**CYBERSECURITY ANALYSIS USING AI**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix

# 1. Load the NSL-KDD dataset (download from https://www.unb.ca/cic/datasets/nsl.html)

# Assume it's preprocessed and stored as 'KDDTrain+.csv'

data = pd.read\_csv("KDDTrain+.csv")

# 2. Basic preprocessing

# Select only a subset of features for simplicity

columns\_to\_use = ['duration', 'protocol\_type', 'service', 'flag', 'src\_bytes', 'dst\_bytes', 'label']

data = data[columns\_to\_use]

# Encode categorical columns

le = LabelEncoder()

for col in ['protocol\_type', 'service', 'flag', 'label']:

data[col] = le.fit\_transform(data[col])

# 3. Split features and labels

X = data.drop("label", axis=1)

y = data["label"] # 0 = normal, 1+ = various attacks

# 4. Train/Test Split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 5. Train a Random Forest model

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

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

# 6. Predict and Evaluate

y\_pred = model.predict(X\_test)

# 7. Results

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

# 8. Visualization

sns.heatmap(confusion\_matrix(y\_test, y\_pred), annot=True, fmt="d", cmap="Blues")

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()