Machine Learning Approaches for Predicting Buchwald-Hartwig Reaction Yields: Comparing different Methods

Hatef Rahimi*

FAU Erlangen-Nürnberg

E-mail: hatef.rahimi@fau.de

Abstract

A comprehensive comparison of machine learning approaches for predicting yields in Buchwald-Hartwig C-N coupling reactions using the Ahneman dataset (4312 reactions) is presented. One-hot encoding with three models (Random Forest, Gradient Boosting, Neural Network) is compared against transformer-based ChemBERTa embeddings. One-hot encoding achieves superior performance across all models compared to ChemBERTa. However, the transformer approach looks promising for novel molecular prediction in the future. Detailed explanations of molecular representation methods, including the Reaction SMILES format, the BERT transformer architecture, and neural network design, are provided. RXNFP (Reaction Fingerprints) was also attempted, but encountered version incompatibility issues. An interactive Gradio web application demonstrates practical yield prediction based on the best-performing model. Molecular identifiers and SMILES strings were cross-validated using the PubChem database[1] to ensure correctness.

Introduction

Problem Statement & Motivation

The Buchwald-Hartwig cross-coupling reaction enables C-N bond formation between aryl halides and amines via palladium catalysis. However, predicting reaction yields could be challenging due to complex interactions between catalysts, substrates, bases, and additives.

Experimentation has generated extensive reaction datasets, but extracting predictive models requires effective molecular representations and machine-learning approaches. Accurate yield prediction can save the chemist from lab work, reduce material waste, and accelerate optimization.

Objectives

Key goals are:

1. Compare multiple ML models with one-hot encoding (Random Forest, Gradient Boosting, Neural Network)
2. Explore and evaluate transformer-based ChemBERTa embeddings as another alternative
3. Deploy an interactive web application for practical use

Overview

This report is structured as follows: Section 2 reviews related work in reaction prediction. Section 3 focuses on the methodology, including the Open Reaction Database, one-hot encoding with model comparison, Reaction SMILES, transformer architecture, ChemBERTa embeddings, and an attempted RXNFP approach. Section 4 presents results and analysis. Section 5 concludes with insights and future directions.

Related Work

The Open Reaction Database repository provides an example implementation[2] that demonstrates machine learning for reaction-yield prediction using one-hot encoding of reaction components with neural networks. Their work on the Suzuki-Miyaura coupling dataset showed that categorical encoding of discrete molecular components can achieve strong predictive performance for high-throughput screening data. Building on this approach, the methodology is extended to compare multiple model architectures (Random Forest, Gradient Boosting, Neural Network) and evaluate transformer-based alternatives (ChemBERTa) to assess whether molecular representations offer advantages over one-hot encoding for the Buchwald-Hartwig dataset.

Methodology

The Open Reaction Database

The Open Reaction Database (ORD)[3] is an open-access initiative for structuring and sharing organic reaction data. Unlike traditional publications, where experimental details are stored as unstructured text in supporting information, the ORD provides a standardized schema implemented using Protocol Buffers that captures comprehensive reaction metadata, including inputs, conditions, workups, outcomes, and analytical data. The database supports various reaction types. All data and code are publicly available on GitHub under open licenses, with web-based interfaces for data submission and searching. The ORD schema[4] enables structured representation of reaction data that can be directly used for machine learning without manual parsing, making it ideal for training predictive models in synthesis planning and reaction optimization.

Dataset & Preprocessing

I use the Ahneman Buchwald-Hartwig dataset from the ORD, containing 4312 C-N coupling reactions with p-toluidine. Each reaction specifies:

- **Catalyst**: 4 Pd-based catalysts
- **Aryl Halide**: 15 aromatic halides
- **Base**: 3 bases
- **Additive**: 24
- **Amine**: p-toluidine (fixed)

This yields 48 unique components. The target is percentage yield (0-100%).

Data splits: 60% training, 10% validation, 30% test.

Approach 1: One-Hot Encoding

Concept

One-hot encoding represents each molecule as a binary indicator. Since there is a fixed library of 48 components, a 48-dimensional sparse vector where exactly 6 positions are "1" (one per component type, including solvent) is created.

