A Machine Learning approach for classification of Protein Residue Interactions

Arianna Pia De Laurentis
ariannapia.delaurentis@studenti.unipd.it

Filippo Bomben
filippo.bomben@studenti.unipd.it

Abstract

In this study, we explore the classification of interactions between protien residues by using machine learning techniques, based on data extracted from the RING tool.

The dataset used includes over 2.4 million labeled interactions from 3,299 Protein Data Bank (PDB) structures, spanning eight interaction classes. Due to a high degree of class imbalance, we applied the Synthetic Minority Oversampling Technique (SMOTE) to improve model generalization across rare interaction types. Preprocessing steps included imputation of missing values, encoding of categorical variables, feature scaling, and dataset stratification.

Two models were developed and compared: a tree-based classifier (Histogram-based Gradient Boosting) and a fully connected neural network designed from scratch using PyTorch. Model evaluation was carried out using accuracy, confusion matrices, and multiclass ROC AUC scores to assess both overall and per-class performance.

Our results demonstrate that both approaches are capable of learning meaningful distinctions between complex residue interaction types, with tree-based models showing strong interpretability and neural networks offering flexibility for further extension.

The code is available at `https://github.com/bombi00/SB-Project`.

1. Introduction

Understanding the interactions that stabilize protein structures is key to decoding their function and activity within biological systems. Among these, non-covalent contacts between amino acid residues (such as hydrogen bonds, ionic interactions, and aromatic stacking) play a central role in shaping protein conformation. To study these intra-molecular relationships, researchers have developed Residue Interaction Networks (RINs): graph-based models in which each node represents a residue, and each edge denotes a non-covalent interaction, inferred from the three-dimensional configuration of the protein.

Residue Interaction Networks (RINs) are useful tools for studying how proteins stay stable, how they change shape to perform their functions, and how their structures have evolved over time. But figuring out and labeling the different types of interactions between residues isn't as simple as measuring how close atoms are. It also requires an understanding of both the 3D structure and the chemical properties of the residues involved.

To address this, RING (Residue Interaction Network Generator) offers a comprehensive solution. RING is a bioinformatics tool that processes three-dimensional protein structures, typically obtained from the Protein Data Bank (PDB), and detects inter-residue contacts based on both spatial and physico-chemical parameters. It classifies these interactions into several biologically meaningful categories, such as Hydrogen Bonds (HBOND), Van der Waals forces (VDW), Salt Bridges (IONIC), π–π Stacking (PIPISTACK), and others, providing a granular view of the interaction landscape within proteins.

While RING uses deterministic, geometry-based heuristics to assign contact types, this project explores an alternative route: leveraging machine learning to predict interaction classes from structural features. The aim is to build a supervised learning framework capable of inferring the type of contact between residue pairs based on statistical patterns in their descriptors. By training predictive models on labeled data extracted from RING, we seek to generalize beyond explicit geometric rules and instead capture sub-trends that might enhance classification accuracy or offer insights into ambiguous cases.

2. Dataset

The dataset used in this work consists of interaction data extracted from 3,299 Protein Data Bank (PDB) structures. Each structure is stored in a separate file that contains detailed residue-residue contact information for that specific protein. These files are organized as tab-separated tables, where each row corresponds to a single contact between two residues.

In total, the dataset includes approximately 2.5 million

residue-residue interactions spanning multiple types of contacts as defined by RING. Each table provides key information for every interaction, such as residue identifiers following BioPython conventions, along with a set of computed features describing the contact. The last column in each table specifies the interaction type, allowing classification of contacts into categories established by RING's framework.

Contact Type	Count
HBOND	901.814
VDW	640.469
PIPISTACK	32.965
IONIC	30.355
SSBOND	1.792
PICATION	7.623
PIHBOND	1.836
Unclassified	860202

Table 1: Contact types distribution in the dataset.

