CST4545: Programming, Systems and Networks for Modern Computing

Final Assessment: Coursework
Deadline: 12/12/2025 – 23:59:00

Coursework Title: AI-Based Intrusion Detection System (IDS) Using Python

Description

The rapid growth of networked systems and the increasing sophistication of cyber threats have made intrusion detection systems (IDS) a critical component in modern computing environments. Machine learning (ML) and deep learning (DL) techniques are increasingly being applied to analyse network traffic and automatically detect anomalous behaviour.

In this coursework, you will develop an AI-powered IDS using Python. The system will classify network traffic as normal or anomalous based on features extracted from a dataset. The project integrates multiple aspects, including:

- Data processing and exploration
- Classical machine learning
- Neural networks
- Real-time prediction simulation using socket programming

This assignment allows you to apply theoretical knowledge to a practical scenario and gain hands-on experience in developing AI-based programs to build intelligent systems for real-world problems.

Two datasets are provided for training and testing AI-based IDS models. They contain labelled records of network connections characterized by 41 features, representing both normal and malicious traffic.

Deliverables

1. **Code (50%)**
    - Submit all Python source files in a single ZIP folder.
    - Ensure the codes are well-commented, and runnable.

2. **Report (30%)**
    - Provide a detailed report explaining each task in the coursework.
    - Include results, visualisations, and graphs where relevant.
    - Maximum length: 2500 words.

3. **Recorded Video (20%)**
    - Submit a video demonstration not exceeding 15 minutes.
    - Showcase the code execution and display the results.
    - Ensure clarity in demonstrating key functionalities of your IDS.

Tasks and Instructions

Part 1 – Exploratory Data Analysis (EDA)

1. Load the training and testing datasets.

- Conduct **EDA** using Python libraries to present the following:
    - The **shape** of the training and testing datasets.
    - **Summary statistics** for the training dataset.
    - The **percentage distribution** of normal and attack records in the training dataset.
    - A **bar chart** visualising the count of normal versus attack records in both the training and testing datasets.
    - A **correlation heatmap** illustrating relationships among the dataset features in the training dataset.
    - A **bar chart** showing the percentage distribution of different attack types in the training dataset.

Part 2 – ML Model Implementation

- Develop and train **two machine learning models** (e.g., Decision Tree, Random Forest, Support Vector Machine, or k-Means) using python libraries.
- **Evaluate** the performance of the models using standard classification metrics, including **accuracy**, **precision**, **recall**, and **F1-score**.

Part 3 – Neural Network Model

- Implement a **neural network** using the **PyTorch** framework.
- **Train** the network on the same dataset used in the previous parts.
- **Evaluate** the model's performance using the same evaluation metrics applied in Part 2 (accuracy, precision, recall, and F1-score).

Part 4 – Model Comparison and Analysis

- **Compare** the performance of the machine learning models and the neural network model on the test dataset using key evaluation metrics: **accuracy**, **precision**, **recall**, and **F1-score**.
- **Discuss** which model demonstrates superior performance and provide reasoning for the observed results.

- **Reflect** on the challenges encountered during the implementation process and suggest potential improvements for future work.

Part 5 – Real-Time IDS Prototype

- Develop a server using Python sockets:
  - Receives data from a client.
  - Uses the trained ML to predict normal/anomalous traffic.
  - Returns the prediction to the client.
- Develop a client:
  - Sends sample test data to the server simulating real-time traffic.

Report Submission Requirements

- **Format**: Submit a structured report including the following sections:

  - **Introduction**: Problem and objectives.
  - **Part 1:** Exploratory Data Analysis (EDA)
  - **Part 2:** ML Model Implementation
  - **Part 3:** Neural Network Model
  - **Part 4:** Model Comparison and Analysis
  - **Part 5:** Real-Time IDS Prototype
  - **Reflection and Conclusion**: Insights and final recommendation

- **File Type**: PDF
- **Words limit:** Max 2500 words

Component	Weight	Assessment Criteria
Code Implementation	50%	- Correctness of Python code - Well-structured, and commented - Successful execution of ML/NN models and IDS pipeline - Acceptable predictions and output
Report	30%	- Clarity, structure, and professionalism - Presentation of results, visualisations, and graphs - Critical analysis and discussion of findings
Video Demonstration	20%	- Clear demonstration of code functionality - Shows running ML/NN models and IDS predictions - Effective communication of workflow and results - Conciseness and adherence to 15-minute limit

Use of AI

AI tools may be used to assist you, but they must not complete the assessment for you. Work generated by AI without your understanding will result in a low mark.