BIOINFORMATICS PROJECT DOCUMENTATION

Group No : 1

Team Members :

1. Navaneeth Sai Patti - S20200010165
2. Nimmalapudi Sai Sriram – S20200010146
3. Yaswanth Reddy Maguluri – S20200010116
4. G.Bhargav reddy -S20200010060
5. M Praveen – S20200010045

Problem Statement :

Predicting Translation Initiation sites from genomic sequence.

A TIS indicates where the translation of genes into proteins starts. In addition to TISs, there are also stop codons (TAA,TAG, or TGA), acting as an endpoint for translation.

Motivation:

Prediction of TIS is a crucial step in understanding gene expression and protein synthesis, and can aid in identifying potential drug targets and disease-causing mutations.

Model Architecture and Diagram:

We have implemented three models :

- Support Vector Machine
- Random Forest Classifier
- CNN with a softmax classifier

Support Vector Machine :

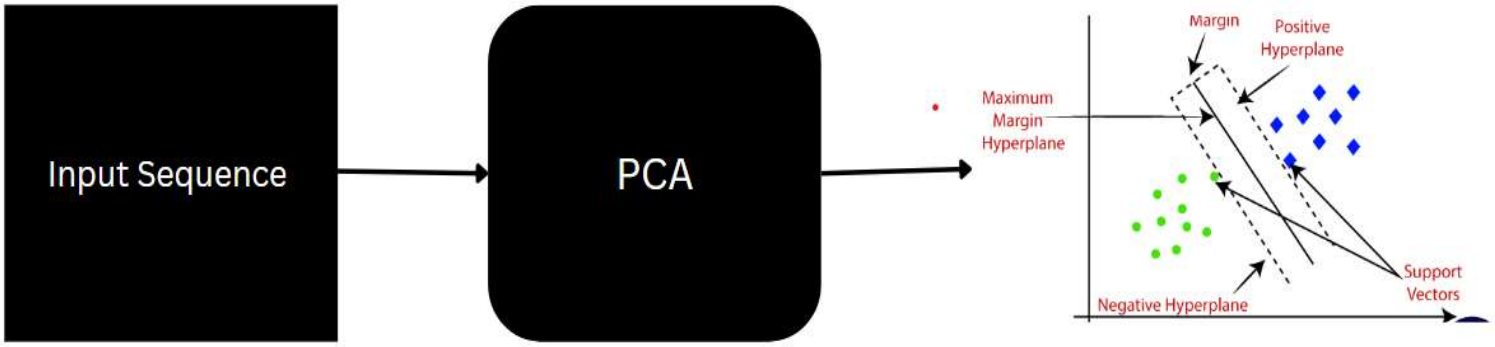
We first extracted features from the sequences like di-nucleotide frequency, tri nucleotide frequency,ATG Frequency

Each Sequence is of length 399 so we get 81 features for each sequence.