3. Data Preprocessing Pipeline

The data preprocessing pipeline begins by loading residue interaction data from multiple tab-separated files stored in the "features_ring" directory. These individual files, each corresponding to a different protein structure, are read into dataframes and concatenated into a single comprehensive dataset. We first remove any rows where the "Interaction" column (our target label) is missing, ensuring that all samples have valid class information for supervised learning. Next, we separate the feature set (X) from the target variable (y), dropping irrelevant identifiers like "pdb_id" and the label column itself from the features.

To ensure data quality, we addressed missing and irrelevant entries. First, interaction labels marked as missing were filled with a placeholder class `Unclassified`. Next, we identified rows with missing numerical features: if a row belonged to a class considered overrepresented (specifically `HBOND`, `VDW`, or `Unclassified`) it was dropped from the dataset if it contained missing numerical data. This selective filtering step removed noisy and incomplete samples while preserving the integrity of more informative classes.

Remaining missing values in numerical columns were imputed using the mean value of each feature within the same interaction class, ensuring that the imputation preserved class-specific distributions. Categorical features were filled with the placeholder string `missing` and subsequently encoded using label encoding, transforming them into integer representations. Numerical features were standardized using a `StandardScaler` to center them around zero and scale to unit variance, which is beneficial for models sensitive to feature magnitude.

The dataset was then split into training and validation subsets using stratified sampling to maintain class distribution. Due to substantial class imbalance where some interaction types were vastly overrepresented compared to others, we applied the `SMOTE` [3] (Synthetic Minority Oversampling Technique) algorithm. A custom sampling strategy was defined to upsample underrepresented classes to target sizes, improving balance without excessively distorting the data distribution.

During the beginning of the preprocessing phase, we experimented with Principal Component Analysis (PCA) as a means of dimensionality reduction. The goal was to reduce feature space complexity while retaining most of the relevant information. We applied PCA and selected the number of components such that approximately 95% of the total variance in the dataset was preserved.

Although this transformation led to a more compact feature representation, it introduced several drawbacks. First, the transformation made it difficult to trace individual features and interpret model outputs. Second, we observed a slight drop in classification performance, especially during training.

For these reasons, despite its potential for reducing dimensionality, we ultimately excluded PCA from the final preprocessing pipeline in favor of a more interpretable and stable feature space.

The preprocessing stage was essential to prepare the residue-residue interaction dataset for machine learning classification. This pipeline ensured that the input features were clean, properly scaled, and representative of all interaction types, enabling more robust and fair model training.

4. Approaches

The model struggles to accurately identify and separate samples from majority classes, despite their larger representation in the dataset. This is mainly due to significant class overlap in the two-dimensional feature space, where data points from different classes are heavily intertwined. Early efforts focused on addressing this issue.

Another challenge involved the representation of interaction features between residue pairs, originally encoded as categorical values. Since most models require numerical input, these features had to be properly transformed to enable effective learning.

To meet the project goals, we implemented two complementary approaches: a tree-based ensemble model for its interpretability and robustness, and a custom neural network built from scratch to capture more complex data pat-

