Mixture-of-Experts (MoE) Architecture Analysis for Cybersecurity

🔍 Dataset Analysis: What Data Structures Do We Have?

☑ Expert 1: Tabular Expert - FULLY SUPPORTED

Current Implementation: ☑ FT-Transformer (already implemented and trained!)

Available Data:

- **CICIDS**: 72 numerical features (flow-level statistics)
- **UNSW**: 39 numerical + 3 categorical features

Features Include:

- Connection-level statistics (packets, bytes, duration)
- Protocol flags (SYN, ACK, FIN, etc.)
- Statistical aggregations (mean, std, min, max)
- Flow characteristics (rates, packet sizes, inter-arrival times)

Status: ☑ **READY** - FT-Transformer model already trained and achieving excellent results

---

⚠ Expert 2: Temporal Expert - PARTIALLY SUPPORTED

Goal: Capture temporal dependencies and attack patterns over time

Challenge: **No explicit timestamps in current datasets!**

What We Have (Flow-Level Temporal Features):

Both CICIDS and UNSW provide **aggregated temporal statistics** but **NOT raw packet sequences**:

CICIDS Temporal Features:

```
temporal_features = [
    "Flow Duration",          # Total duration of flow (microseconds)
    "Flow IAT Mean",          # Mean inter-arrival time between packets
    "Flow IAT Std",           # Std dev of inter-arrival times
    "Flow IAT Max",           # Maximum inter-arrival time
    "Flow IAT Min",           # Minimum inter-arrival time
    "Fwd IAT Total/Mean/Std/Max/Min",  # Forward direction timing
    "Bwd IAT Total/Mean/Std/Max/Min",  # Backward direction timing
    "Active Mean/Std/Max/Min",         # Active time before idle
    "Idle Mean/Std/Max/Min",           # Idle time statistics
]
```

UNSW Temporal Features:

```
temporal_features = [
    "dur",              # Flow duration
    "sjit", "djit",     # Source/dest jitter (timing variance)
    "sinpkt", "dinpkt",   # Inter-packet arrival times
    "tcprtt",           # TCP round-trip time
    "synack", "ackdat",   # TCP handshake timing
]
```

Workaround: Temporal Feature Expert (RECOMMENDED) ⭐

- Create a specialized MLP/Transformer that focuses on **temporal statistics**
- Input: IAT features, duration, jitter, timing-related columns
- Purpose: Learn temporal attack signatures (e.g., DDoS = low IAT variance, Port Scan = regular IAT patterns)
- **This is NOT true time-series but captures temporal characteristics**

❌ Expert 3: Graph Expert - NOT SUPPORTED

Goal: Capture communication topology (IP → IP, port relationships)

Challenge: **NO IP ADDRESS DATA in datasets!**

What's Missing: No Source IP, No Destination IP, No explicit port pairs

Workaround Option: Use UNSW's connection count features (`ct_*`) as pseudo-graph statistics

- `ct_srv_src`: Count of flows to same service from same source
- `ct_dst_ltm`: Count of flows to same dest in last time window
- **NOT a true GNN, but captures graph-like relational properties**

🎯 Recommended MoE Architecture: **2-Expert System**

```
Input Flow Data
    ↓
    ├→ Expert 1: Tabular Expert (FT-Transformer)
    │   - All features (flow statistics, flags, rates)
    │   - Output: Z_tab (embedding)
    │
    └→ Expert 2: Temporal Feature Expert (MLP/Transformer)
        - Focus: IAT, duration, jitter, timing features
        - Output: Z_temporal (embedding)

    ↓
Gating Network (MLP)
```

```
    - Input: [Z_tab, Z_temporal]
    - Weights: softmax([w_tab, w_temporal])
    - Output: Z_final = w_tab * Z_tab + w_temporal * Z_temporal

    ↓
Classifier Head
    - Input: Z_final
    - Output: [Normal, Attack]
```

Why This Works:

- ☑ **Honest**: Respects dataset limitations (no IPs, no packet sequences)
- ☑ **Novel**: Temporal expert learns timing-based attack signatures
- ☑ **Practical**: Both experts use available data effectively
- ☑ **Academic**: Demonstrates MoE concept without inventing fake data

---

📋 Implementation Plan

Components to Build:

1. **Temporal Expert** (new)
2. **Gating Network** (new)
3. **MoE Wrapper** (new)
4. **Feature Splitter** in preprocessing (new)
5. **train_moe.py** (new unified training script)
6. **MLflow integration** for gating weights

Next Steps:

Would you like me to:

1. ☑ **Implement 2-Expert MoE** (Tabular + Temporal) - RECOMMENDED
2. ⚠ **Attempt 3-Expert MoE** (add Relational expert with ct_* features)
3. 📊 **Just analyze and document** the architecture design

Ready to start coding when you give the go-ahead! 🚀