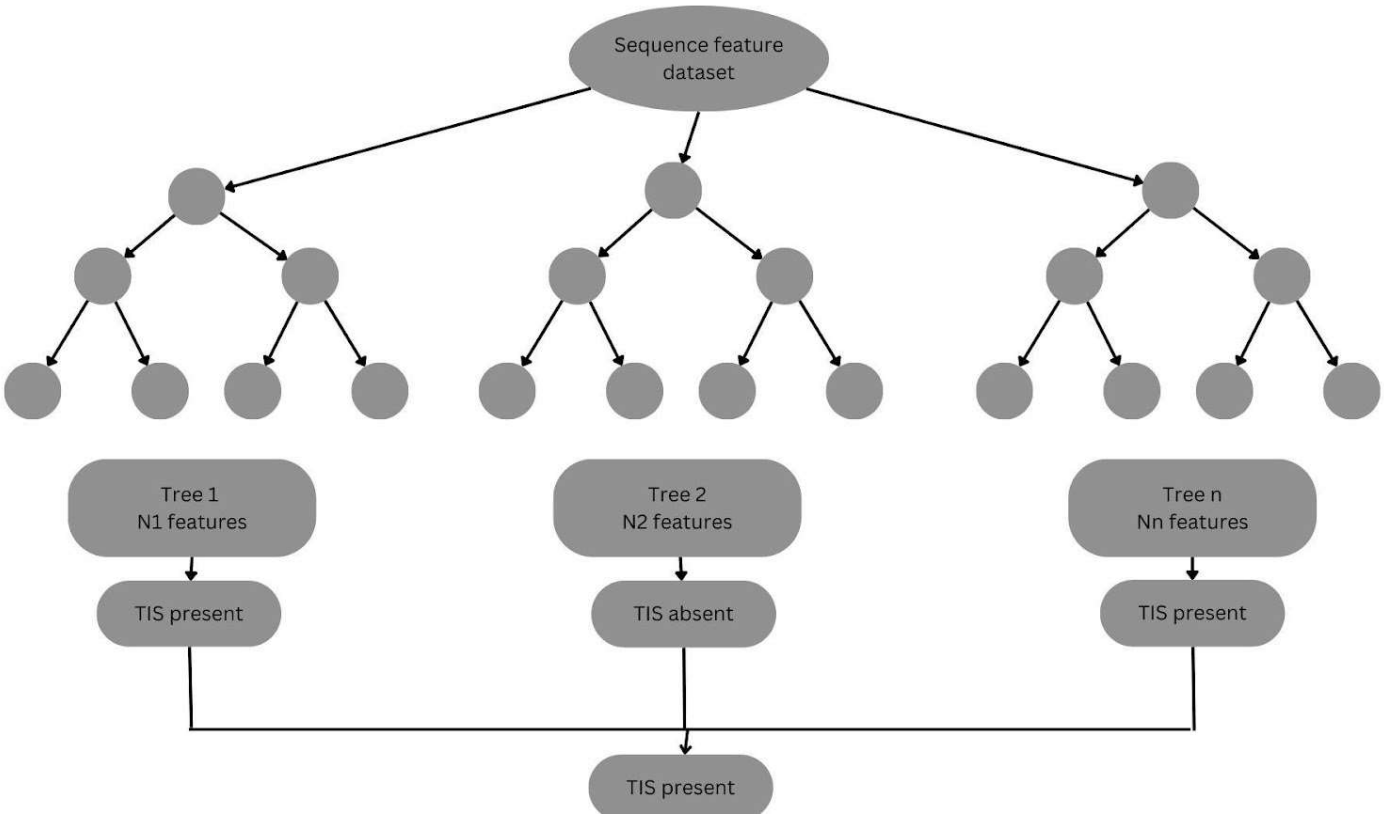
We are Standardize features by removing the mean and scaling to unit variance.

We are using PCA for dimensionality reduction 81 dimensional space to 20 dimensional space.

Finally we are applying SVM on this dataset with kernel as linear and penalty = 1 we got accuracy as 97%.



Random Forest Classifier:

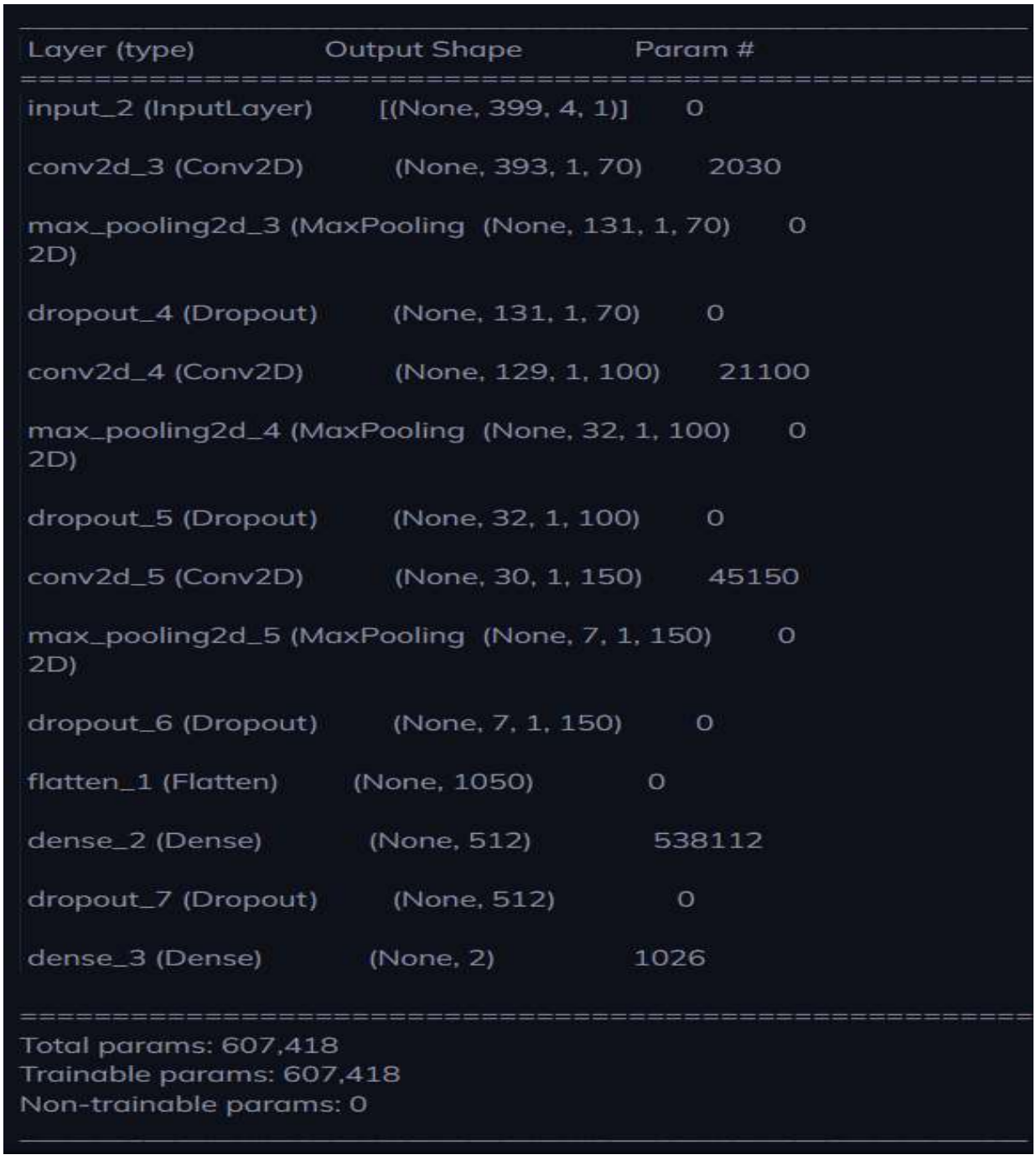


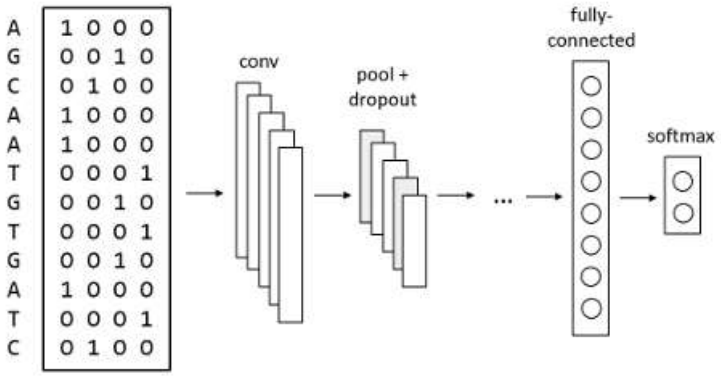
Once 20 features extracted from PCA . Dataset is splitted into multiple partitions in our case we are splitting them into 50 trees.

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

We got accuracy around 98.99%

CNN Architecture :



