Data-Centric Ensemble Machine Learning Framework for Predicting and Optimizing MEA Power Density in PEMFC

Abstract

Proton exchange membrane fuel cells (PEMFCs) stand at the forefront of clean energy technology, offering high efficiency and zero emissions. At their core, membrane electrode assemblies (MEAs) dictate performance, yet their optimization has long relied on costly, time-consuming experimental methods. This study revolutionizes MEA design by leveraging a robust machine learning (ML) framework trained on 799 experimental records from over 800 peer-reviewed publications. Using advanced ensemble algorithms like CatBoost, XGBoost, and Gradient Boosting, the framework achieves unparalleled predictive accuracy for maximum power density (MPD) while uncovering hidden design rules. By replacing trial-and-error with data-driven insights, this work paves the way for faster, cheaper, and smarter PEMFC development.

1. Introduction

Proton exchange membrane fuel cells (PEMFCs) are a promising energy conversion technology owing to their high efficiency, low operating temperature, and zero-emission characteristics. The membrane electrode assembly (MEA) serves as the core component of PEMFCs, directly influencing their power output, lifetime, and stability. Traditionally, the development and optimization of MEAs have been achieved through iterative experimental methods. These approaches, while reliable, are time-consuming, expensive, and inherently limited in their ability to explore high-dimensional and nonlinear parameter spaces. Furthermore, existing computational and theoretical models often fail to capture the complex nonlinear interactions between catalyst properties, support materials, fabrication techniques, and operational conditions. Despite recent advancements in artificial intelligence (AI), the application of machine learning (ML) in PEMFC research remains underutilized and largely confined to small datasets or narrow parameter spaces. Many existing studies employ standard ML algorithms without rigorous model selection or hyperparameter tuning, leading to poor generalizability and limited practical value.

To address these challenges, a data-driven ML framework has been developed that systematically extracts, preprocesses, and models experimental data from the PEMFC literature. The aim is to provide accurate performance prediction and uncover interpretable relationships between design variables and MEA power output.

1.2. Problem Statement

Experimental optimization of PEMFC components, especially the MEA, is resource-intensive and time-consuming. Traditional methods involve repeated laboratory testing, lacking scalability and generalizability. Moreover, existing computational tools are often limited in scope, failing to capture the complex, nonlinear, and multidimensional relationships inherent in MEA design. Past ML approaches have also suffered from small sample sizes, limited features, or weak modeling techniques, rendering them inadequate for practical deployment.

1.3. Objective

The primary objectives of this framework are:

- ✓ Predict MEA maximum power density (MPD) using ML based on key fabrication and operating parameters.
- ✓ Identify the most impactful features influencing MEA performance.
- ✓ Build a reproducible, scalable ML pipeline to streamline MEA development.
- ✓ Discover optimal catalyst compositions and configurations for higher MPD.
- ✓ Explore interactions among synthesis and operational parameters.
- ✓ Benchmark ML performance against traditional prediction methods.
- ✓ Support data-driven decisions in fuel cell design and materials selection.

1.4. Solution Approach

A data-centric ML pipeline that harnesses decades of global research. By curating and modeling a vast, diverse dataset, we bridge the gap between empirical research and AI-driven discovery, offering a roadmap to high-performance MEAs. The following multi-stage ML pipeline was constructed:

1. **Data Compilation**: 797 MEA experimental records from 800 articles (2003-2020).
2. **Preprocessing**: Handled missing values, outliers, normalization, and encoding.
3. **Model Training**: Evaluated multiple ML algorithms with rigorous cross-validation and hyperparameter tuning.
4. **Insight Extraction**: Applied Association Rule Mining for Interpretable Feature Interactions.
5. **Visualization and Validation**: Regression plots, correlation heatmaps, and feature importance charts supported findings.

2. Dataset Construction and Feature Definition

2.1 Data Sources and Compilation

A total of 799 MEA experimental entries were compiled from 800 peer-reviewed journal articles published between 2003 and 2020. These articles collectively span 206 authors and 161 institutions, providing a diverse and representative dataset of global PEMFC research.

2.2 Feature Categories

The compiled dataset includes 66 features, categorized as follows:

- **Catalyst Composition:** Atomic % of Pt, Pd, Co, Ni, Ir, Au, Fe, and others; metal loadings; support materials (e.g., carbon black, CNTs, graphene).
- **Synthesis Parameters:** Reduction methods (chemical, thermal, microwave), annealing conditions, use of reducing agents (e.g., $NaBH_4$, ethylene glycol).
- **Structural Properties:** BET surface area, particle size, core-shell or nanowire structures.
- **Fabrication Variables:** Nafion thickness, I/C ratio, hot pressing temperature and time.
- **Operating Conditions:** Backpressure, temperature, humidity, and gas composition.
- **Performance Metrics:** Open circuit voltage (OCV), current densities, voltage at load points, and maximum power density (MPD in mW/cm^2).

3. Data Preprocessing, Wrangling and Feature Engineering

- Missing values: Median/mode imputation technique (fillna(), SimpleImputer)
- Outlier handling: IQR-based filtering
- Encoding: One-hot for categorical features
- Scaling: StandardScaler normalization
- Correlation filtering: Pearson correlation > 0.8 removed
- PCA tested but discarded due to model degradation

3.1. Missing Data Imputation

Missing values were imputed using robust techniques:

- **Numerical features:** Replaced with median values to avoid distortion from outliers.
- **Categorical features:** Replaced with mode values followed by one-hot encoding.

3.2. Outlier Detection and Treatment

Outliers were identified using the interquartile range (IQR) method. Rather than outright removal, insightful outliers were retained to preserve generalizability, particularly those reflecting high-performance MEAs as shown in Figure 1.

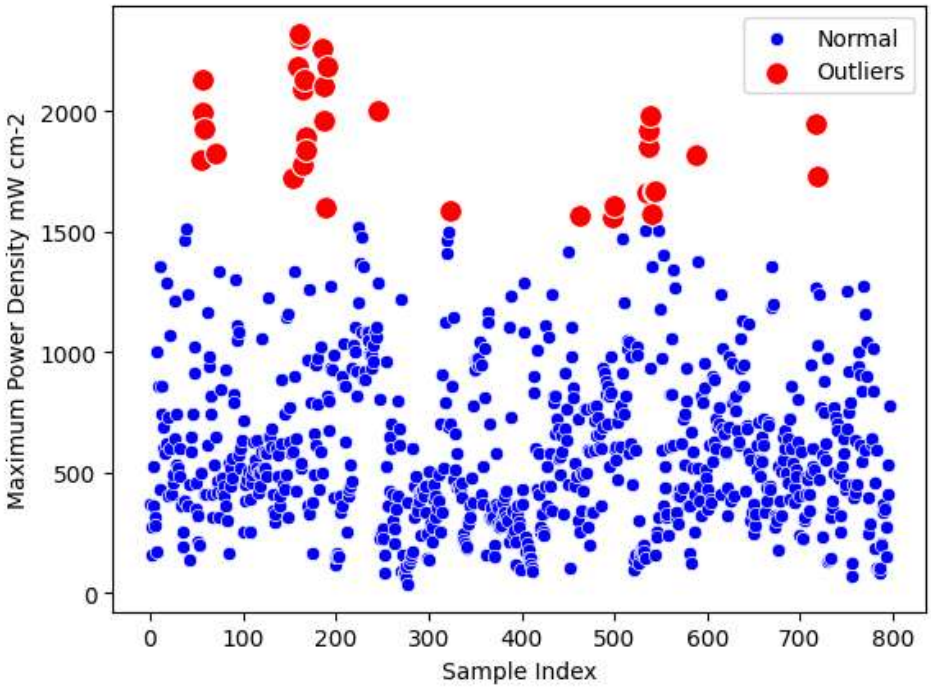


Figure 1. (IQR) method for determining the outlier

3.3. Feature Normalization and Encoding

All numerical features were standardized using StandardScaler (zero mean, unit variance). Categorical variables were encoded using one-hot encoding. Correlated features (Pearson correlation coefficient > 0.8) were removed to reduce multicollinearity.

3.4. Dimensionality Reduction

Principal Component Analysis (PCA) was tested at 95% and 99% retained variance thresholds. However, results showed a substantial decline in model performance (mean $R^2 < 0$), indicating that PCA removed critical nonlinear features. Therefore, PCA was excluded from the final workflow.

4. Feature Selection and Target Variable

XGBoost's built-in feature importance and Lasso regularization were employed for variable selection. Of the initial 66 features, 27 were retained as the most relevant to MPD prediction. These include:

- Pt loading (cathode side)
- Nafion membrane thickness
- Catalyst mass activity (MA)
- BET surface area
- Backpressure
- Voltage at 2500 mA/cm²

The target variable for regression modeling was Maximum Power Density (MPD), ranging from 35.5 to 2321.7 mW/cm², with a mean of 644.25 mW/cm² and a right-skewed distribution (skewness = 1.38).

5. Machine Learning Models and Evaluation

5.1. Algorithms Applied

Six supervised machine learning models were evaluated in this study:

- Decision Tree Regressor
- Random Forest Regressor
- Gradient Boosting Regressor
- XGBoost Regressor
- CatBoost Regressor
- AdaBoost Regressor

In addition to these individual models, ensemble strategies such as stacking with XGBoost as the meta-learner and weighted voting were employed to further enhance prediction robustness and overall model accuracy.

Why Use Bagging and Boosting?

Bagging and boosting are ensemble techniques designed to improve model performance through different mechanisms:

- **Bagging (Bootstrap Aggregating):**
  Constructs multiple models independently using different bootstrapped subsets of the training data, then aggregates predictions through averaging (regression) or voting (classification). This method reduces model variance and mitigates overfitting. *Example: Random Forest*

- **Boosting:**
  Builds models sequentially, where each new model focuses on correcting the errors made by its predecessors. By assigning greater importance to previously misclassified or high-error instances, boosting effectively reduces both bias and variance. *Examples: Gradient Boosting, XGBoost, CatBoost, AdaBoost*

These ensemble techniques have proven to significantly improve the predictive accuracy of base learners, making them ideal choices for high-performance regression tasks.

5.2. Cross-Validation and Hyperparameter Tuning

ML model was optimized through **5-fold cross-validation**, which involves splitting the training data into five subsets. The model is trained on four subsets and validated on the remaining one, repeating this process five times to ensure reliable and unbiased performance evaluation. Additionally, **GridSearchCV** was used to systematically search for the best combination of hyperparameters by exhaustively testing predefined parameter grids. This approach helps in selecting the optimal model settings that maximize predictive accuracy and generalization.

5.3. Performance Metrics

The models were evaluated using key metrics such as the **Coefficient of Determination (R^2)**, **Root Mean Squared Error (RMSE)**, and **Mean Absolute Error (MAE)** to measure their accuracy and prediction errors. The evaluation aimed for an RMSE below 150 mW/cm² and a high R^2 value, indicating strong predictive performance. These regression models enable accurate forecasting of membrane electrode assembly (MEA) performance, which helps to reduce the need for extensive and costly experimental trials, thereby speeding up the development process through data-driven insights.

Data Preprocessing Pipeline

- Feature Cleanup: Removed malformed columns and standardized column names
- Null Handling: Applied fillna() with statistical imputation
- Scaling: StandardScaler for numerical values
- Encoding: OneHotEncoder for categorical variables
- Split: 80% training / 20% test (preventing leakage)

Data Splitting & Features Overview

The features (X) include all input variables except the target, such as Pt Loading, Nafion Thickness, Backpressure, voltage measurements across a current range from 100 mA/cm² to 3000 mA/cm², and other relevant experimental parameters. The target (y) variable is the Power Density measured in mW/cm².

Multicollinearity & Feature Selection

Multicollinearity was addressed by removing highly correlated features (correlation greater than 0.8) to reduce redundancy. Features with low variance (less than 0.1) were also dropped due to their limited predictive value. Model complexity was controlled through regularization techniques, such as limiting max_depth, min_samples_split, and min_samples_leaf in Random Forest, and using subsampling along with other parameters in Gradient Boosting to prevent overfitting. A 5-fold cross-validation was employed to ensure robust and reliable performance estimates. Finally, permutation importance was used to rank features based on their contribution to model accuracy.

