15.095 Machine Learning Under Optimization Lens Project

COVID-19/SARS B-cell Antibody Valence Prediction

Sierra Gong, Phillip Nelson

Abstract

The accurate prediction of B-cell epitopes, critical regions of proteins triggering immune responses, plays a vital role in vaccine development. This study explores a multimodal machine learning framework integrating ProtBERT-generated sequence embeddings and tabular biological features to improve antibody reaction prediction. By addressing challenges such as dimensionality, embedding integration, and effective ways to incorporate protein sub-sequence importance, this work compares the performance of interpretable and uninterpretable models across multiple metrics, including accuracy, precision, recall, and AUC. The results demonstrate that multimodal approaches using peptide-only embeddings, combined with tree-based methods—particularly XGBoost—outperform tabular-only models, achieving a 7.2% accuracy improvement over the baseline. These findings highlight the value of our developed framework in advancing therapeutic designs and improving vaccine specificity.

---

1. Introduction

The development of vaccines relies heavily on accurately predicting useful B-cell epitopes, which are specific regions of proteins that trigger an immune response. This project focuses on enhancing antibody reaction prediction accuracy by addressing the unique challenges of integrating sequential and tabular data. The significance of this work lies in its potential to improve vaccine design by providing precise predictions. The project aims to create a scalable, high-performance framework for epitope performance classification by leveraging a multimodal approach and comparing advanced machine learning methods.

---

2. Data

The dataset comprises 12,076 epitopes, their immune response, and two primary types of inputs: **protein sequence data** and **tabular features**. The sequence data includes **parent protein sequences** and **sub-peptide sequences** derived from the parent proteins, capturing biological context and long-range dependencies. We believe B-cells inducing antigen-specific immune responses in vivo produce large amounts of antigen-specific antibodies by recognizing the subregions (epitope regions) of antigen proteins. The tabular features, derived from the

Immune Epitope Database (IEDB), represent 8 covariates associated with IgG antibody types. **Chou-Fasman** (β turn, peptide feature), **Emini** (relative surface accessibility, peptide feature), **Kolaskar-Tongaonkar** (antigenicity, peptide feature), **Parker** (hydrophobicity, peptide feature), **isoelectric point** (protein feature), **aromaticity** (protein feature), **hydrophobicity** (protein feature), and **stability** (protein feature).

Labels were simplified into binary categories, "Positive" for epitope-inducing and "Negative" for non-epitope-inducing, ensuring a focus on actionable outcomes. Preprocessing steps included removing high-homology peptides (40% similarity threshold) and splitting the data into training and testing sets to avoid overlapping parent proteins. These steps ensured a robust dataset for evaluating multimodal classification models. The processed dataset now has roughly 27.2% negative labels and 72.8% positive.

---

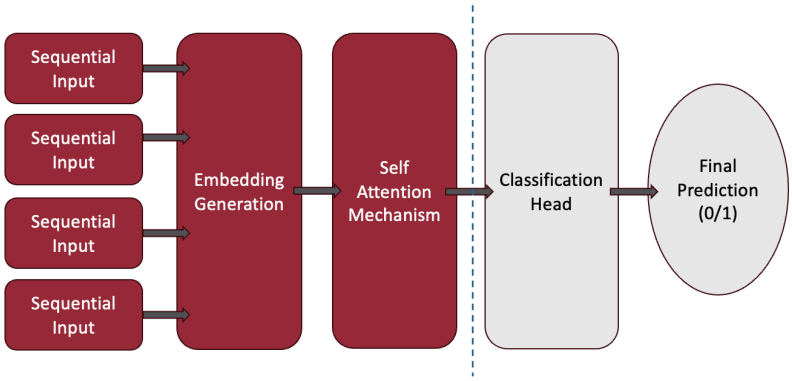
3. Challenges

This project presents several challenges and intriguing aspects. First, the **generation of embeddings** requires selecting from numerous available models for biological data, each offering different strengths and limitations. Identifying the most suitable model is crucial for capturing the nuanced features of the data. Second, handling **subsequences** is particularly challenging because the model must effectively integrate both parent and embedded sequence data, which may exhibit high collinearity. This requires careful preprocessing and feature engineering to avoid redundancy while maintaining critical information. Finally, **comparing classification models** adds a layer of complexity, as some models **do not support backpropagation**, preventing the joint training of attention layers. This necessitates creative approaches to combine the strengths of attention mechanisms with classification techniques that lack gradient-based optimization. These challenges make the project a compelling blend of computational biology, machine learning, and feature engineering.

