1. Descriptive statistics and data preprocessing

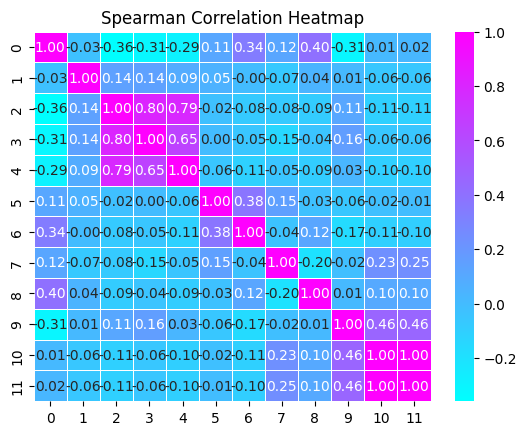
Based on the detailed explanation of VDJDB[1], we have summarized the meanings of each attribute. Due to the detailed explanations in the VDJDB documentation, the article will not repeat the meanings of each attribute. For the task of TCR specificity testing, the following 12 attributes are considered valuable: complex.id, cdr3, gene, v.segm, j.segm, antigen.epitope, antigen.gene,species, vdjdb.score, mhc.a, mhc.b, mhc.class.

We first need to detect missing values in the database, as shown in Table 1.1:

**Table 1.** Missing value detection

|  |  |  |
| --- | --- | --- |
| Index | Name | Missing values |
| 0 | complex.id | 0 |
| 1 | cdr3 | 0 |
| 2 | gene | 0 |
| 3 | v.segm | 0 |
| 4 | j.segm | 0 |
| 5 | antigen.epitope | 0 |
| 6 | antigen.gene | 0 |
| 7 | species | 0 |
| 8 | vdjdb.score | 0 |
| 9 | mhc.a | 0 |
| 10 | mhc.b | 0 |
| 11 | mhc.class | 0 |

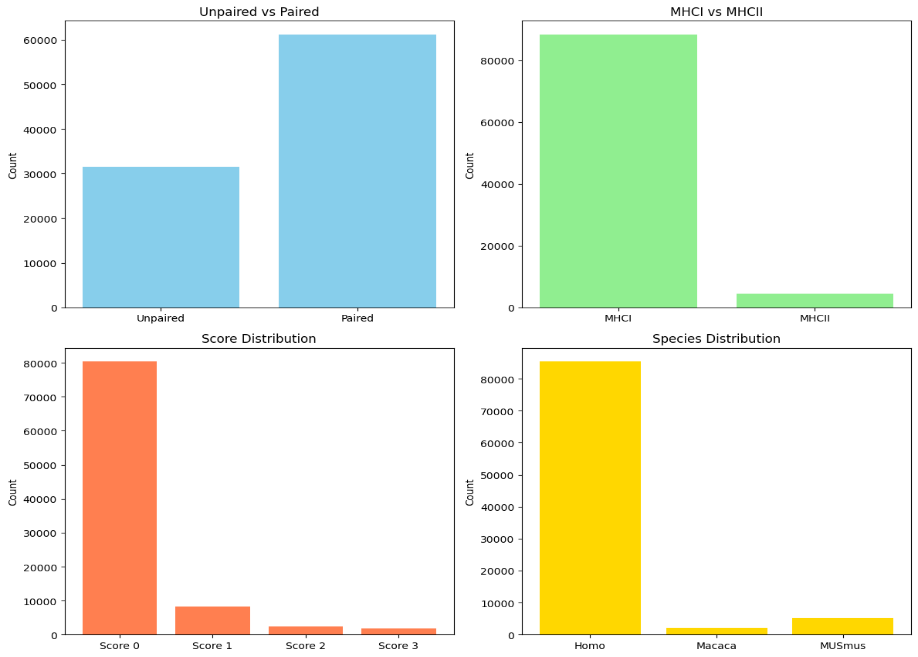
From Table 1.1, it can be seen that there are no missing values in the database for the attributes we need. Next, we conduct Spearman correlation analysis on these 12 attributes, and the results of the correlation analysis can help us screen for attributes with collinearity, thereby reducing the number of data attributes that the model needs to use. Our Spearman correlation analysis results are shown in Figure 1:



**Fig. 1.** Correlation Analysis

From the graph, it can be seen that mhc.class and mhc.b are completely correlated, and j.segm, v.segm, and gene are highly correlated. So in order to avoid the problem of high collinearity in the model, we need to avoid using highly correlated data simultaneously.