Example

Consider a reaction with Catalyst 1 (Pd-XPhos), Aryl Halide 1 (4-CF_3-phenyl chloride), Base 1 (P2Et), Additive 1 (DMSO), Amine 1 (p-toluidine), and Solvent (THF). The SMILES strings are:

```
Catalyst:     CC(C)c1cc(C(C)C)c(-c2ccccc2P(C2CCCCC2)...)...
Aryl Halide: FC(F)(F)c1ccc(Cl)cc1
Base:         CCN=P(N=P(N(C)C)(N(C)C)N(C)C)(N(C)C)N(C)C
Additive:     CS(=O)C
Amine:        Cc1ccc(N)cc1
Solvent:      C1CCOC1
```

The one-hot vector:

$$\mathbf{x} = [\underbrace{1,0,0,0}_{4},\underbrace{1,0,...,0}_{15},\underbrace{1,0,0}_{3},\underbrace{1,0,...,0}_{24},\underbrace{1}_{1},\underbrace{1}_{1}] \tag{1}$$

Model Comparison

Three different machine learning models are evaluated on the one-hot encoded features:

1. Random Forest

- Ensemble of 500 decision trees
- Max depth: None (unlimited)
- Max features: sqrt (random feature subsampling)

2. Gradient Boosting

- 100 sequential boosting iterations
- Learning rate: 0.1 (default)
- Max depth: 5

3. Neural Network

```
Input:    48 features (binary)
Dense:    64 neurons, ReLU
Dropout:  30%
Dense:    32 neurons, ReLU
Dropout:  30%
Output:   1 neuron, Linear (no activation)
```

Neural Network Training: Adam optimizer (lr=0.005), MSE loss, batch size 100, 300 epochs with early stopping (patience=30).

All three models achieved excellent performance.

Approach 2: ChemBERTa Embeddings

What is BERT?

BERT (Bidirectional Encoder Representations from Transformers)[5] is a pre-trained language model. It learns contextual representations by predicting masked words. The key innovation is *bidirectional attention*: BERT reads sequences in both directions simultaneously.

What is a Transformer?

Transformers[6] use self-attention mechanisms instead of recurrence. The core self-attention operation is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

where Q (query), K (key), and V (value) are learned projections, and d_k is the dimension. This computes relevance scores between all token pairs.

Simple Example: Consider the sentence "The cat sat"

```
Tokens: ['The', 'cat', 'sat']
```

Self-attention allows each word to look at all other words:

- 'cat' computes: How relevant is 'The'? 'cat'? 'sat'?
- High attention to 'sat' (verb-subject relationship)
- Moderate attention to 'The' (determiner)

Attention scores might be:

```
'The': attends to The=0.2, cat=0.7, sat=0.1
'cat': attends to The=0.3, cat=0.2, sat=0.5
'sat': attends to The=0.1, cat=0.6, sat=0.3
```

This captures relationships (subject-verb) without explicit grammar rules.

Chemistry Example: For SMILES string "CCO" (ethanol):

```
Tokens: ['C', 'C', 'O']
```

Self-attention works similarly:

- First 'C' computes relevance with all atoms
- High attention to bonded neighbor 'C'
- Lower attention to distant 'O'

Attention scores:

```
C[0]: attends to C[0]=0.3, C[1]=0.5, O[2]=0.2
C[1]: attends to C[0]=0.5, C[1]=0.2, O[2]=0.3
O[2]: attends to C[0]=0.2, C[1]=0.5, O[2]=0.3
```

Each atom's representation incorporates information from all other atoms, allowing the model to learn chemical bonding patterns and functional groups.

Advantages: Parallel processing, long-range dependencies, contextual understanding.

What is ChemBERTa?

ChemBERTa[7,8] is BERT pre-trained on 77 million SMILES strings from the ZINC database. It learns:

- Chemical syntax and grammar
- Molecular patterns and functional groups
- Structural similarity relationships

ChemBERTa outputs 768-dimensional dense vectors capturing molecular properties.

Reaction SMILES Format

Unlike one-hot encoding treating molecules independently, the *entire reaction* is presented as:

$$\text{Reaction SMILES} = \text{reactants} > \text{agents} > \text{products} \tag{3}$$

Example (same reaction as before):

```
Reactants: FC(F)(F)c1ccc(Cl)cc1.Cc1ccc(N)cc1
Agents: CC(C)c1cc(C(C)C)c(...)c(C(C)C)c1.CCN=P(...)N(C)C.CS(=O)C
Products: Cc1ccc(Nc2ccc(C(F)(F)F)cc2)cc1

Full: FC(F)(F)c1ccc(Cl)cc1.Cc1ccc(N)cc1>CC(C)c1cc(...)>Cc1ccc(Nc2ccc...)cc1
```

This 200-400 character string encodes the complete reaction.