---

4. Methods

The methodology for this project involves two primary steps. The first step leverages **transformers** to extract embeddings from the sequential data, providing a high-dimensional representation of the protein sequences. A **self-attention layer** is then added to refine these embeddings, enabling the model to account for the specific context of this project by emphasizing the importance of different structural features within the sequence. In the second step, these specialized embeddings are combined with the numerical dataset (tabular covariates) to perform the final classification. Various classification methods discussed in class, including CART, Random Forest, XGBoost, Logistic Regression, Simple Neural Networks, and Optimal Classification Trees (OCT), are applied to evaluate model performance and determine the most effective approach for predicting antibody responses.



4.1 Train Embeddings and Attention Layer

We utilized ProtBert, a pre-trained model based on Bert, to generate our sequence embeddings, to capture information from both the parent protein sequence and peptide subsequence. Each embedding is composed of a 1x1024 representation of every amino acid within the original sequence for the entire sequence. We tested 5 different methods of embedding extraction:

1. Parent Only: Only the parent sequence is embedded
2. Peptide Only: Only the peptide subsequence is embedded
3. Subsequence: The parent is embedded, and the indices corresponding to the subsequence are extracted
4. Masking: Amino acids outside of the subsequence are masked during embedding generation
5. Concatenated: Both the parent sequence and peptide subsequence are embedded and the concatenated

These high-dimensional ProtBert embeddings are then refined for the binary classification task through an attention mechanism and integration of additional tabular covariates. We developed a classification neural network composed of an input layer, an embedding layer, an attention layer, 2 dense layers, and a Linear output layer using BCEwithLogitsLoss which applies a sigmoid activation to our outputs. Due to processing constraints, we trained each model for 5 epochs to finetune the BERT embedding weights.

Attention Layer Integration

To align embeddings with the classification task, an attention layer was added to emphasize biologically relevant regions within the sub-sequences. Our attention mechanism acts as a weighted pooling operation, where each token contributes to its attention score. The attention scores are used to perform a weighted summation of the token embeddings reducing the sequence dimension to 1 x 1024. The attention weights were learned during training, enabling the model to prioritize key features in the embeddings. Tabular covariates (e.g., hydrophobicity, isoelectric point) were concatenated with the attention-refined embeddings after this step, allowing the model to utilize both sequential and non-sequential data.

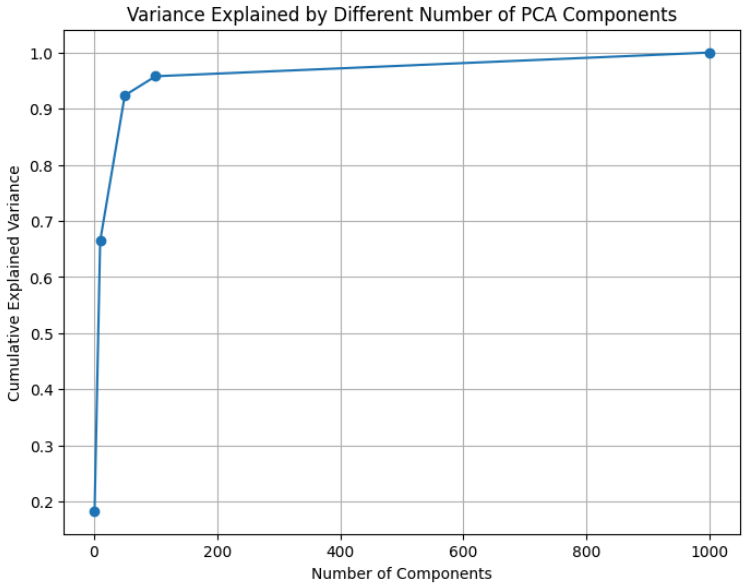
Embedding Extraction for Model Comparison

We utilized an 80-20 train test split, allowing us to test the performance of each embedding method, and used accuracy and loss as a metric for comparison. Once the classification neural network was trained, we then passed the entire dataset of sequences through our model to extract our final embeddings. We extract our final embeddings after they are passed through the attention layer to incorporate relationships between tokens across the sequence, rather than just the initial positional or token-level embeddings from the embedding layer. Additionally, the attention layer helps adapt the embeddings for the classification task under our project context.