Next, we counted the data categories of some attributes as shown in the Figure 2:



**Fig. 2.** Data categories for certain attributes

Lu Tianshi's team[2] also focused on the data in the above figure when using deep learning to predict TCR specificity. When processing the data, they filtered out paired data that only selected MHC category as MHCI and belonged to human species. Rudolph, M. G [3] also pointed out in their article that the interaction process between TCR and MHC is different between MHCI and MHCII. Kim, S. M. [4] emphasized the importance of paired data in the article, pointing out that α-and β-chains play a significant role in the function of T cells.

Based on the above analysis, our preprocessing steps are as follows:

Step 1, retain only the complex.id, cdr3, gene, antigen.epitope, species, vdjdb.score, and mhc.class attributes.

Step 2, filter out data with mhc.class as MHCI.

Step 3, filter out paired data with complex.id not 0.

Step 4, filter out the data with species as HomoSapien.

Step 5, filter out data with vdjdb.score not being 0.

Step 6, rearrange the paired datapoint and merge them into the same datapoint(cdr3 with TRA and cdr3 with TRB).

After introducing the preprocessing methods, we will introduce the first part of the model.

1. Part ⅠSelecting antigens based on Editing Distance

Vujovic M[5] used the distance matrix of TCR for clustering in their paper. Inspired by this, the first part of our model framework also selects candidate antigens through the distance matrix of TCR.

When we calculate the distance between sequences, we take into account the edit distance, which is the minimum number of times a sequence can be changed into another sequence by adding characters, deleting characters, and modifying characters. For each change in the sequence, three cases are considered. Through calculation, we can get the formula for editing distance as follows:

\*(1)(2)(3)(4) and represent the two sequences compared, respectively, and D[i,j] represents the distance from the first character to the i character in and the distance from the first character to the j character in .

And, since the distance from any string to the empty string is the length of the string itself. We have the following formula for initializing D:

From (1.1) (1.2)(1.3)(1.4)(1.5) we can get the edit distance of the two sequences.

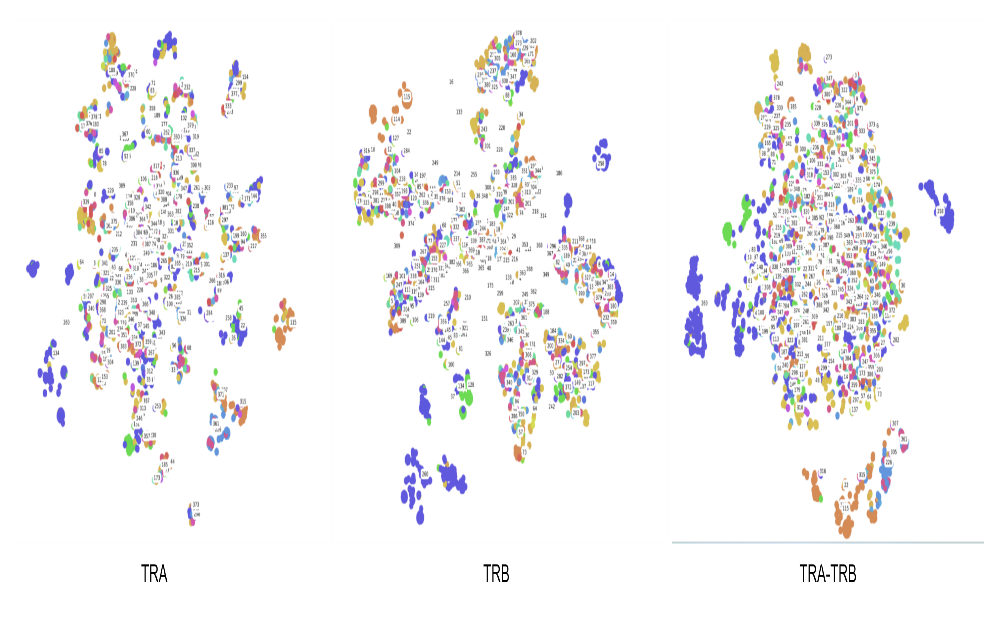
* 1. TCR visualization based on PCA and t-SNE