From Reaction SMILES to Embeddings

The complete process of converting a Reaction SMILES string to a 768-dimensional embedding involves several steps:

Step 1 - Input: Reaction SMILES

```
Input: FC(F)(F)c1ccc(Cl)cc1.Cc1ccc(N)cc1>CC(C)c1cc(...)>Cc1ccc(Nc2ccc...)cc1
```

Step 2 - Tokenization

The tokenizer breaks the SMILES string into individual tokens:

```
Tokens: ['F','C','(','F',')','(','F',')','c','1','c','c','c','(','C','l',')','c','c','1',
        '>','C','C','(','C',')','c','1',...] (up to 512 tokens)
```

Each token represents an atom, bond, branch marker, or structural element.

Step 3 - Transformer Processing (ChemBERTa)

The tokens are passed through 12 transformer layers. Each layer consists of:

- **Self-Attention**: Each token computes attention with all other tokens
- **Feed-Forward**: Non-linear transformation of each token

After 12 layers, each token has a 768-dimensional contextual embedding that captures:

- Its identity (what atom/symbol it is)
- Its context (what other tokens surround it)
- Chemical patterns (functional groups, bonding)

Step 4 - Mean Pooling

Average of all token embeddings to get a single reaction embedding:

$$\mathbf{e}_{\text{reaction}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{e}_{\text{token}_i} \tag{4}$$

Step 5 - Final Output

A single 768-dimensional vector representing the entire reaction:

```
Output: [-0.234, 0.891, -0.456, 0.123, ..., 0.678]  (768 numbers)
```

This embedding captures the chemical semantics of the reaction and serves as input to our neural network for yield prediction.

Visual Summary:

```
Reaction SMILES (string)
        |
    Tokenizer
        |
512 Tokens
        |
Transformer Layer 1 (Self-Attention + Feed-Forward)
        |
Transformer Layer 2
        |
    ...
        |
Transformer Layer 12
        |
512 x 768 Token Embeddings
        |
    Mean Pooling
        |
1 x 768 Reaction Embedding
        |
Neural Network
        |
    Yield Prediction
```

The key advantage of this approach is that ChemBERTa has already learned chemical patterns from 77 million molecules, so the 768-dimensional embeddings contain meaningful chemical information without requiring us to train on millions of reactions.

Results and Discussion

Quantitative Performance

Table 1 compares all approaches on the test set.

Key Findings:

- All one-hot models achieve excellent performance ($R^2 = 0.93$)

Table 1: Model Performance on Test Set (1293 reactions)

Approach	RMSE (%)	MAE (%)	R^2
One-Hot Encoding			
Neural Network	**7.8**	**5.6**	**0.9**
Gradient Boost	11.1	8.3	0.8
Random Forest	11.1	8.2	0.8
ChemBERTa (NN)	16.2	11.5	0.6

- Model choice (RF, GB, NN) has minimal impact with one-hot encoding
- ChemBERTa performs moderately but not as good as one-hot encoding

Why Does One-Hot Win?

Advantages for This Dataset

1. **Fixed Library**: Only 48 molecules means one-hot is perfectly suited
2. **Direct Mapping**: Each component directly corresponds to one feature
3. **No Information Loss**: Binary encoding preserves all component identity
4. **Best with Neural Networks**: NN significantly outperforms tree-based models (RMSE 6.98% vs 11.1%)
5. **Efficiency**: 48 input features vs 768 - simpler input representation
6. **Interpretability**: Can directly see which components affect yield

ChemBERTa Limitations Here

1. **Dataset Size**: 4312 reactions is small for fine-tuning transformers
2. **Training Mismatch**: ChemBERTa was pre-trained on individual molecular SMILES rather than reaction SMILES.
3. **Complexity Overhead**: 768-D + deeper network = more parameters
4. **Dense Embeddings**: May capture irrelevant molecular features for this task

When Would ChemBERTa or RXNFP Excel?

Advanced embedding approaches would outperform one-hot in:

- **Novel Molecules**: Predicting yields for unseen components
- **Larger Datasets**: 10,000+ reactions to learn representations
- **Diverse Reactions**: Multiple reaction types requiring transfer learning
- **Structural Tasks**: Problems requiring molecular substructure understanding
- **Reaction-Specific Models**: RXNFP trained on reactions may capture mechanistic patterns

Model Comparison Analysis

The nearly identical performance of Random Forest, Gradient Boosting, and Neural Network (all RMSE 7.0%, R^2 0.93) demonstrates that **feature representation matters more than model architecture** for this dataset. One-hot encoding provides such clear, informative features that even simple tree-based models can achieve excellent results. This contrasts with dense embeddings like ChemBERTa, where more complex neural architectures are necessary to decode the information.