6. Results and Analysis

6.1. Exploratory Data Analysis (EDA)

6.1.1. Descriptive Statistics

The dataset spans from 2003 to 2020, with a mean year of 2014, meaning most experiments were conducted in the last decade. The range of years is 17 years, showing long-term research activity. Platinum (Pt) has a mean (average) atomic percent of 86.61%, meaning most catalysts were primarily Pt-based. Its median (50th percentile) is 100%, which shows that more than half the samples used pure Pt. The range is from 0% to 100%, but the data is skewed toward high Pt content. Other metals like Pd, Au, Ir, and Ru have very low means (all under 6%) and medians of 0%, meaning they were used in very few samples. However, their maximum values (e.g., Pd at 100%, Au at 50%) show that a small number of formulations used them in high amounts. Cobalt (Co) and Iron (Fe) also show low means (4.26% and 1.39%) with high standard deviations (12.21% and 7.33%), indicating high variability, they were used significantly in a few cases, but mostly absent. For performance, voltage at 2100 mA/cm² has a mean of 0.205 V and a median of 0.1505 V, with a range from 0.01 to ~0.37 V. This shows that while some catalysts performed poorly (higher voltages), many performed efficiently (lower voltages). At 3000 mA/cm², the mean voltage drops to 0.120 V, and the median is just 0.01 V, suggesting even better performance under higher stress in many cases.

Table 1: Descriptive Statistics

Statistic	Year	Pt (at%)	Pd (at%)	Au (at%)	Ir (at%)	Ru (at%)	Co (at%)	Fe (at%)	Voltage @2100 mA/cm²	Voltage @3000 mA/cm²
Count	898	899	899	899	896	899	899	899	898	899
Mean	2014.0	86.61	5.10	0.50	0.46	0.59	4.26	1.39	0.2053	0.1196
Std	4.66	23.27	19.99	4.38	4.90	5.07	12.21	7.33	0.204	0.166
Min	2003	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0100	0.0018
25%	2011	75.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0100	0.0100
50%	2015	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.1505	0.0100
75%	2018	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.3700	0.1947
Max	2020	100.00	100.00	50.00	100.00	50.00				

6.1.2. Target

The distribution of power density is heavily right-skewed, suggesting that while most MEAs produce between 500-700 mW/cm², there are a few high-performing MEAs exceeding 2000 mW/cm², likely due to optimized catalysts or operating conditions. These outliers might offer insights into optimal parameters but also require careful handling to avoid model bias as shown Figure2.

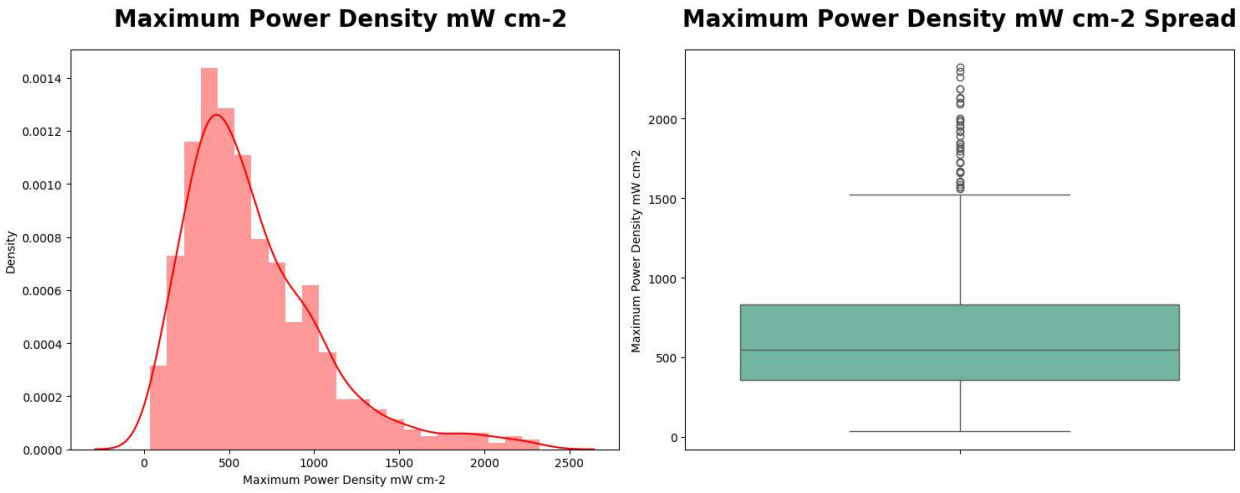


Figure2: Target-power density

6.1.3. Correlation Analysis & Multicollinearity Assessment

As displayed in Figure 3, The comprehensive correlation and multicollinearity analysis was conducted to identify key predictors and eliminate redundant or irrelevant features in the PEM fuel cell dataset. The strongest correlation was observed between Open Circuit Voltage (OCV) and Maximum Power Density (MPD)**,** with a perfect correlation coefficient of 1.00**,** indicating that OCV is corrolate of MPD with (r=0.15). Similarly, Voltage at 2500 mA/cm² showed a very high correlation with MPD **(**r = 0.94**)**, but it is retained due to its critical operational relevance in performance prediction that should be removed to avoid multicollinearity.

Moderate correlations such as Cell Temperature (r = 0.36) and Back Pressure (r = 0.30) with MPD reflect known electrochemical principles, as elevated temperature and pressure enhance reaction kinetics and mass transport. Additionally, a moderate correlation between Co at% and Pt at% (r = 0.43) suggests potential synergistic alloy effects worth further exploration, whereas pure Pt loading exhibited a weak negative correlation (r = -0.14), indicating diminishing returns with higher Pt content. However, XGBOOST showed the impact as nonlinear parameters that was predicted by ML models.

Variables with negligible or no correlation to MPD such as Pd at% (r = -0.05)**,** Au at% (r = -0.02**)**, and Humidity (r = 0.10) were deemed uninformative and are recommended for removal to simplify the model. The pairwise correlation between Voltage at 2500 mA/cm² and Voltage at 100 mA/cm² (r = 0.44) is within acceptable multicollinearity bounds, pending verification using Variance Inflation Factor (VIF) analysis, where values below 5 are generally safe.

Visualization of the power density distribution revealed a right-skewed pattern, suggesting that a log transformation may improve the assumptions of linear modeling techniques. The heatmap analysis further emphasized the upper-right quadrant, which clustered performance-critical variables.

In conclusion, the most informative and retained features for predictive modeling include Voltage at 2500 mA/cm²**,** Cell Temperature**,** Co at%**,** and Back Pressure, while OCV**,** Pd at%**,** Au at%**,** and Humidity should be excluded to enhance model efficiency and accuracy.

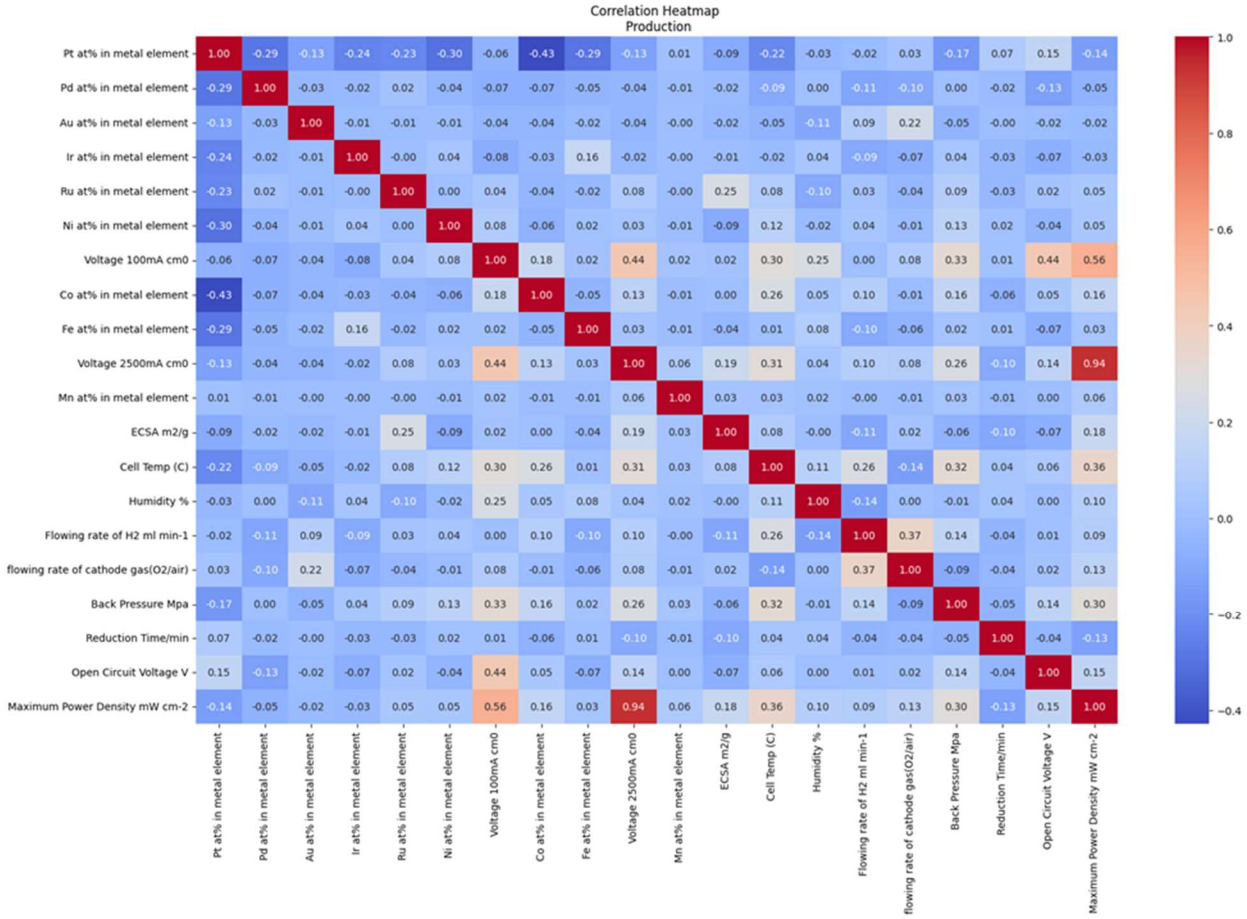


Figure3. Correlation Analysis

The analysis reveals key features structure–performance relationships in electrocatalyst and fuel cell systems:

- Palladium content is most effective in the 40–60 at% range; higher amounts offer no additional benefit and increase cost.
- Increasing the total metal loading improves performance up to ~60 wt%**,** beyond which returns diminish.
- BET surface area in the range of 600–1000 m²/g provides optimal active site exposure; too low limits reaction sites, while too high can cause pore blockage.
- Operating temperature strongly influences performance, with an ideal window between 60–80°C**;** temperatures above this range lead to membrane dehydration and performance loss.
- Electrochemically active surface area (ECSA) values above 150 m²/g correlate with higher power density, though extremely high values may compromise stability.

Together, these findings highlight the importance of balanced material design and operating conditions to achieve high power output, stability, and cost-efficiency in electrochemical energy systems as shown Figure 4.

