Project Logic Behind Anomaly Detection in Network Traffic

Objective:
Detect **unusual or malicious network activity** without prior labels using **unsupervised learning techniques**.

---

1. Dataset Understanding & Preprocessing

- Used **KDD Cup 1999 dataset**: contains labelled instances of normal and attack traffic.
- The dataset is CSV without header row (i.e 0,1,2,3… as column names). Therefore, it doesn't have defined column names. Hence, we **assigning proper column names**.
- Dropped redundant/irrelevant features to reduce noise.
- Converted categorical data to numeric (using LabelEncoder or OneHotEncoder).
- Applied **MinMax Scaling** to normalize features for better model performance.

---

2. Unsupervised Model Logic

- In real-world security, new types of attacks emerge constantly.
- Hence, we don't always have labeled data — so models must learn to detect unusual patterns.

---

3. Isolation Forest

- **Logic:** Anomalies are rare and different — hence, easier to isolate.
- Model builds decision trees that randomly split features.
- Points that are **isolated quickly** (i.e., fewer splits) are likely anomalies.
- **Output:** Binary predictions **(0 = normal, 1 = anomaly).**

---

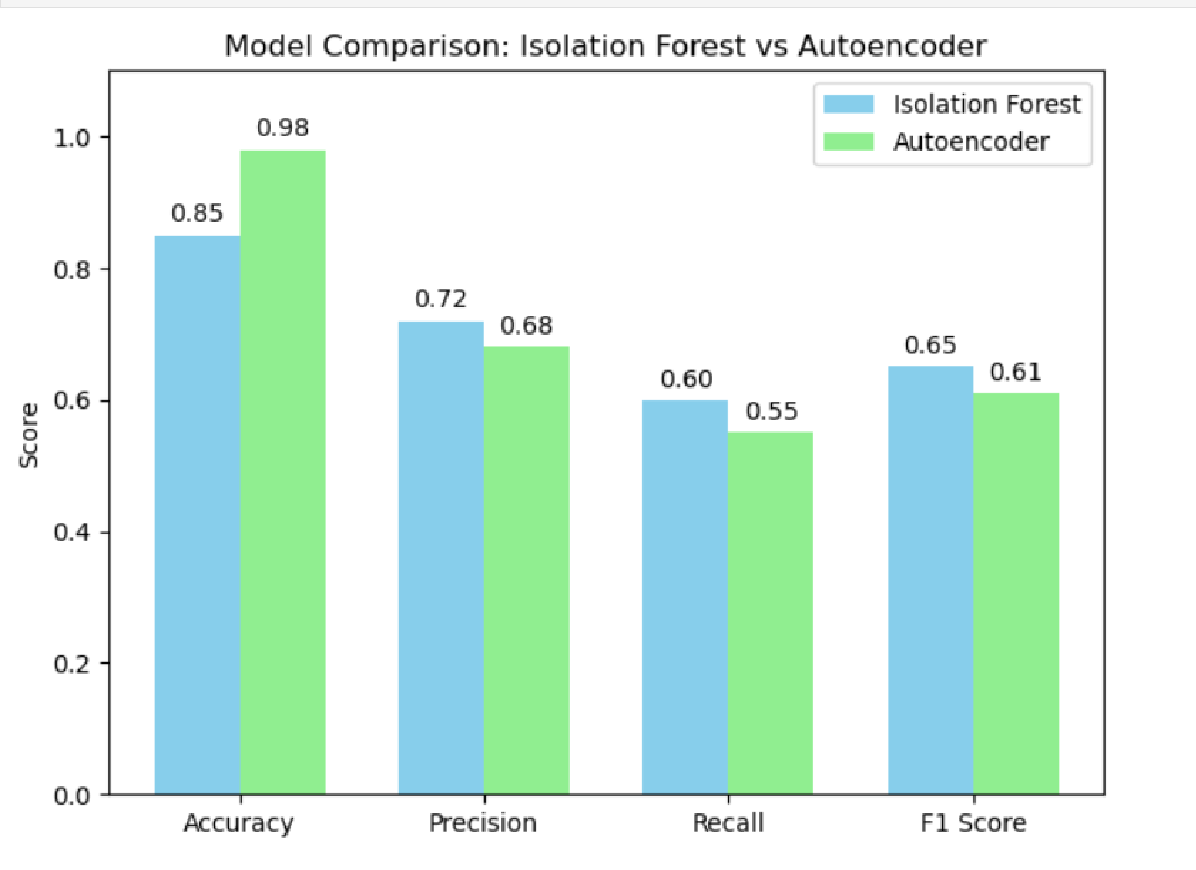
4. Autoencoder

- **Logic:** Neural network trained to reconstruct input data.
- Learns the normal pattern of network traffic.
- **If reconstruction error is high, the input is likely an anomaly (i.e., unusual pattern).**
- **Output:** Mean squared error between input and reconstructed data → threshold used to flag anomalies.

---

5. Evaluation & Interpretation

- Used original labels (only for evaluation) to compare model predictions.
- Accuracy, confusion matrix, and histograms used to assess results.
- Autoencoder showed higher accuracy (98%) compared to Isolation Forest (85%).



Conclusion:

- ✓ This project used KDD Cup 1999 dataset to detect network anomalies. Specifically, *kddcup.data_10_percent_corrected* file was used.
- ✓ We implemented two unsupervised models—Isolation Forest and Autoencoder.
- ✓ Both models effectively distinguished between normal and malicious traffic without labeled training data.
- ✓ Autoencoder performed better in minimizing reconstruction error and identifying subtle attacks with an accuracy of **98%**.
- ✓ Proper preprocessing, feature scaling, and model evaluation were key in improving accuracy.
- ✓ The project also visualized key insights and anomalies for clearer interpretation.