Task – Network Traffic Anomaly Detection (UNSW-NB15 Dataset)

AI for Cybersecurity – Home Assignment

Objective:

Develop anomaly detection and intrusion classification models using the UNSW-NB15 network security dataset**.** You will explore network traffic features, build both classical and deep learning anomaly detectors, and evaluate their performance in identifying malicious attacks.

Dataset link: *(Google Drive folder)*

https://drive.google.com/drive/folders/1tYI7T0dzBtKVRvKHtTqWx1W-2Ls5ajSC

Dataset Overview:

The dataset contains real modern network traffic including normal activity and multiple attack types (e.g., DoS, Exploit, Fuzzers, Reconnaissance, Worms). It includes:

- Flow features **(**duration, bytes, packets, flags)
- Categorical features (protocol, state, service)
- Labels**:**
  - Binary: normal (0) vs. attack (1)
  - Multi-class: attack categories (9 classes)

Assignment Task:

1. Data Pre-processing & Exploration

Perform an initial analysis of the dataset:

- Load dataset and inspect features, data types, label distribution
- Check for missing values and apply an appropriate imputation strategy
- Encode categorical features (protocol, service, state) using One-Hot or target encoding
- Compute summary statistics (mean, std, percentiles)
- Visualize feature distributions (histograms, boxplots, correlation heatmap)
- Analyse class imbalance and consider SMOTE or under-sampling

2. Classical Unsupervised Anomaly Detection

Train at least one anomaly detection model on normal traffic only, such as: Isolation Forest, One-Class SVM**,** Local Outlier Factor (LOF):

- Split dataset into normal (train) and mixed normal + attack (test)
- Train anomaly model on only normal samples
- Compute anomaly scores and determine threshold
- Evaluate using test set against ground-truth attack labels

3. Supervised Intrusion Classification

Train at least two models: Random Forest / Decision Tree / Logistic Regression / Naive Bayes / SVM.

4. Deep Learning: Autoencoder Anomaly Detection

Build a neural Autoencoder to learn normal network patterns.

5. Model Comparison & Final Discussion

- Compare Unsupervised, Supervised, and Autoencoder performance
- Explain tradeoffs: accuracy vs. interpretability vs. generalization
- Discuss cyber-security implications of false positives/negatives
- Suggest improvements (feature selection, embedding, hyperparameter tuning, etc.)

Deliverables (ZIP Submission):

Your submission should include the following components packaged in a single ZIP file:

1. **Python Implementation:**
   - A well-documented Python script or Jupyter Notebook (`anomaly_detection.py`) that includes all steps from data pre-processing to model selection.
   - Ensure your code is organized, with clear comments explaining each section and function.
2. **Project Report (PDF):**
   - A comprehensive PDF document detailing the steps you took to complete the project.
   - **Report Structure:**
     - **Introduction:** Brief overview of the project and its objectives.
     - **Data Pre-processing:** Describe the methods used to clean and prepare the data, including handling missing values and feature engineering.
     - **Modelling:** Explain the classification algorithms chosen and the rationale behind selecting them.
     - **Evaluation:** Present the performance metrics and confusion matrices for each model.
     - **Model Tuning:** Discuss the hyperparameter tuning process and how it improved model performance.
     - **Model Selection:** Justify the final model choice based on your evaluations.

        - **Conclusion:** Summarize your findings and suggest potential improvements or future work.

Submission Instructions:

1. **Prepare Your Work:**
    - Ensure your Python script or Jupyter Notebook runs without errors and includes all necessary components.
    - Compile your project report into a PDF document, ensuring it is well-formatted and free of typos.
2. **Create a ZIP File:**
    - Include both your Python implementation and PDF report in a single ZIP archive.
    - Name the ZIP file following any specific guidelines provided (e.g., `YourName_AnomalyDetection_Project.zip`).
3. **Upload:**
    - Submit the ZIP file through the designated submission portal or as instructed by your course guidelines.

Additional Guidelines:

- **Reproducibility:** Ensure that your code can be executed on a different machine without issues. Include any necessary instructions or requirements.
- **Documentation:** Provide clear comments in your code and ensure your report is thorough and well-organized.
- **Academic Integrity:** Do your own work and cite any external resources or references you use.