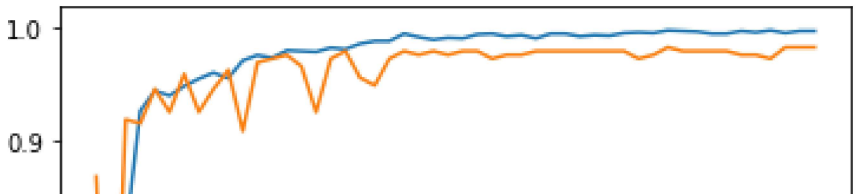


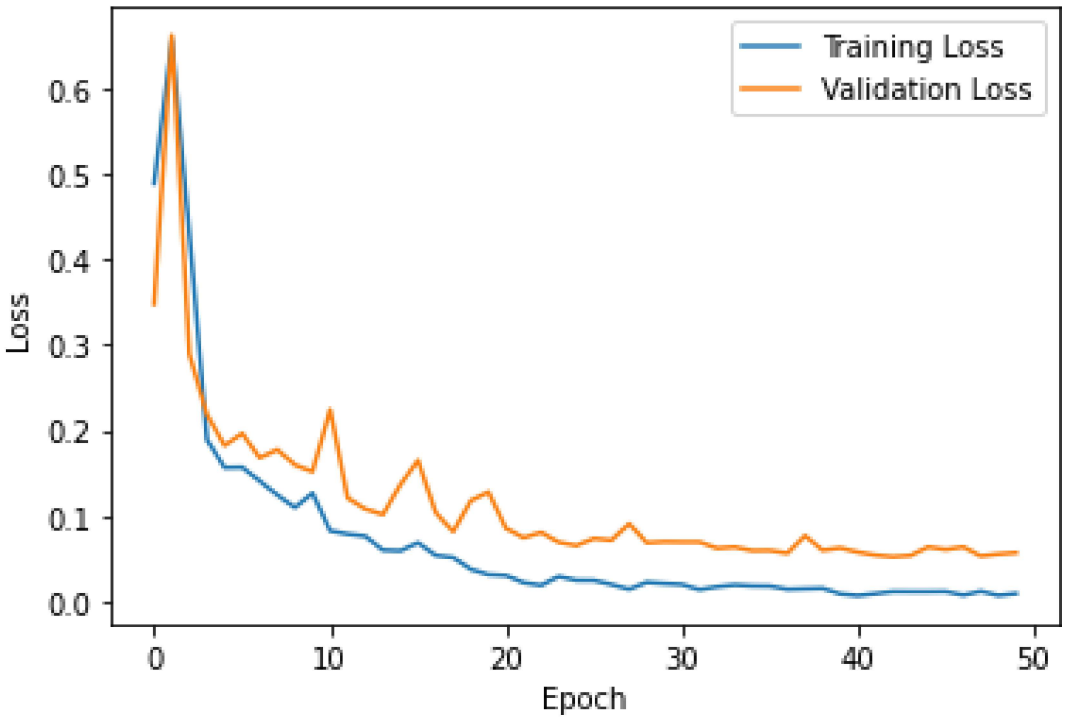
Model Architecture in a high level

It consists of three convolutional layers containing 70, 100, and 150 filters, respectively, of size 74, 31, and 31, respectively. Each of these layers is followed by a max-pooling layer (size 31 or 41, depending on the input size) and a dropout layer (p = 02). The output of this last dropout layer is then connected to a fully-connected layer with 512 neurons and another dropout layer (p = 02). Finally, a softmax layer is added, generating a probability that leads to a positive or negative classification of a candidate TIS. In addition, every convolutional and fully connected layer is succeeded by a rectified linear unit (ReLu) for non-linearization.

The weights of the model are initialized using a Glorot uniform distribution, while biases are initialized at zero. The training procedure minimizes the categorical cross-entropy cost function over the training set, using stochastic gradient descent with nesterov momentum. At the start of training, the learning rate is 005, whereafter it is halved every ten epochs, for a total of 50 epochs. After training, the validation set is used to select the predictive model that performed best during training. Finally, we evaluate the selected model against a testing set.

We got an accuracy around 97.88%




Training Loss
Validation Loss
Loss
Epoch
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
10
20
30
40
50

c)Results And Model Significance

In the Tis analysis, we used CNN, random forest, and SVM for predicting whether Tis is there in the sequence or not.

First, we have extracted features like 2-mer and 3-mer frequencies. And we have done c2 encoding and found atg frequencies. we have passed them to PCA to reduce features and finally used them on the models. Lets discuss each model below and the results.

Support Vector Machine(SVM):

SVM is a supervised learning algorithm that uses a technique called kernel trick to transform data into higher dimensions, where it becomes easier to separate the data into different classes or groups.

In classification problems, SVM tries to find the hyperplane that best separates the data points of different classes. The hyperplane is chosen in such a way that it maximizes the margin, i.e., the distance between the hyperplane and the nearest data points of each class.

After getting PCA Matrix we use train test split and split the model to pass them through the model and the architecture will be taken place as mentioned above. SVM Uses Gradient descent. It forms two support Vectors and Predicts the Tis.

We have changed the test set size relatively 5 times and have generated accuracies. For test size 0.3 We have the accuracy of around 97 and we have generated confusion matrix .

