



Project Proposal

Network Intrusion Detection Challenge

Artificial Intelligence and Cyber Security

MSCS 640 - Fall 2026

Dr. ElMouatez Billah Karbab

1 PROJECT OVERVIEW

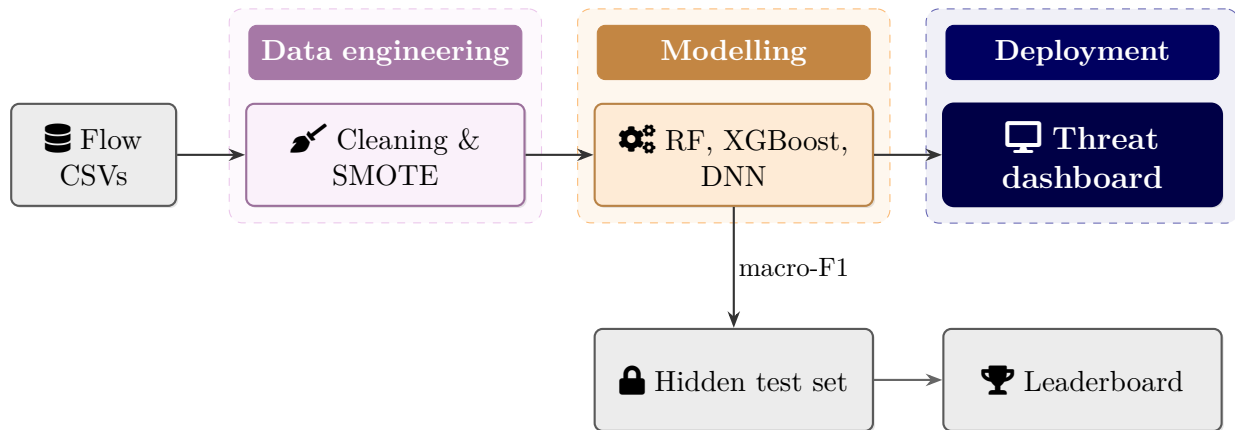


Figure 1: **Project pipeline.** Teams clean the CSE-CIC-IDS2018 flows, train their models and ship a threat dashboard; the Phase 2 models are also scored by macro-F1 on a hidden test set to set the leaderboard rank.

In modern Security Operations Centers (SOCs), analysts are overwhelmed by the sheer volume of network traffic logs. Traditional signature-based firewalls fail to catch zero-day attacks and sophisticated anomalies.

YOUR MISSION: You must act as AI Security Engineers to build a **Machine Learning-based Network Intrusion Detection System (NIDS)**. Your model must analyze raw network flow data and classify traffic as either “Benign” (Normal) or a specific type of “Attack” (e.g., DDoS, Port Scan, Botnet) with the highest possible accuracy.

This project is a **competition**. All teams will compete on the same dataset. The teams with the highest performance metrics on the hidden test set will receive the highest marks (the “State-of-the-Art” Bonus).

2 THE DATASET

We will use **CSE-CIC-IDS2018**, produced by the Canadian Institute for Cybersecurity and described in [1]. It captures ten days of traffic and ships the CICFlowMeter flow features as ML-ready CSVs.

- **Data Source:** the public, anonymously readable bucket <https://cse-cic-ids2018.s3.amazonaws.com/> under Processed Traffic Data for ML Algorithms/. The instructor’s whole-bucket mirror is `aws s3 sync --no-sign-request s3://cse-cic-ids2018/ .`
- **Where to start:** develop and debug on the smallest day, `Thursday-01-03-2018_TrafficForML_CICFlowMeter.csv` (103 MB), then pull the remaining nine days once your pipeline runs end to end. The largest day is about 4 GB, so plan for chunked reading.
- **Features:** 79 flow statistics per record (e.g. Flow Duration, Flow Byts/s, Flow Pkts/s, Tot Fwd Pkts, and the TCP flag counts).
- **Target:** multi-class classification on the `Label` column. Label spellings are taken verbatim from the CSVs (the misspelling `Infiltration` is the dataset’s own): `Benign`, `FTP-BruteForce`, `SSH-Bruteforce`, `DoS attacks-GoldenEye`, `DoS attacks-Slowloris`, `DoS attacks-Hulk`, `DoS attacks-SlowHTTPTest`, `DDoS attacks-LOIC-HTTP`, `DDOS attack-HOIC`, `DDOS attack-LOIC-UDP`, `Brute Force -Web`, `Brute Force -XSS`, `SQL Injection`, `Infiltration`, `Bot`. Each day carries `Benign` plus only the attacks staged that day, so build your training set across days rather than from one file.