One-hot predictions are consistently closer to experimental values.

Web Application

An interactive Gradio interface for practical yield prediction using the one-hot model is developed. Features include:

- Dropdown menus for component selection
- Instant yield prediction with confidence
- Reaction component summary display

The app converts user selections to SMILES, creates one-hot vectors, loads the trained Keras model, and displays formatted predictions.

Limitations & Insights

Limitations:

1. Dataset limited to Buchwald-Hartwig with p-toluidine
2. One-hot cannot extrapolate to novel components
3. Reaction conditions (temperature, concentration) not included
4. ChemBERTa not fine-tuned on reaction data
5. RXNFP version incompatibilities prevented evaluation

Insights:

1. Feature representation dominates model choice for fixed component sets
2. Simpler models can outperform complex ones with limited data
3. Data efficiency crucial: one-hot requires less data
4. Reaction-specific pre-training (RXNFP) may be key for transformers

Conclusion

Summary of Key Findings

Multiple machine learning approaches for Buchwald-Hartwig yield prediction are compared. One-hot encoding achieved excellent performance across Random Forest, Gradient Boosting, and Neural Network models, outperforming ChemBERTa embeddings.

For the Ahneman dataset with fixed molecular components, one-hot encoding is superior due to direct component representation, model flexibility, fewer parameters, better generalization with limited data, and high interpretability. The comparable performance across different models demonstrates that feature representation matters more than model architecture for this dataset.

Limitations & Challenges

Main constraints include dataset scope (single reaction type, single amine), the inability of one-hot to predict novel components, missing reaction condition variables, lack of ChemBERTa fine-tuning on reactions, and technical challenges with RXNFP version compatibility.

Future Work

Promising directions include:

1. **Fine-tune BERT**: Train ChemBERTa specifically on reaction data
2. **RXNFP Integration**: Resolve dependency conflicts to evaluate reaction-specific embeddings pre-trained on reaction SMILES data.
3. **Hybrid Approach**: Combine one-hot with molecular descriptors

The choice of molecular representation should be guided by application needs, dataset size, and whether predictions are needed for novel components.

Individual Contribution

In this project, I was responsible for the aspects including dataset acquisition from the Open Reaction Database, data preprocessing and extraction of reaction components, implementation of one-hot encoding with multiple ML models (Random Forest, Gradient Boosting, Neural Network), ChemBERTa embedding pipeline implementation, attempted RXNFP integration with dependency troubleshooting, neural network architecture design and hyperparameter tuning, training and evaluation of the models, development of the Gradio web application, and comprehensive result analysis.

During development, AI-assisted coding tools for debugging and guidance on implementation were used, but design decisions, experimental methodology, and scientific interpretations were made independently. Where I used existing models (ChemBERTa from Hugging Face), datasets (Ahneman dataset from ORD), or libraries (TensorFlow, scikit-learn, RDKit), I have appropriately cited them and adapted them as needed for this specific project. All interpretations and conclusions in this report reflect individual work.

Code Availability

To ensure reproducibility, all code is available in a GitHub repository.

- **Repository Link**: `https://github.com/HatefRahimi/DigitalAlchemy`
- **GitHub Username**: Hatef Raheeme
- **Repository Name**: Digital Alchemy

The repository contains:

- Data processing notebooks for ORD extraction
- One-hot encoding implementation
- ChemBERTa implementation
- Gradio web application
- Trained models (`yield_model.keras`)
- Reaction Smiles CSV file

Software Requirements and Installation

The project requires Python 3.9 or higher. All required packages can be installed using pip:

Core Dependencies:

- NumPy: Numerical computing
- Pandas: Data manipulation
- Matplotlib: Visualization
- Seaborn: Statistical visualization
- Scikit-learn: Machine learning models
- TensorFlow: Neural network implementation

ChemBERTa Implementation:

- Transformers: HuggingFace transformers library
- PyTorch: Deep learning framework
- RDKit: Chemical informatics

Web Application:

- Gradio: Interactive web interface

Installation:

```
pip install numpy pandas matplotlib seaborn
pip install scikit-learn tensorflow
pip install transformers torch
pip install gradio
pip install rdkit
```

Running the Web Application:

```
# Ensure yield_model.keras is in the same directory
python yield_predictor_Gradio.py

# Application will launch at http://localhost:7860
```

All code has been tested on Ubuntu 24.04 and Windows 11.

