CIS600 Internet of Things Security and Privacy

Fall 2025

Assignment II

Due: 11:59pm on Thursday, November 20, 2025

Total: 10 Points

Submission

ALL submissions should be via Blackboard. No hand delivery will be accepted. Late submissions will be graded out of 50%. The work submitted more than 3 days late gets automatic ZERO.

You are required to submit your solution as a **.** `ipynb` file and it should be fully commented and reproducible.

1. Objective

Students will design, implement, and evaluate a machine learning model capable of distinguishing between benign and malicious software samples based on extracted features. The goal is to explore supervised learning techniques for cybersecurity applications.

2. Learning Outcomes

By completing this assignment, students will be able to:

- Understand feature extraction from malware binaries or metadata.
- Apply classification algorithms to cybersecurity datasets.
- Evaluate model performance using appropriate metrics.
- Interpret model results and discuss practical implications in malware detection.

3. Dataset

The provided data describes a dataset with the following characteristics:

- Total samples: Approximately 138,047
- Legitimate (non-malicious) samples: 41,323 (e.g., .exe, .dll files)
- Malicious samples: 96,724

This represents a class-imbalanced dataset, with more than double the number of malicious files compared to legitimate ones. This type of imbalance is common in cybersecurity and needs to be considered during any subsequent analysis or machine learning model training

4. Assignment Tasks:

1. **Data Exploration:**
    - Load and analyze dataset structure, feature types, and class distribution.
    - Handle missing or categorical data appropriately.
2. **Feature Engineering:**
    - Normalize or standardize features.
    - Perform dimensionality reduction (e.g., PCA, feature selection).
3. **Model Development:**
    - Implement and compare several classifiers (e.g., Random Forest, XGBoost, SVM, Neural Network).
    - Tune hyperparameters using grid search or cross-validation.
4. **Model Evaluation:**
    - Use metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
    - Include confusion matrix and feature importance visualization.
5. **Discussion:**
    - Interpret results, discuss trade-offs (e.g., false positives vs. detection rate).
    - Reflect on limitations and possible improvements.

5. Assessment Criteria

Criteria	**Weight**
Data Preprocessing & Understanding	15%
Model Design & Implementation	35%
Evaluation & Analysis	25%
Interpretation & Discussion	15%
Code Quality & Reproducibility	10%