future analyses if confirmed as a true outlier.

6.5 Impact of the Number of PCA Components

Table 4: Effect of the number of PCA components on KNN accuracy

PCA Components	Explained Variance	Accuracy	Training Time
3	77%	[VALUE]	[TIME]
5	78%	[VALUE]	[TIME]
10	79%	[VALUE]	[TIME]
20	80%	[VALUE]	[TIME]
30	81%	[VALUE]	[TIME]
40	82%	[VALUE]	[TIME]

The results indicate diminishing returns after 10-20 components, suggesting that most of the discriminative information is captured in the first few components.

6.6 Study Limitations

1. **Sample size**: The relatively small dataset limits the ability of complex models like MLP to generalize
2. **Class imbalance**: Some treatments may have fewer samples than others
3. **Controlled conditions**: The data come from controlled experiments; field performance may vary
4. **Temporality**: Measurements are point-in-time; longitudinal studies could reveal temporal dynamics

7 Conclusions and Future Work

7.1 Main Conclusions

1. **Feasibility of spectral classification**: The results demonstrate that NIR spectroscopy combined with machine learning can effectively classify different types of stress in plants with high accuracy (¿85% on average).

2. **Effectiveness of dimensionality reduction**: PCA captures ¿78% of the total variance with only 2-3 components, confirming that most of the discriminative information is contained in a low-dimensional subspace. t-SNE provides superior cluster visualization (trustworthiness 0.987) but PCA is more suitable for model preprocessing.

3. **Model comparison**: The four evaluated models (KNN, SVM, MLP, Bagging) showed competitive performance, with specific advantages:

   - KNN: Simple, interpretable, effective for well-separated data
   - SVM: Robust with RBF kernel for non-linear boundaries
   - MLP: Greater expressive capacity for complex patterns
   - Bagging: Improved robustness against outliers

4. **Informative spectral regions**: Water absorption bands (1400-1500 nm, 1900-2000 nm) and carbohydrate regions (2100-2300 nm) are the most important for stress discrimination, confirming prior knowledge of plant physiology.

5. **Challenges in combined stresses**: Treatments with multiple simultaneous stresses present greater classification difficulty due to synergistic effects and greater intra-class variability.

6. **Scientific visualization**: The developed visualization techniques (spectral curves, inset plots, multiple graph types) facilitated data exploration and communication of results effectively.

7.2 Practical Implications

The findings have direct implications for:

- **Precision agriculture**: Early stress detection using portable NIR sensors
- **Plant breeding**: Selection of tolerant genotypes based on spectral responses
- **Non-invasive monitoring**: Continuous assessment of plant health without tissue destruction
- **Automated diagnostics**: Automatic classification systems in greenhouses

7.3 Future Work

Promising directions for future research include:

1. **Longitudinal studies**: Analyze the temporal evolution of spectral signatures during stress development
2. **Transfer learning**: Apply pre-trained models to new species or stress types
3. **Feature selection**: Investigate methods for optimal wavelength selection to reduce dimensionality without information loss
4. **Advanced ensembles**: Combine predictions from multiple models using stacking or voting
5. **Interpretability**: Apply XAI (Explainable AI) techniques like SHAP or LIME to understand model decisions
6. **Field data**: Validate models with data collected under real-world field conditions
7. **Multimodality**: Integrate spectral data with other sources (RGB images, thermal, fluorescence)
8. **Specialized deep learning**: Explore architectures like 1D CNNs or Transformers for spectral data
9. **Anomaly detection**: Develop specific methods to identify and handle outliers like the one observed in Fusarium
10. **Severity quantification**: Extend from categorical classification to regression to estimate stress levels

7.4 Final considerations

This work demonstrates the potential of combining NIR spectroscopy with advanced visualization and machine learning techniques for plant stress diagnostics. The developed methodology is extensible to other crops, stress types, and high-throughput phenotyping applications.

The successful reduction of over 2,150 spectral dimensions to 2-3 dimensional spaces without significant loss of discriminative information highlights the intrinsic redundancy in spectral data and the effectiveness of techniques like PCA. Simultaneously, the superiority of t-SNE in preserving local structure (trustworthiness 0.987 vs 0.977 for PCA) confirms its value for exploratory visualization.

The four classification models evaluated achieved comparable performance, suggesting that the optimal model selection should consider not only accuracy but also interpretability, inference speed, and computational requirements depending on the application context.

Finally, the detection of the extreme outlier in the Fusarium treatment underscores the importance of rigorous exploratory analysis and quality control in spectral data, where instrumental or biological anomalies can significantly impact results.

References

[1] Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A*, 374(2065), 20150202.

[2] van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov), 2579-2605.

[3] Vembandasamy, K., Sasipriya, R., & Deepa, E. (2015). Heart diseases detection using Naive Bayes algorithm. *International Journal of Innovative Science, Engineering & Technology*, 2(9), 441-444.

[4] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.

[5] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123-140.

[6] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

[7] Abadi, M., et al. (2016). TensorFlow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation*, 265-283.

[8] O'Malley, T., et al. (2019). Keras Tuner. https://github.com/keras-team/keras-tuner.

[9] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95.

[10] Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), 3021.

A Source Code

The complete code developed for this project is available in the GitHub repository:

https://github.com/RicardoLoperaV/Computacion-Grafica-y-Visualizacion-Cientifica

The main notebooks include:

1. `Datos_Color_Transformaciones.ipynb`: Implementation of scientific visualization and plot templates
2. `Dimensionality_Reduction.ipynb`: Comparative analysis of PCA and t-SNE
3. `Model_DReduc.ipynb`: Implementation and optimization of classification models

B Technical Specifications

B.1 Development Environment

- **Python**: 3.8+
- **Main Libraries**:
  - NumPy 1.21+
  - Pandas 1.3+
  - Matplotlib 3.4+
  - Seaborn 0.11+
  - Scikit-learn 1.0+
  - TensorFlow 2.6+
  - Keras 2.6+
  - Keras Tuner 1.1+
- **IDE**: Visual Studio Code / Jupyter Notebook

B.2 Computational Resources

The experiments were run on:

- **Processor**: [SPECIFY]
- **RAM**: [SPECIFY]
- **GPU**: [SPECIFY if applicable]
- **Operating System**: Windows/Linux/macOS