The results for confusion matrix is array([[350, 6], [6, 334]])

For Svm for

Test size accuracy

0.3 0.97

0.15 0.98

0.4 0.97

For Random Forest:

It is an ensemble learning technique that combines multiple decision trees to create a more accurate and robust model.

In a random forest, each decision tree is built independently by randomly selecting a subset of the features and data points. This reduces overfitting and increases the diversity of the individual trees. The final prediction is then made by combining the outputs of all the decision trees, either by taking the majority vote in classification problems or by averaging the outputs in regression problems Random Forest.

We got Random forest with better accuracy compared to svm. In random Forest also we change n_estimators which means generating n decision trees

For n_eztimators 500means 500 decision trees are generated. We got Accuracy is 0.989

The confusion matrix is. [[332 2] [3 359]]

CNN:

CNN stands for Convolutional Neural Network, which is a deep learning algorithm commonly used in computer vision tasks

such as image classification, object detection, and segmentation.

A CNN consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers. In the convolutional layers, the input image is convolved with multiple filters to extract features such as edges, corners, and textures. The pooling layers reduce the dimensionality of the output of the convolutional layers, by taking the maximum or average value of small regions. The fully connected layers are used for classification or regression, by mapping the extracted features to the output classes.

The backpropagation is done using kernel. We got a competitive accuracy with random forests. Here we have done c4 encoding for the sequence. And we have used 4 cnn layers. For each layer, we have decreased the matrix size and finally, we have got a 7*1 layer and later using the sgd and leaning rate we have converting it into 2*1 and changed predicting whether it is there or not.

Later we got an accuracy around 0.99 and the confusion matrix generated is

[166 1] [2 179].

Out of the three models we got random forest and CNN as the best and svm we got less compare to them.

Results.

For Svm we accuracys

[0.979,0.985,0.981,0.982,0.978]

and confusion matrix array([[350, 6], [6, 334]])

For Random forest

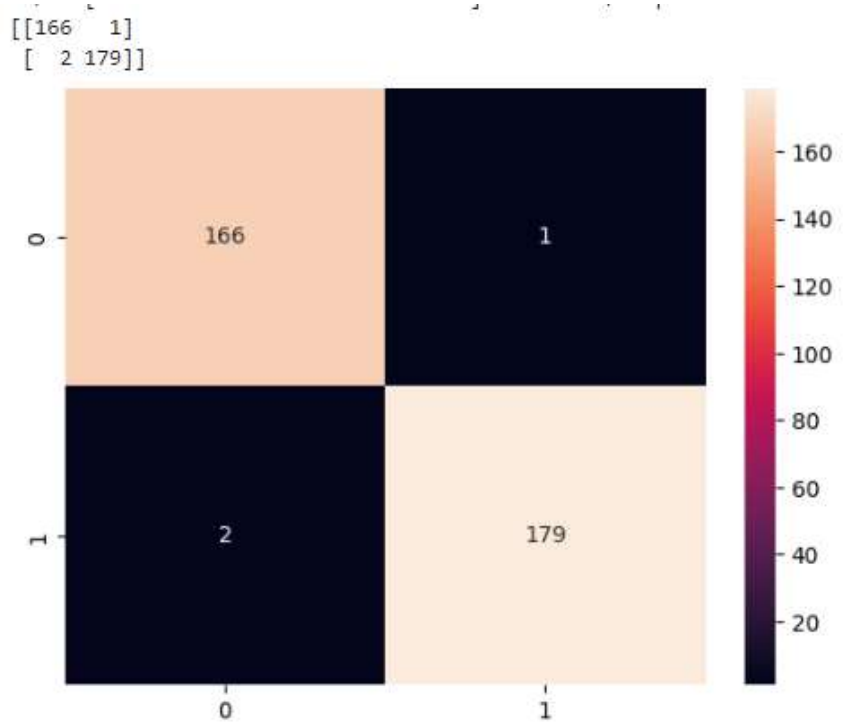
[0.985,0.989,0.998,0.997,0.992]

and confusion matrix is [[332 2] [3 359]]