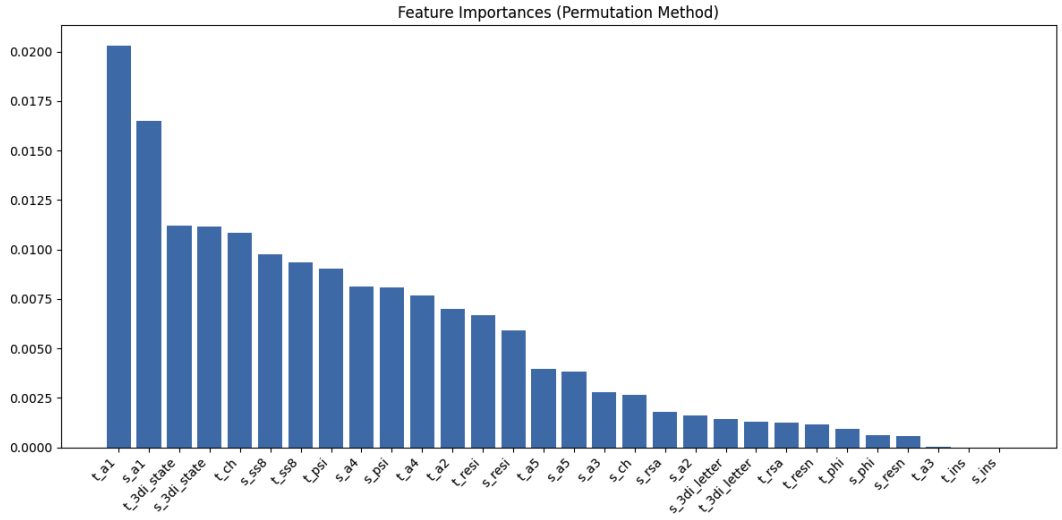

Figure 1: Graph representing feature importances in the dataset.

terns. This dual strategy enables a direct comparison between traditional machine learning and deep learning methods for residue interaction prediction.

4.1. Gradient Boosting-Based Classification Approach

The first strategy adopted to address the classification of protein residue interactions was based on tree ensembles, specifically using a gradient boosting model based on the design of LightGBM [1]. This approach leverages the predictive power of decision trees while addressing their limitations through boosting, which sequentially corrects the errors of previous trees to build a robust overall model.

After training the model on the resampled dataset (balanced using SMOTE), performance was evaluated both on the training set and on a separate validation set. The training and validation accuracy were computed to assess the model's generalization capability. Additionally, we computed the **Matthews Correlation Coefficient (MCC)**, a metric particularly effective for evaluating multi-class classifiers in imbalanced settings. Unlike accuracy, MCC takes into account true and false positives and negatives, providing a more informative score for evaluating classifier performance on skewed datasets.

.

4.2. Neural Network Architecture

In this project, we developed a fully connected feedforward neural network [2] , named `Predictor`, aimed at performing multi-class classification of protein residue interactions. The model is structured as a sequence of three dense layers, each carefully configured with specific functions and activation strategies to extract meaningful patterns from the input features and improve classification accuracy.

The key components of `Predictor` include:

- **Input Layer**: Receives the preprocessed feature vector representing residue interactions, with input size matching the number of extracted features.
- **Hidden Layers**: Two fully connected layers with 128 and 64 neurons, respectively, designed to extract increasingly abstract representations and capture nonlinear patterns.
- **Batch Normalization & ReLU**: Each hidden layer is followed by batch normalization and a ReLU activation, which stabilize training and introduce nonlinearity.
- **Dropout**: Applied after each activation (rate = 0.3) to reduce overfitting and enhance generalization by randomly deactivating neurons during training.
- **Output Layer**: A final fully connected layer outputs logits for each interaction class, which are passed to a classification loss function (e.g., cross-entropy).

This architecture balances depth and regularization, enabling the model to capture subtle distinctions among residue interaction types while minimizing the risk of overfitting on the training data. .

4.2.1 Model Training Process

The training process was designed for robustness and reproducibility. Key hyperparameters (batch size = 32, epochs

= 30, learning rate = 0.001, dropout = 0.3) were loaded from a configuration file. Random seeds were set using `torch.manual_seed` and `numpy` to ensure consistent results.

The dataset was split into training and validation sets (90/10) and wrapped with `TensorDataset` and `DataLoader`, with shuffling enabled for training data.

Training was performed on the most suitable device available (MPS, CUDA, or CPU), using the `Predictor` model. The loss function was `CrossEntropyLoss`, and optimization was handled by the Adam optimizer. Each epoch included forward and backward passes, with gradient clipping (max norm 1.0) applied to improve stability.

4.2.2 Model Evaluation Process

Evaluation plays a fundamental role in monitoring the learning process and assessing the generalization ability of a neural network. During training, after each epoch, the model was evaluated on a held-out validation set that was never seen during parameter updates. This validation phase ensures that the model is not simply memorizing the training data (i.e., overfitting), but is instead learning patterns that generalize to unseen examples.

The accuracy was computed as the percentage of correct class predictions relative to the total number of samples in the validation set. This way we were able to monitor improvements and identify potential stagnation or overfitting.

