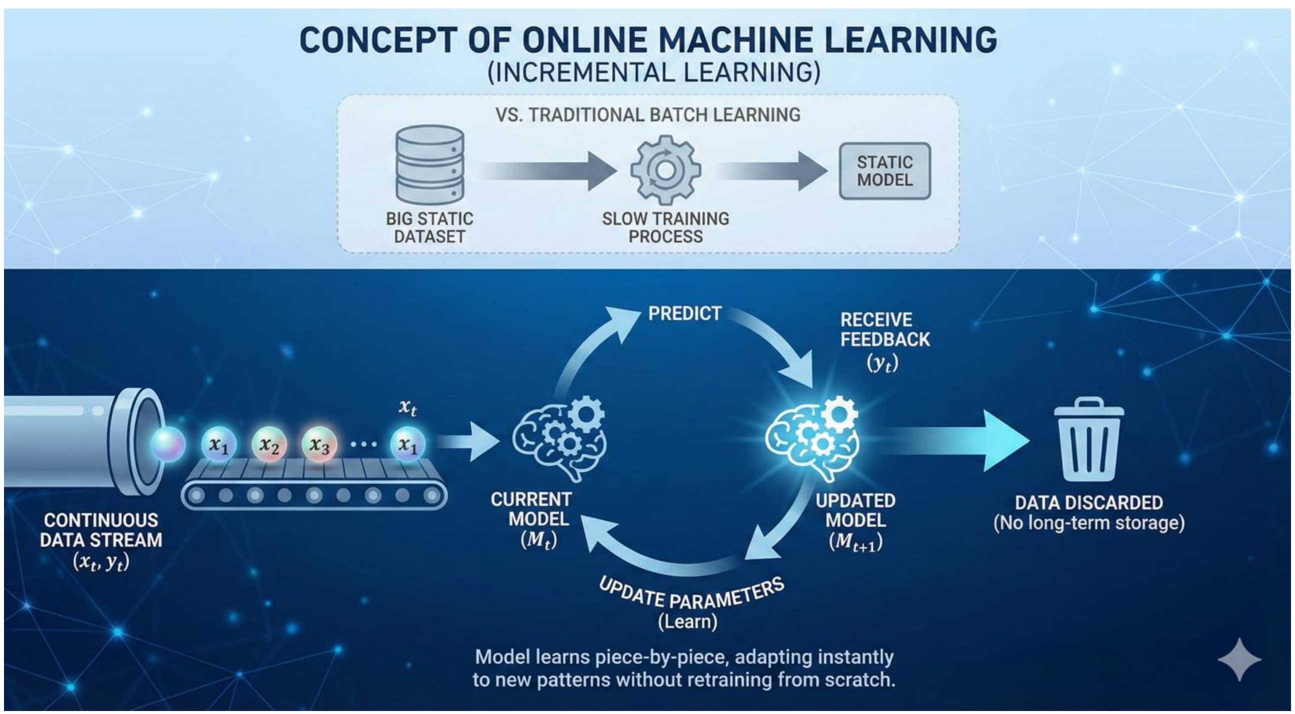
# Introduction to Online Machine Learning

Prof.Venki Muthukumar

# Batch vs. Online

- **Traditional ML (Batch Learning):**
  - **Static:** Model is trained once on a fixed dataset (e.g., CSV file).
  - **Offline:** Training can take hours/days; model is deployed as a read-only artifact.
  - **Equation:** Model = Train($X_{all}$, $y_{all}$})
  - **Limitation:** If data changes, you must retrain from scratch.
- **Online ML (Incremental Learning):**
  - **Dynamic:** Model updates continuously as new data arrives.
  - **Real-Time:** Training happens in milliseconds per sample.
  - **Equation:** $\text{Model}_t$ = Update($\text{Model}_{t-1}$, $x_{new}$, $y_{new}$)
  - **Advantage:** Adapts immediately to new trends; requires less memory (doesn't store history).



# Why Online ML?



- **Big Data Volume:** Dataset is too large to fit in RAM (e.g., years of server logs).
- **Dynamic Environments:** User/Sensor behavior changes.
- **Edge Computing (IoT):** Devices like Raspberry Pi have limited memory; cannot store huge datasets for batch training.
- **Cold Start:** Start learning immediately without waiting to accumulate a "training set."