For cnn

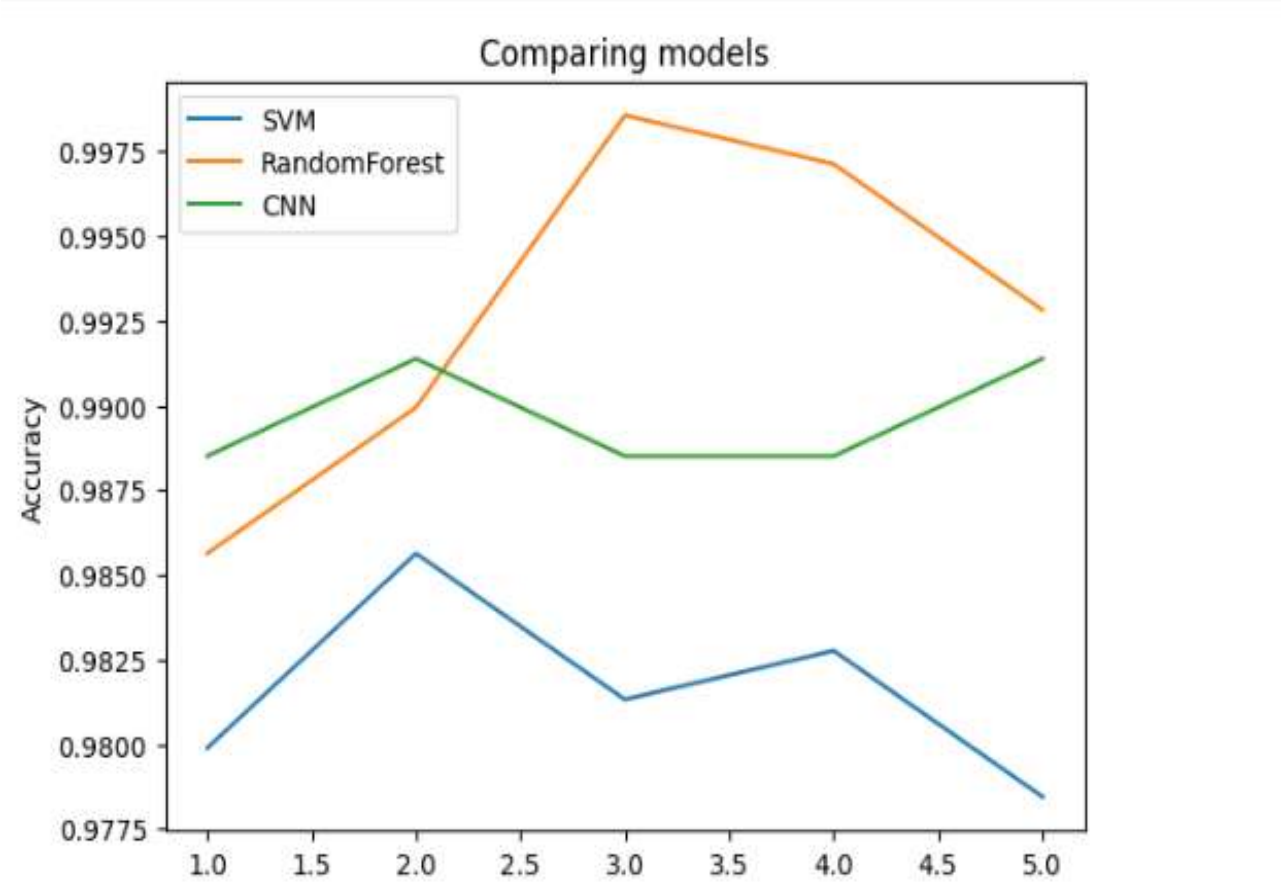
[0.988,0.991,0.988,0.988,0.991]

