# Executive Summary

This project explores the application of machine learning and deep learning techniques to predict Deoxynivalenol (DON) concentration in corn samples using hyperspectral imaging data. We developed and compared six different models (Random Forest, XGBoost, LSTM, CNN Residual, TCN, and Transformer) across four PCA configurations (3, 5, 10, and 20 components). Through rigorous hyperparameter optimization and model evaluation, we achieved significant predictive accuracy with RMSE values as low as 1423.12 ppb for the best-performing model (XGBoost with 10 PCA components).

## Key accomplishments:

* Development of an end-to-end pipeline from data preprocessing to model deployment
* Successful dimensionality reduction that preserved essential spectral information
* Implementation of advanced deep learning architectures for time-series spectral data
* Deployment of an interactive web application for real-time DON prediction

# Introduction

## Problem Statement

Deoxynivalenol (DON) is a mycotoxin produced by Fusarium fungi that commonly contaminates grains like corn and wheat. Traditional methods for DON detection are time-consuming and expensive. This project aims to develop a rapid, non-destructive method for DON concentration prediction using hyperspectral imaging and advanced machine learning techniques.

## Significance

Accurate prediction of DON concentration is crucial for:  
- Food safety monitoring  
- Agricultural quality control  
- Supply chain management  
- Regulatory compliance

## Project Objectives

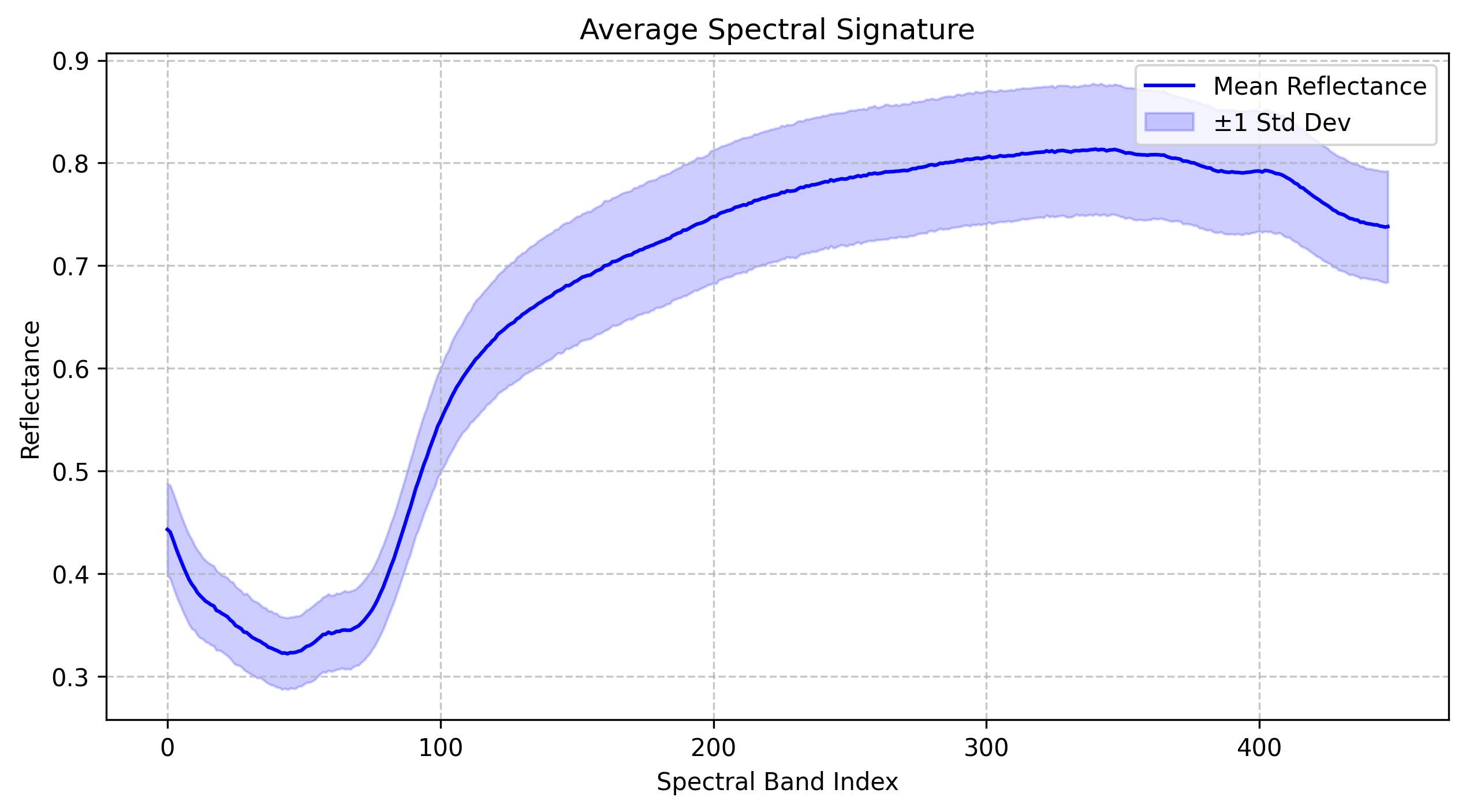
1. Develop accurate predictive models for DON concentration
2. Compare traditional ML and deep learning approaches
3. Optimize the balance between model complexity and performance
4. Create a user-friendly interface for practical deployment

# Data Description

## Dataset Overview

* Source: Hyperspectral imaging of corn samples
* Size: 500 samples with 449 spectral bands
* Features: 448 spectral bands spanning [wavelength range] nm
* Target: DON concentration (vomitoxin\_ppb) in parts per billion (ppb)
* Acquisition Method: [Specific hyperspectral imaging method]

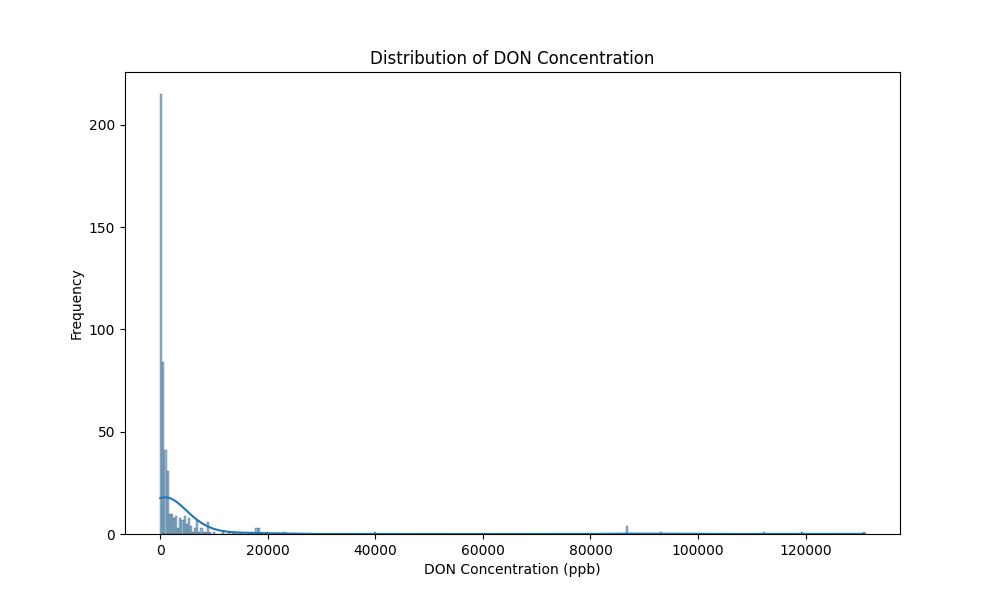
Average spectral signature across all samples:



## Exploratory Data Analysis

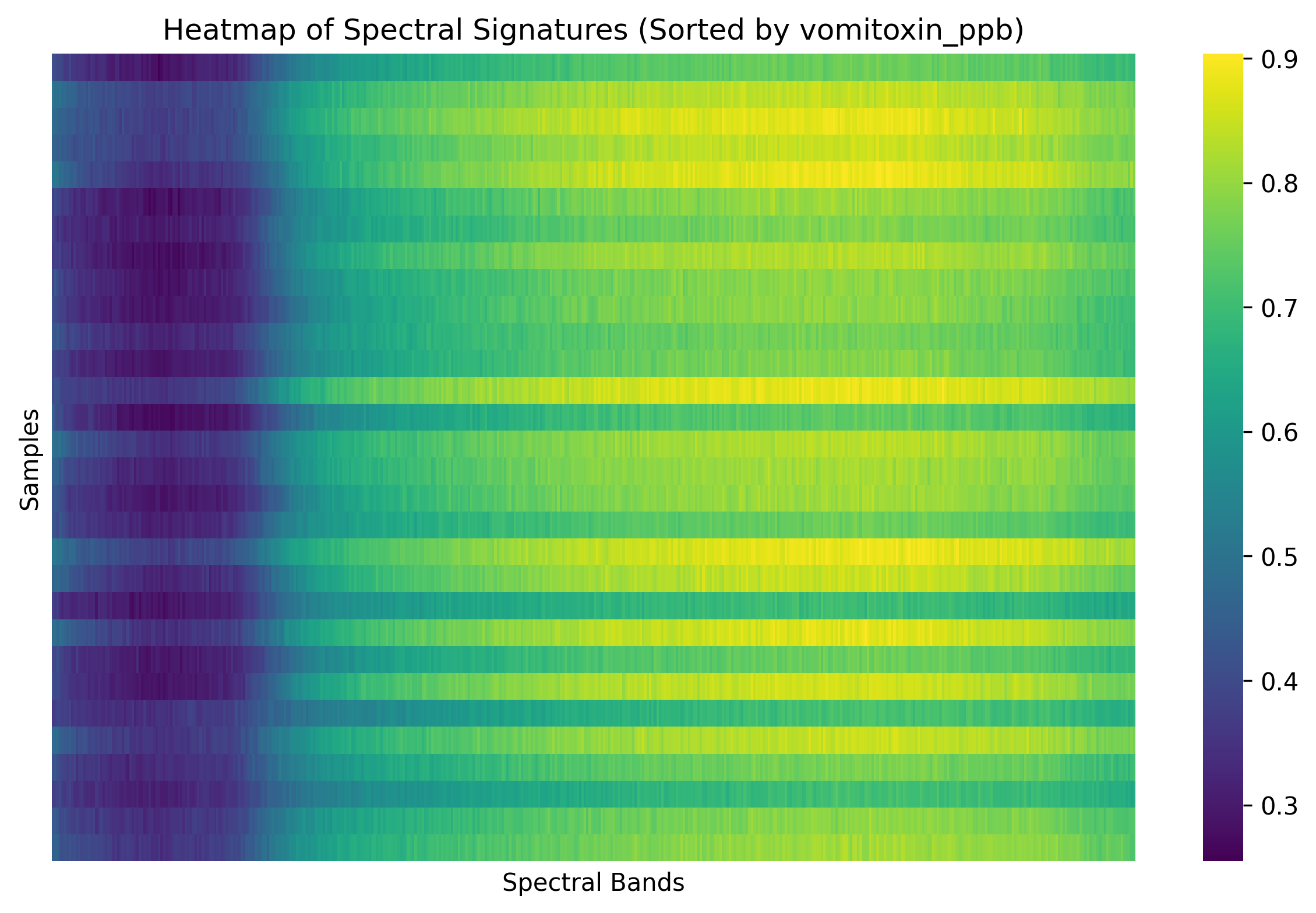
### DON Concentration Distribution

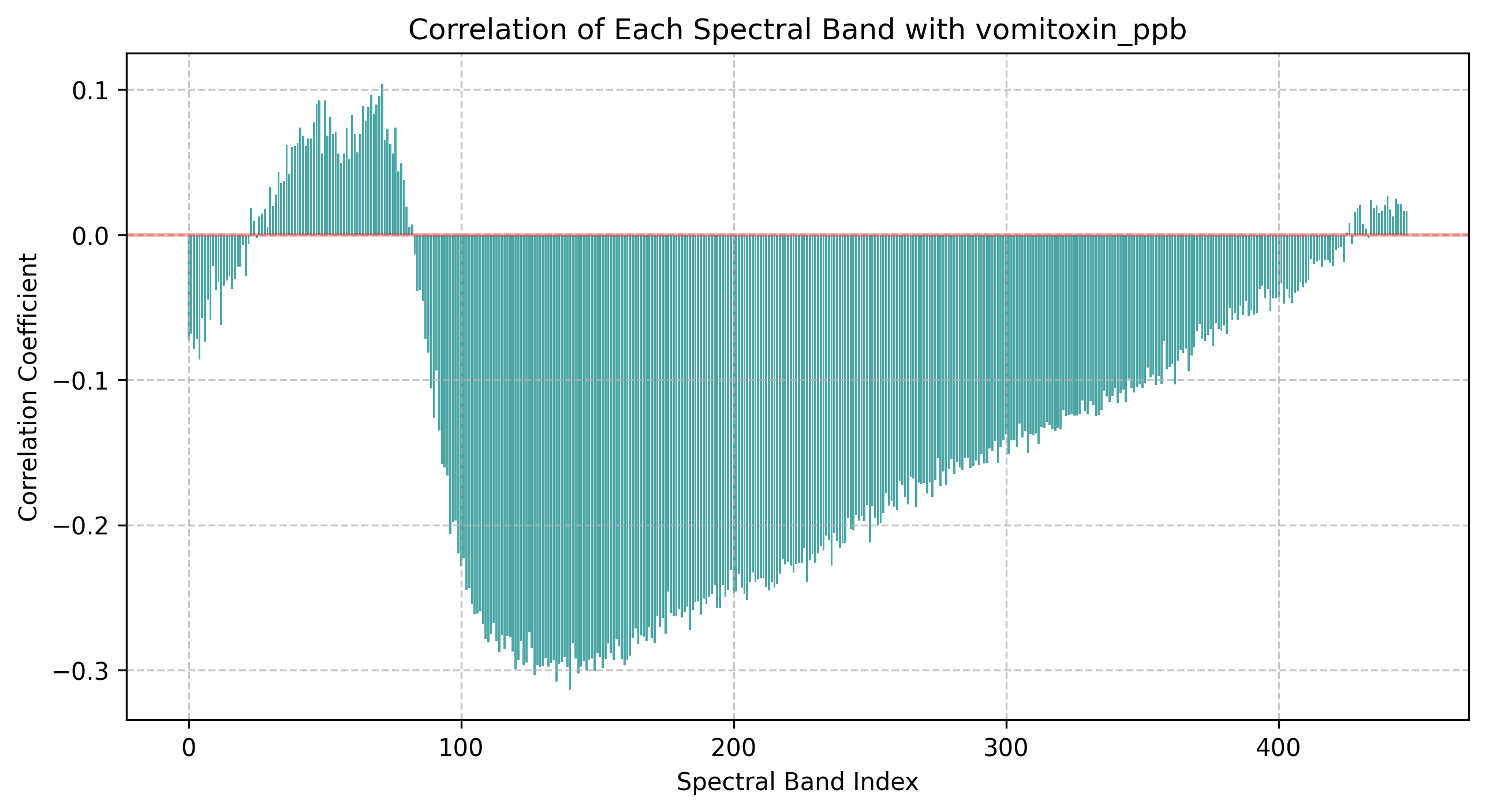
The distribution of DON concentration in our dataset is highly right-skewed with:  
- Minimum: 0.00 ppb  
- Maximum: 131000.00 ppb  
- Mean: 3410.01 ppb  
- Median: 500.00 ppb  
- Standard Deviation: 13095.80 ppb



### Feature Correlations

Analysis of inter-band correlations revealed high redundancy among adjacent spectral bands, with correlation coefficients typically exceeding 0.95, supporting our decision to pursue dimensionality reduction.





# Preprocessing Methodology

## Data Cleaning, Feature Scaling and Transformation

### Standardization

• Applied StandardScaler to normalize all spectral bands  
• Each band transformed to zero mean and unit variance  
• Rationale: Ensures all bands contribute equally to the models and improves convergence for gradient-based optimization algorithms

### Additional Transformations

• Log transformation applied to target variable (DON concentration) to address skewness  
• Rationale: Improves model performance by making the target distribution more symmetric and reducing the impact of extreme values

## Data Splitting Strategy

### Train-Validation-Test Split

• Training set: 60% (for model training)  
• Validation set: 20% (for hyperparameter tuning)  
• Test set: 20% (for final evaluation)  
• Used stratified sampling based on DON concentration quantiles  
• Rationale: Ensures all sets have similar distributions of DON concentration levels