# Key Terminology



- **Instance/Observation:** A single data point (x, y) arriving at time t.
- **Concept Drift:** When the statistical properties of the target variable change over time (e.g., "Normal" temperature changes from Winter to Summer).
- **Forgetting Factor / Window:** Mechanism to discard old, irrelevant data (fading memory).
- **Interleaved Test-Then-Train:** The standard evaluation method—predict first, then use the true label to update the model, then measure error.
- **Regret:** The difference in performance between the online model and the best possible static model trained on all data in hindsight.

# Methodologies & Evaluation

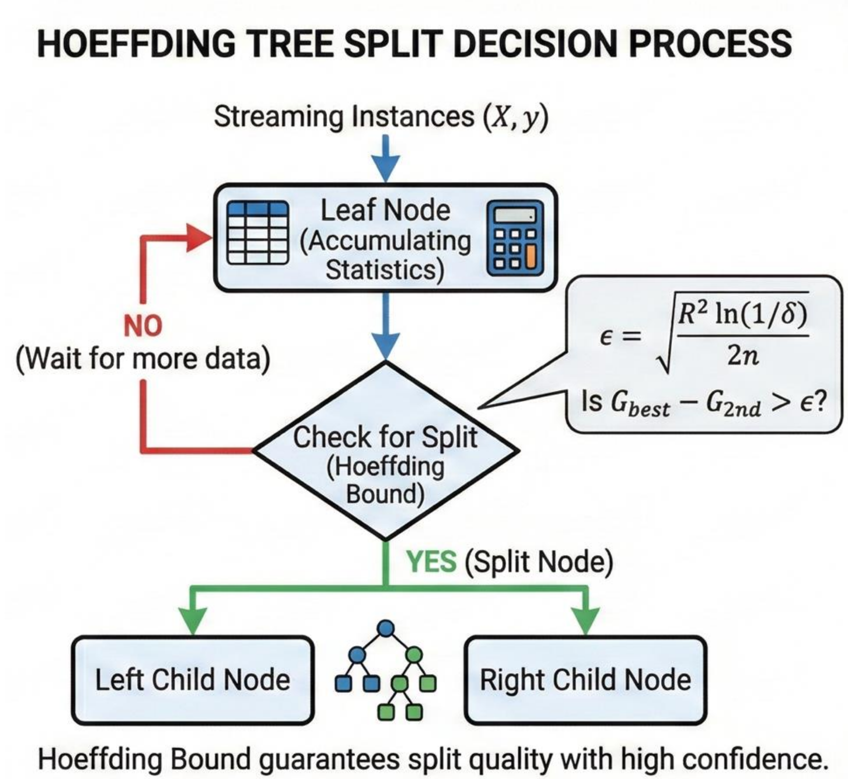
Embedded ML

- **Prequential Evaluation (Predictive Sequential):**
  - Instead of "Train/Test Split", every sample serves as a test sample *before* it becomes a training sample.
  - Accuracy is a running average over time.
- **Windowing Strategies:**
  - **Landmark Window:** All data since the beginning (rarely used, too heavy).
  - **Sliding Window:** Only the last ‘N’ samples matter (FIFO).
  - **Damped Window:** Recent samples have higher weight; older samples decay exponentially.

