**Implementation Plan**

Here, I select the Random Forest model for Real-Time Implementation

### **I. Model Building**

**Data Collection and Preprocessing:**

1. Input Dataset on IDE

2. Preprocess the data: handle missing values, encode categorical variables, normalise/standardise numerical features, and split the data into training and validation sets.

**Feature Engineering:**

1. Create new features that might enhance the model's performance.

2. Remove or transform features that are not useful.

**Model Training:**

1. Import Random Forest

2. Train the model using the training dataset.

3. Perform hyperparameter tuning using techniques like GridsearchCV or RandomsearchCV.

4. Evaluate the model using the validation dataset.

**Model Export:**

1. Save the trained model to a file for later use in the real-time system.

model.save\_model('Random\_Forest\_model.pkl')

### **II. Develop a Real-Time System**

**Set Up the Environment:**

1. Use a cloud service (AWS, GCP, Azure) or on-premise servers.

2. Set up necessary tools and libraries.

**Create an API:**

1. Use Flask, FastAPI, or another web framework to create an API for real-time predictions.

**Integrate with Network Traffic Monitoring Tools:**

1. Use Wireshark, Suricata, or Snort to capture network traffic data.

2. Preprocess the captured data to match the model's input format.

**Deploy the API:**

1. Containerize the application using Docker.

2. Deploy the container to a cloud service or on-premise server.

3. Use orchestration tools like Kubernetes for scaling and managing the deployment.