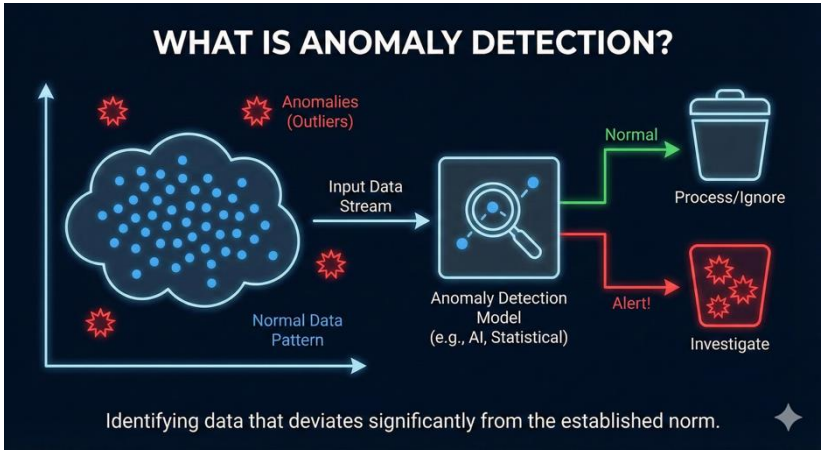
Deep Learning for Anomaly Detection

What is Anomaly Detection?

- **Definition:** Identifying data points, events, or observations that deviate significantly from the dataset's normal behavior.
- **The "Needle in a Haystack" problem:** Anomalies are rare (often <1% of data).
- **Types of Anomalies:**
  - **Point Anomalies:** A single data point is far from the rest (e.g., credit card fraud).
  - **Contextual Anomalies:** Abnormal only in a specific context (e.g., 90°F is normal in summer, anomaly in winter).
  - **Collective Anomalies:** A sequence of data points is abnormal (e.g., irregular heart rhythm).