References

(1) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **2019**, 47 (D1), D1102–D1109. https://doi.org/10.1093/nar/gky1033.

(2) Open Reaction Database. Machine Learning Examples. *ord-schema repository.*

Available at: `https://github.com/open-reaction-database/ord-schema`.

(3) Open Reaction Database. Available at: `https://open-reaction-database.org`.

(4) Kearnes, S. M.; Maser, M. R.; Wleklinski, M.; Kast, A.; Doyle, A. G.; Dreher, S. D.; Hawkins, J. M.; Jensen, K. F.; Coley, C. W. The Open Reaction Database. *J. Am. Chem. Soc.* **2021**, 143, 18820–18826. https://doi.org/10.1021/jacs.1c09820.

(5) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint* **2018**, arXiv:1810.04805.

(6) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2017; Vol. 30.

(7) Chithrananda, S.; Grand, G.; Ramsundar, B. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. *arXiv* **2020**, arXiv:2010.09885. https://arxiv.org/abs/2010.09885.

(8) Author1, A.; Author2, B.; Author3, C. Title of the Article. *Digital Discovery* **2025**, Volume, Page–Page. https://doi.org/10.1039/D5DD00348B.

Appendix

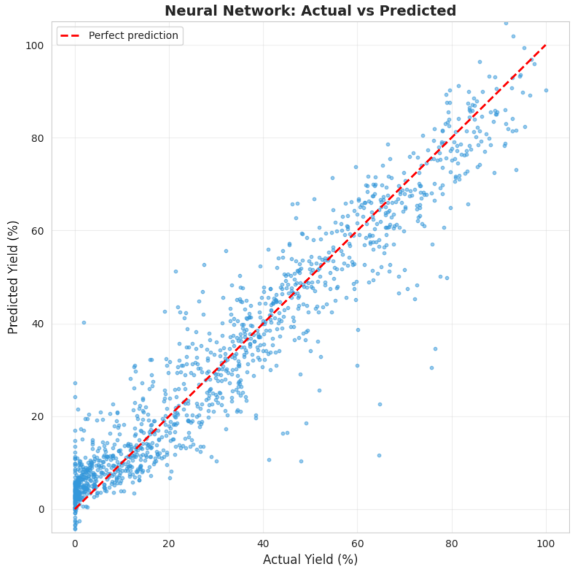


Figure 1: one-hot encoding predictions: Scatter plot of actual vs predicted yields on test set showing excellent correlation ($R^2 = 0.93$).

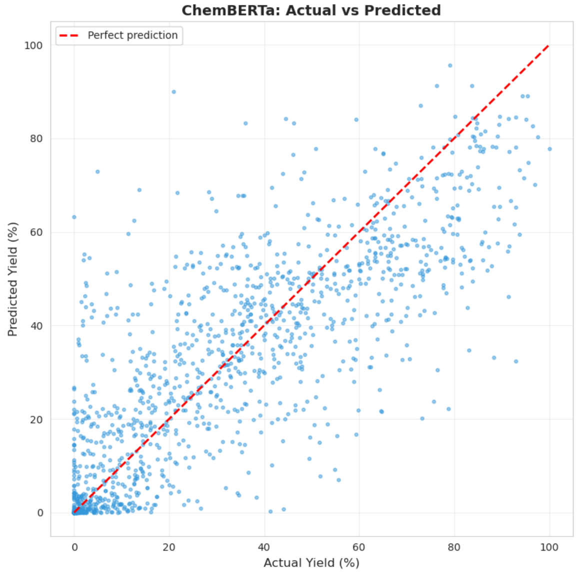


Figure 2: ChemBERTa predictions: Scatter plot showing moderate correlation ($R^2 = 0.65$) with larger spread than one-hot encoding.

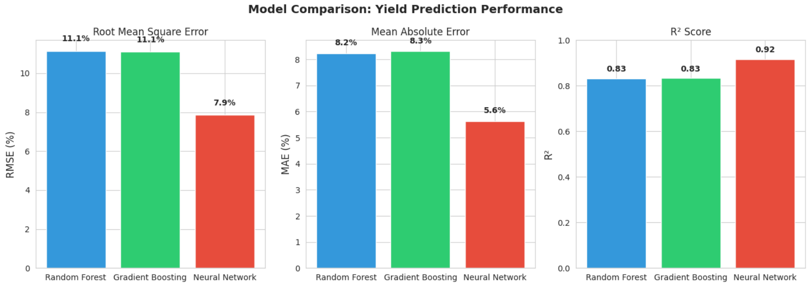


Figure 3: Model comparison for one-hot encoding: Random Forest, Gradient Boosting, and Neural Network show different performance levels, with Neural Network achieving best results compared to tree-based models.