After calculating TRA, TRB, and TRA-TRB separately, we obtained three different distance matrices based on edit distance. Then we first use PCA to reduce the distance matrix to 50 dimensions and explain the variance as shown in the table:

**Table 1.** PCA explained variance

|  |  |  |  |
| --- | --- | --- | --- |
| Matrix | TRA | TRB | TRA-TRB |
| variance | 0.96041 | 0.965254 | 0.93969 |

It can be seen that after using PCA to reduce dimensionality to 50 dimensions, most of the information was still retained. Then, we used t-SNE for dimensionality reduction, and the results of dimensionality reduction are shown in the following figure:



**Fig. 1.** Visualization after dimensionality reduction

It can be intuitively seen that the clusters of TRB after dimensionality reduction are more prominent, and more TCRs of the same antigen category are assigned to the same cluster.

* 1. Clustering TCR based on editing distance

We can directly use clustering algorithms to cluster the distance matrix and obtain clustering results, as shown in the Table 2,Table 3:

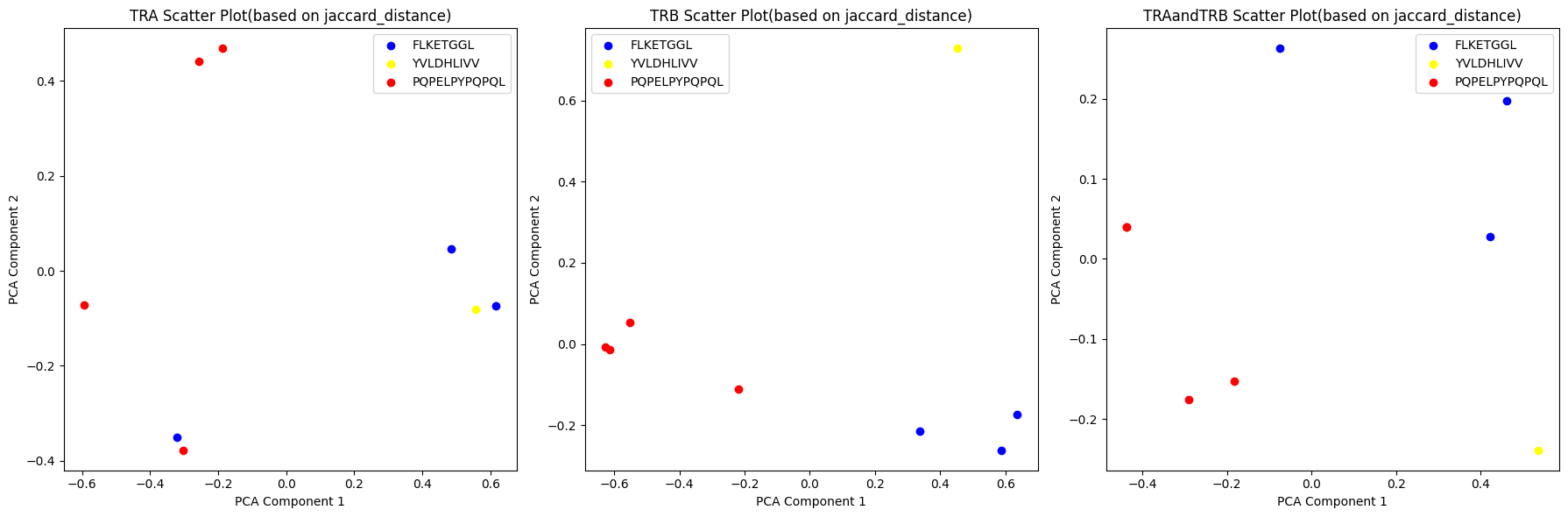
**Table 2.** Cluster performance based on k-means

|  |  |  |  |
| --- | --- | --- | --- |
|  | TRA | TRB | TRA-TRB |
| NMI | 0.01507 | 0.02621 | 0.10711 |
| ARI | 0.112103 | 0.113624 | 0.022367 |

**Table 3.** Cluster performance based on DBSCAN

|  |  |  |  |
| --- | --- | --- | --- |
|  | TRA | TRB | TRA-TRB |
| NMI | 0.41187 | 0.405358 | 0.352910 |
| ARI | 0.01912 | 0.014908 | 0.00949 |

From the table, it can be seen that directly using clustering algorithms has poor clustering performance on the entire distance matrix. This is because TCRs with specificity for similar antigens will be grouped into the same cluster, and there are many such TCRs in the database. If clustering is only performed on TCRs with dissimilar antigens, the effect will be relatively ideal, as shown in the following figure:



**Fig. 2.** Dimensionality reduction display of minority points with dissimilar antigens

For a small number of TCRs with dissimilar antigens, the clustering effect will be very good, but for databases with a large number of TCRs, it is not appropriate to directly use clustering to predict specificity. However, we can still conclude that TCRs with close editing distances have similar antigens.