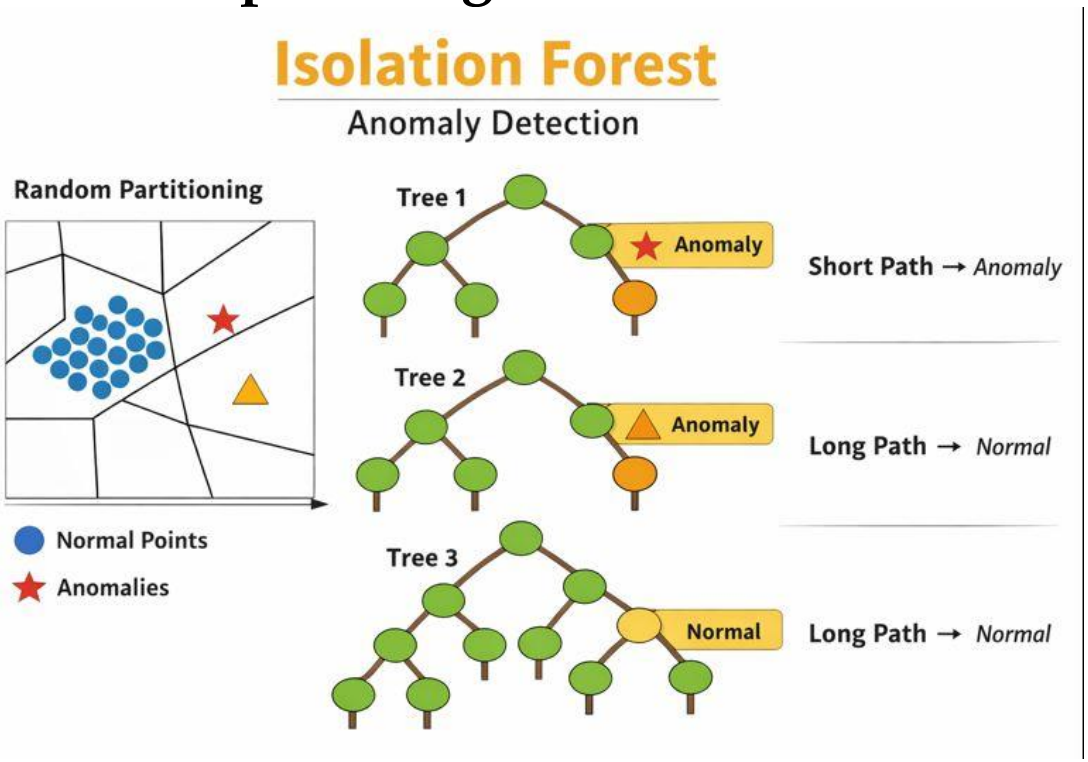


Why Deep Learning? (vs. Traditional ML)

- **Traditional Methods:** (Isolation Forest, One-Class SVM) struggle with high-dimensional data (images, video) and complex non-linear relationships.
- **Deep Learning Advantages:**
  - **Feature Learning:** Automatically learns hierarchical representations (no manual feature engineering).
  - **Scale:** Handles massive datasets effectively.
  - **Unstructured Data:** Excellence in processing Images, Audio, and Time Series.

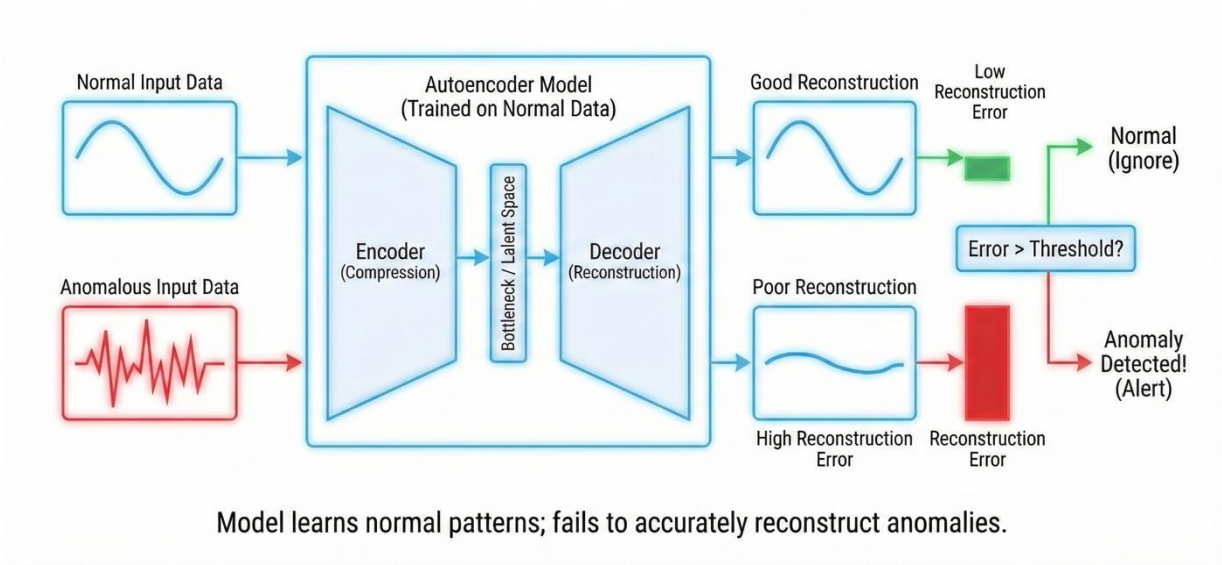
Isolation Forest

- Isolation Forest is an **unsupervised anomaly detection algorithm** that identifies anomalies by **isolating data points** instead of modeling normal behavior.
  - Measure **path length** (number of splits to isolate a point)
  - Points with **short path lengths = anomalies**



Reconstruction-Based Detection

- **The Hypothesis:** A model trained *only* on normal data will learn to reconstruct normal data well.
- **The Mechanism:** When the model encounters an anomaly, it fails to reconstruct it accurately.
- **The Metric: Reconstruction Error** (e.g., Mean Squared Error between Input and Output).
  - Low Error = Normal.
  - High Error = Anomaly.