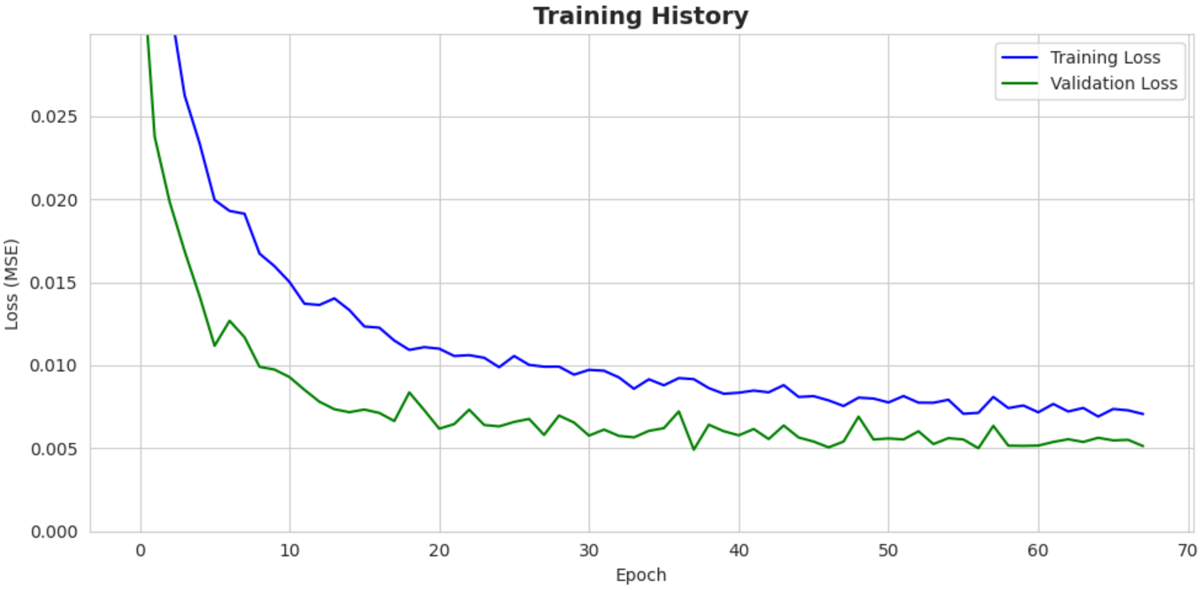


Figure 4: Training history for one-hot encoded neural network showing fast convergence. Model converges in approximately 30-40 epochs with minimal overfitting.

