Network Intrusion Detection using Machine Learning

Gotam Sai Varshith - 22BCE1605
PAVAN KRISHNA R - 24BCE5294
MRINAL SWAIN - 24BCE5357

Presentation Agenda

1. Introduction: The Problem and Our Motivation
2. Background: What is a Network Intrusion Detection System (NIDS)?
3. The Challenge: Why Use Machine Learning?
4. Methodology: Our End-to-End Project Workflow
5. The Dataset: A Look at NSL-KDD
6. The Models: KNN, LDA, and SVM
7. Implementation: The Training Pipeline and Streamlit App
8. Live Demonstration: A Walkthrough of the Dashboard
9. Results: Comparative Analysis of Model Performance
10. Conclusion & Future Work

The Modern Cybersecurity Challenge

- **Hyper-Connected World:** Our modern infrastructure (finance, healthcare, energy) is built on complex, interconnected networks.
- **Evolving Threat Landscape:** Cyber threats are no longer static. We face sophisticated, automated, and "zero-day" attacks that have never been seen before.
- **The Need for Intelligent Defense:** We need security systems that are not just reactive, but proactive, adaptive, and intelligent.
- **Project Goal:** To build and evaluate an intelligent, anomaly-based Network Intrusion Detection System (NIDS) using classical machine learning algorithms.

The Digital Security Guard

- Core Function: A NIDS is a system that continuously monitors network traffic for suspicious activity, policy violations, or outright malicious threats.
- **Analogy:**
  - A Firewall is like a locked door. It blocks known bad traffic.
  - A NIDS is like a security guard who watches for suspicious behavior—someone picking the lock, looking in windows, or acting abnormally.
- It acts as a critical second line of defense, catching threats that may have bypassed the firewall.

Two Main Approaches to Detection

- **Signature-Based Detection:**
  - **How it Works:** Matches network traffic against a database of known attack patterns (signatures).
  - **Pros:** Very fast and accurate for known threats. Low false positive rate.
  - **Cons:** Completely blind to new, "zero-day" attacks. Requires constant updates to its signature database.
- **Anomaly-Based Detection:**
  - **How it Works:** First, it learns a baseline of what "normal" network behavior looks like. Then, it flags any significant deviation from that baseline as a potential attack.
  - **Pros:** Can detect novel and zero-day attacks. More adaptive to new threats.
  - **Cons:** Can have a higher false positive rate if the "normal" model isn't perfect.
- **Our Focus:** This project focuses on building a powerful Anomaly-Based NIDS using Machine Learning.

The Power of Data-Driven Security

- **The Problem with Manual Systems:**
  - Modern networks are massive and generate terabytes of data. Manual analysis is impossible.
  - Traditional, rule-based systems can't keep up with the speed and creativity of new attacks.
- **How Machine Learning Helps:**
  - Scalability: ML algorithms can analyze massive volumes of network data efficiently.
  - Pattern Recognition: They can autonomously learn the complex patterns of both normal and malicious behavior without being explicitly programmed.
  - Adaptability: ML models can identify new and evolving threats, making them ideal for anomaly detection.
  - Reduced Manual Effort: Reduces the need for security experts to constantly write and update thousands of detection rules.