# Algorithms – Linear & Tree-Based



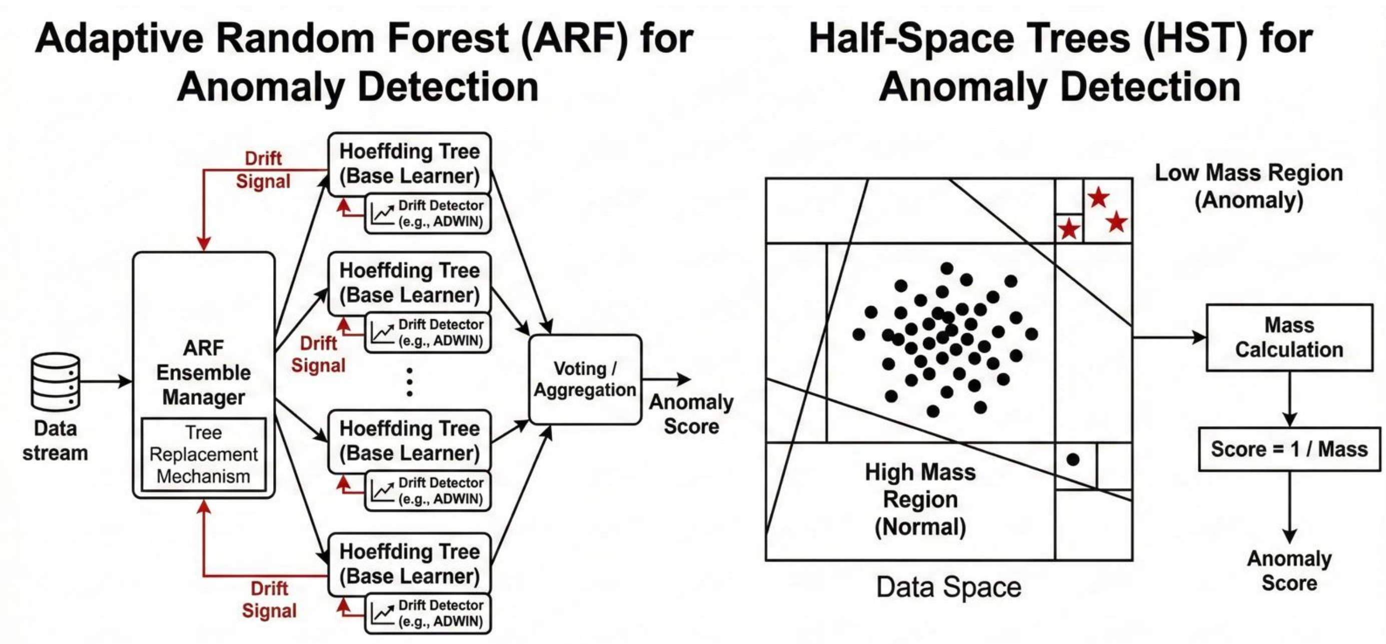
- **Stochastic Gradient Descent (SGD):** The backbone of online learning. Updates weights slightly for every error.
  - *Used for:* Regression, SVM, Basic Neural Networks.
- **Hoeffding Trees (Very Fast Decision Trees - VFDT):**
  - Wait for enough statistical evidence (using Hoeffding Bound) to split a node.
  - Hoeffding Bound calculates the number of samples (n) needed to guarantee that the observed best split is truly the best.
  - *Result:* A decision tree that is nearly identical to a batch tree but built incrementally.
  - *Used for:* Classification in changing streams.



# Ensemble & Anomaly Detection



- **Adaptive Random Forest (ARF):**
  - A forest of Hoeffding Trees.
  - Includes drift detection per tree; replaces trees that perform poorly due to drift.
- **Half-Space Trees (HST):**
  - Used for **Anomaly Detection**.
  - Randomly partitions space; anomalies fall into sparse regions (low mass).
  - Fast, efficient, no labels required (Unsupervised).