4.2 Classification Model

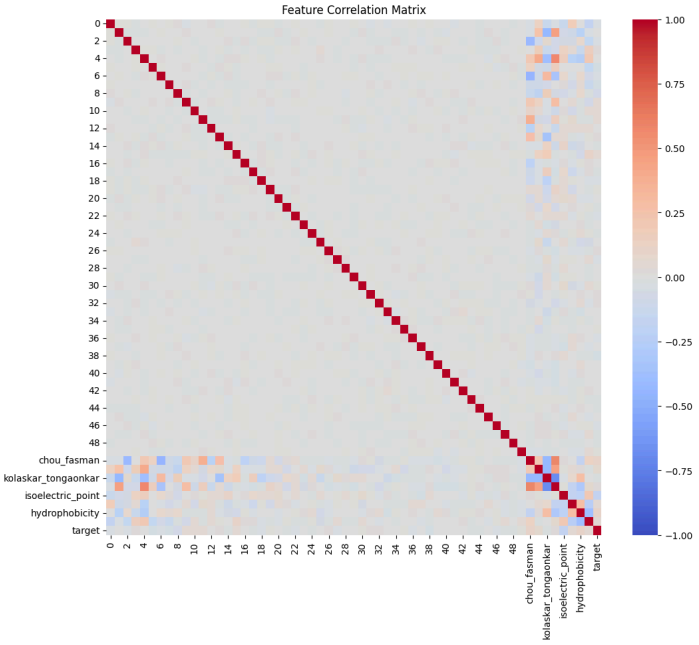
In this step, we utilized the optimal embeddings method from **Section 4.1** as the numerical input to our models. These embeddings represent high-dimensional feature vectors derived from protein sequences, providing a rich source of information for downstream classification tasks. By integrating these embeddings with tabular covariates, we aimed to leverage the complementary strengths of both data modalities in predicting antibody activity.

To ensure model performance and generalizability, we addressed the challenge of high dimensionality. Specifically, ProtBERT outputs embeddings of size 1024 per sequence, which remains unchanged in the attention layer. This results in protein sequence features that are much larger in dimensionality compared to the additional covariates and the dataset size, leading to the **curse of dimensionality**. To mitigate this, we applied **Principal Component Analysis (PCA)** for dimensionality reduction.



Using PCA, we compressed the 1024-dimensional embeddings to reduce redundancy while preserving critical variance. The data was first scaled to standardize feature distributions, and we applied the **elbow method** to determine the optimal number of components, finding that **50**

components explained the majority of the variance (92.4%). The reduced-dimensional embeddings were then concatenated with the additional covariates, forming a new, combined dataset. This concatenated dataset was **downsampled** to address class imbalance and **normalized** to ensure consistency across features. We have also checked the covariance between the new features, to make sure that they are not highly correlated, so the extracted genomic embeddings does not hinder the original performance of the dataset.



After preparing the final dataset, we conducted hyperparameter tuning for each of the six classifiers—CART, Random Forest, XGBoost, Logistic Regression, Simple Neural Networks, and Optimal Classification Trees (OCT)—to ensure optimal model performance for final prediction. The models were trained on two versions of the data: (1) tabular data alone and (2) the multimodal dataset combining ProtBERT-generated embeddings with tabular covariates. This setup enabled us to compare the predictive performance of these models across the two data types and assess the value of including peptide sequence embeddings.

---

5. Results

This section summarizes the performance of the sub-sequence extraction methods and classification models trained on **multimodal data** (embeddings + covariates) and **tabular-only data**, evaluating their classification accuracy, precision, recall, F1 score, AUC, and confusion matrices. Additionally, specific observations from the **Optimal Classification Trees (OCT)** models are highlighted.