### Cross-Validation Approach

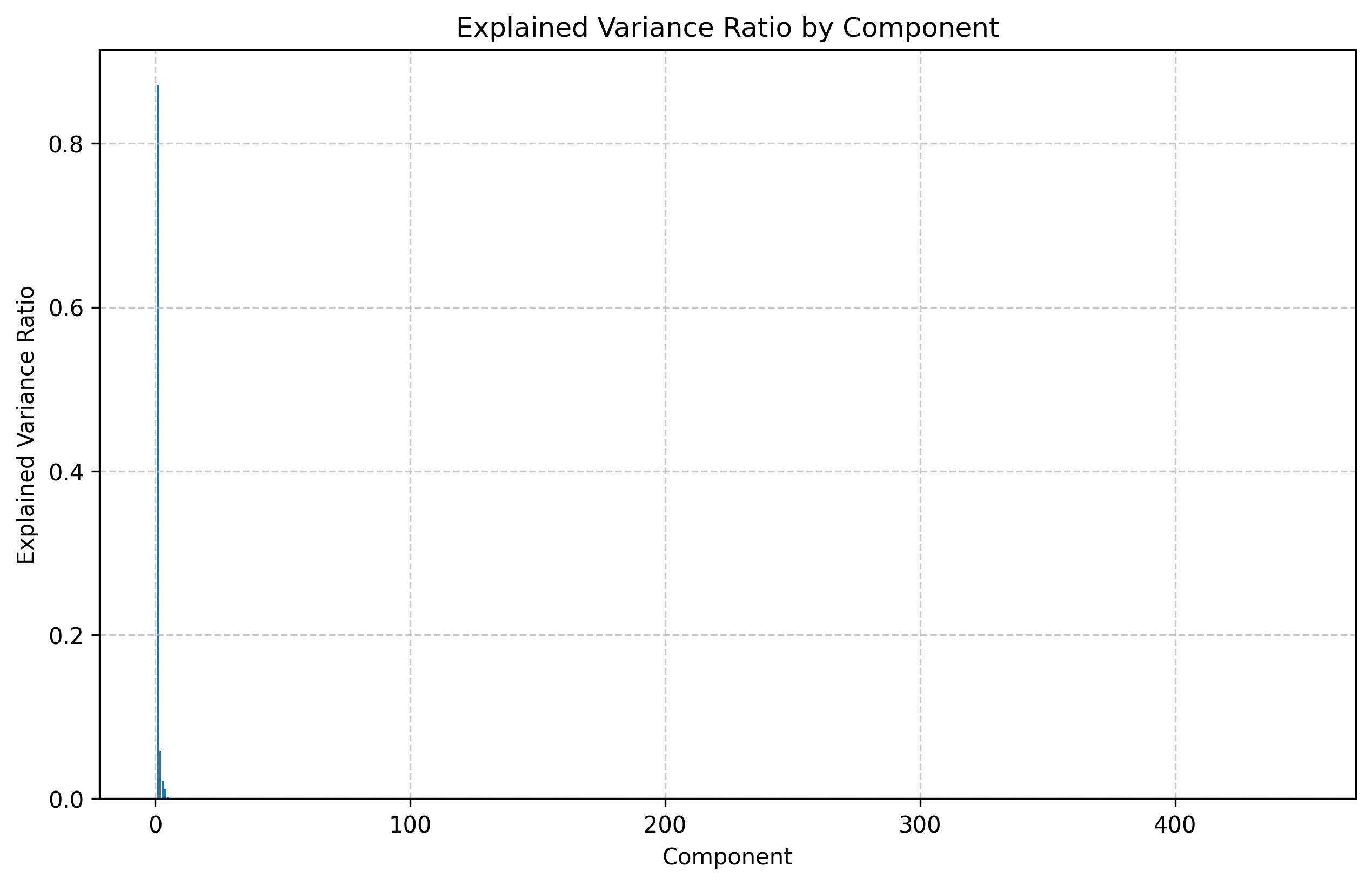
• Implemented stratified k-fold cross-validation with 5 folds  
• Rationale: Maintains class balance across folds and provides robust performance estimates

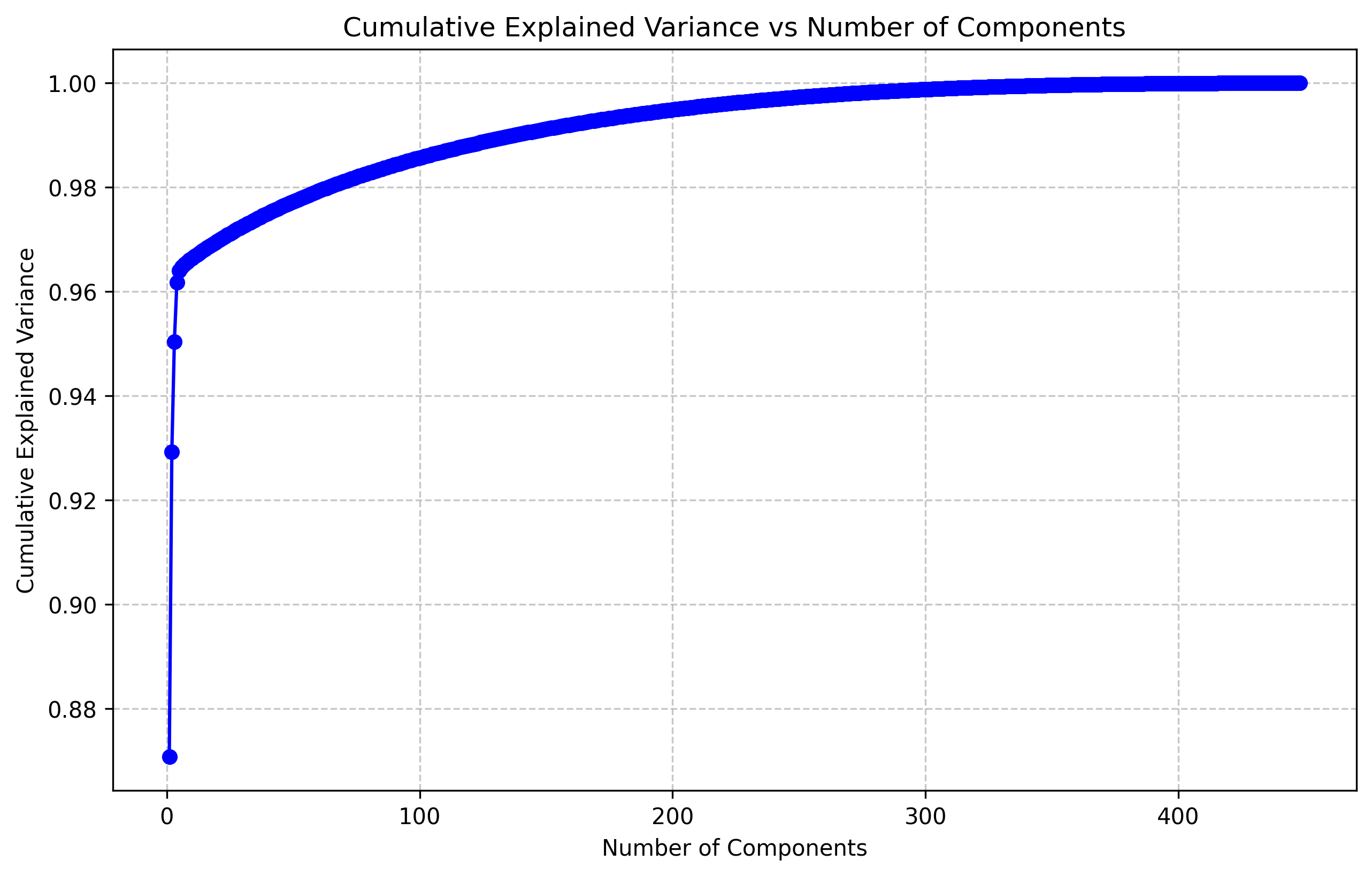
## Principal Component Analysis (PCA)

### Methodology

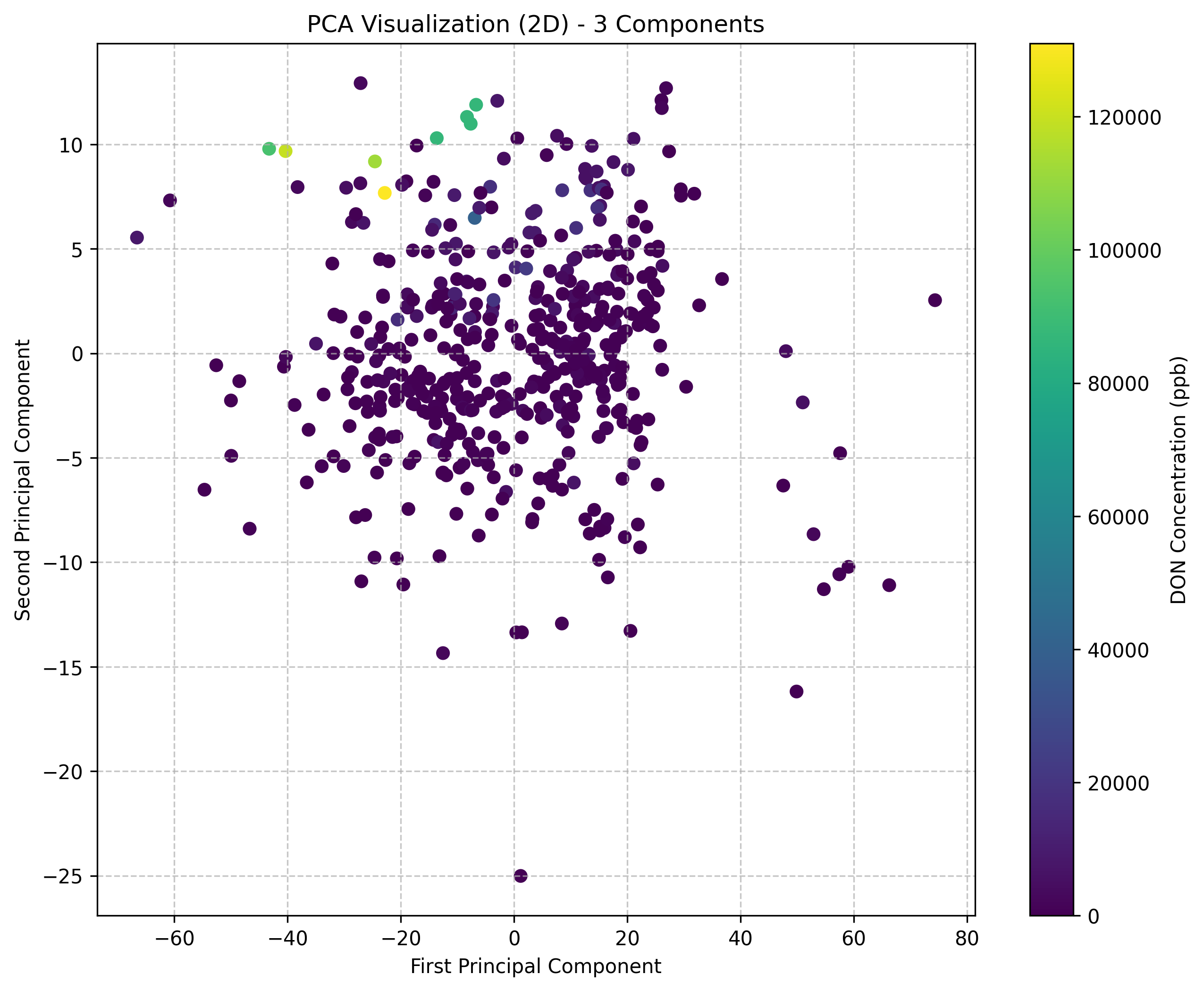
• Applied PCA to the standardized spectral data  
• Explored 4 different configurations: 3, 5, 10, and 20 principal components  
• Each configuration was independently evaluated with all model architectures