5. Model Evaluation Metrics

To thoroughly assess our classifier's performance in a multi-class setting, we employed a range of metrics beyond simple accuracy.

ROC AUC Score: Using a one-vs-rest (OvR) strategy, we computed the ROC AUC to evaluate the model's ability to distinguish each class from the rest. This metric is especially useful for imbalanced data, as it focuses on ranking quality rather than raw accuracy.

ROC Curves: ROC curves were plotted for each class to visualize the trade-off between True Positive Rate and False Positive Rate at various thresholds. These curves highlight class-specific performance and help detect potential weaknesses.

Matthews Correlation Coefficient (MCC): MCC was used for its robustness in imbalanced settings. It considers all elements of the confusion matrix and provides a balanced evaluation. Values range from -1 (total disagreement) to 1 (perfect prediction), with 0 indicating random guessing.

Confusion Matrix: The confusion matrix reveals misclassifications by comparing predicted vs. true labels. It is useful for spotting systematic errors and analyzing per-class precision and recall.

Combined, these metrics offer a comprehensive evaluation of the model's ability to classify and separate interaction types across diverse scenarios.

6. Results and Discussion

The classification models developed in this project achieved moderate overall accuracy, reflecting the inherent complexity of the task and the limitations posed by the dataset. The results, summarized below, indicate that while accuracy remains relatively low, most interaction classes excluding `PIBOND` and `PICATON` show a concentration of true positives within their correct categories.

Metric	HGBC	Neural Network
Training Accuracy	0.5231	–
Validation Accuracy	0.5027	0.4861
MCC	0.2622	0.2388
ROC AUC Score	0.8623	0.6625

Table 2: Performance comparison between Gradient Boosting Classifier (HGBC) and Neural Network.

Although the models do not achieve high accuracy, the ROC AUC values, especially for the tree-based model, suggest that the classifiers can rank predictions effectively in a one-vs-rest setup. However, the imbalance in the dataset continues to affect performance. Some classes, such as `PIBOND` and `PICATON`, are notably underrepresented, which led to poorer performance and fewer correctly predicted instances.

Despite applying SMOTE to synthetically balance class distributions, the artificial samples generated do not always reflect realistic biological patterns. This limits the generalizability of the model, especially for complex or rare interaction types. Furthermore, the disparity in training sample sizes between classes means that dominant categories (e.g., hydrogen bonds and van der Waals interactions) still exert a stronger influence during learning.

Overall, the models establish a functional baseline, demonstrating that residue interaction classification is feasible with machine learning, but also highlighting the need for more refined sampling strategies or domain-specific feature engineering to boost performance.

7. Predictor

The prediction program follows the complete preprocessing pipeline described in the earlier section, ensuring that the input data is prepared in the same

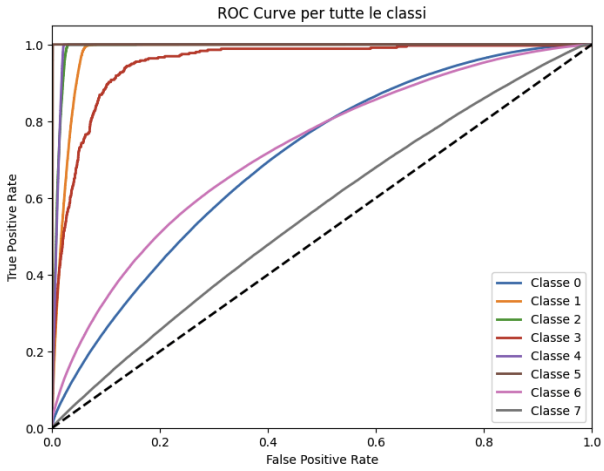

Figure 2: ROC curve for Gradient Boosting-Based Classification Approach.