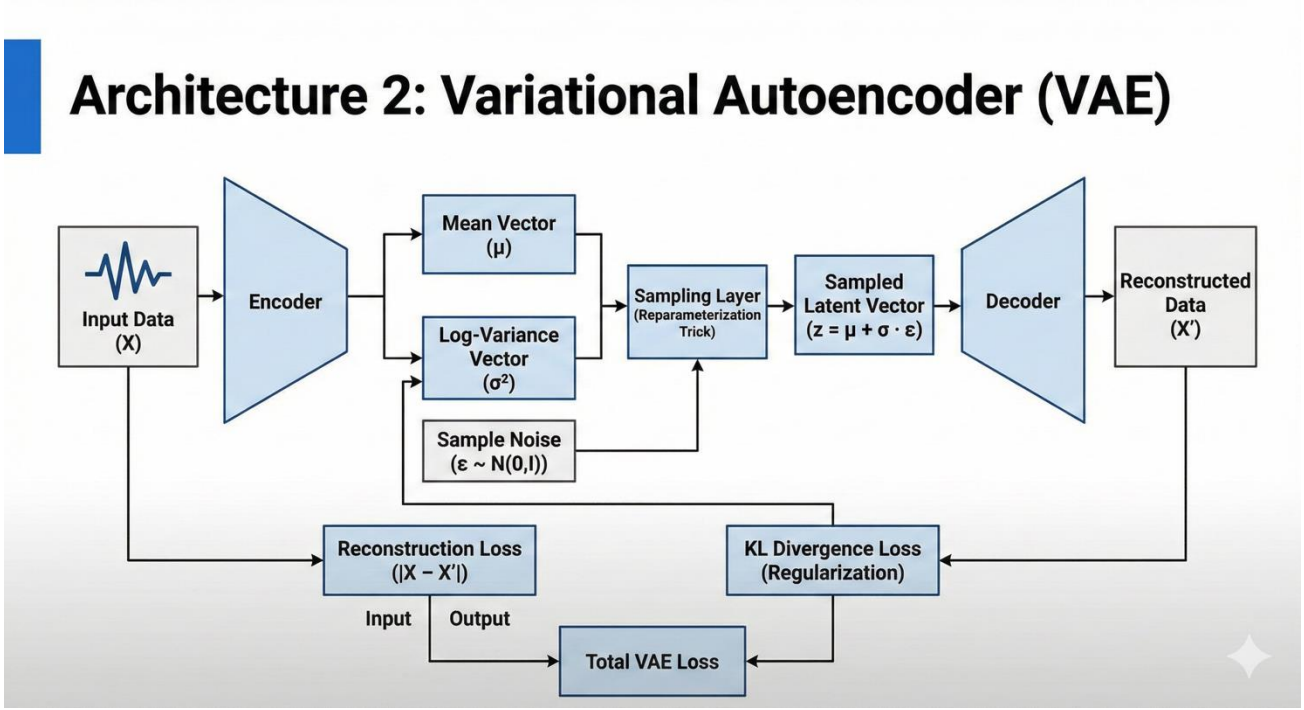


Architecture 1 – The Autoencoder (AE)

- **Structure:** Encoder (compresses input to latent space) -> Bottleneck -> Decoder (reconstructs input).
  - **Encoder** - take the complex, high-dimensional input data (like an image with thousands of pixels or a vibration signal with thousands of data points) and compress it. Find patterns.
  - **Bottleneck** - single layer in the middle of the network – passes most essential features.
  - **Decoder** - takes the compressed code from the Bottleneck and attempts to reconstruct the original input data as accurately as possible.
- **Training:** Trained exclusively on "Normal" samples to minimize reconstruction loss.
- **Usage:** During inference, if the reconstruction loss is above a threshold, flag as anomaly.

