import numpy as np

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report

import tensorflow as tf

from tensorflow.keras import layers, models

data = np.load('processed\_data.npz')

X, y = data['X'], data['y']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.15, random\_state=42)

# SVM

svm = SVC(kernel='rbf', C=1.0, gamma='scale')

svm.fit(X\_train, y\_train)

print("SVM Report:")

print(classification\_report(y\_test, svm.predict(X\_test)))

# RF

rf = RandomForestClassifier(n\_estimators=100, max\_depth=15)

rf.fit(X\_train, y\_train)

print("Random Forest Report:")

print(classification\_report(y\_test, rf.predict(X\_test)))

# CNN

X\_cnn = X.reshape(-1, X.shape[1], 1)

X\_train\_cnn, X\_test\_cnn = train\_test\_split(X\_cnn, test\_size=0.15, random\_state=42)

cnn = models.Sequential([

layers.Conv1D(32, 3, activation='relu', input\_shape=(X.shape[1], 1)),

layers.Conv1D(64, 3, activation='relu'),

layers.Flatten(),

layers.Dense(64, activation='relu'),

layers.Dense(3, activation='softmax')

])

cnn.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

cnn.fit(X\_train\_cnn, y\_train, epochs=10, validation\_split=0.1, verbose=2)

cnn.evaluate(X\_test\_cnn, y\_test)