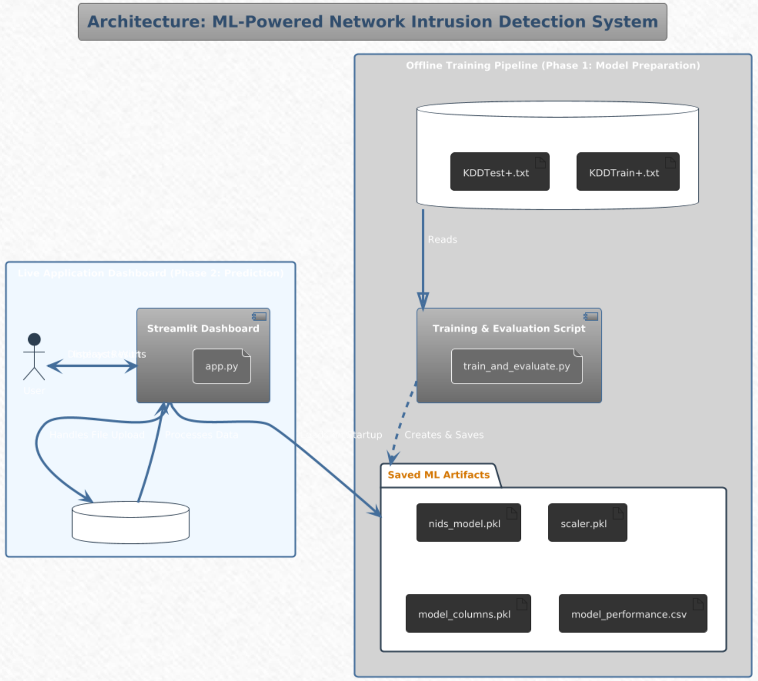
A Structured, End-to-End Approach

- **Phase 1: Data Acquisition & Preprocessing**
  - Procured the benchmark NSL-KDD dataset.
  - Performed rigorous data cleaning, encoding of categorical features, and normalization of numerical features.
- **Phase 2: Model Training & Evaluation**
  - Trained three different ML models: KNN, LDA, and SVM.
  - Evaluated each model on an unseen test set using standard metrics (Accuracy, Precision, etc.).
- **Phase 3: Asset Serialization**
  - Identified the best-performing model based on evaluation results.
  - Saved the trained model, data scaler, and feature list as .pkl files.
- **Phase 4: Application Development**
  - Built an interactive web dashboard using Streamlit.
  - Integrated the saved ML assets into the app for real-time predictions.

The Benchmark for NIDS Research

- **Source:** An improved and refined version of the classic KDD Cup 1999 dataset.
- **Key Improvement:** Removes redundant records, providing a more realistic and unbiased environment for model evaluation.
- **Structure:**
  - Each record represents a network connection.
  - Each connection is described by 41 features (e.g., duration, protocol_type, src_bytes).
- **Classes:** Labeled as either 'Normal' or one of four main attack categories:
  - DoS (Denial of Service)
  - Probe (Probing)
  - R2L (Remote to Local)
  - U2R (User to Root)
- For our project, we simplified this to a binary classification problem: 'Normal' vs. 'Attack'.