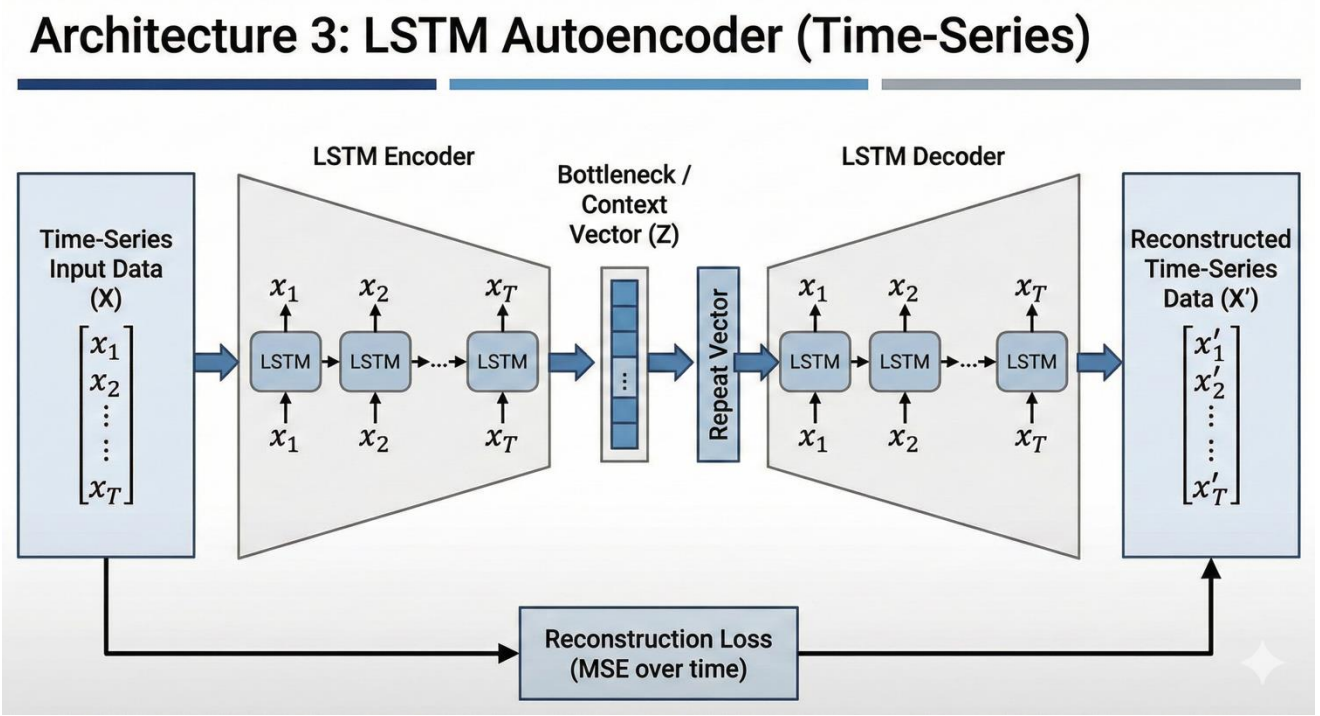
Architecture 2 – Variational Autoencoders

- **Limitation of Standard AE:** Can overfit and simply "memorize" inputs.
- **VAE Solution:** Encodes inputs into a probability distribution (mean and variance) rather than a fixed point.
- **Benefit:** Provides a smoother latent space and probabilistic anomaly scores (Evidence Lower Bound - ELBO).
- **Use Case:** Complex image data or manufacturing defect detection.