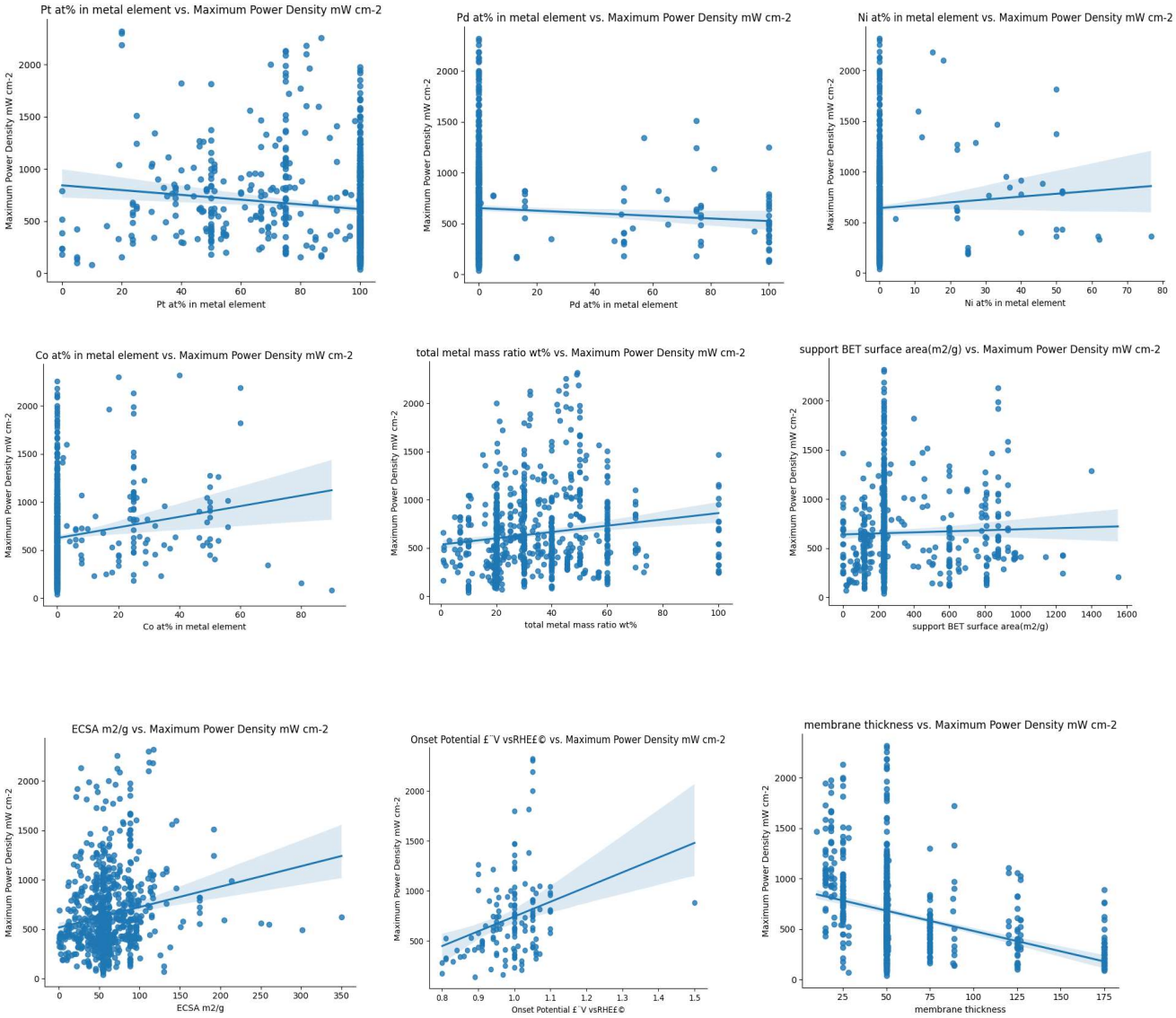


Figure 4. Relation the key features

6.1.4. Power Density vs. Feature (Membrane): The Battle of Efficiency

This dataset compares power density across various fuel cell membranes, with Nafion-based membranes clearly dominating the performance landscape as shown in Figure 5.

✓ **Top Performers**

Membranes labeled Nafion 212 consistently reach or exceed 200 mW/cm², marking them as benchmarks for high-efficiency performance. The repeated success of these entries highlights Nafion 212 as a go-to standard in PEM fuel cell research.

✓ **Nafion Leads the Field**

Multiple variants, Nafion 115, 112, HP, and XL appear frequently, confirming Nafion's status as the industry standard. Modified versions such as Nafion/TiO_2 composites indicate ongoing efforts to enhance conductivity, mechanical stability, or water retention.

✓ **Emerging Alternatives**

Experimental membranes like SPEEK/PP-Cs2.5 and sulfonated polystyrene/PVC represent attempts to develop cost-effective or higher-performing alternatives. These underdog materials suggest a healthy innovation pipeline aimed at competing with or surpassing Nafion.

✓ **Data Integrity Notes**

Inconsistencies like Nation vs. Nafion likely reflect typographical errors or informal naming. This highlights the need for careful data cleaning in meta-analyses or AI-driven materials discovery.

Conclusion:
While Nafion remains the gold standard, the data shows active research into modified Nafion membranes and alternative materials. The drive toward membranes exceeding 200 mW/cm² reflects the field's focus on balancing efficiency, durability, and cost. The search for the next-generation fuel cell membrane is very much alive.

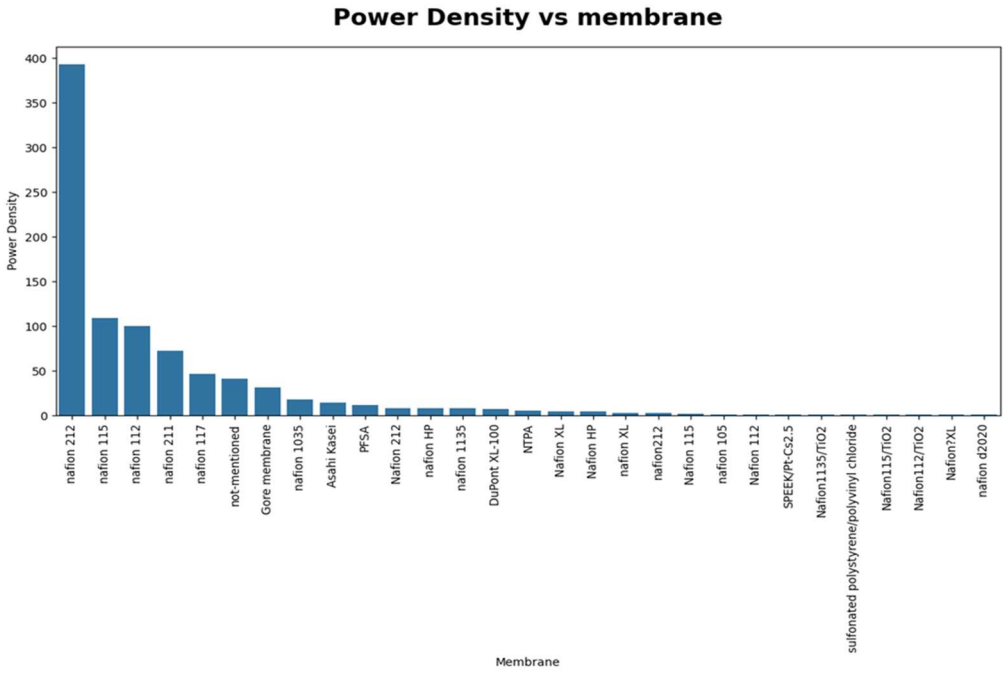


Figure 5. Power Density vs. Feature (Membrane)

6.1.5. The Story of Fuel Cell Structures & Substrates

About 86% of anode samples use plain Pt/C catalysts (Figure 6) due to their cost-effectiveness and scalability. Core-shell structures (6.6%) offer better performance but face durability issues. Exotic nanostructures like nanowires and nanodendrites are rare and mostly experimental. Vulcan carbon (XC-72R) dominates as the carbon support (48.5%) because it's cheap and reliable but degrades over time. Carbon nanotubes (11.5%) provide high conductivity but are costly and hard to process. Nitrogen-doped carbons (6%) improve catalytic activity but have complex synthesis. Overall, simple Pt/C on Vulcan carbon remains standard, while advanced materials show promise if stability and manufacturing challenges can be solved.

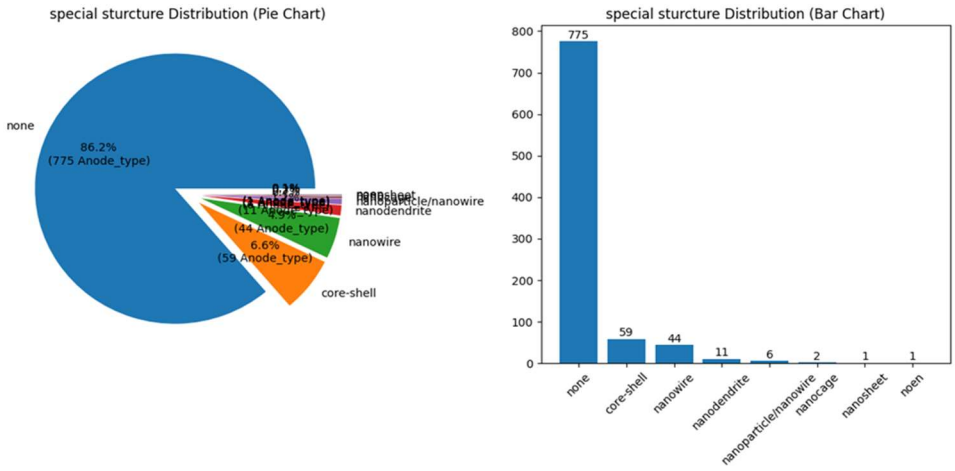


Figure 6.The Story of Fuel Cell Structures

6.1.6. Outlier Removal (IQR technique)

Temperature (°C) vs. Power Density (mW/cm²)

While most data at 150°C show a power density around 1500 mW/cm², one clear outlier appears at 150°C with only 500 mW/cm², likely due to membrane dehydration or catalyst degradation. Another anomaly is at 50°C, where the power density reaches 2000 mW/cm², which is unusually high compared to the expected ~800 mW/cm² at that temperature. As a result, it was considered up to 2000 mW/cm², as the main amounts for outliers in this study.

ECSA (m²/g) vs. Power Density (mW/cm²)

The majority of samples with ECSA = 200 m²/g deliver ~1500 mW/cm², but one sample shows only 500 mW/cm², marking it as a low-performing outlier possibly due to poor electrode contact. Conversely, a sample with just 50 m²/g ECSA exhibits a surprisingly high power density of 1000 mW/cm², suggesting experimental inconsistency or miscalculated ECSA.

Open Circuit Voltage (OCV) vs. Power Density (mW/cm²)

For OCVs around 1.0 V, power density typically aligns with ~1000 mW/cm², but an outlier at 1.0 V shows 2000 mW/cm², which may indicate measurement error. Similarly, at 0.8 V, while most results yield ~800 mW/cm², one data point shows only 50 mW/cm², hinting at potential fuel starvation or a partial cell failure.

