DDoS Detection System

CODE:

```
import math
import random

# Function to calculate the entropy of a given data sequence
def calculate_entropy(data):
    entropy = 0
    total_count = len(data)

    # Count the frequency of each unique element in the data
    frequencies = {}
    for item in data:
        if item in frequencies:
            frequencies[item] += 1
        else:
            frequencies[item] = 1

    # Calculate the entropy using the frequency distribution
    for count in frequencies.values():
        probability = count / total_count
        entropy += probability * math.log2(probability)

    entropy = -entropy
    return entropy

# Predefined threshold for DDoS detection
ddos_threshold = 4.5

# List to store packet data and their labels
```

```
packet_data = []
labels = []

# Function to send packet and update packet_data list
def send_packet():
    packet = random.randint(0, 255)
    packet_data.append(packet)
    labels.append(0)  # Label 0 indicates normal traffic

# Function to generate DDoS packets and update packet_data list
def generate_ddos_packet():
    packet = random.randint(256, 511)
    packet_data.append(packet)
    labels.append(1)  # Label 1 indicates DDoS traffic

# Generate and send normal packets
for _ in range(1000):  # Adjust the number of packets as needed
    send_packet()

# Generate and send DDoS packets
for _ in range(200):  # Adjust the number of DDoS packets as needed
    generate_ddos_packet()

# Combine the normal and DDoS packets
combined_data = list(zip(packet_data, labels))
random.shuffle(combined_data)
packet_data, labels = zip(*combined_data)

# Split the data into training and testing sets
train_data = list(packet_data[:800])
train_labels = list(labels[:800])
```

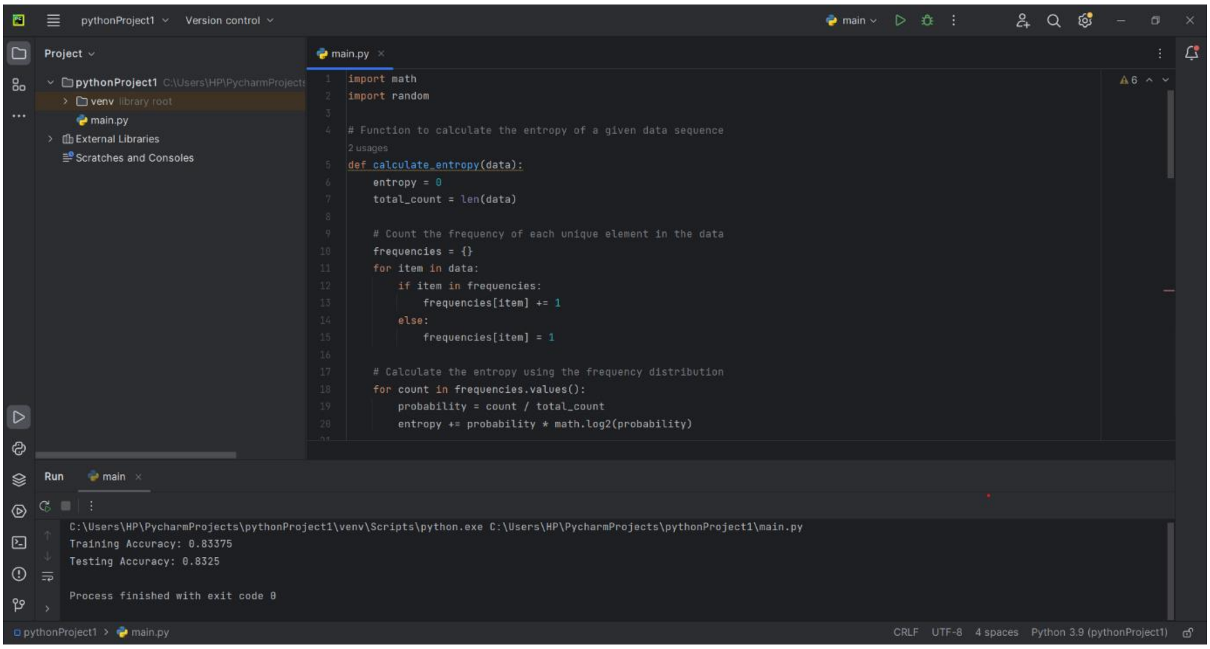
```
test_data = list(packet_data[800:])
test_labels = list(labels[800:])

# Train the DDoS detection algorithm
train_entropies = [calculate_entropy([data]) for data in train_data]
train_ddos_detected = [1 if entropy > ddos_threshold else 0 for entropy in train_entropies]

# Test the DDoS detection algorithm
test_entropies = [calculate_entropy([data]) for data in test_data]
test_ddos_detected = [1 if entropy > ddos_threshold else 0 for entropy in test_entropies]

# Calculate accuracy
train_accuracy = sum(train_ddos_detected[i] == train_labels[i] for i in range(len(train_labels))) / len(train_labels)
test_accuracy = sum(test_ddos_detected[i] == test_labels[i] for i in range(len(test_labels))) / len(test_labels)
print("Training Accuracy:", train_accuracy)
print("Testing Accuracy:", test_accuracy)
```

OUTPUT:



```
import math
import random

# Function to calculate the entropy of a given data sequence
def calculate_entropy(data):
    entropy = 0
    total_count = len(data)

    # Count the frequency of each unique element in the data
    frequencies = {}
    for item in data:
        if item in frequencies:
            frequencies[item] += 1
        else:
            frequencies[item] = 1

    # Calculate the entropy using the frequency distribution
    for count in frequencies.values():
        probability = count / total_count
        entropy += probability * math.log2(probability)
```

```
C:\Users\HP\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject1\main.py
Training Accuracy: 0.83375
Testing Accuracy: 0.8325

Process finished with exit code 0
```