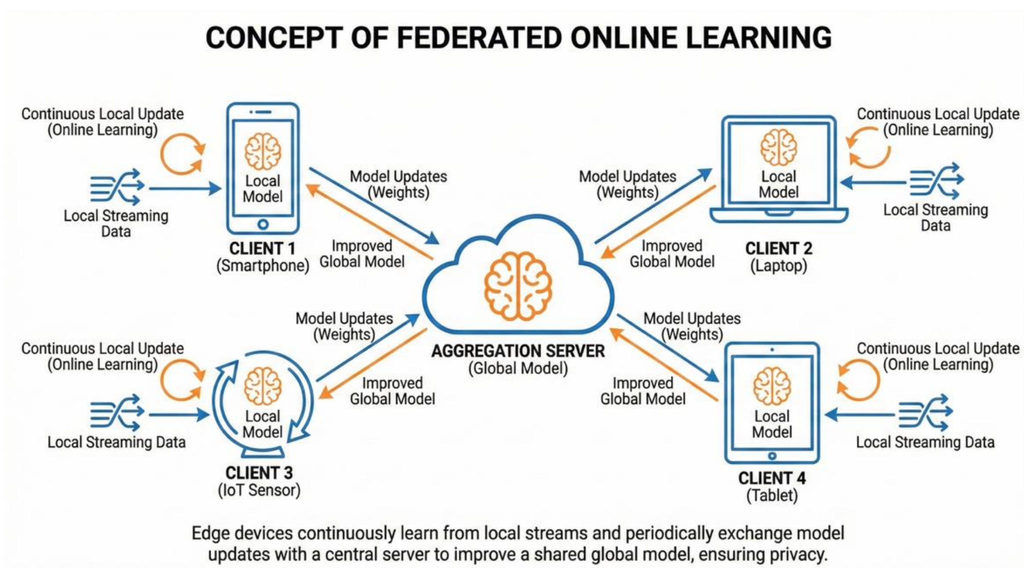

Ensemble of online trees that adapts to concept drift by replacing individual learners.

Randomly partitions space; anomalies fall into sparse regions with low data mass.

# Federated Online Learning

Embedded ML

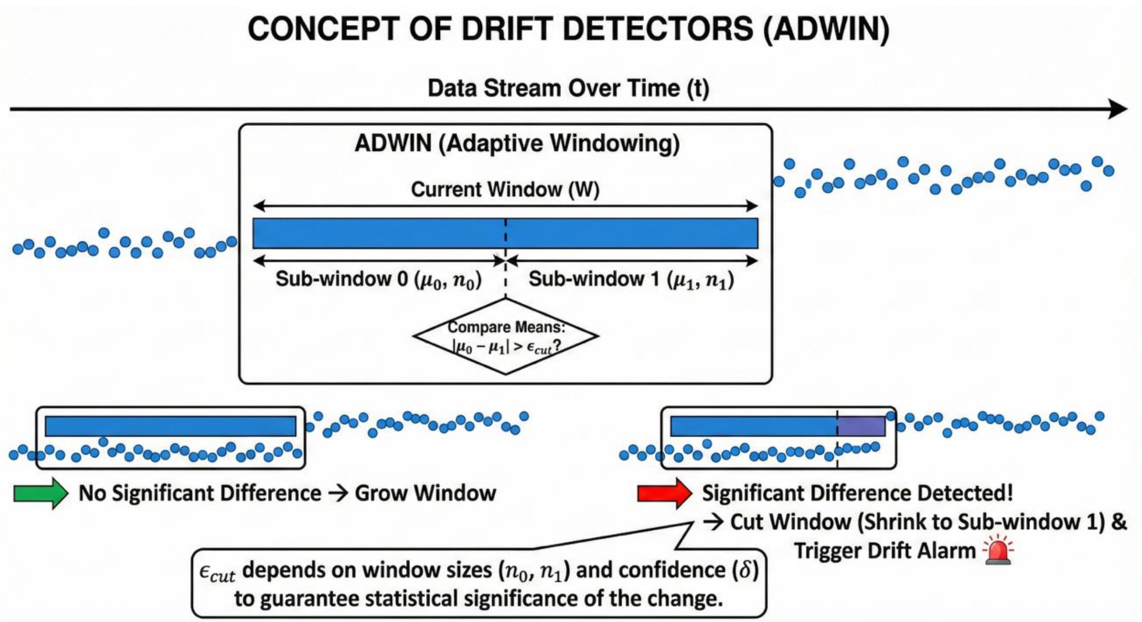
- Combines two advanced concepts: **Federated Learning + Online Learning**.
- Example
  - Your Raspberry Pi receives a stream of sensor data.
  - It updates its local model immediately (Online Learning) to adapt to current conditions.
  - Periodically, it sends just the *updates* (weight changes) to a central server.
  - The server aggregates these updates from 1,000 devices and sends back a better global model.
  - The Pi continues learning online, now starting from this smarter global baseline.
- Highlights
  - Devices (Edge) learn online locally.
  - Only model updates (weights) are sent to the cloud, not data.
  - Preserves privacy while learning from millions of streams.