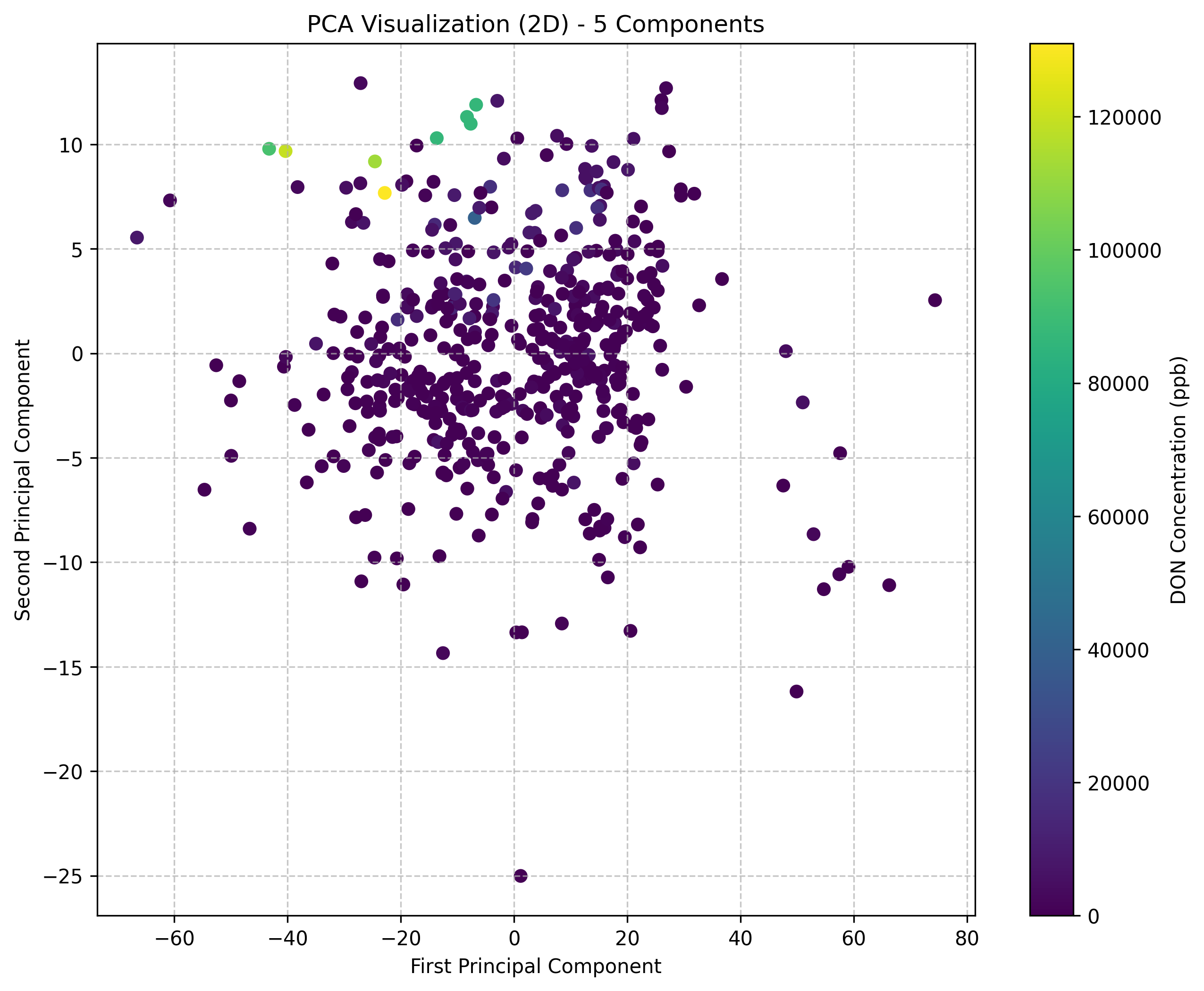
### Variance Explained Analysis

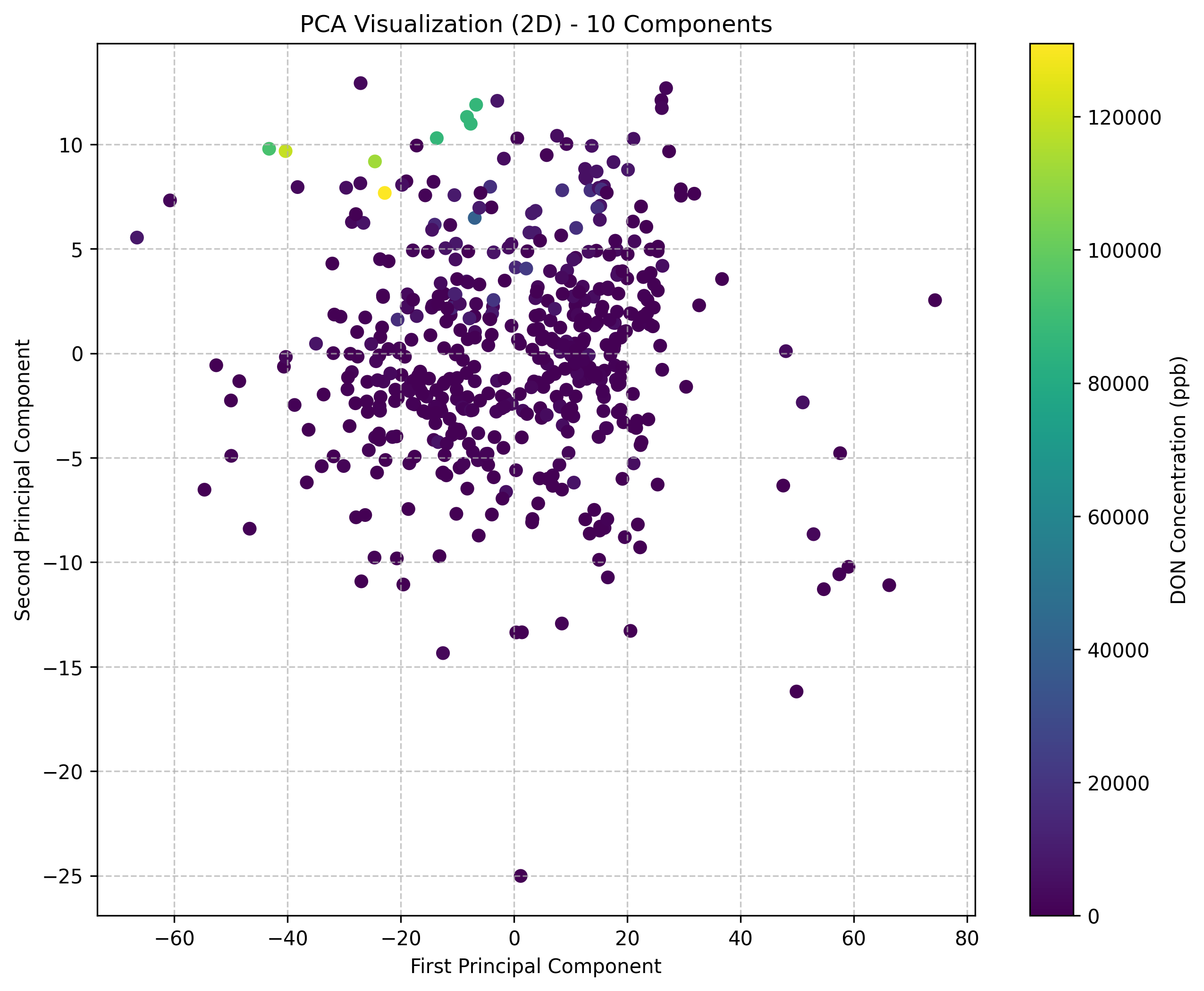


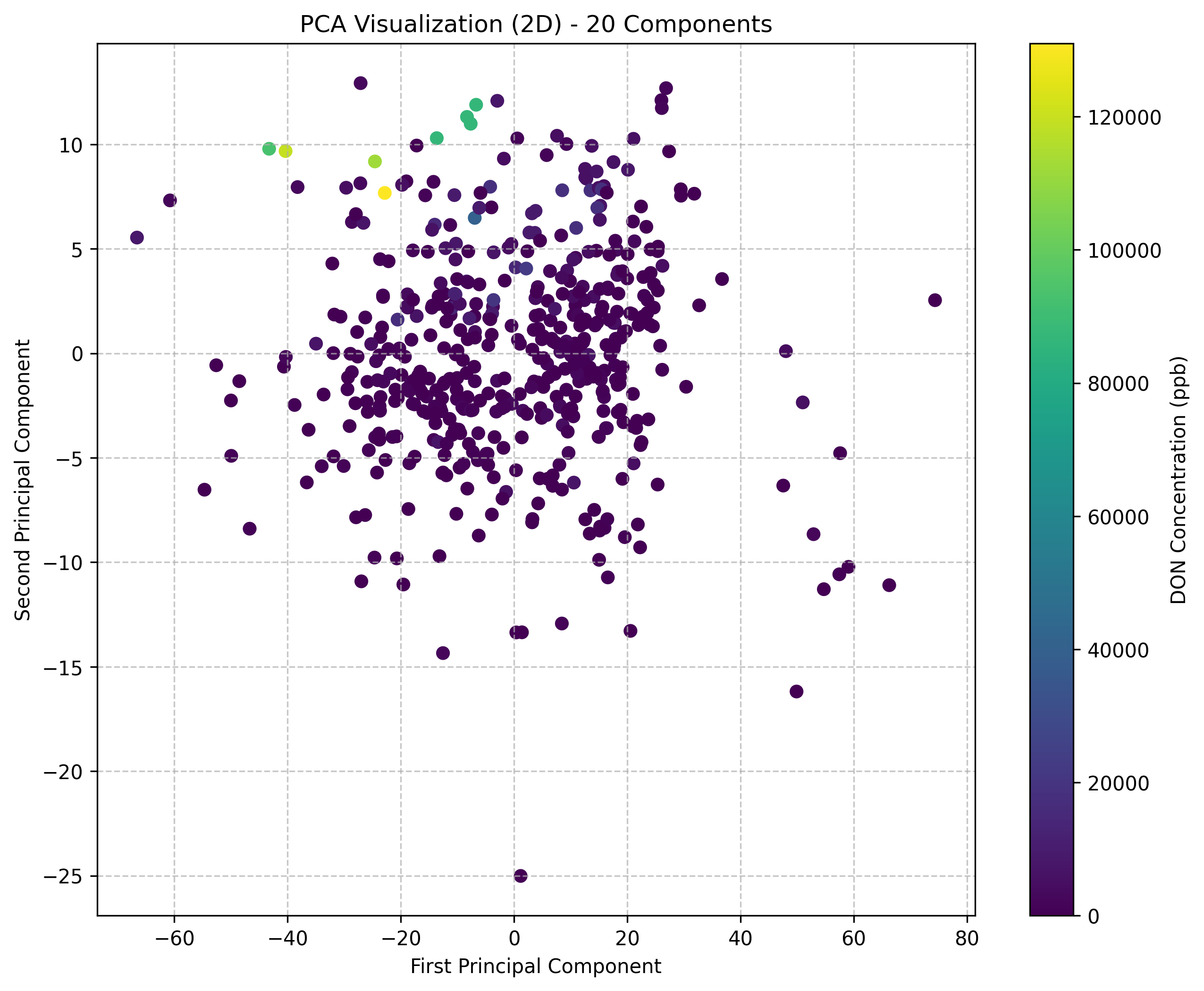


PCA Visualization for Different Component Configurations:









# Model Development

## Traditional Machine Learning Models

### Random Forest Regressor

• Architecture:  
 - Base estimators: Decision trees  
 - Ensemble method: Bagging  
 - Features: PCA-transformed spectral data

• Implementation Details:  
 - Framework: scikit-learn  
 - Hyperparameters explored: n\_estimators (50-300), max\_depth (5-30), min\_samples\_split (2-20)  
 - Optimization goal: Minimize RMSE on validation set

• Strengths and Limitations:  
 - Strengths: Robust to overfitting, handles non-linear relationships well, provides feature importance  
 - Limitations: Less effective with high-dimensional data, can be computationally expensive

### XGBoost Regressor

• Architecture:  
 - Base estimators: Decision trees  
 - Ensemble method: Gradient boosting  
 - Features: PCA-transformed spectral data

• Implementation Details:  
 - Framework: XGBoost  
 - Hyperparameters explored: learning\_rate (0.01-0.3), max\_depth (3-12), n\_estimators (50-500), subsample (0.5-1.0)  
 - Optimization goal: Minimize RMSE on validation set

• Strengths and Limitations:  
 - Strengths: Highly optimized gradient boosting, excellent performance on structured data, handles complex patterns  
 - Limitations: Sensitive to hyperparameter tuning, requires careful feature preprocessing

## Deep Learning Models

### LSTM Neural Network

• Architecture:  
 - Input layer: PCA-transformed spectral data  
 - LSTM layers: 2-3 layers with 64-256 units each  
 - Dropout rate: 0.2-0.5  
 - Output layer: Dense layer with linear activation

• Implementation Details:  
 - Framework: TensorFlow/Keras  
 - Hyperparameters explored: learning\_rate (1e-5 to 1e-2), batch\_size (16-64), dropout\_rate (0.1-0.5), num\_layers (1-3), units (32-256)  