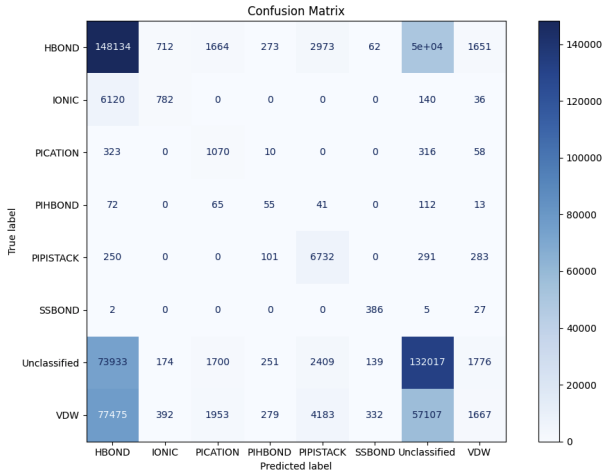

Figure 3: Confusion Matrix for Gradient Boosting-Based Classification Approach.

way as during training. It loads both the trained `HistogramGradientBoostingClassifier` model and the weights of the custom neural network, depending on the selected prediction method.

An important additional step involves loading the mean value for each of the 32 feature columns computed from the training dataset. This step is essential to handle cases where the newly downloaded PDB file may be missing some features. Since the models require the same input structure and number of features used during training, any missing columns would otherwise lead to errors or unreliable predictions.

Instead of filling missing values with zeros, which could introduce bias or mislead the model, the program replaces them with the corresponding column means. This approach ensures that the input remains within a realistic and expected range, helping maintain prediction accuracy and consistency with the training conditions.

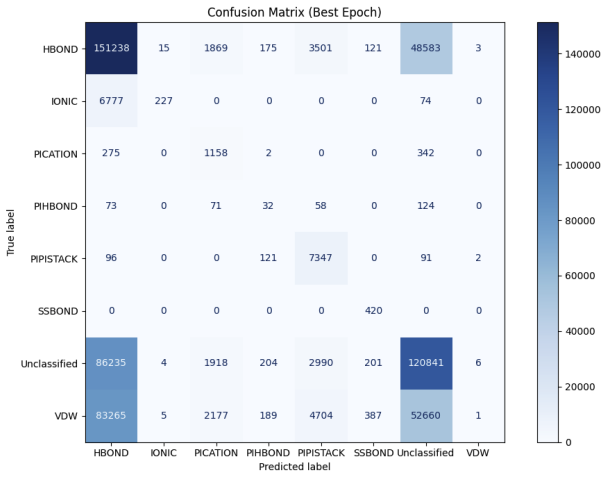

Figure 4: Confusion Matrix for Neural Network.

8. Conclusion

In this study, we explored machine learning approaches to classify protein residue interactions using features extracted from Residue Interaction Networks (RINs). Our methodology involved thorough data preprocessing, including cleaning, encoding, normalization, and synthetic oversampling with SMOTE to address significant class imbalance. Despite these efforts, the dataset posed several challenges, including overlapping feature spaces between classes and the limited realism of synthetic samples for underrepresented categories.

We implemented and evaluated two classification models: a gradient boosting classifier and a feedforward neural network. While neither model achieved high accuracy, the gradient boosting approach showed better generalization performance, particularly reflected in its higher ROC AUC and Matthews Correlation Coefficient (MCC) scores. Notably, most interaction classes achieved a majority of true positives in their respective categories, with the exception of minority classes such as `PIBOND` and `PICATON`, which suffered from insufficient data and less effective oversampling.

Our findings highlight the difficulties inherent in multiclass classification of complex biological interactions and underscore the importance of high-quality, balanced data.

Overall, this work provides a solid baseline and framework for future improvements in the automated classification of residue-residue interaction types.

References

[1] Thomas Finley Taifeng Wang Wei Chen Weidong Ma Qiwei Ye Tie-Yan Liu Guolin Ke, Qi Meng. Lightgbm: A highly efficient gradient boosting decision tree, 2017.

[2] Liliana Perescu-Popescu Nikos Mastorakis Marius-Constantin Popescu, Valentina E. Balas. Multilayer perceptron and neural networks, 2009.

[3] Lawrence O. Hall W. Philip Kegelmeyer Nitesh V. Chawla, Kevin W. Bowyer. Smote: Synthetic minority over-sampling technique, 2002.