System Architecture



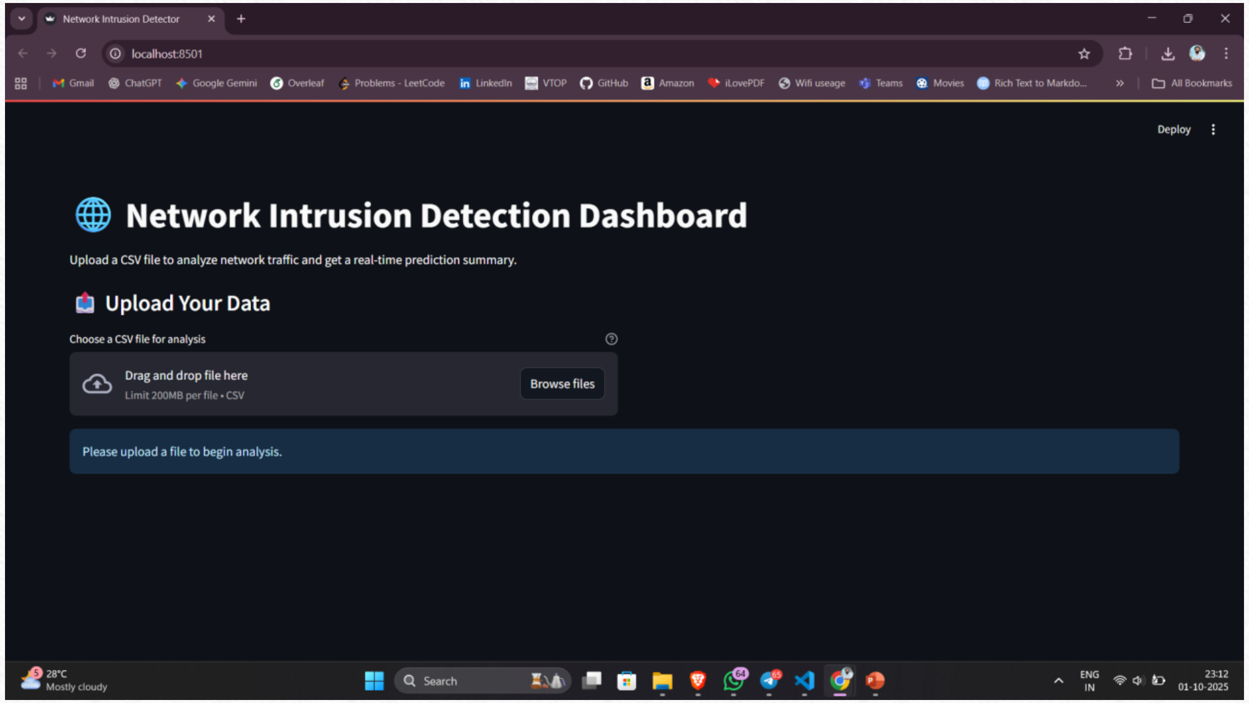
Our Chosen Algorithms

- **K-Nearest Neighbors (KNN):**
  - An instance-based “lazy learner.“
  - Classifies a connection based on the majority class of its 'k' closest neighbors in the feature space.
  - Simple, yet surprisingly effective for pattern recognition.
- **Linear Discriminant Analysis (LDA):**
  - A statistical method that finds a linear combination of features to best separate the classes.
  - Computationally fast and efficient, but assumes a linear relationship in the data.
- **Support Vector Machine (SVM):**
  - A powerful classifier that finds the optimal "hyperplane" or decision boundary that best separates normal and attack data points.
  - Known for its robustness in high-dimensional spaces. We used a linear kernel for efficiency.