3 CORE OBJECTIVES

PHASE 1: DATA ENGINEERING (THE FOUNDATION)

Real-world security data is dirty and imbalanced. You are required to:

1. **Clean the Data:** Handle missing and infinite values common in network logs.

2. **Handle Imbalance:** Network attacks are rare (approx. 1% of traffic) compared to normal traffic. You must apply techniques like **SMOTE** (Synthetic Minority Over-sampling Technique) or Class Weighting to ensure your model detects the attacks, not just the normal traffic.
3. **Feature Selection:** Identify which network features (e.g., **Total Fwd Packets**, **Flow Duration**) are actually relevant to security.

PHASE 2: MODEL DEVELOPMENT (THE ENGINE)

Develop a Deep Learning or Ensemble Machine Learning model.

- **Baseline:** Start with a Random Forest or XGBoost.
- **Advanced:** Implement a Neural Network (MLP, CNN-1D for sequence, or Autoencoder for anomaly detection).
- **Metric:** We optimize for **Macro F1-Score**. Accuracy is misleading in imbalanced security datasets.

PHASE 3: DEPLOYMENT (THE PRODUCT)

A model in a notebook is useless to a SOC analyst. You must build a **web-based dashboard** (using Streamlit or Flask) that:

1. Accepts a CSV upload of new network traffic.
2. Runs your pre-trained model.
3. Displays a “Threat Report” highlighting which flows are malicious.

4 DELIVERABLES

Each group must submit the following by the deadline:

1. **Git Repository:** Must contain `main.py`, `requirements.txt`, and a `README.md` with instructions. Code must be clean and modular.
2. **Project Paper (4 Pages, IEEE Format):**
 - **Abstract:** Summary of your approach.
 - **Methodology:** Handling class imbalance and algorithm choice.
 - **Results:** Confusion Matrix, ROC Curves, and F1-Scores.

- **Conclusion:** Future improvements.
3. **Live Demo/Website:** A URL or local demo of your Threat Detection Dashboard.

5 THE COMPETITION (EVALUATION CRITERIA)

Grades are awarded based on a mix of technical rigor and competitive performance.

Total Points: 100

Component	Points	Description
Data Pipeline	15	Quality of cleaning, EDA, and handling imbalance.
Methodology	15	Justification of model choice and hyperparameter tuning.
Reproducibility	15	Can the TA clone the repo and run it without errors?
The Website	15	UI/UX, functionality, and visualization of results.
The Paper	15	Clarity, structure, and academic quality.
Presentation	15	Demo-day delivery: the talk, the live demo, and handling questions.
Performance Rank	10	Based on the Hidden Test Set Leaderboard .

Leaderboard Bonus:

- **1st Place:** Automatic 10 on Performance + **2% Overall Bonus**.
- **2nd Place:** Automatic 10 on Performance + **1% Overall Bonus**.
- **3rd Place & Below:** Graded on a sliding scale relative to the baseline.

6 RULES OF ENGAGEMENT

1. **Open Source:** You may use open-source libraries (Scikit-Learn, TensorFlow, PyTorch), but you cannot use a pre-trained “black box” API without explanation.
2. **Collaboration:** This is a team project (2 members). All members must commit code to the Git repository.

3. **AI Use:** You may use AI tools, but you must disclose how you used them in a short statement at the end of the paper. You are responsible for every claim: check each statement and each reference against its original source.
4. **Citation Integrity:** Every cited work must exist and must support the sentence that cites it. A fabricated or misattributed reference is treated as academic misconduct.
5. **Plagiarism:** Copying text from any source without quotation and citation is not accepted.

7 TIMELINE

- ☐ **Week 6:** Data Cleaning & Feature Engineering Checkpoint.
- ☐ **Week 9:** Model Training & Optimization Checkpoint.
- ☐ **Week 11:** Website Development.
- ☐ **Week 13:** Final Submission & Demo Day.

REFERENCES

- [1] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, pages 108–116, 2018.