So the idea of the first part of our model is to generate the target antigen through a distance matrix: calculate the edit distance matrix of the TRB between the target TCR and other TCRs in the training database, calculate the top five closest training data, and use our subsequent classifier for further more accurate discrimination.

After calculation, if we only use the antigen of the closest training data as the antigen of the target TCR, the accuracy is only 66.6%, and even if we use the first five closest training data as the antigen of the target TCR, the accuracy is only 71%. So we need to introduce the second part of our model.

1. Part Ⅱ Encoding of TCR classifier

For the encoding of cdr3 and antigen sequences, we adopt the N-gram encoding method, which has the advantage of capturing the local structural information of cdr3 and antigen well, while also retaining the length distribution information of cdr3.The formula is as follows:

According to the formula, we can adjust our encoding method by searching for the optimal N and feature size.

In addition to the N-gram encoding method, we also discussed the encoding methods of one-hot representation(1.7), improved one-hot representation(1.8), and word bags representation(1.9), and compared the performance of each encoding method in the experiment. s

In the discussion section, we demonstrated the excellent performance of N-gram encoding by comparing other encoding methods.

1. Part Ⅱ Specificity Classifier Based on Random Forest
   1. Data preparation

We use data with vdjdb.score 0 as negative class data (0 labels) and data with vdjdb.score 1, 2, and 3 as positive class data (1 label). Due to the large number of negative class data compared to positive class data, in order to reduce model bias caused by class imbalance, we adopt a random sampling method to sample negative class data, randomly selecting the same amount of negative class data as positive class data as training data.

Next, we used N-gram models to encode the CDR3 sequence and antigen receptor sequence, respectively. The hyperparameters of the model are as follows:

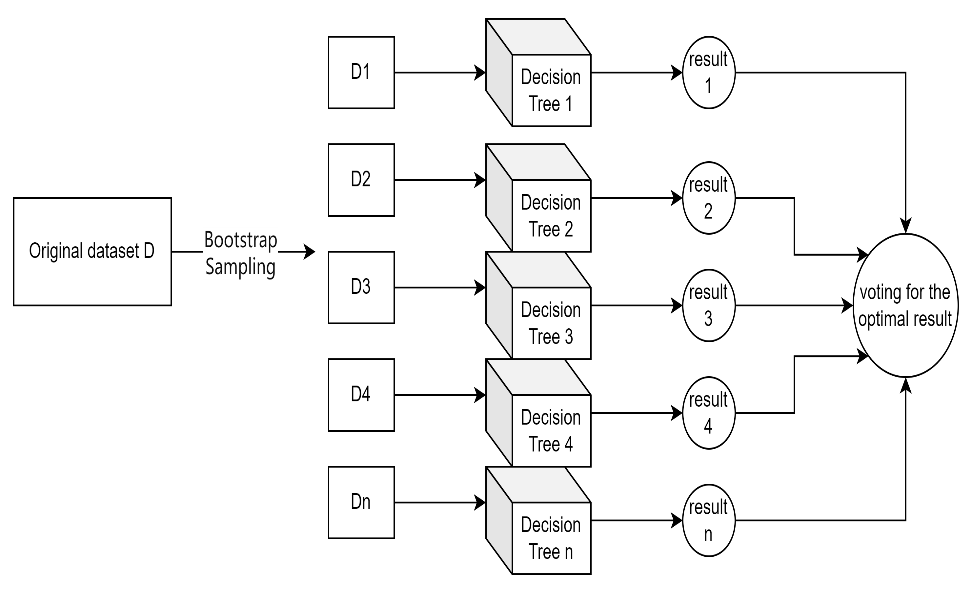
**Table 1.** Hyperparameter settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| parameter | analyzer | max features | ngram\_range | lowercase |
| value | char | 2000 | (5,5) | False |

* 1. Construction of Random Forest

We will use the encoded CDR3 and antigen sequence as feature inputs and labels as model outputs to construct a random forest model(RF).