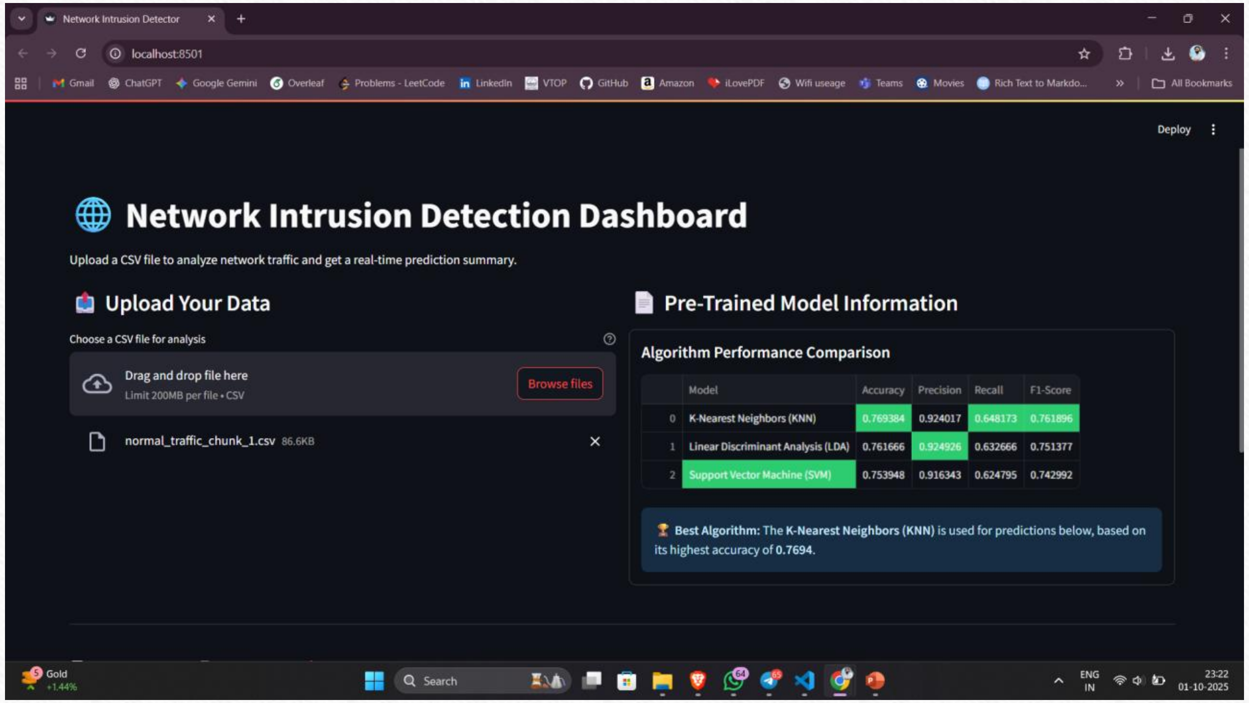
Bringing the Model to Life

- **Technology:** We used Streamlit, a Python framework for building beautiful, interactive web apps for machine learning and data science.
- **Key Features of Our Dashboard:**
  - **File Uploader:** Allows users to upload one or more CSV files for analysis.
  - **Dynamic UI:** The analysis results only appear after a file is uploaded, keeping the interface clean.
  - **Real-Time Prediction**: The app preprocesses the uploaded data and uses the saved best model to make predictions instantly.
  - **Tabbed Results:** The output is neatly organized into tabs:
    - Prediction Summary: High-level metrics and an "Attack Detected!" warning.
    - Detailed Results: A row-by-row table with predictions, with attacks highlighted.
    - Input Data: A preview of the user's original data.