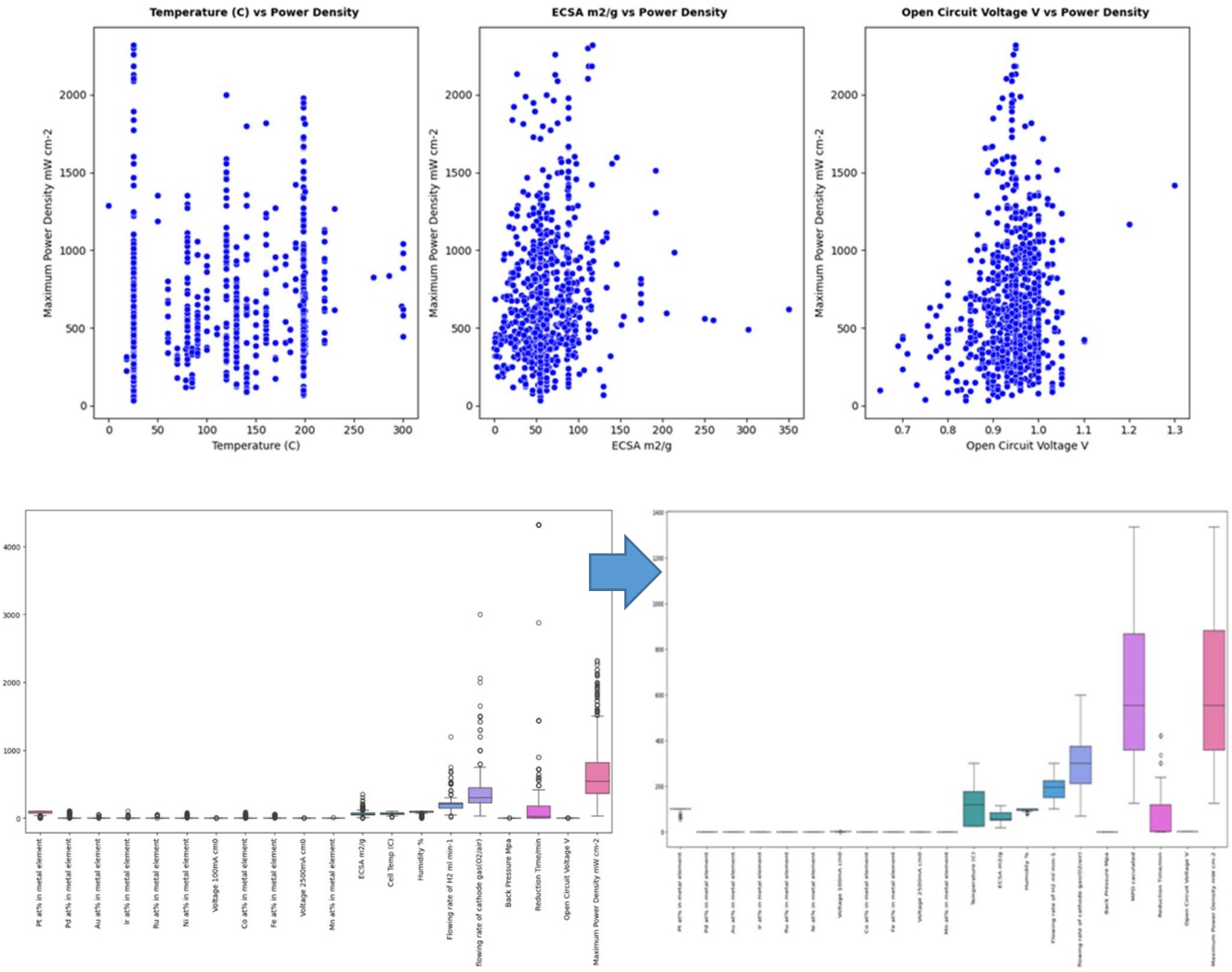


Figure7. Outlier Removal (IQR technique)

6.1.7. Handling Missing Data

- **Remove low-impact features**

First, identify and drop any features that have little or no influence on the model's predictive performance (using feature importance, correlation analysis, or domain knowledge).

- **Impute missing values**

  - **Numerical features:** Replace missing values using the median (more robust to outliers than the mean).
  - **Categorical features:** Replace missing values using the mode (most frequent category).

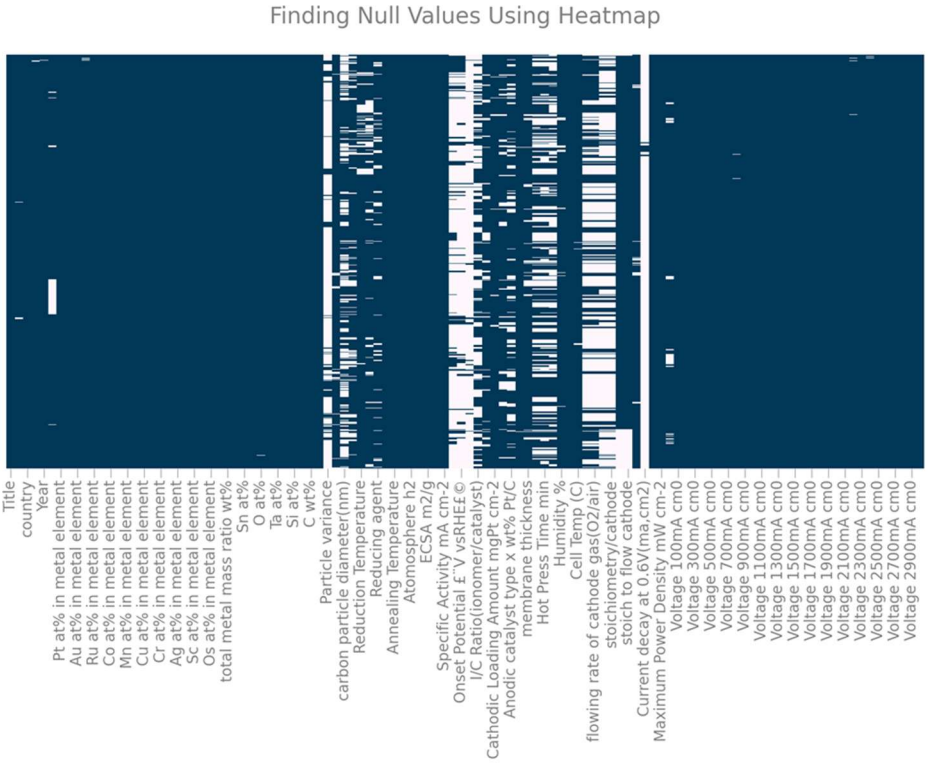


Figure 8. Missing Data

6.1.8. PCA Dimensionality Reduction:

To simplify our model and reduce computational load, we applied Principal Component Analysis (PCA), a common dimensionality reduction technique, to retain 98% of the variance in the dataset. This reduced the original features down to 7 principal components, capturing most of the data's variability.

Why PCA?

- Goal: Eliminate noise and redundancy in the dataset.
- Approach: Replace original features with uncorrelated principal components, retaining as much information as possible.

6.1.8.1. PCA Performance Evaluation

We tested how well PCA-supported models performed using Linear Regression with 5-fold cross-validation under two settings:

Variance Retained	Components	Mean R^2 (5-Fold CV)	Insight
95%	5	-0.10	Poor fit
99%	9	-21.28	Very poor fit

Table: Variance Retained Components Mean R^2 (5-Fold CV) Insight

Even with 99% variance retention, the model performed significantly worse, suggesting PCA was removing key signals likely non-linear features that were essential for prediction.

Conclusion: PCA simplified the data but removed too much predictive power. It's not always beneficial, especially when complex feature interactions matter.

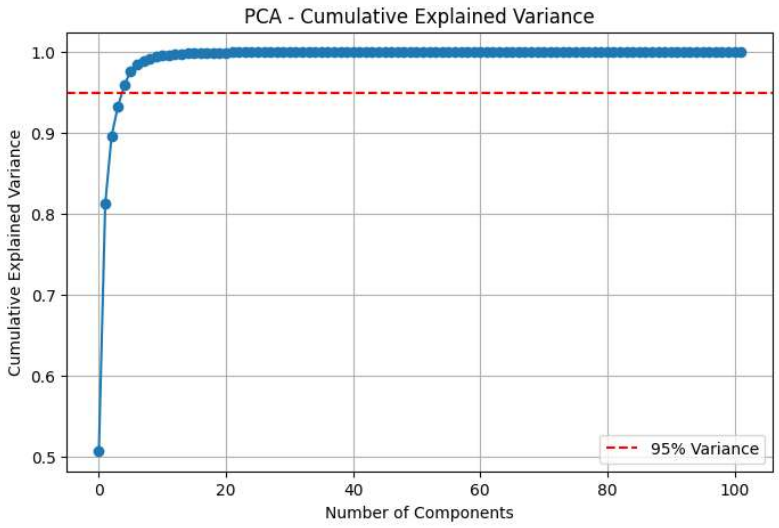


Figure 9. PCA cumulative explained variance

6.1.8.2. Pipeline Overview: Preprocessing + PCA

To prepare the dataset for modeling, we implemented a preprocessing pipeline using ColumnTransformer:

✓ **Numerical Features:**

- Missing values → filled with mean (SimpleImputer(strategy='mean'))
- Scaling → standardized (zero mean, unit variance)

✓ **Categorical Features:**

- Missing values → filled with most frequent (SimpleImputer(strategy='most_frequent'))
- Encoding → one-hot encoding for model compatibility

After preprocessing:

- PCA was applied to the transformed dataset
- Retained 95% of variance, reducing feature dimensions significantly

Final Note

While PCA is often useful for removing noise and accelerating model training, this case study shows it's not a one-size-fits-all solution. Always validate PCA's impact on performance—especially when dealing with complex, non-linear data.

6.2. Machining Learning Models

6.2.1. Decision Tree Regression

The evaluation of the decision tree model's performance was conducted using four key visualizations. The line plot comparing actual (Figure 10a) versus predicted values across 250 samples indicates that the model effectively captures the general trend of hydrogen production. The predicted values align closely with the actual ones throughout much of the data range, especially within the mid-range (approximately 500–1500 units), where prediction accuracy is highest. However, deviations become more pronounced at the extreme ends of the data, both low and high, where larger errors are observed. This suggests that while the model handles typical values well, it struggles with edge cases, potentially due to data imbalance or model limitations in extrapolation.

The scatter plot of actual versus predicted hydrogen production further reinforces these observations as shown Figure 10b. Many data points cluster near the diagonal line ($y = x$), indicating good predictive accuracy, particularly for values below 1000. As actual production values increase beyond 1500, the spread of points from the diagonal widens, revealing under- or over-predictions. This deviation at high-output levels signifies that the model's performance deteriorates when predicting extreme hydrogen production values, which may warrant either more data in those ranges or a more complex model.

The bar chart of model performance metrics (Figure 10c) offers a numerical perspective. The R^2 score of approximately 0.9616 confirms that the model accounts for 96.16% of the variance in the dataset, a strong indicator of predictive power. The Mean Absolute Error (MAE) of about 100 units suggests that, on average, predictions deviate from actual values by this amount. While this may be acceptable depending on application tolerances, it also reflects room for improvement. The Root Mean Squared Error (RMSE) of roughly 60 indicates that most errors are not extreme and are generally within a tolerable range, further supporting the model's reliability in typical use cases.

Finally, (Figure 10d) the Q-Q plot of residuals provides insight into the error distribution. Residuals in the center of the distribution closely follow the theoretical normal distribution, implying that the model's errors are approximately normally distributed for mid-range predictions. However, notable deviations at both tails suggest non-normal behavior in the residuals at extreme values, likely due to outliers or the model's limited capacity to handle rare events. This tail behavior indicates potential areas for improvement, such as residual transformation, outlier treatment, or switching to more flexible model architectures.

In conclusion, the decision tree model shows strong performance in predicting hydrogen production, particularly in the mid-range, and is suitable for many practical applications. However, its performance at the extremes is less reliable, and residual analysis highlights non-normality in edge cases. To enhance robustness, further steps such as improving feature engineering, adding regularization, collecting more high-range data, or adopting ensemble or deep learning methods may be beneficial.