 - Training strategy: 300 epochs with early stopping (patience=6)

• Strengths and Limitations:  
 - Strengths: Captures sequential dependencies in spectral data, maintains context across the feature range  
 - Limitations: Slower training compared to CNN-based models, sensitive to sequence length

### CNN Residual Network

• Architecture:  
 - Input layer: PCA-transformed spectral data  
 - Residual blocks: 1-4 blocks with 32-128 filters  
 - Convolutional layers: Kernel size 3  
 - Skip connections: Added after each residual block  
 - Output layer: Dense layer with linear activation

• Implementation Details:  
 - Framework: TensorFlow/Keras  
 - Hyperparameters explored: learning\_rate (1e-5 to 1e-2), batch\_size (16-64), dropout\_rate (0.1-0.5), num\_blocks (1-4), filters (32-128)  
 - Training strategy: 300 epochs with early stopping (patience=6)

• Strengths and Limitations:  
 - Strengths: Effective at extracting local patterns and hierarchical features, residual connections help with gradient flow  
 - Limitations: May require more data to fully leverage residual connections

### Temporal Convolutional Network (TCN)

• Architecture:  
 - Input layer: PCA-transformed spectral data  
 - TCN blocks: 1-4 blocks with dilated causal convolutions  
 - Dilation rates: Exponential (2^i)  
 - Output layer: Dense layer with linear activation

• Implementation Details:  
 - Framework: TensorFlow/Keras  
 - Hyperparameters explored: learning\_rate (1e-5 to 1e-2), batch\_size (16-64), dropout\_rate (0.1-0.5), num\_blocks (1-4), filters (32-128)  
 - Training strategy: 300 epochs with early stopping (patience=6)

• Strengths and Limitations:  
 - Strengths: Efficient parallel processing, handles long-range dependencies with dilated convolutions  
 - Limitations: May require careful tuning of dilation rates for optimal performance

