Model Research & Selection

The objective of this to identify, evaluate, implement a **continuous learning pipeline** and select the mathematical and machine learning models that will power the Interactive Cyber Threat Visualization Dashboard. For a real-time dashboard, the models must prioritize low-latency inference, high throughput, and the ability to handle unstructured streaming data.

Unlike static models, these algorithms process data in a single pass (O(1) or O(n) complexity per sample) and are robust to **Concept Drift**—the phenomenon where cyber attack patterns change over time.

To meet the project objectives, the research is divided into four functional modeling areas:

A. Real-Time Anomaly Detection (Unsupervised)

Purpose: To detect "Zero-Day" exploits and sudden traffic spikes in live streams without pre-labeled data.

- **Primary Model: Half-Space Trees (HS-Trees)**
  - **Logic:** A fast, one-pass anomaly detection algorithm designed for data streams. It builds a set of random trees where the depth of a leaf determines the anomaly score. Unlike Isolation Forest, which often requires a full batch to calculate scores, HS-Trees update their internal "mass" counters in real-time.
  - **Streaming Advantage:** It has a constant memory footprint and updates instantly as each network packet arrives.
- **Alternative: Streaming Local Outlier Factor (Streaming LOF)**
  - **Logic:** Uses a "sliding window" approach to compare the density of the current data point against its immediate temporal neighbors.
  - **Streaming Advantage:** Ideal for detecting "Point Anomalies" (e.g., a single login attempt from a forbidden IP) within a moving time-frame.

B. Attack Classification (Online Supervised)

Purpose: To categorize live threats into specific MITRE ATT&CK techniques (e.g., T1110 - Brute Force).

- **Primary Model: Hoeffding Tree (Very Fast Decision Tree - VFDT)**
  - **Logic:** This is the gold standard for streaming classification. It uses the **Hoeffding Bound** to decide how many samples are needed to split a node. It only looks at each sample once and grows the tree as more data flows in.
  - **Streaming Advantage:** It can process millions of records per second and is theoretically guaranteed to converge to the same result as a batch decision tree.
- **Ensemble Model: Adaptive Random Forest (ARF)**
  - **Logic:** An evolved version of Random Forest that includes **Drift Detectors** (like ADWIN) for each tree. If the model detects that an attack pattern has changed (e.g., a new variant of a DDoS), it replaces the underperforming trees with new ones.
  - **Streaming Advantage:** Handles high-dimensional logs better than a single tree and automatically adapts to "adversarial drift."

C. Temporal Trend Forecasting (Incremental Time-Series)

Purpose: To predict the next hour's threat volume based on live incoming velocity.

- **Primary Model: Online SNARIMAX**
  - **Logic:** An incremental version of ARIMA that incorporates Seasonal (S), Network (N), and Exogenous (X) variables. It updates its coefficients using **Stochastic Gradient Descent (SGD)** rather than solving complex matrix equations.
  - **Streaming Advantage:** Unlike Meta's Prophet, which needs a full history to refit, SNARIMAX adapts to the most recent "velocity" of attacks instantly.
- **Alternative: Passive-Aggressive Regressor**
  - **Logic:** A "lazy" learner that stays passive if the prediction is correct but becomes aggressive (updates weights) if the error exceeds a threshold.

- **Streaming Advantage:** Extremely low latency, making it perfect for real-time dashboard refresh rates.

D. Geospatial Risk Scoring (Dynamic Heuristic)

Purpose: To provide a live-updating "Heat Score" for the global map.

- **Algorithm: Exponentially Weighted Moving Average (EWMA) Risk Score**
- **Formula:**

$$Risk_t = \alpha\text{-}(Severity \times Frequency)_{current} + (1-\alpha) \cdot Risk_{t-1}$$

- α **(The Memory Factor):** A higher α (e.g., 0.8) makes the map hyper-reactive to instant changes. A lower α (e.g., 0.2) makes the map smoother but slower to clear.
- **The "Wash-Out" Effect:** This mathematical approach ensures that if attacks stop, the term $(1-\alpha) \cdot Risk_{t-1}$ will eventually shrink the score to zero, effectively "cleaning" the map of old alerts automatically.
- **Deep Explanation:** By using an α (decay factor), the dashboard prioritizes the *most recent* 5 minutes of data while slowly phasing out older incidents. This prevents the map from staying "red" for hours after an attack has been mitigated.

To support these models, you must implement **Online Feature Extraction**:

1) **Hashing Trick:** Converts categorical features (like User-Agents) into fixed-size vectors without needing a pre-defined vocabulary.
2) **Rolling Statistics:** Calculates running mean, variance, and entropy of packet sizes using **Welford's Algorithm** (updates mean/variance in a single step).