The confusion matrix



```
Training the model...
Learning rate:  0.05  for epoch:  0
Epoch 1/50
52/53 [============================>.] - ETA: 0s - loss: 0.4166 - accuracy: 0.7939 - auc_4: 0.8856 - precision_4: 0.7985 - recall_4: 0.7596 - true_positives_4: 1264.0000 - true_ne
Epoch 1: val_accuracy improved from -inf to 0.92905, saving model to best_model.h5
53/53 [==============================] - 5s 71ms/step - loss: 0.4146 - accuracy: 0.7951 - auc_4: 0.8868 - precision_4: 0.7997 - recall_4: 0.7611 - true_positives_4: 1274.0000 - tr
Learning rate:  0.05  for epoch:  1
Epoch 2/50
52/53 [============================>.] - ETA: 0s - loss: 0.1737 - accuracy: 0.9381 - auc_4: 0.9805 - precision_4: 0.9398 - recall_4: 0.9387 - true_positives_4: 1562.0000 - true_ne
Epoch 2: val_accuracy improved from 0.92905 to 0.94595, saving model to best_model.h5
53/53 [==============================] - 3s 60ms/step - loss: 0.1730 - accuracy: 0.9385 - auc_4: 0.9807 - precision_4: 0.9402 - recall_4: 0.9391 - true_positives_4: 1572.0000 - tr
Learning rate:  0.05  for epoch:  2
Epoch 3/50
52/53 [============================>.] - ETA: 0s - loss: 0.1480 - accuracy: 0.9447 - auc_4: 0.9858 - precision_4: 0.9405 - recall_4: 0.9495 - true_positives_4: 1580.0000 - true_ne
Epoch 3: val_accuracy did not improve from 0.94595
53/53 [==============================] - 4s 77ms/step - loss: 0.1472 - accuracy: 0.9450 - auc_4: 0.9860 - precision_4: 0.9408 - recall_4: 0.9498 - true_positives_4: 1590.0000 - tr
Learning rate:  0.05  for epoch:  3
Epoch 4/50
52/53 [============================>.] - ETA: 0s - loss: 0.1758 - accuracy: 0.9387 - auc_4: 0.9808 - precision_4: 0.9359 - recall_4: 0.9381 - true_positives_4: 1561.0000 - true_ne
Epoch 4: val_accuracy did not improve from 0.94595
53/53 [==============================] - 3s 59ms/step - loss: 0.1761 - accuracy: 0.9385 - auc_4: 0.9808 - precision_4: 0.9356 - recall_4: 0.9379 - true_positives_4: 1570.0000 - tr
Learning rate:  0.05  for epoch:  4
Epoch 5/50
52/53 [============================>.] - ETA: 0s - loss: 0.1501 - accuracy: 0.9411 - auc_4: 0.9859 - precision_4: 0.9400 - recall_4: 0.9423 - true_positives_4: 1568.0000 - true_ne
Epoch 5: val_accuracy did not improve from 0.94595
53/53 [==============================] - 3s 59ms/step - loss: 0.1493 - accuracy: 0.9415 - auc_4: 0.9860 - precision_4: 0.9404 - recall_4: 0.9427 - true_positives_4: 1578.0000 - tr
Learning rate:  0.05  for epoch:  5
Epoch 6/50
52/53 [============================>.] - ETA: 0s - loss: 0.1101 - accuracy: 0.9585 - auc_4: 0.9922 - precision_4: 0.9557 - recall_4: 0.9603 - true_positives_4: 1598.0000 - true_ne
Epoch 6: val_accuracy did not improve from 0.94595
53/53 [==============================] - 3s 60ms/step - loss: 0.1101 - accuracy: 0.9588 - auc_4: 0.9922 - precision_4: 0.9554 - recall_4: 0.9606 - true_positives_4: 1608.0000 - tr
```

The comparison of the three models



Conclusion

In this project, we used support vector machines, random forests, and convolutional neural networks to predict translation initiation sites on DNA sequences. Our results showed that all three methods performed well, with the convolutional neural network achieving the highest accuracy. Our findings suggest that machine learning can be effective in predicting translation initiation sites, and further research is needed to explore generalizability and other factors that impact accuracy. Overall, our project lays the foundation for future research in this area.

References

- DeepTIS: Improved translation initiation site prediction in genomic sequence via a two-stage deep learning model. Chao Wei, Junying Zhang , Xiguo Yuan (2021)
- Gene and translation initiation site prediction in metagenomic sequences Doug Hyatt1,2,*, Philip F. LoCascio1, Loren J. Hauser1,2 and Edward C. Uberbacher(2012)
- Global sequence features based translation initiation site prediction in human genomic sequences Neelam Goel , Shailendra Singh , Trilok Chand Aseri .(2020)
- NeuroTIS: Enhancing the prediction of translation initiation sites in mRNA sequences via a hybrid dependency network and deep learning framework Chao Wei, Junying Zhang, Xiguo Yuan, Zongzhen He, Guojun Liu, Jinhui Wu
- Interpretable Convolutional Neural Networks for Effective Translation Initiation Site Prediction Jasper Zuallaert, Mijung Kim, Yvan Saeys‡§, and Wesley De Neve