5.1 Embedding Methods Performance

Evaluation of Different Sub-Sequence Incorporation Methods

Method	Evaluation Accuracy (%)	Average Loss
Parent Only	72.1	0.5514
Subsequence	73.18	0.518
Peptide Only	74.57	0.5079
Concatenated	73.91	0.5251
Masked	71.65	0.5527

Above is the accuracy and loss of the neural network with each embedding method. “Peptide Only” outperformed the other methods, suggesting the intrinsic properties of the peptide carry the most significant information regarding an epitope’s immune response. Although it may lack the additional context derived from the entire protein, this method gains accuracy due to its reduction in noise from the significant length of the parent sequence. Additionally, many epitopes share the same parent sequence, so this information alone may not provide much predictive insight. We observe methods including the parent and peptide sequence show a balanced performance, retaining context-specific information. However, their marginally lower accuracy compared to “Peptide Only”, indicates the entire parent sequence’s context might not be critical for this task.

5.2 Classification Model Performance

The results in the **Appendix** of this study highlight the comparative performance of multimodal models (combining ProtBERT embeddings and tabular features) and tabular-only models across key evaluation metrics, including accuracy, precision, recall, F1 score, AUC, and confusion matrices. Multimodal models consistently outperformed tabular-only models, particularly in metrics such as AUC and recall, demonstrating the added value of incorporating ProtBERT embeddings. **XGBoost** emerged as the top-performing model in the multimodal setting, achieving the highest AUC (0.8761) and F1 score (0.8059), showcasing its ability to effectively leverage both tabular features and embeddings to capture complex relationships in the data.

The results also reveal several key insights into model performance and the effectiveness of different data types. Firstly, interpretable models, such as CART and OCT, achieved performance metrics comparable to non-interpretable ones, such as XGBoost and Neural Networks. This suggests that interpretable approaches can provide meaningful predictions without sacrificing significant accuracy, offering a clear advantage in applications where understanding model decisions is critical. Secondly, while multimodal models incorporating ProtBERT embeddings outperformed tabular-only models, the improvement was modest. This may be because the eight covariates used in the study were carefully selected properties derived from specific peptide sequences, effectively representing the biological and genetic

structures that contribute to predictive performance. As such, the embeddings offered limited additional value.

The results further highlight the importance of model tuning. Performance across models was highly sensitive to hyperparameters, demonstrating that achieving optimal results requires careful and systematic parameter selection. This sensitivity underscores the critical role of optimization during model development.

Overall, these findings underscore the comparative advantages of multimodal approaches in tasks requiring the integration of diverse data types. ProtBERT embeddings proved to be a valuable addition, enriching the predictive capabilities of the models. Ensemble-based methods like XGBoost stood out as particularly effective, balancing high performance with the ability to handle complex data inputs. While tabular-only models remain viable for simpler tasks, the integration of embeddings offers significant improvements. Under the context of this project, multimodal models are the preferred choice for future experimentation efforts.

---

6. Discussion and Conclusion

6.1 Experiment Other Models

Future research opportunities lie in leveraging advanced transformer models such as AlphaFold's and ESM (Evolutionary Scale Modeling) and testing how different embeddings fit into our project's context. These models provide highly specialized embeddings by incorporating structural and evolutionary insights into protein representation. Other creative ways to utilize the peptide sequence (subsequence) information could also potentially enhance model performance, enabling improved prediction accuracy by capturing unique nuanced biological shapes and relationships.

6.2 Additional Data

The study faced challenges related to a small sample size relative to the possible sequence variations, as well as an imbalanced dataset, which further reduced the number of effective learnable samples. These limitations likely hindered the model's capacity to generalize. To develop a universally applicable model, future work could focus on collecting larger datasets with more peptide samples. Additionally, incorporating infection-specific responses as new categorical features may help the model better capture how antibody activity varies under different viral invasions for the model to scale.