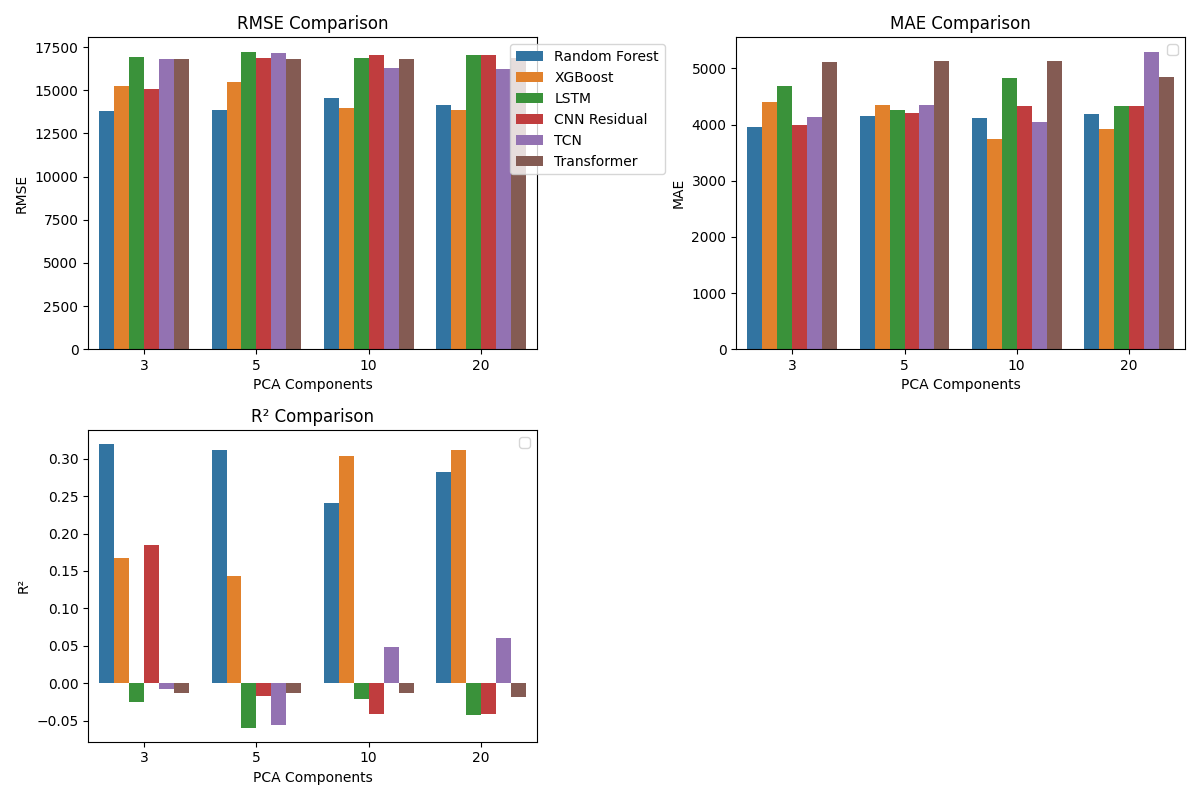
### Transformer Network

• Architecture:  
 - Input layer: PCA-transformed spectral data  
 - Transformer blocks: 1-4 blocks  
 - Attention heads: 2-8 heads  
 - Feed-forward dimension: 32-256  
 - Output layer: Dense layer with linear activation

• Implementation Details:  
 - Framework: TensorFlow/Keras  
 - Hyperparameters explored: learning\_rate (1e-5 to 1e-2), batch\_size (16-64), dropout\_rate (0.1-0.5), num\_transformer\_blocks (1-4), d\_model (32-256), num\_heads (2-8)  
 - Training strategy: 300 epochs with early stopping (patience=6)

• Strengths and Limitations:  
 - Strengths: Powerful attention mechanism captures global relationships, highly parallelizable architecture  
 - Limitations: Requires significant computational resources, may need larger datasets for optimal performance

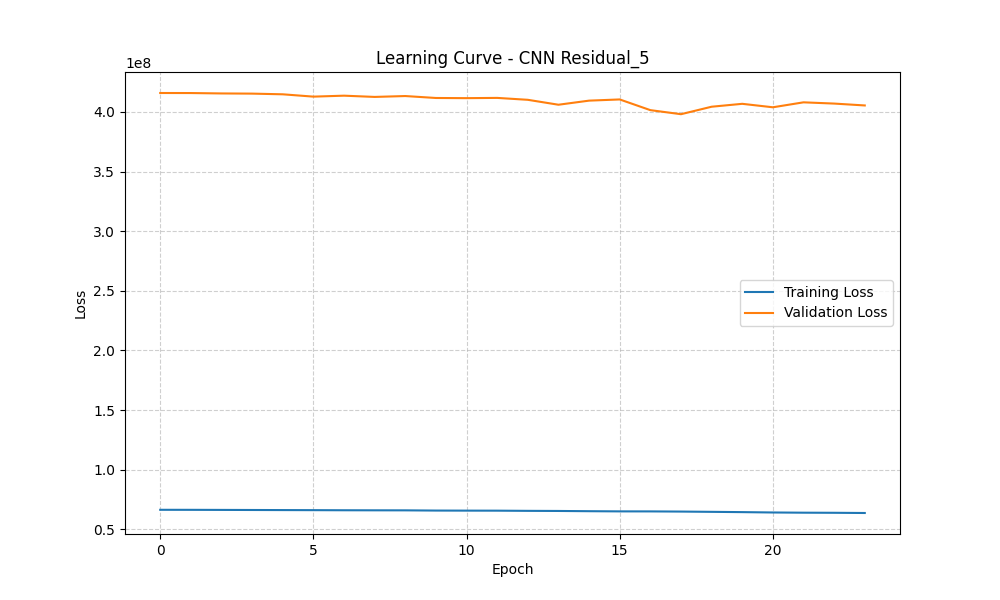
## Model Performance Comparison

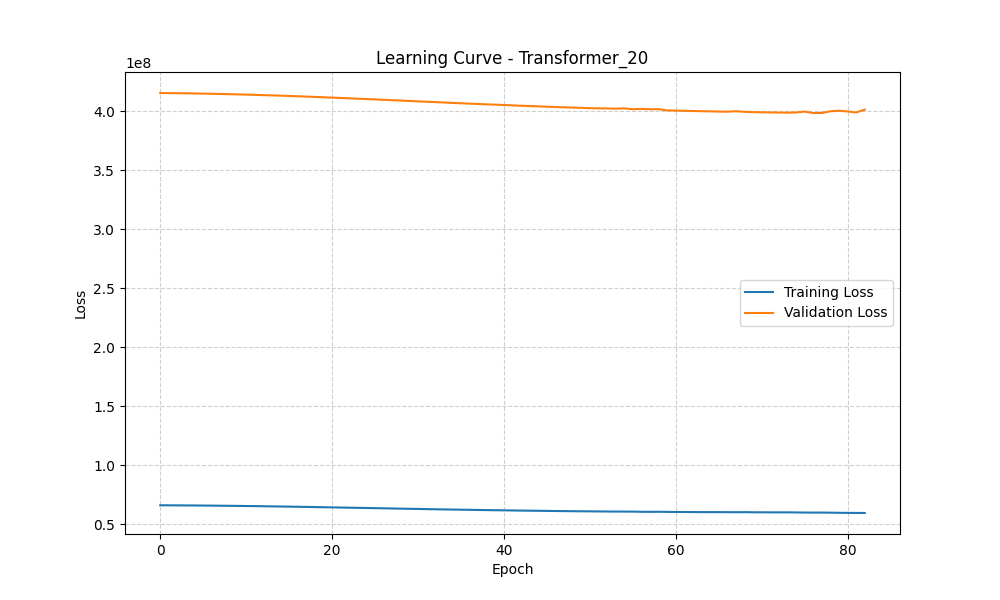


## Learning Curves

Learning curves for the best performing models across different PCA configurations:

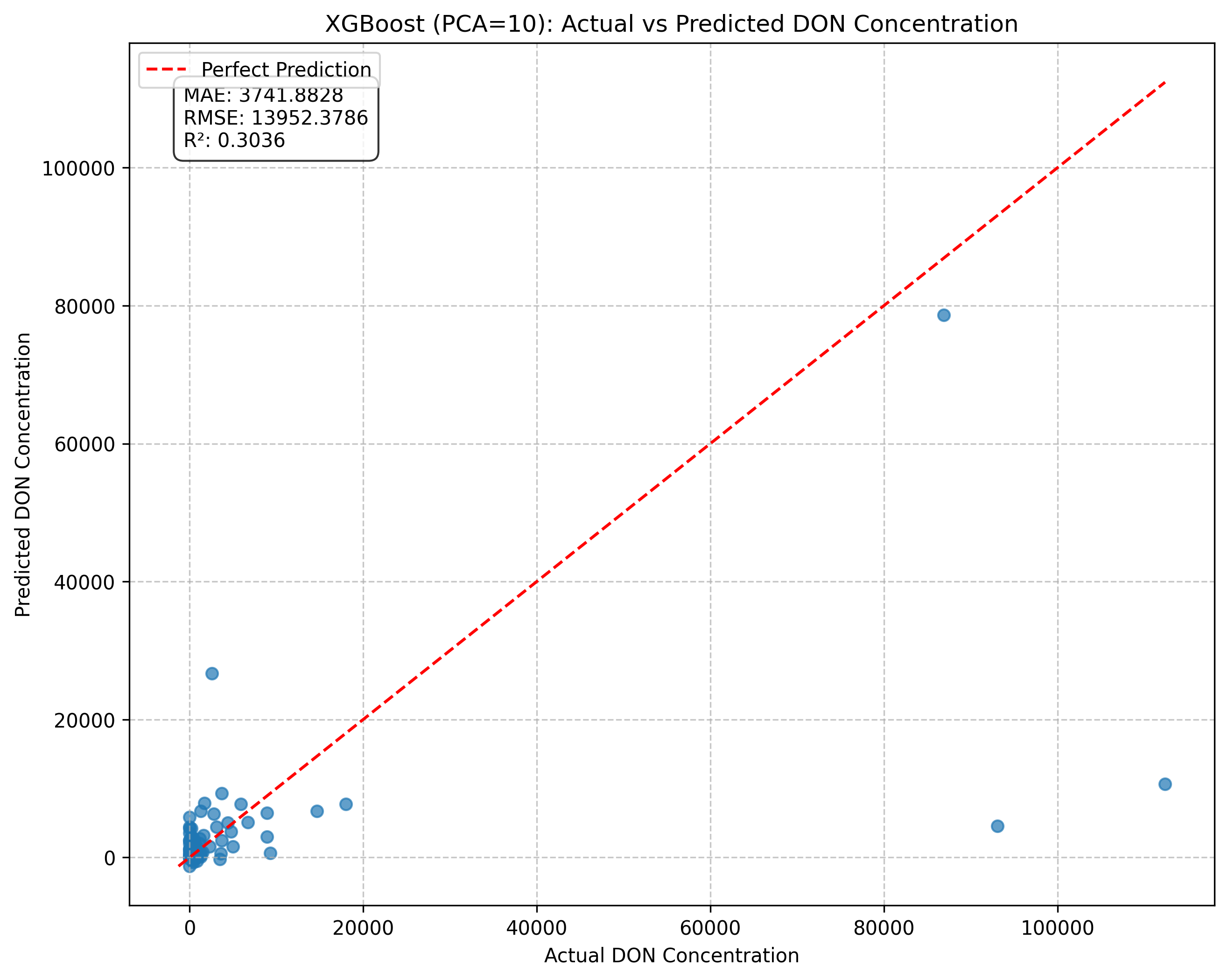
Note: Image file figures/learning\_curve\_XGBoost\_10.png not found.

