Project Title: **AI-Based Cyber Security Threats Prediction AI Agent**

1□ Offline Phase — Model Training

Step A: Data Collection - Collect dataset: UNSW-NB15. Includes packet/session features and timestamps for temporal sequences.

Step B: Preprocessing - Clean data: remove useless columns (IDs, duplicates, etc.). Encode categorical features (protocol, service, state). - Normalize numerical features.

For CNN-LSTM: create **time-windowed sequences** (e.g., 50 packets per session/IP).

2□ Machine Learning Path (Static Features)

Input: Single packet/session features
Model: XGBoost

Steps: 1. Train model on offline static features.
2. Evaluate metrics: Accuracy, Precision, Recall, F1, ROC-AUC.
3. Save trained model (`xgboost_model.pkl`).

Live Data Integration: - Capture live packets using `scapy` or logs.
- Extract the same static features.
- Feed to XGBoost → get prediction (normal / attack).
- Take action (alert/block/log).

3□ Deep Learning Path (Temporal / Sequence Features)

Input: Sequences of packets per session/IP (time windows)
Model: CNN-LSTM using PyTorch

Steps: 1. Train CNN-LSTM on **sequence data** (offline).
2. Evaluate metrics: Accuracy, F1, etc.
3. Save trained model (`cnn_lstm_model.h5`).

Live Data Integration: - Capture live traffic.
- Create sequences (same window size as training).
- Feed sequences to CNN-LSTM → get prediction.
- Take action (alert/block/log).

> □ **Important:** Sequence creation is mandatory. CNN-LSTM cannot use single static packets.

4□ Real-Time Agent Logic

- Both paths feed into an **agent layer**:
  - Attack probability > 0.9 → block + alert
  - 0.6–0.9 → monitor + log
  - < 0.6 → safe
- Logs can be stored or displayed in a **dashboard**.

5□ Deployment (Optional)

- Use **FastAPI / Flask**: expose `/predict` endpoint.
- Model(s) loaded in backend.
- Live traffic features/sequences sent to endpoint → returns prediction.
- Deploy on **Render** or any cloud service.

Key Takeaways

1. **XGBoost = fast, easy, static data, live-ready.**
2. **CNN-LSTM = handles temporal patterns, sequence-based, live data requires sequence creation.**
3. **Both models can feed the same agent logic for real-time alerts.**
4. **Deployment**: FastAPI + Render exposes models for real-time testing.

Visual Pipeline Diagram (Text Representation)

```
              +-------------------+
              |  Raw Network Data|
              +--------+----------+
                       |
                       v
              +-------------------+
              |  Feature Extraction|
              +--------+----------+
                       |
          +------------+-------------+
          |                          |
          v                          v
+-------------------+         +-------------------+
|  Static Features |          | Sequence Features|
+-------------------+         +-------------------+
          |                          |
          v                          v
+-------------------+         +-------------------+
|  XGBoost Model    |         |  CNN-LSTM Model  |
+-------------------+         +-------------------+
          |                          |
          v                          v
+-------------------+         +-------------------+
|  Prediction &     |         | Prediction &    |
|  Agent Logic      |         | Agent Logic     |
+-------------------+         +-------------------+
          |                          |
          v                          v
     Alerts / Actions           Alerts / Actions
          |                          |
          +------------+-------------+
                       v
               Monitoring Dashboard
```