# Concept Drift



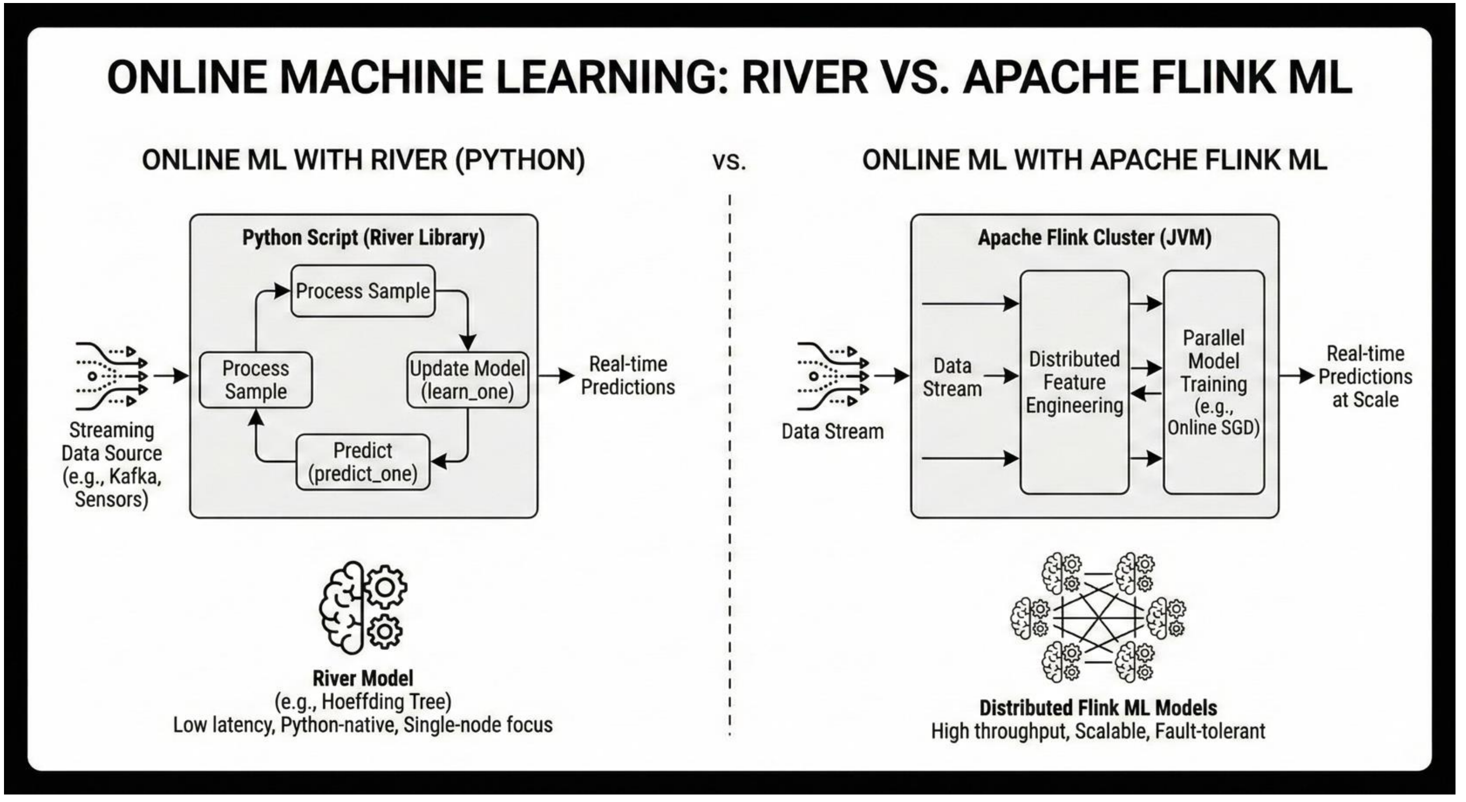
- **Definition:** The relationship between input data and output label changes.
- **Types:**
  - **Sudden Drift:** Immediate change (e.g., accidents changed traffic patterns).
  - **Gradual Drift:** Slow evolution (e.g., mechanical wear on a sensor).
  - **Recurring Drift:** Seasonal patterns (e.g., seasonal temperature).
- **Impact:** A static model's accuracy plummets.
- **Solution:** Drift Detectors (ADWIN, DDM) trigger model resets or adaptive window sizing.



# Tools & Ecosystem

Embedded ML

- **River (Python):** (Previously Creme/Scikit-Multiflow). The standard for Python. Scikit-learn like API (learn_one, predict_one).
- **Apache Flink ML:** For distributed streaming ML at enterprise scale.



# Handson