The model construction is shown in the following figure:



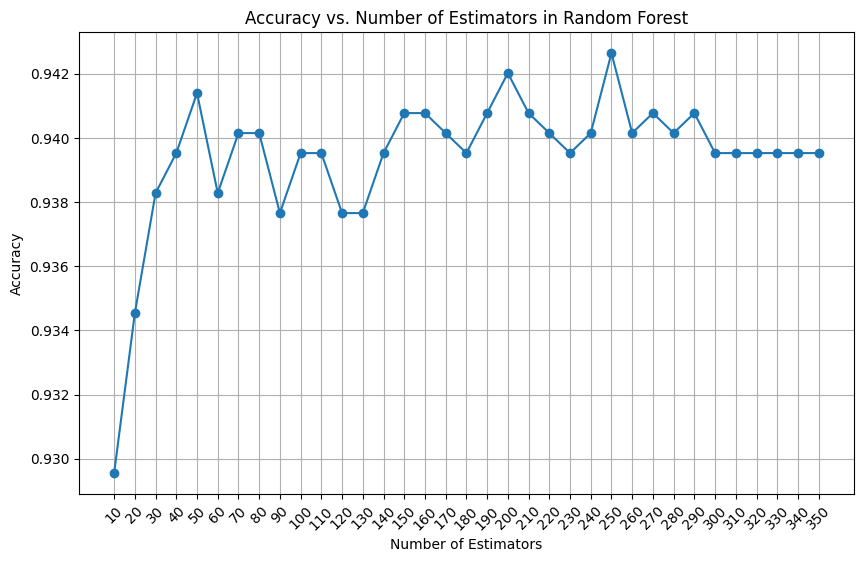
**Fig. 1.** RF classification principle

For the construction of decision trees, we use the GINI coefficient for evaluation, as shown by the following formula:

In formula (1.1), D represents the dataset, k represents the number of categories, and represents the proportion of samples belonging to category i in the dataset.

* 1. Selection of hyperparameters

For a random forest, selecting how many evaluators (number of decision trees) is an important hyperparameter. We use k-fold validation to calculate the average accuracy of samples for different numbers of random forest models, as shown in the following figure:



**Fig. 2.** Accuracy vs. Number of Estimators in Random Forest

From the graph, we can intuitively see that when the number of decision trees is 250, the model achieves the best accuracy. So we have chosen 250 evaluators, and our choices for other hyperparameters and k-fold validation settings are shown in the table below:

**Table 2.** hyperparameters settings

|  |  |
| --- | --- |
| parameter | setting |
| max\_depth | [None, 5, 10, 15] |
| max\_features | ['auto', 'sqrt', 'log2'] |
| min\_samples\_split | [2, 5, 10] |
| cv | 5 |
| scoring | accuracy |

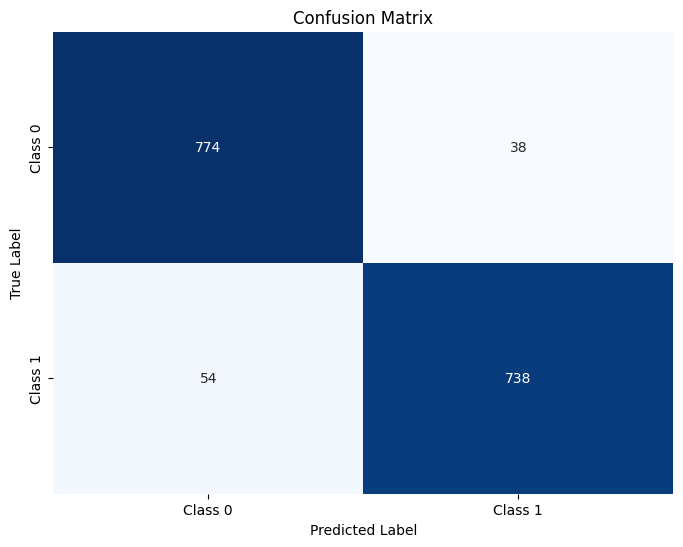
The final hyperparameter design determined by grid search is shown in the table below:

**Table 3.** final hyperparameter design

|  |  |
| --- | --- |
| parameter | setting |
| max\_depth | None |
| max\_features | log2 |
| min\_samples\_split | 2 |
| n\_estimators | 250 |

1. Results and Discussions
   1. Results

Based on our test set, we compared the results of our generative model with the true antigen of TCR. The confusion matrix of the model results is shown in the following figure:



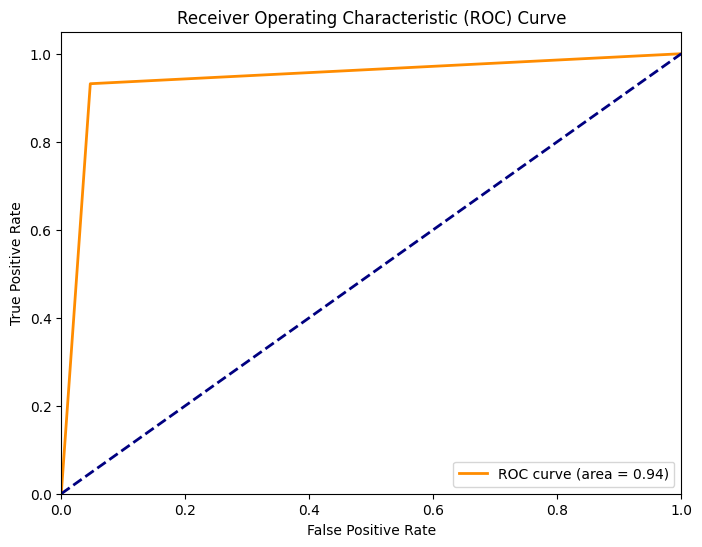
**Fig. 7.** Confusion matrix

Meanwhile, we calculated the common performance indicators as follows:

**Table 8.** Model performance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indicator | Accuracy | Recall | Precision | F1 score |
| value | 0.9426 | 0.9318 | 0.9510 | 0.9413 |

Meanwhile, we plotted the ROC curve of the model as shown in the following figure:



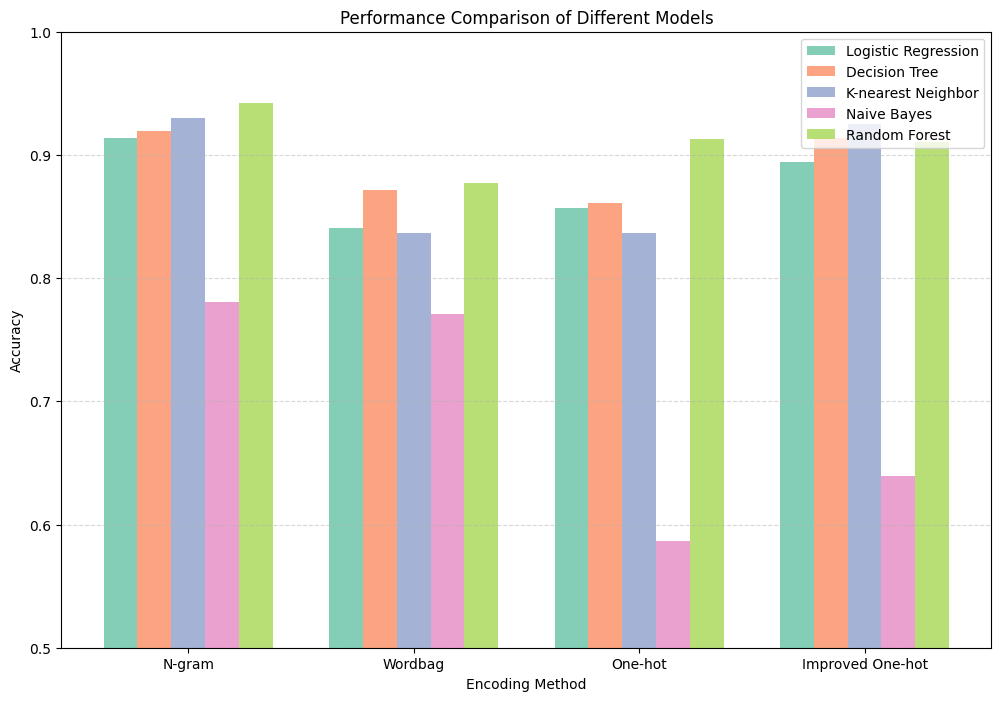
**Fig. 8.** ROC Curve

From the above model results, we can conclude that compared to antigen prediction using distance matrix, adding a classifier for discrimination is quite effective.