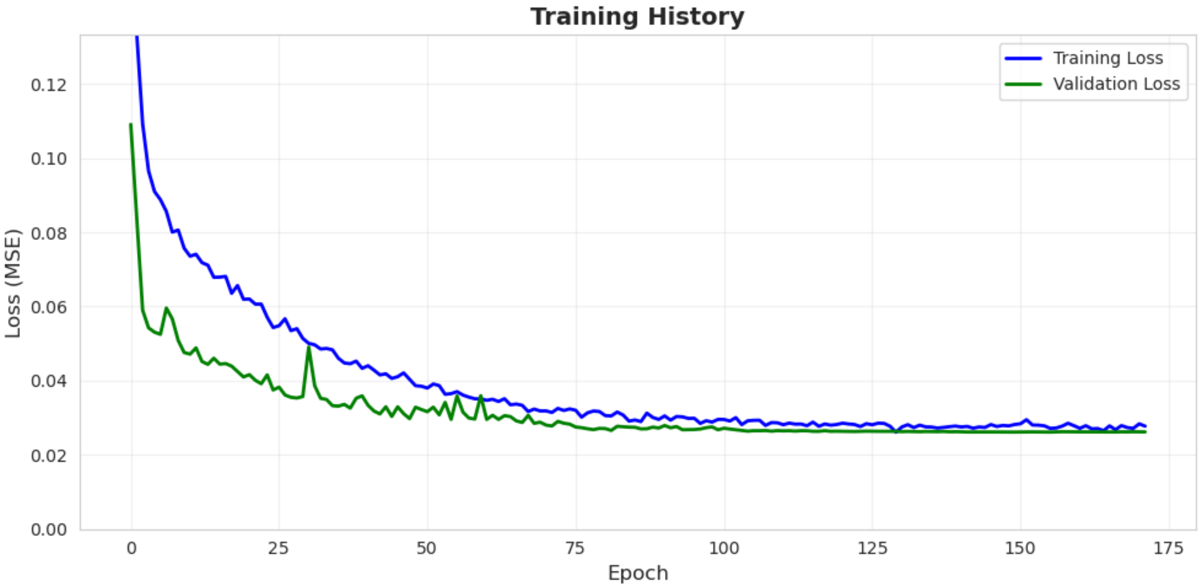


Figure 5: Training history for ChemBERTa neural network showing slower convergence. The model requires approximately 100 epochs to converge, with a larger gap between training and validation loss, indicating a more complex optimization landscape.

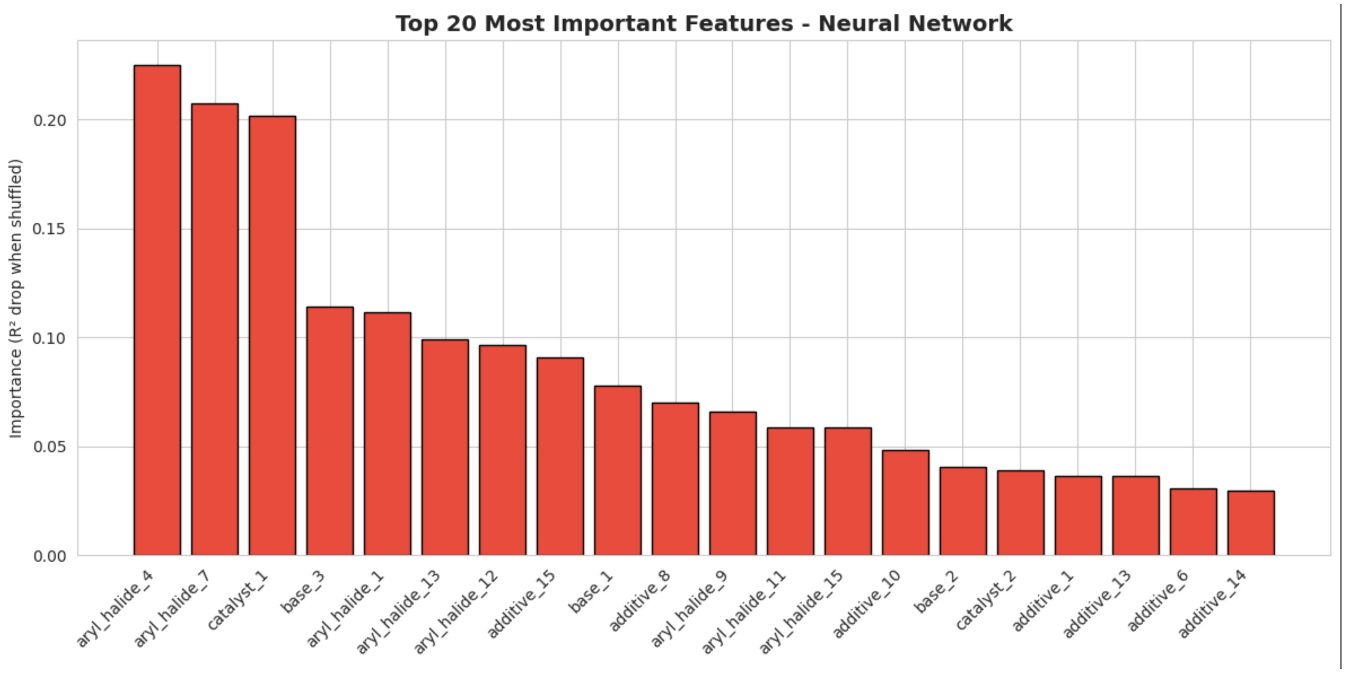
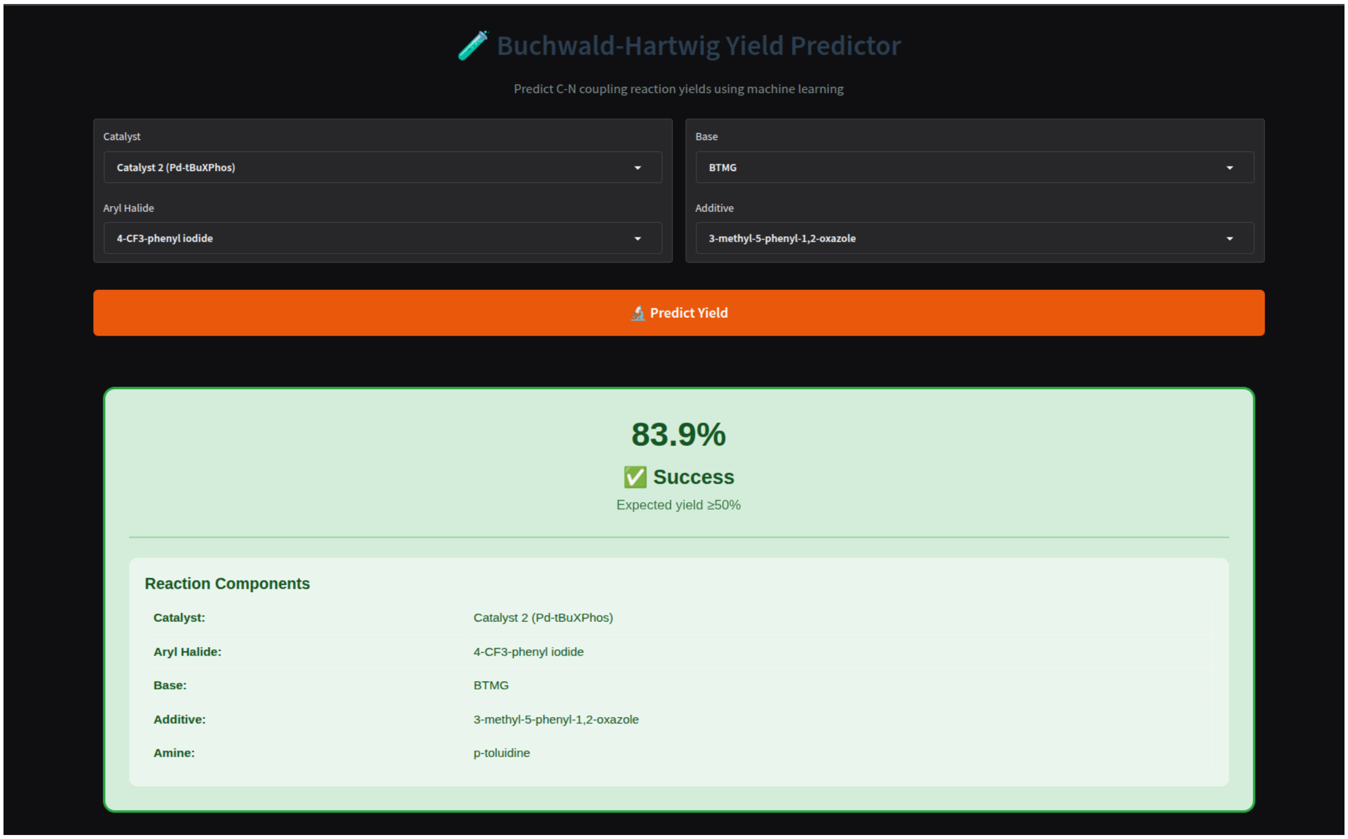


Figure 6: Top 20 most important features (permutation importance). Top 5: (1) 1-chloro-4-methoxybenzene, (2) 1-chloro-4-ethylbenzene, (3) X-Phos catalyst, (4) MTBD base, (5) 1-chloro-4-(trifluoromethyl)benzene.



🧪 Buchwald-Hartwig Yield Predictor

Predict C-N coupling reaction yields using machine learning

Catalyst: Catalyst 2 (Pd-tBuXPhos)

Aryl Halide: 4-CF3-phenyl iodide

Base: BTMG

Additive: 3-methyl-5-phenyl-1,2-oxazole

🔬 Predict Yield

83.9%

✅ Success

Expected yield ≥50%

Reaction Components

Catalyst:	Catalyst 2 (Pd-tBuXPhos)
Aryl Halide:	4-CF3-phenyl iodide
Base:	BTMG
Additive:	3-methyl-5-phenyl-1,2-oxazole
Amine:	p-toluidine

Figure 7: Gradio web application interface showing component selection dropdowns and yield prediction display. The interface allows users to select reaction components from dropdown menus and instantly predicts the expected yield using the trained neural network model.