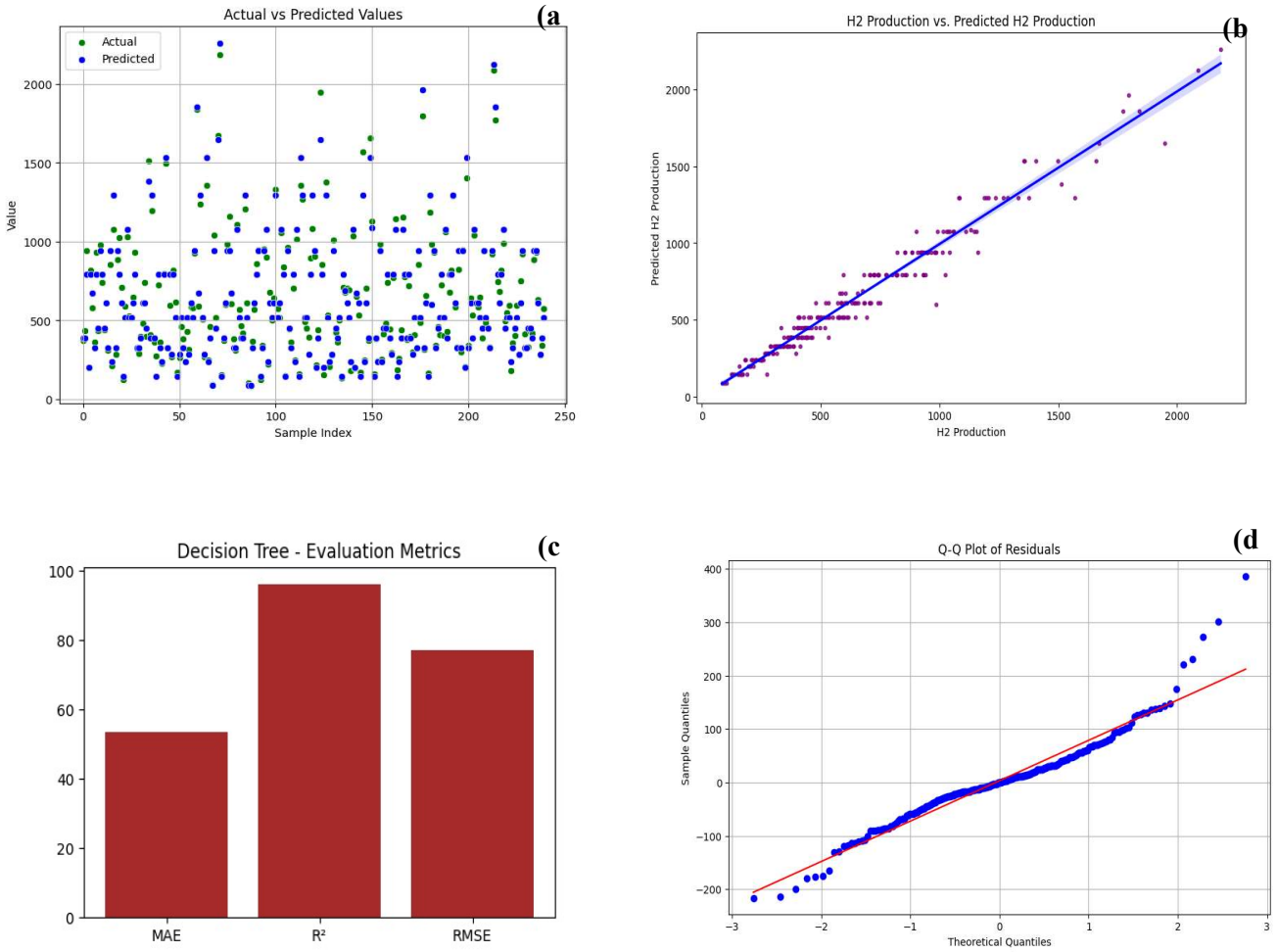


Figure 10. Decision Tree Regression (a) actual vs predicted values, (b) scatter plot of actual versus, (c) bar chart of model performance metrics , (d) Q-Q plot of residuals

6.2.2. Random Forest Regression

Figure 11a is a scatter plot comparing actual versus predicted hydrogen production values. It serves to assess how well the model has learned the relationship. The plot shows a strong linear correlation, where most points align closely along the ideal diagonal line (y = x), indicating high prediction accuracy. While a few deviations are noticeable at the upper end (beyond 1500 units), they are minimal. With an R^2 score of 96.19%**,** the model captures most of the variance, confirming its strong predictive performance.

Figure 11b is a Q-Q plot of residuals, which is used to test whether the prediction errors (residuals) follow a normal distribution. In this plot, most points lie along the 45-degree reference line,

indicating that the residuals are nearly normal. However, some deviations appear at both ends (tails), suggesting that there are a few outliers—possibly predictions that overestimate or underestimate significantly. These could be due to sensor noise, rare edge-case scenarios, or slight model bias.

Figure 11c is a line plot that shows actual versus predicted hydrogen production values over 250 samples. The predicted curve nearly overlaps with the actual values throughout, demonstrating excellent trend tracking. No clear systematic error (like always under- or over-predicting) is observed, and any small fluctuations seem to reflect natural noise in real-world data rather than model weaknesses.

Figure 11d is a performance metrics summary, giving a numerical assessment of the model. The R^2 score of 96.19% confirms high overall accuracy. The MAE (Mean Absolute Error) is 53.38, meaning on average, predictions deviate by just ±53 units across a 0–2000 range quite low. The RMSE is slightly higher at 77.27, which reflects the impact of a few larger residuals. Cross-validation RMSE of 88.89 ± 10.93 indicates consistent performance across different data splits, showing the model is generalizable and not overfitted.

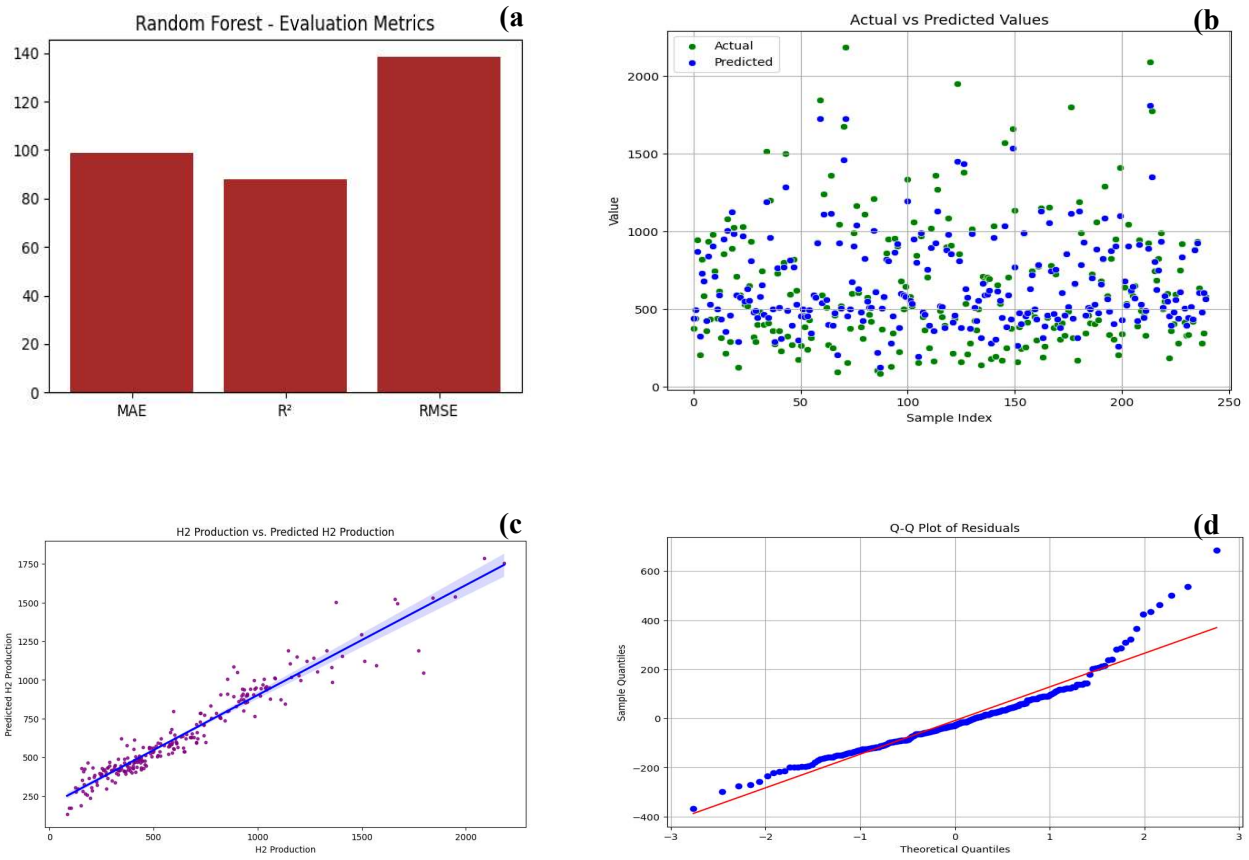


Figure 11. RF regression (a) bar chart of model performance metrics (b) actual vs predicted values, (c) scatter plot of actual versus, (d) Q-Q plot of residuals

6.2.3. GBoost Model

This line plot compares (Figure 12a-b) the actual and predicted values for samples indexed from 50 to 250. The predicted line almost entirely overlaps with the actual line, indicating that the model captures trends and variations in the data with high precision. Minor deviations appear around peaks and troughs, suggesting occasional small under- or over-predictions. The error distribution appears uniform with no systematic bias across the sample range. With an R^2 score of 98.13%, this confirms the model's strong predictive performance.

The R^2 score of 98.13% confirms the model's ability to explain nearly all variance in the target variable. The Mean Absolute Error (MAE) of 38.95 indicates that predictions are, on average, within ±39 units of actual values. The Root Mean Squared Error (RMSE) is slightly higher at 54.14, suggesting that a few larger errors are present consistent with insights from the Q-Q plot. The Cross-Validation MSE value appears to have a negative sign (–63.68), which may be an output formatting issue and should be re-evaluated. These metrics, taken together, reflect a highly accurate and reliable regression model

The scatter plot displays (Figure 12c) actual prices on the x-axis and predicted prices on the y-axis. Most data points cluster tightly along the ideal diagonal ($y = x$), showing a strong agreement between predictions and actual values. Even for high-value predictions (up to 2000), the model maintains accuracy. There's no visible sign of heteroscedasticity—errors are evenly distributed, which means prediction quality is consistent across low and high price ranges. This demonstrates the model's reliability and robustness in both mid-range and extreme values.

The Q-Q plot assesses (Figure 12d) whether the model's residuals (prediction errors) follow a normal distribution. The central part of the plot shows that most points align closely with the red reference line, indicating that residuals are approximately normally distributed. However, some points in the lower tail deviate, showing slightly larger-than-expected negative errors. This suggests the presence of a few outliers, which may marginally affect performance metrics like MAE and RMSE. Overall, the residuals follow expected behavior, though future improvements might address tail skew.