6.3 Curse of Dimensionality

The imbalance in feature representation—50 tabular features after PCA vs. 8 covariates and 1024 ProtBERT embeddings vs. 11,500 samples—poses challenges. Certain models assign equal importance to all features when testing, leading to potential overfitting or underutilization of the 8 critical features (CART, OCT). This highlights the potential for additional-tailored

dimensionality reduction techniques, feature selection, or regularization to ensure models effectively prioritize relevant information of the data.

6.4 Impact

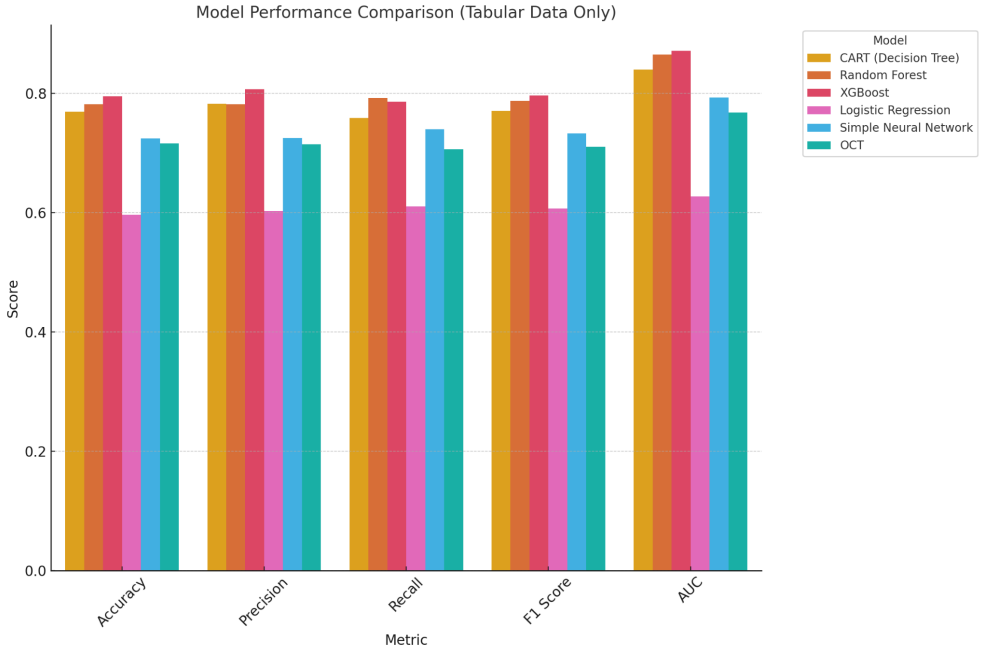
Despite these challenges, the integration of multimodal data led to a 7.2% increase in accuracy over the baseline (predicting the majority class). While, in practice, relying solely on a baseline model would be unhelpful as it does not allow for the study of vaccine effectiveness, this higher accuracy highlights the effectiveness of using the provided biological features combined with embeddings to capture both sequence and contextual information. Overall, these advancements directly contribute to enhanced vaccine specificity, offering valuable insights into antigenicity and immune responses. The ability to achieve higher accuracy and specificity has significant implications for improving therapeutic designs and developing targeted vaccines.