The disadvantage of classifiers is that we need to obtain both the antigen sequence and the TCR cdr3 sequence in order to make predictions, and the application scenarios are relatively small. Our model can better adapt to more application scenarios.

* 1. Discussions

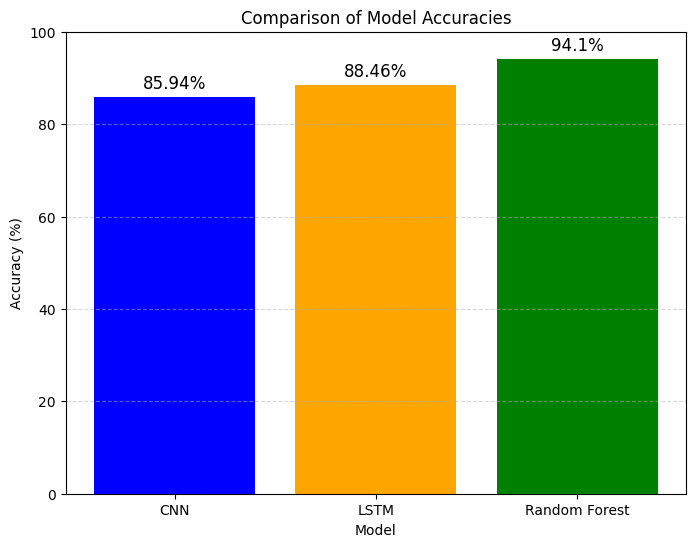
Next, we will discuss other methods to compare the performance of different models. We calculated the accuracy of other classifier models and encoding methods through k-fold validation as follows:



**Fig. 9.** performance comparison

From the graph, we can see that N-gram+RF is the best performing classifier.

At the same time, we also constructed two deep learning models, LSTM and CNN, for training. The accuracy of the test set is as follows:



Even though we adjusted the number of parameters, model structure, regularization techniques, and deep learning multiple times, they did not achieve superior performance than RF on this database.

1. Further work and improvement
   1. Enhancement of Datasets

For the task of TCR specificity detection, there are too few high-quality paired data with high reliability in the database (only a few hundred high-quality paired data with score=3). So we need to enhance our dataset. Based on the idea of DeepImmuno-GAN [6], we can also enhance the dataset through machine learning methods (GAN) or by using data from multiple databases. After enhancing the dataset, deep learning models may have surprising performance.

* 1. Improvement of antigen generation algorithm

Even though our algorithm for generating antigens based on edit distance has acceptable performance, for some TCRs, the algorithm efficiency is not high due to excessive search times. We can use lower time complexity and more reliable distance algorithms or other generation algorithms to form the first part of the model.

1. Conclusion
2. References
3. Antigenomics. (2023, June). vdjdb-db. GitHub. <https://github.com/antigenomics/vdjdb-db/blob/2023-06-01/README.md>.
4. Lu, T., Zhang, Z., Zhu, J., Wang, Y., Jiang, P., Xiao, X., ... & Wang, T. (2021). Deep learning-based prediction of the T cell receptor–antigen binding specificity. Nature machine intelligence, 3(10), 864-875.
5. Rudolph, M. G., & Wilson, I. A. (2002). The specificity of TCR/pMHC interaction. Current opinion in immunology, 14(1), 52-65.
6. Kim, S. M., Bhonsle, L., Besgen, P., Nickel, J., Backes, A., Held, K., ... & Prinz, J. C. (2012). Analysis of the paired TCR α-and β-chains of single human T cells. PloS one, 7(5), e37338.
7. Vujovic, M., Degn, K. F., Marin, F. I., Schaap-Johansen, A. L., Chain, B., Andresen, T. L., ... & Marcatili, P. (2020). T cell receptor sequence clustering and antigen specificity. Computational and Structural Biotechnology Journal, 18, 2166-2173.
8. Li, G., Iyer, B., Prasath, V. S., Ni, Y., & Salomonis, N. (2021). DeepImmuno: deep learning-empowered prediction and generation of immunogenic peptides for T-cell immunity. Briefings in bioinformatics, 22(6), bbab160.