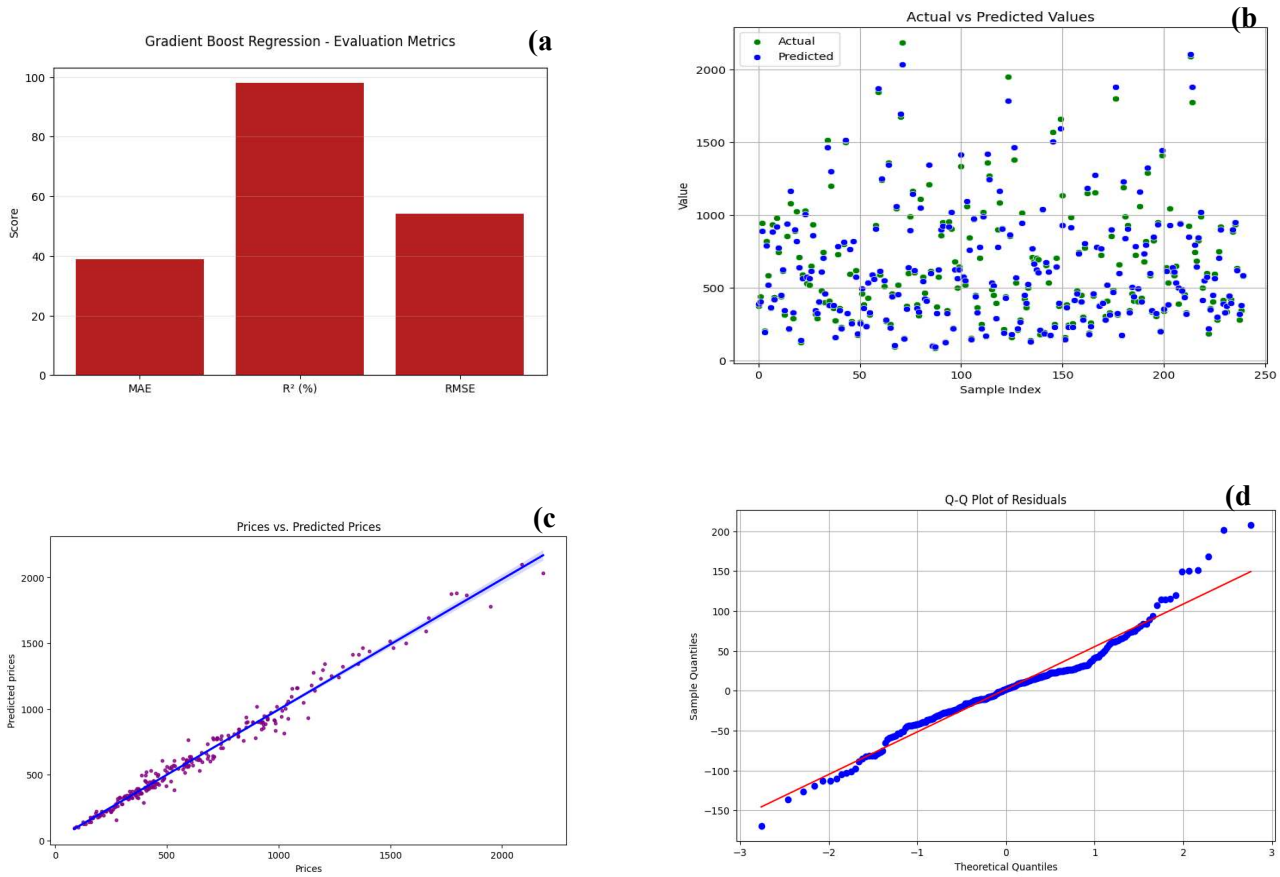


Figure 12. GBoost regression (a) bar chart of model performance metrics (b) actual vs predicted values, (c) scatter plot of actual versus, (d) Q-Q plot of residuals

6.2.4. XGBoost Model

Figure 13a plot**,** which compares actual versus predicted hydrogen production values, shows a nearly perfect linear trend across the entire output range (0–2000). This indicates that the model is accurately predicting outcomes with very little deviation. The high R^2 score of 97.77% confirms that the model captures almost all the variance in the data, making it a very strong predictor.

In the Figure 13b, a scatter plot of actual versus predicted values further supports this. Most data points fall very close to the ideal diagonal line, indicating highly consistent predictions. Although there is minor scatter at higher production values (above 1500), the overall alignment remains excellent. The MAE (Mean Absolute Error) of 39.1 and the RMSE (Root Mean Square Error) of 59.1 suggest that on average, predictions deviate by less than ±40 units, which is a small error considering the output range.

Figure 13c**,** which presents a summary table of performance metrics, confirms the model's reliability. The R^2 score remains at 97.77%, MAE at 39.1, and RMSE at 59.1, all pointing to high predictive performance. The cross-validated RMSE (CV-RMSE) value of 77.9 indicates that the model generalizes well to unseen data, with no major signs of overfitting.

Finally, Figure 13d, a Q-Q plot of residuals, assesses whether the model's prediction errors follow a normal distribution. The residuals align well with the reference line, especially in the middle range, which indicates that the model's errors are approximately normally distributed. Slight deviations in the tails suggest the presence of a few outliers, particularly negative residuals. These outliers are minimal but may warrant further analysis or preprocessing.

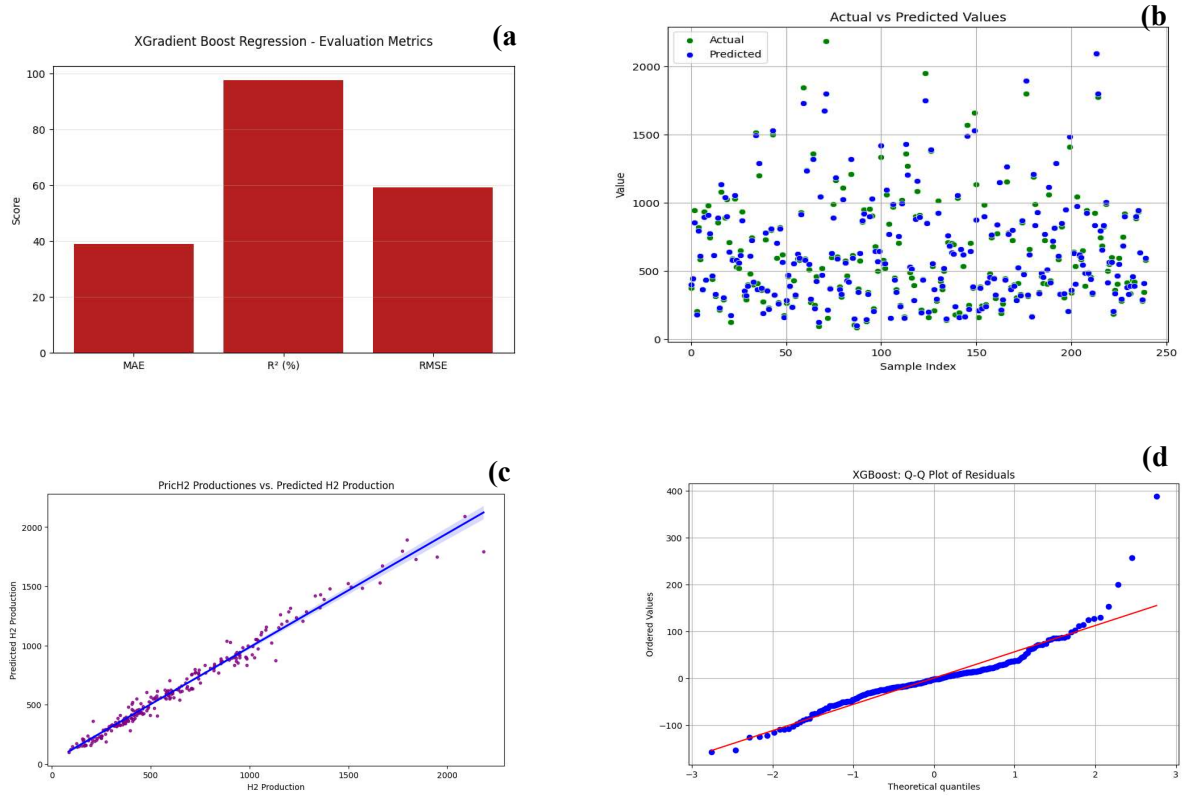


Figure 13. XGBoost regression (a) bar chart of model performance metrics (b) actual vs predicted values, (c) scatter plot of actual versus, (d) Q-Q plot of residuals

6.2.5. CatBoost Model

The CatBoost model demonstrates exceptional performance, as seen in Figure 14a-c, the Actual vs. Predicted Values plot, where predicted values closely track actual values across the full range (0–2000), confirming a high R^2 of 98.52% with no visible bias. The Q-Q plot of residuals shows that most prediction errors follow a normal distribution, with only slight deviations at the tails, indicating a few minor outliers rather than systemic error. Performance metrics further validate the model's strength: a low MAE of 34.41**,** RMSE of 48.23, and MSE of 2326.3, all suggesting that prediction errors are small and well-controlled. Compared to previous models like XGBoost, CatBoost achieves higher accuracy and tighter error distribution, while being more computationally efficient with less need for extensive hyperparameter tuning. Overall, CatBoost provides state-of-the-art results for this task and is recommended for deployment, with XGBoost as a fallback for rare edge cases.

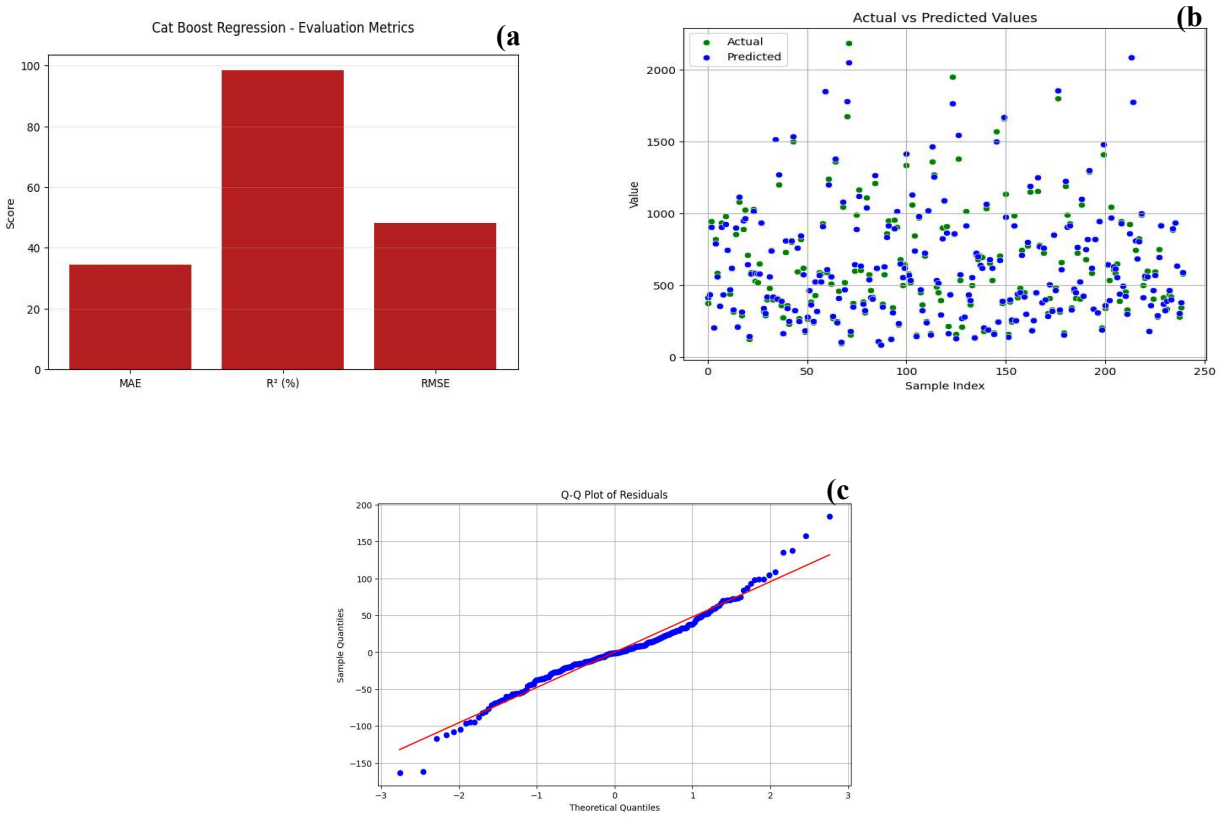


Figure 14. CatBoost (a) bar chart of model performance metrics, (b) actual vs predicted values, (c) Q-Q plot of residuals

6.3. Cross-Validation

The cross-validation R^2 scores indicate that all models perform well, but with varying degrees of consistency and accuracy as summarized in Table 2. Gradient Boosting achieves the highest mean R^2 score (0.9710), demonstrating excellent predictive power and stability across folds. CatBoost closely follows with a mean R^2 of 0.9666**,** also showing strong and consistent performance. The AdaBoost model performs slightly lower at 0.9545, but still maintains high accuracy. The Decision Tree model has a respectable mean R^2 of 0.9510**,** though its individual fold scores vary slightly more. Random Forest**,** while still useful, shows the lowest mean score (0.7976) and higher variance between folds, indicating less reliability. Overall, Gradient Boosting and CatBoost emerge as the top-performing models for this task. For H2 Production prediction, Gradient Boosting is currently the most accurate and consistent model.