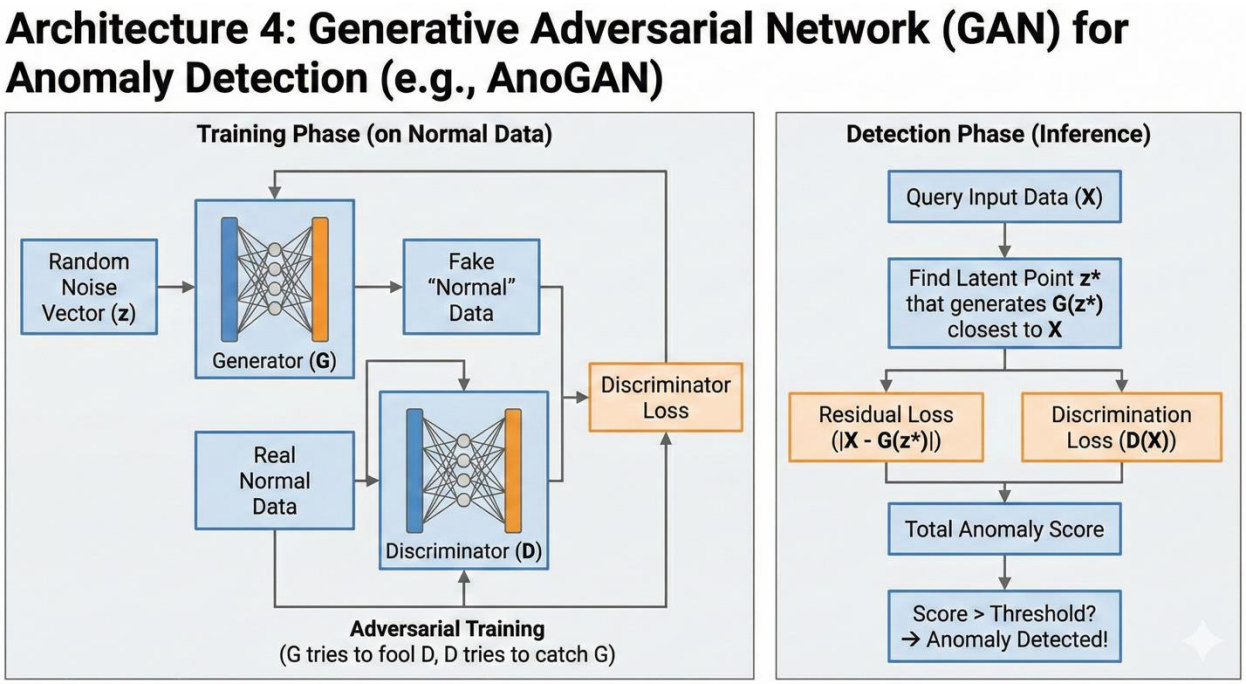
Architecture 3 – LSTM Autoencoders (TS)

- **Challenge:** Detecting anomalies in sequential data (logs, sensor readings, stock prices).
- **Solution:** Long Short-Term Memory (LSTM) networks capture temporal dependencies.
- **Structure:** An LSTM Encoder compresses the time sequence; an LSTM Decoder reconstructs the sequence.