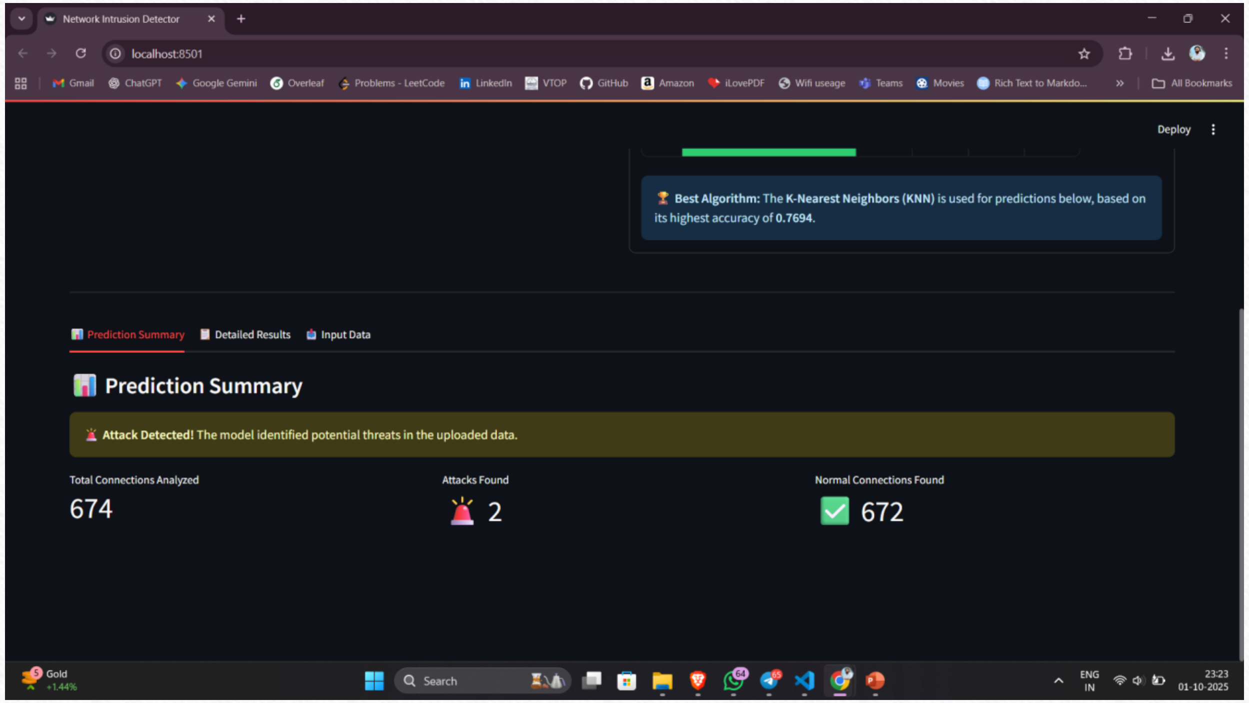
Bringing the Model to Life



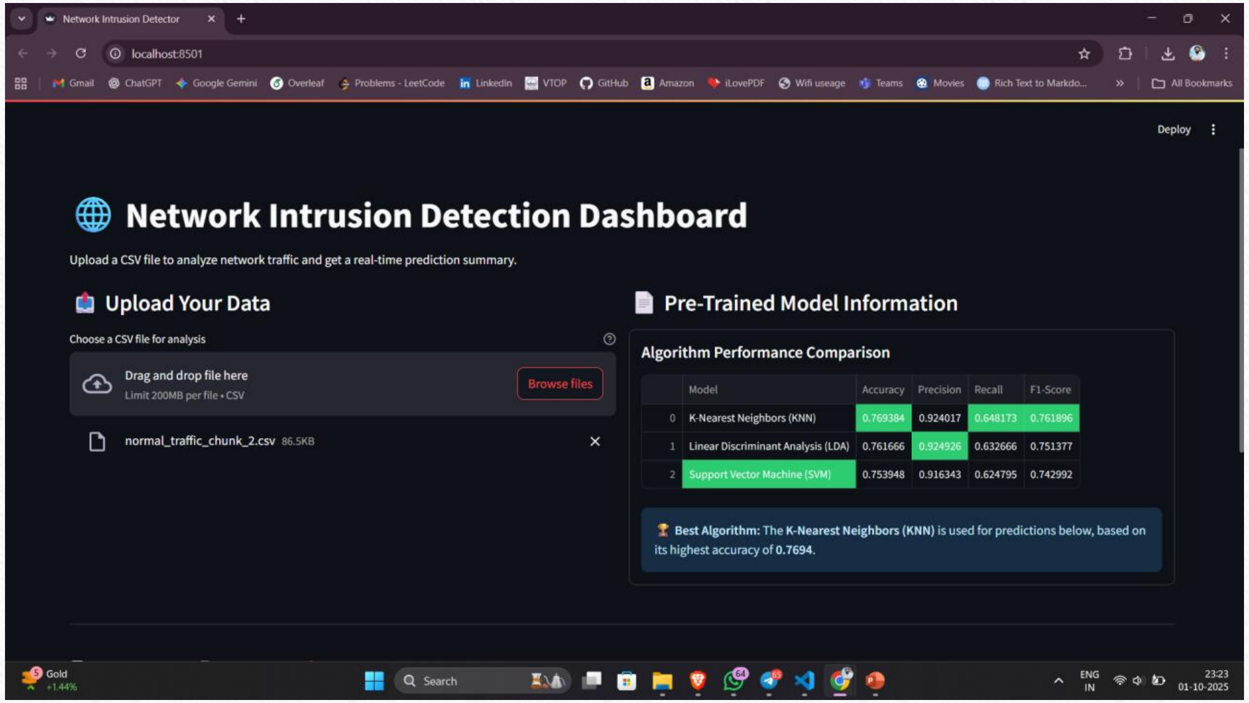
Sample Outputs



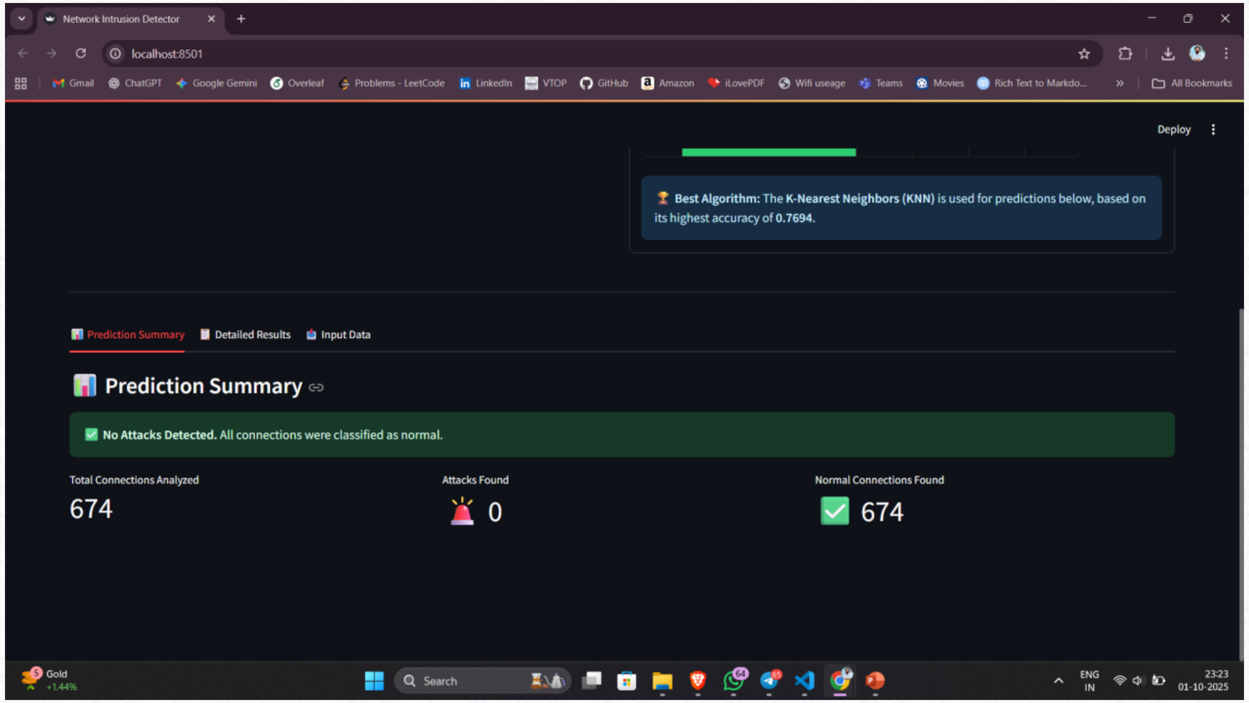
Sample Outputs



Sample Outputs



Sample Outputs



Which Model Performed the Best?

- We evaluated all three models on the unseen NSL-KDD test set. The results are as follows:

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.7694	0.9240	0.6482	0.7619
LDA	0.7617	0.9249	0.6327	0.7514
SVM	0.7539	0.9163	0.6248	0.7430

Summary of Achievements and Next Steps

- **Conclusion:**
  - Successfully demonstrated that classical machine learning provides a powerful and effective foundation for building anomaly-based NIDS.
  - Developed a complete, end-to-end project, from data preprocessing to a fully functional and interactive web application.
  - Empirically proved that KNN was the most suitable model for this specific task on the NSL-KDD dataset.
- **Future Work:**
  - Integrate Deep Learning Models: Explore advanced models like LSTMs or Autoencoders, which might capture more complex patterns.
  - Test on Modern Datasets: Evaluate the models on newer datasets like CIC-IDS2017 to ensure relevance against modern threats.
  - Develop a Real-Time Pipeline: Enhance the application to capture and analyze live network traffic directly, instead of relying on file uploads.
  - Implement Multi-Class Classification: Upgrade the model to not just detect attacks, but also classify the type of attack (e.g., DoS, Probe).

Thank you!