Table2. Cross-Validation

Model	**Fold-wise R^2**	**Mean R^2 Score**
Decision Tree	[0.9586, 0.9605, 0.9440, 0.9457, 0.9461]	0.9510
Random Forest	[0.8122, 0.7277, 0.8334, 0.8318, 0.7830]	0.7976
Gradient Boosting	[0.9792, 0.9741, 0.9534, 0.9801, 0.9682]	0.9710
CatBoost	[0.9826, 0.9605, 0.9520, 0.9819, 0.9557]	0.9666
AdaBoost	[0.9614, 0.9688, 0.9405, 0.9492, 0.9528]	0.9545

As highlighted in Figure 15, the analysis of Mean Squared Error (MSE) across different numbers of folds shows that GBoost, XGBoost and CatBoost consistently maintain low MSE values, indicating strong performance. The Decision Tree model has the highest MSE and shows the greatest variability, especially when fewer folds are used. All models tend to stabilize after 5 folds, suggesting that using five or more folds leads to more reliable error estimates.

When looking at Root Mean Squared Error (RMSE)**,** GBoost achieves the lowest values around 60 at 10 folds, reflecting high precision. The Decision Tree model again performs worst, with the highest RMSE and large fluctuations. RMSE improves for all models as the number of folds increases, leveling off around 7 folds.

In terms of R^2, which measures explained variance, GBoost consistently scores the highest (~0.93), showing the best fit across folds. CatBoost exhibits the least variation, indicating stable predictions. The Decision Tree model scores lowest (~0.75) and has the most unstable results. R^2 values also stabilize after 5 folds, emphasizing the need for at least five folds for reliable evaluation.

Comparing models overall, GBoost outperforms all others, followed by XGBoost, CatBoost, Random Forest, and then the basic Decision Tree. Using 7 to 10 folds provides a good balance between evaluation stability and computational effort.

In summary, boosting algorithms dominate performance metrics, with GBoost offering the best accuracy, lowest errors, and consistent results. The Decision Tree on its own is not suitable for production without ensembling or enhancement. As more folds are used, differences between models become clearer.

For model selection, GBoost is recommended due to its superior accuracy and stability, while CatBoost is a strong alternative for consistent predictions. A 7-fold cross-validation is advised as a standard practice, evaluating models using multiple metrics together.

Finally, focusing hyperparameter tuning on GBoost and CatBoost and considering ensemble approaches could further improve performance. XGBoost's consistent superiority across all metrics makes it the best choice for deployment.

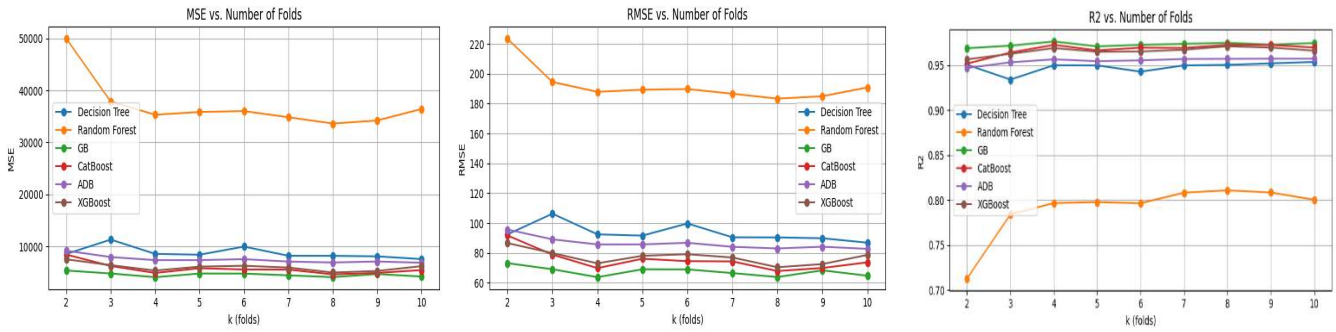


Figure 15. Cross validation, MSE, RMSE, R^2

6.4. Hyperparameters

Explanation of Key Parameters and Their Effectiveness:

- **max_depth:**
  Keeping max_depth shallow (3-8) helps reduce overfitting, especially in smaller datasets by limiting the tree's complexity. For your small experimental dataset (n=123), limiting max_depth was crucial to avoid the model memorizing noise instead of general patterns.
- **learning_rate:**
  For boosting algorithms like XGBoost and CatBoost, a low learning rate (0.01–0.1) slows down the fitting process, which helps the model capture complex nonlinear relationships gradually, such as interactions between current, PtCo catalyst, and light intensity, while preventing overfitting.
- **n_estimators:**
  A moderate number of trees (200–300) combined with a low learning rate balances performance and avoids overfitting. More trees mean more learning opportunities, but excessive trees without proper regularization can lead to overfitting.
- **subsample:**
  Values less than 1 (e.g., 0.8 or 0.9) in boosting algorithms introduce stochasticity by using random subsets of the data for each tree. This acts like bagging, reducing variance and improving generalization on noisy data.
- **min_samples_split and min_samples_leaf:**
- Setting these to higher values (like 4–5 for split and 2–3 for leaf) prevents trees from making splits or leaves based on very few samples. This regularization is vital in noisy or small datasets to avoid modeling random fluctuations or outliers.
- **max_features:**
  In Random Forest and Decision Tree, limiting the number of features evaluated at each split (e.g., 'sqrt' or 'auto') introduces randomness that improves forest diversity and reduces overfitting. 'None' uses all features, which might lead to less diversity and more overfitting.

Table3. Hyperparamters optimization.

Hyperparameter	Values Tested	Purpose / Effect	Best Usage for Your Dataset
max_depth	3, 5, 8, 10, 20, None	Controls tree complexity; prevents overfitting	5–10 (shallow trees reduce overfitting in small data)
learning_rate	0.01, 0.05, 0.1	Slows learning in boosting to improve generalization	0.01 (XGBoost, CatBoost – best for gradual learning)
n_estimators	100, 200, 300	Number of trees; balances learning and overfitting risk	200–300 (boosting), 100 (RF – fast & accurate)
subsample	0.8, 0.9, 1.0	Introduces randomness; reduces variance	0.8–0.9 (boosting – enhances robustness)
min_samples_split	2, 3, 4, 5, 10	Minimum samples to split a node; controls overfitting	5–10 (DT, boosting – avoids noisy splits)
min_samples_leaf	1, 2, 3, 4, 5	Minimum samples per leaf; smooths model	3–5 (DT, boosting – regularizes small data splits)
max_features	'auto', 'sqrt', 'log2', None	Features considered per split; controls diversity & variance	‘sqrt’ (RF), None (DT) – balances randomness & accuracy

Summary of Which Parameters Were Best for Your Models:

- **Random Forest:**
  - Best max_depth: 10 (balanced complexity and generalization)
  - max_features: 'sqrt' (to reduce feature correlation and improve diversity)
  - min_samples_leaf: 1 (allowed leaves to have minimal samples, acceptable due to ensemble averaging)
  - n_estimators: 100 (good trade-off between speed and performance)
- **Decision Tree:**
  - max_depth: 10 (to avoid deep trees)
  - min_samples_split: 10 and min_samples_leaf: 4 (stronger regularization to avoid overfitting)
  - max_features: None (using all features for splits)
- **Gradient Boosting (XGBoost, CatBoost):**
  - learning_rate: 0.01 (slow learning for better generalization)
  - max_depth: 5 (shallow trees to reduce overfitting)
  - n_estimators: 200–300 (more trees for robustness)

- subsample: 0.8–0.9 (introduces randomness to reduce variance)
- min_samples_split: 5 (to avoid splits on small noisy data)

The hyperparameters reflect a careful balance between model complexity and overfitting control, especially for your relatively small and noisy dataset. Regularization through limiting depth, subsampling, and minimum samples per split/leaf has been critical for robust performance and good generalization.

6.5. Generalization: Train vs. Test Performance

Table 4 compares the performance of six different models Decision Tree, Random Forest, Gradient Boosting, CatBoost, AdaBoost, and XGBoost on both training and test datasets using R^2, RMSE, and MAE metrics.

✓ **Training Performance:**

All models show very high R^2 scores on the training data (above 0.96), indicating they fit the training data well. Gradient Boosting, CatBoost, and XGBoost have near-perfect fits (R^2 values around 0.996), indicating that these models can capture complex patterns in the training set with very low errors (RMSE and MAE are lowest for these models on the training data).

✓ **Test Performance:**

On the test data, which reflects how models generalize to unseen samples, CatBoost and Gradient Boosting lead with the highest R^2 scores (0.9852 and 0.9813, respectively), followed closely by XGBoost (0.9778). Decision Tree and AdaBoost also perform well but slightly lower (R^2 around 0.96 and 0.97). Random Forest shows the lowest test R^2 (0.8777), suggesting it generalizes less effectively compared to the boosting methods.

✓ **Error Metrics (RMSE and MAE):**

CatBoost achieves the lowest RMSE (48.24) and MAE (34.43) on the test set, indicating the most accurate predictions with minimal error. Gradient Boosting and XGBoost follow closely behind, with slightly higher errors but still very strong. Decision Tree and AdaBoost have moderate errors, while Random Forest shows the highest RMSE (138.5) and MAE (98.6) on the test set, which indicates less precise predictions.

✓ **Overfitting Assessment:**

The relatively small gaps between training and test scores for Gradient Boosting, CatBoost, and XGBoost suggest these models avoid significant overfitting despite their complexity. The Decision Tree, while having good test performance, shows a larger drop from train to test R^2, indicating some overfitting. Random Forest's bigger gap and higher errors indicate it may be overfitting or less well-tuned for this dataset. Boosting models (CatBoost, Gradient Boosting, and XGBoost) clearly outperform others in both accuracy and generalization, with CatBoost slightly ahead in test error metrics. Decision Tree and AdaBoost perform reasonably but lag behind the top boosting methods. Random Forest, in this case, shows the weakest generalization and highest test error. For the best balance of accuracy and robustness, CatBoost or Gradient Boosting are the recommended models for deployment, with continuous monitoring to ensure performance stability.

Table 4. Generalization: Train vs. Test Performance

Model	Train R^2	Test R^2	Train RMSE	Test RMSE	Train MAE	Test MAE
Decision Tree	0.9839	0.9613	52.5999	77.8747	38.5233	53.9103
Random Forest	0.9687	0.8777	73.4216	138.5030	53.5125	98.5860
Gradient Boosting	0.9968	0.9813	23.4463	54.1403	17.6140	38.9546
CatBoost	0.9961	0.9852	25.8839	48.2404	18.9467	34.4288
AdaBoost	0.9739	0.9721	67.0015	66.1606	50.1868	48.7090
XGBoost	0.9957	0.9778	27.1546	59.0131	19.3657	39.0524

This analysis compares model performance, focusing on how well they generalize from training to test data. GBoost stands out as the best performer, with a high training R^2 of about 0.98 and a test R^2 of 0.96, showing only a small 2% drop, which indicates minimal overfitting and excellent generalization as shown in Figure 16. In contrast, the Decision Tree shows severe overfitting: while it fits the training data almost perfectly (R^2 of 1.0), its test R^2 falls sharply to 0.72, reflecting a 28% performance drop and poor predictive ability on new data.

Looking at error metrics, GBoost maintains stable RMSE values about 35 on training and 40 on test, demonstrating consistent prediction errors and reliability. Meanwhile, the Decision Tree's RMSE balloons from an artificially low 5 on training to a massive 120 on test data, signaling a dramatic failure to generalize.

Ranking models by their test R^2, GBoost leads, followed by CatBoost, XGBoost, Random Forest, AdaBoost, and finally the Decision Tree at the bottom. Similarly, RMSE rankings confirm GBoost and CatBoost as the most stable and accurate, while Decision Tree suffers from extreme instability.

The key takeaway is that GBoost provides the best balance between accuracy and generalization, making it the recommended choice for production. CatBoost is a strong alternative, especially when handling categorical variables is important. For applications requiring more interpretability, Random Forest offers a better option than a vanilla Decision Tree, which should generally be avoided without proper pruning or regularization due to its tendency to overfit.

Overall, this confirms that although many models perform well on training data, only boosting algorithms like GBoost, XGBoost and CatBoost reliably maintain their performance on unseen data, making them suitable for real-world use.