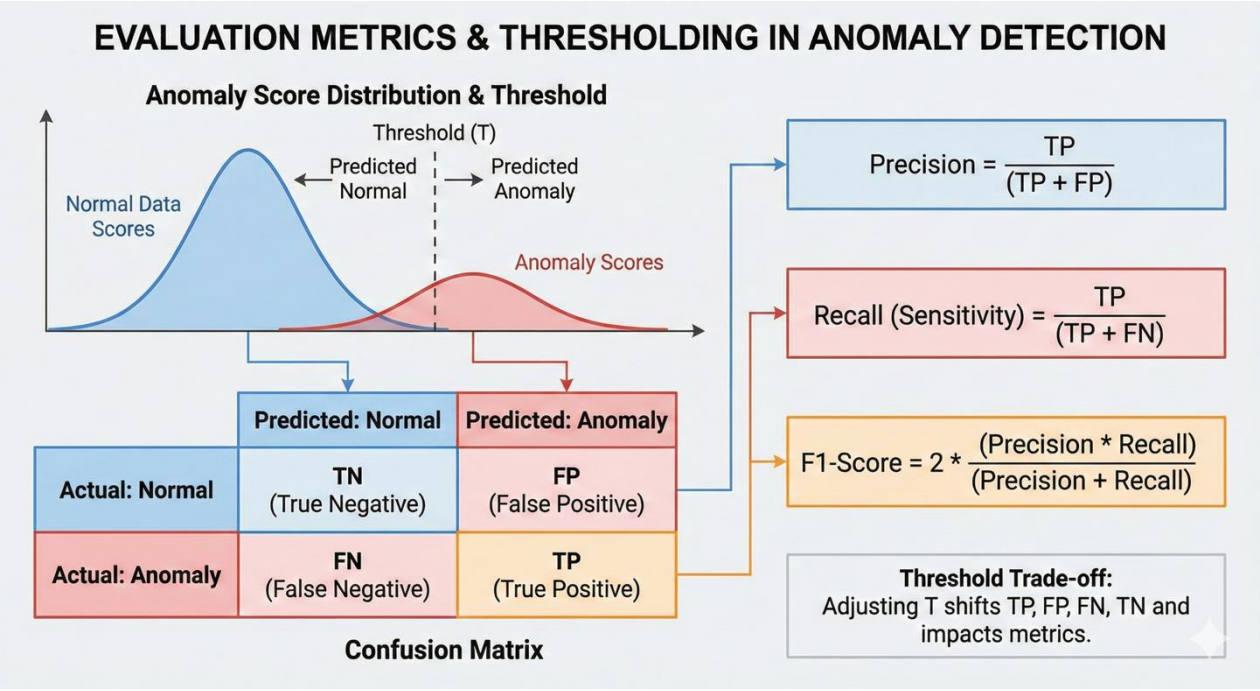
Architecture 4 – Generative Adversarial Networks

- **Concept (AnoGAN):** Train a Generator (G) to create "normal" data and a Discriminator (D) to distinguish real from fake.
- **Detection Logic:**
  - In inference, we try to find the latent point that generates an image closest to the query image.
  - If the query image cannot be generated by the model (high residual loss) or the Discriminator rejects it (discrimination loss), it is an anomaly.
- **Pros/Cons:** High performance on images, but difficult to train (unstable).



Evaluation Metrics & Thresholding

- **The Challenge:** You usually don't have labeled anomalies to test with easily.
- **Defining the Threshold:**
  - **Fixed:** Manually set (e.g., "Error > 0.05").
  - **Statistical:** Mean + 3 * Standard Deviation of normal errors.
- **Metrics (if labels exist):**
  - **Accuracy is misleading** (99% accuracy is useless if 1% are anomalies).
  - Use **Precision, Recall, F1-Score**.
  - **AUROC** (Area Under Receiver Operating Characteristic Curve).



Summary & Best Practices

- **Data Preparation:** Normalize/Scale data to [0,1] range (crucial for Neural Networks).
- **Model Selection:**
  - Tabular Data -> Autoencoder / VAE.
  - Time Series -> LSTM / Transformer.
  - Images -> Convolutional AE / GANs.
- **Pitfalls:** "Contaminated" training data (if training data has anomalies, the model learns them as normal).
- **Conclusion:** Deep Learning offers robust, automated feature extraction for finding anomalies in complex, high-volume data.

HandsOn

- Raspberry Pi (Sender) to collect and stream data, and the Host PC (Receiver) to process that stream and detect anomalies.
- Use BMI160 as a vibration sensor from the M5Core or RPI to transmit the data at high-speed to the PC to run your anomaly detection.
- Initially keep the IMU sensor idle (under no disturbance), cause vibrations by slightly moving the sensor in z axis.
- Evaluate the performance of the anomaly detector.