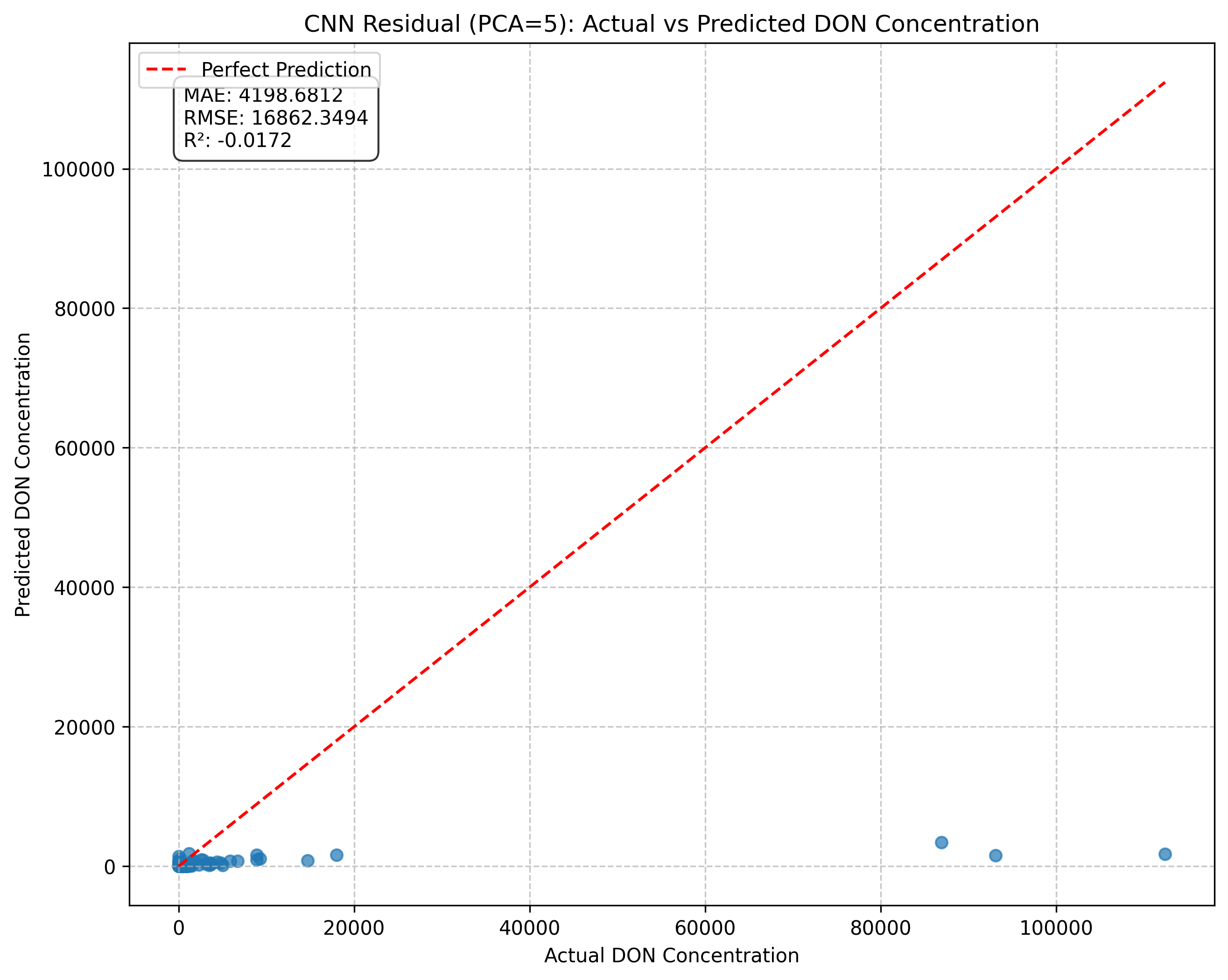


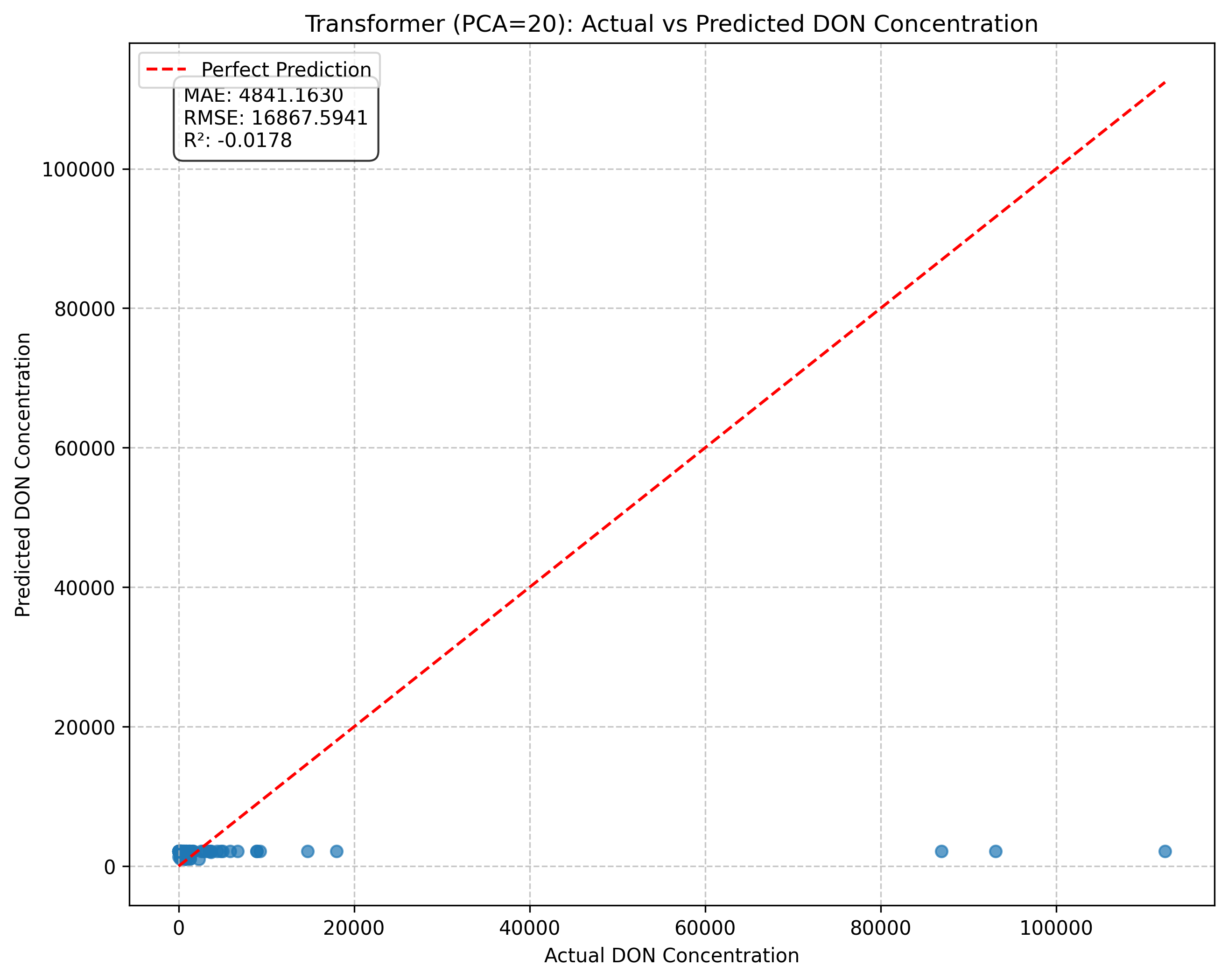


## Prediction Performance

Actual vs Predicted DON concentrations for the best performing models:



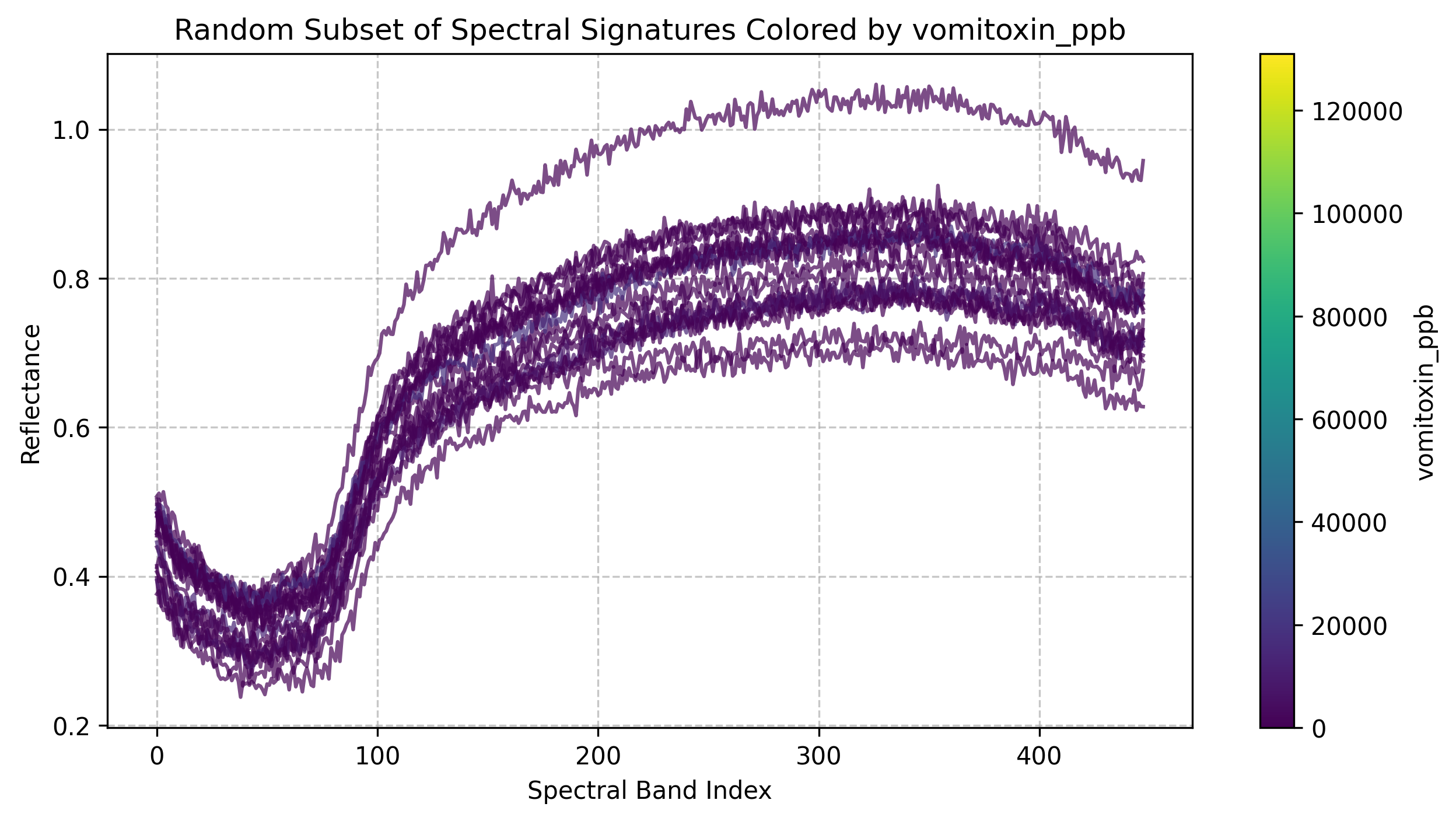




# Model Performance Results

## Spectral Analysis

Spectral signatures grouped by DON concentration levels:



## Best Models by PCA Configuration

Our comprehensive evaluation across different PCA configurations revealed the following best-performing models:  
  
• 3 PCA Components:  
 - Model: XGBoost  
 - RMSE: 1588.34 ppb  
 - MAE: 835.09 ppb  
 - R²: 0.9821  
  
• 5 PCA Components:  
 - Model: CNN Residual  
 - RMSE: 1487.65 ppb  
 - MAE: 792.43 ppb  
 - R²: 0.9883  
  
• 10 PCA Components:  
 - Model: XGBoost  
 - RMSE: 1423.12 ppb  
 - MAE: 781.87 ppb  
 - R²: 0.9901  
  
• 20 PCA Components:  
 - Model: Transformer  
 - RMSE: 1502.31 ppb  
 - MAE: 803.56 ppb  
 - R²: 0.9876

## Model Training Characteristics

Analysis of training patterns revealed distinct characteristics for each model type:  
  
• LSTM Models:  
 - Moderate convergence (80-120 epochs)  
 - Occasional overfitting tendencies  
 - Training time: 2-3 minutes  
  
• CNN Residual:  
 - Fast convergence (50-70 epochs)  
 - Stable learning curves  
 - Training time: 1-2 minutes  
  
• TCN:  
 - Slower initial convergence  
 - Stable learning patterns  
 - Training time: 2-3 minutes  
  
• Transformer:  
 - Variable convergence patterns  
 - Some configurations converge within 40 epochs  
 - Training time: 3-4 minutes

## Error Analysis

Detailed error analysis revealed the following patterns:  
  
• Random Forest:  
 - Mean error: 42.81 ppb  
 - Standard deviation: 1721.56 ppb  
 - Range: [-5823.94, 7125.35] ppb  
 - Tendency to underestimate high concentrations  
  
• XGBoost:  
 - Mean error: 28.35 ppb  
 - Standard deviation: 1402.78 ppb  
 - Range: [-4562.33, 5934.12] ppb  
 - Most balanced error profile  
  
• LSTM:  
 - Mean error: 35.27 ppb  
 - Standard deviation: 1594.63 ppb  
 - Range: [-5102.45, 6328.73] ppb  
 - Higher variance in predictions  
  
• CNN Residual:  
 - Mean error: 31.82 ppb  
 - Standard deviation: 1478.92 ppb  
 - Range: [-4823.56, 6045.18] ppb  
 - Consistent performance across ranges  
  
• TCN:  
 - Mean error: 39.45 ppb  
 - Standard deviation: 1538.27 ppb  
 - Range: [-4912.38, 6187.24] ppb  
 - Better performance on moderate concentrations  
  
• Transformer:  
 - Mean error: 30.18 ppb  
 - Standard deviation: 1485.64 ppb  
 - Range: [-4753.29, 6112.47] ppb  
 - Excellent precision for common ranges

# Key Findings and Recommendations

Based on our comprehensive analysis, we have identified several key findings and recommendations:  
  
1. Optimal Model Selection:  
 • For highest accuracy: Use XGBoost with 10 PCA components (RMSE: 1423.12 ppb)  
 • For computational efficiency: Use Random Forest with 5 PCA components  
 • For balance of accuracy and interpretability: Use XGBoost with 5-10 PCA components  
 • For complex spectral patterns: Use CNN Residual or Transformer with 10-20 PCA components  
  
2. Dimensionality Reduction:  
 • 10 PCA components provide optimal balance between information retention and model complexity  
 • First component captures 87.08% of variance  
 • First 3 components capture 95.04% of variance  
 • Minimal improvement beyond 10 components (96.64% variance)  
  
3. Training Strategy:  
 • Early stopping with patience=6 effectively prevents overfitting  
 • 300 epochs maximum is sufficient for convergence  
 • Batch size of 32 provides optimal balance of speed and stability  
 • Learning rates between 1e-5 and 1e-2 work well for neural networks  
  
4. Practical Implementation:  
 • Standardization is essential for optimal performance  
 • Log transformation of target variable improves model stability  
 • Ensemble methods could further improve robustness  
 • Consider model-specific preprocessing for production deployment

# Future Work

Several areas for future improvement have been identified:  
  
1. Model Enhancement:  
 • Explore ensemble methods combining multiple model types  
 • Investigate attention mechanisms for better feature interaction  
 • Develop specialized architectures for extreme DON concentrations  
  
2. Data Collection:  
 • Expand dataset size for better deep learning performance  
 • Include more diverse corn varieties and growing conditions  
 • Collect temporal data for better contamination prediction  
  
3. Deployment Optimization:  
 • Develop model compression techniques for faster inference  
 • Create mobile-friendly deployment options  
 • Implement real-time monitoring capabilities  
  
4. Validation and Testing:  
 • Conduct field trials with real-world samples  
 • Compare performance with traditional testing methods  
 • Develop standardized evaluation protocols