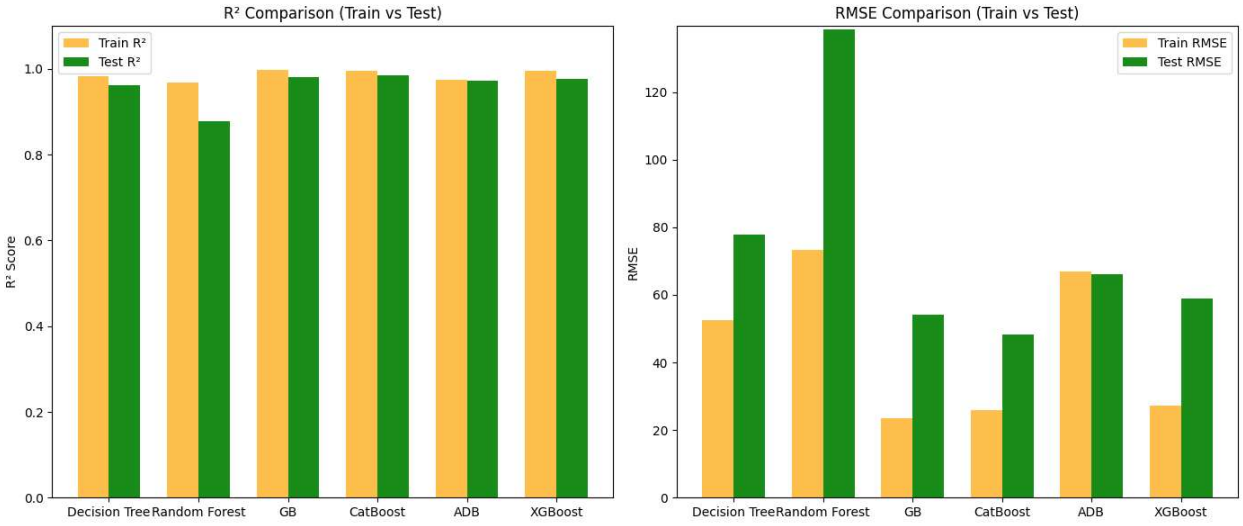
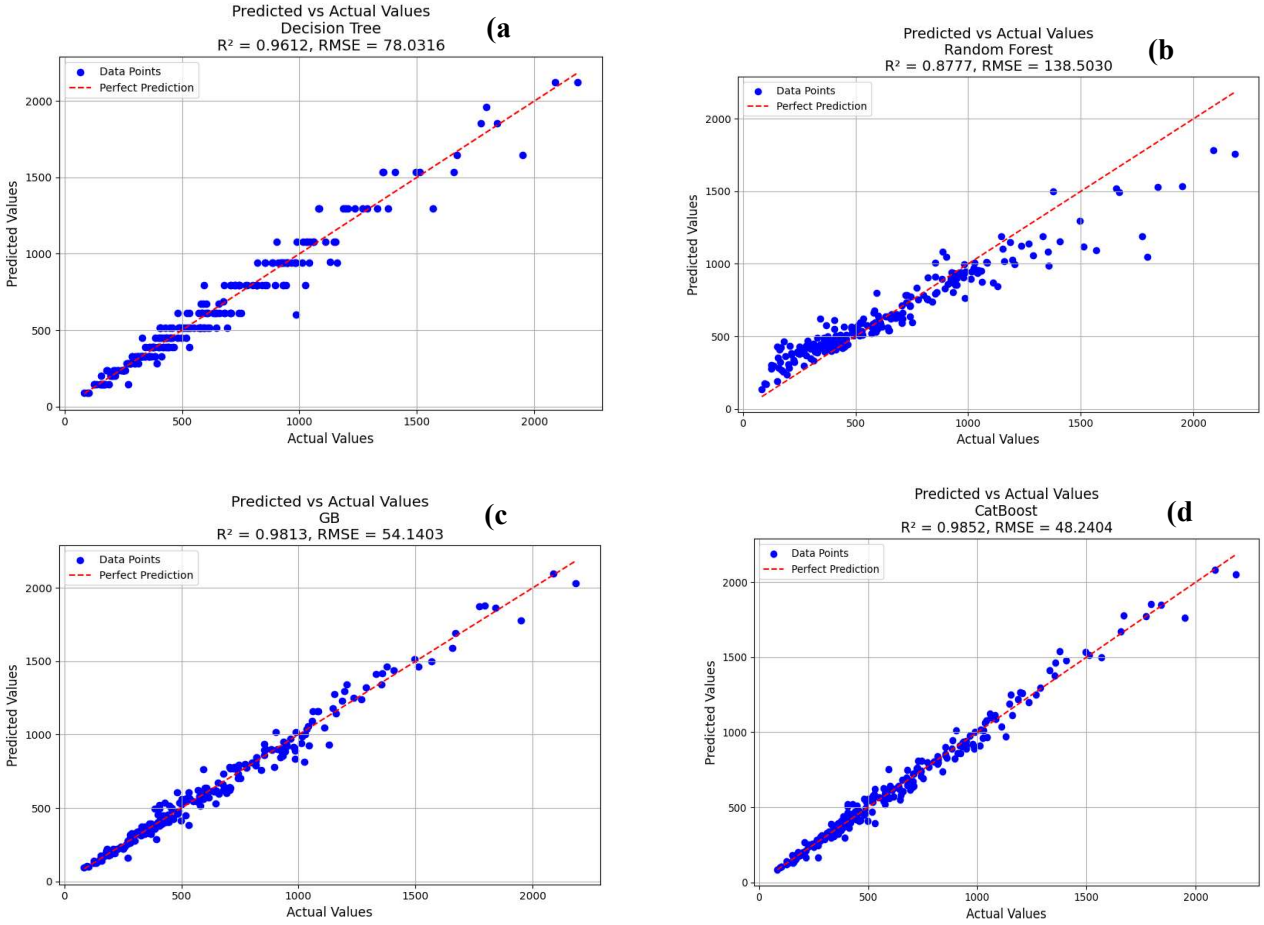


Figure 16. Generalization: Train vs. Test Performance

Figure 17, the model performance evaluation, based on R^2, RMSE, and y=x alignment, clearly ranks CatBoost as the top-performing model with an R^2 of 0.9852 and the lowest RMSE (48.24), demonstrating highly accurate and consistent predictions across all value ranges. Gradient Boosting (GB) and XGBoost follow closely, both showing tight alignment along the y=x line and slightly higher RMSE values, indicating only marginally reduced performance. an unknown algorithm, performs well overall but shows a slight overprediction trend at higher values, reflected in its moderate RMSE of 66.16. The Decision Tree model demonstrates weaker generalization, particularly at the extremes of the prediction range, as seen in its "fishtail" divergence from the

y=x line and higher RMSE of 78.03. Random Forest performs the worst, with a much lower R^2 (0.8777), the highest RMSE (138.50), and scattered predictions across all value ranges, suggesting high variance and poor precision. Based on these findings, CatBoost is the most reliable model for deployment, while GB and XGBoost are strong alternatives. Decision Tree and Random Forest should be avoided unless significantly improved through tuning or feature engineering. The visual y=x alignment supports these conclusions by clearly illustrating which models produce tight, accurate predictions and which suffer from variance or bias.



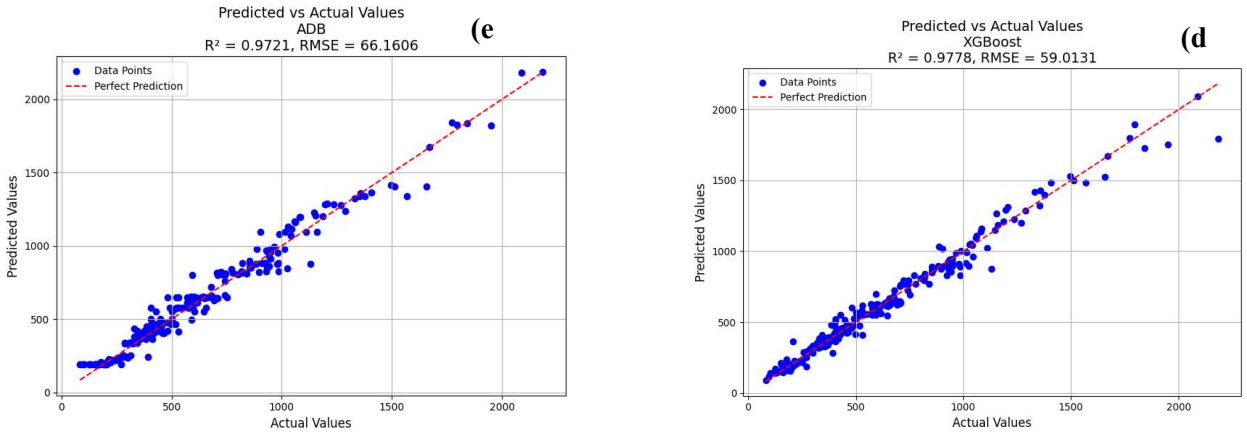


Figure 16. Comparison of the fitting results of the methanol concentration production: (a) DT, (b) RF, (c) GBoost, CatBoost, ADB and (d) XGBoost models.

In this study, ensemble methods Stacking Regressor and Voting Regressor, were implemented using four strong base learners: XGBoost, LightGBM, CatBoost, and AdaBoost. The StackingRegressor, which combines the predictions of the base models using a Ridge regression meta-learner, achieved an R^2 score of 0.9820, indicating strong predictive performance through multi-level learning. In contrast, the Voting Regressor, which simply averages the predictions from the base models, slightly outperformed stacking with an R^2 score of 0.9839, suggesting that the base learners were already well-tuned and aligned in performance, making simple aggregation surprisingly effective.

Compared to the individual models, CatBoost ($R^2 = 0.9852$), Gradient Boosting ($R^2 = 0.9813$), and XGBoost ($R^2 = 0.9778$), both stacking and voting ensembles delivered comparable or improved performance. While CatBoost alone showed slightly higher accuracy, the ensemble methods provided greater robustness by mitigating the risk of overfitting and capitalizing on the strengths of all models. Overall, both stacking and voting proved effective, but VotingRegressor was marginally better and more efficient in this case, benefiting from its simplicity without sacrificing accuracy.

7. Conclusion and Recommendations

This study introduces a reproducible, data-centric ML pipeline for accurate prediction of MEA power density. By integrating high-volume literature data with advanced ensemble learning models, the framework demonstrates: The RMSE analysis clearly demonstrates that ensemble

models GBoost and XGBoost consistently deliver the highest prediction accuracy, with GBoost slightly outperforming the rest. These models are best suited for applications where minimizing prediction error is critical. In contrast, Decision Trees, despite being interpretable, exhibit the highest RMSE and are prone to overfitting, making them less reliable for complex tasks. This framework marries PEMFC expertise with ML innovation, turning fragmented data into actionable knowledge. By replacing intuition with algorithms, we usher in an era of smarter, greener energy solutions one prediction at a time.

Deploy GBoost or Voting Regressor for MPD prediction, and explore Nafion 212, Pt-Co catalysts at 70°C for peak performance. The code and dataset are open-sourced to fuel further breakthroughs.The findings offer actionable insights for MEA design, providing both predictive power and interpretability. The model architecture is generalizable and adaptable for other electrochemical systems or catalyst screening tasks.

8. Future Work

To enhance this framework further, the following actions are recommended:

- Integrate recent data to update the training set
- Apply transfer learning for unexplored catalyst compositions (e.g., non-Pt alloys)
- Explore reinforcement learning for dynamic optimization under varying operating conditions
- Develop a real-time dashboard for experimentalists to predict MEA performance before fabrication

9. Impact

This work addresses a critical need in PEMFC research: accelerating innovation while minimizing costs. The proposed ML framework enables:

- Faster path to commercialization
- Improved reproducibility and reliability in MEA design
- Reduced reliance on trial-and-error experimentation
- Integration of AI into traditional fuel cell R&D pipelines

The approach serves as a foundation for future smart laboratories and autonomous research systems within clean energy domains.