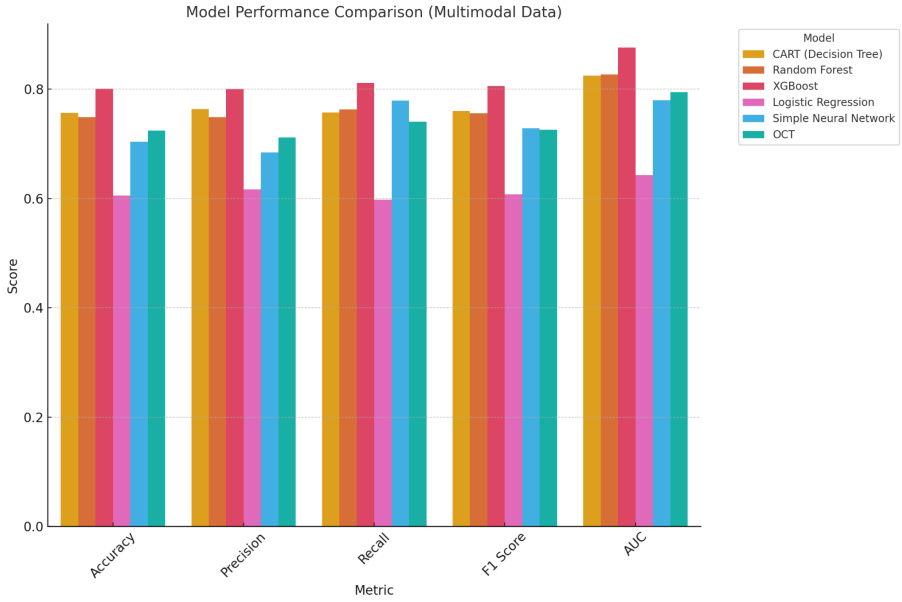
---

7. Contributions

Both authors collaborated on developing the embedding methods to test and establish the general framework for the project. Phillip Nelson was primarily responsible for coding the deep learning embeddings section of the model, while Sierra Gong focused on the final classification and result evaluation.

Appendix


Model Performance Comparison (Tabular Data Only)
Model
CART (Decision Tree)
Random Forest
XGBoost
Logistic Regression
Simple Neural Network
OCT
Score
0.0
0.2
0.4
0.6
0.8
Accuracy
Precision
Recall
F1 Score
AUC
Metric


Model Performance Comparison (Multimodal Data)
Model
CART (Decision Tree)
Random Forest
XGBoost
Logistic Regression
Simple Neural Network
OCT
Score
0.0
0.2
0.4
0.6
0.8
Accuracy
Precision
Recall
F1 Score
AUC
Metric

Model	Data Type	Best Hyperparameters	Accuracy	Precision	Recall	F1 Score	AUC	Confusion Matrix
CART (Decision Tree)	Multimodal	{'max_depth': 10, 'min_samples_leaf': 50, 'min_samples_split': 10}	0.7566	0.7633	0.7575	0.7604	0.8249	[[578, 187], [193, 603]]
CART (Decision Tree)	Tabular Only	{'max_depth': 5, 'min_samples_leaf': 5, 'min_samples_split': 20}	0.7694	0.7824	0.7588	0.7704	0.8402	[[597, 168], [192, 604]]
Random Forest	Multimodal	{'max_depth': None, 'min_samples_leaf': 5, 'min_samples_split': 5, 'n_estimators': 50}	0.7488	0.7488	0.7632	0.7562	0.827	[[561, 204], [188, 608]]
Random Forest	Tabular Only	{'max_depth': None, 'min_samples_leaf': 10, 'min_samples_split': 20, 'n_estimators': 25}	0.7816	0.7819	0.7927	0.7873	0.8652	[[589, 176], [165, 631]]
XGBoost	Multimodal	{'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 100}	0.8007	0.8005	0.8116	0.8059	0.8761	[[604, 161], [150, 646]]
XGBoost	Tabular Only	{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100}	0.795	0.8067	0.7864	0.7964	0.8714	[[615, 150], [170, 626]]
Logistic Regression	Multimodal	{'C': 10, 'penalty': 'l2', 'solver': 'liblinear'}	0.6053	0.6166	0.598	0.6074	0.643	[[469, 296], [320, 476]]
Logistic Regression	Tabular Only	{'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}	0.5964	0.6028	0.6106	0.6067	0.627	[[445, 320], [310, 486]]
Simple Neural Network	Multimodal	{'epochs': 20, 'batch_size': 32, 'learning_rate': 0.001}	0.704	0.6843	0.7788	0.7285	0.7795	[[479, 286], [176, 620]]
Simple Neural Network	Tabular Only	{'epochs': 20, 'batch_size': 32, 'learning_rate': 0.001}	0.7245	0.7254	0.7399	0.7326	0.7928	[[542, 223], [207, 589]]
OCT (Optimal Tree)	Multimodal	Default Hyperparameters	0.7239	0.7116	0.7407	0.7256	0.7948	[[570, 231], [200, 560]]
OCT (Optimal Tree)	Tabular Only	Default Hyperparameters	0.7162	0.7148	0.7065	0.7106	0.7677	[[544, 217], [226, 574]]