Project Summary

CyberSentinel: Anomaly Detection in Network Traffic Using Isolation Forest & Autoencoder

Objective

To design and deploy a system that detects abnormal patterns in network traffic which may indicate potential cyberattacks, using unsupervised machine learning models trained on the KDD Cup 1999 dataset.

Problem Statement

Traditional rule-based intrusion detection systems struggle to detect zero-day attacks or subtle behavior anomalies. CyberSentinel aims to overcome this by learning the underlying patterns of normal traffic and flagging deviations without needing labeled data.

Technologies & Tools

- **Language:** Python 3.10+
- **Libraries:** Scikit-learn, TensorFlow, Pandas, Seaborn, Streamlit
- **Models:** Isolation Forest, Autoencoder
- **Platform:** Streamlit (interactive web UI)
- **Dataset:** KDD Cup 1999 (10%) from Kaggle

Learning Outcome

- Practical experience with anomaly detection in cybersecurity
- Working with unsupervised models and evaluation metrics
- Building a real-time dashboard using Streamlit
- Data preprocessing, feature selection, and model deployment

Key Features

- Upload CSV and detect anomalies instantly
- Choose model (IF / AE) dynamically
- Smart anomaly explanations
- Download labeled predictions

Results, Demo & Future Scope

Model Performance

Metric	Isolation Forest	Autoencoder
Precision	97.33%	98.77%
Recall	98.86%	98.45%
F1-Score	98.09%	98.61%
ROC-AUC	93.90%	96.72%

- Autoencoder slightly outperformed Isolation Forest in overall recall and F1-score.

Streamlit Dashboard Highlights

- Upload `.csv` with required 10 features
- Detect anomalies instantly with Isolation Forest or Autoencoder
- Filter, sort, and download anomalies
- Smart reasoning engine for flagged records
- *Try it here : https://cybersentinel-ecymuxdunrynbdpdfexvu4.streamlit.app*

Real-World Applications

- Detect DoS, port scanning, spoofing, unauthorized access
- Extendable to modern network datasets and APIs

Future Scope

- Real-time data streaming support
- Model ensemble voting logic
- REST API backend (Flask/FastAPI)
- UI improvements with session tracking
- Timeline-based anomaly trends

Contact Info

Nitesh Yadav
Contact: niteshyadav0604@gmail.com
