Deep Learning - Instructor: Vito Walter Anelli, Ph.D.
M.D. in Computer Engineering - Politecnico di Bari
Test code: 2025_IV - Released on April 27th 2025
Deadline: May 31st, 2025

I Cybersecurity Threat Mitigation: Advanced Behavioral Anomaly Detection in User Authentication Logs Using Deep Learning

1 Task Overview

Anomaly detection in sequential authentication event data is a critical component of modern cybersecurity frameworks. With the escalating frequency of cyberattacks, compromised accounts, and insider threats, organizations require proactive systems to identify suspicious activities that evade traditional rule-based security measures. This project focuses on developing Deep Learning models to detect anomalous user behaviors in authentication logs, a vital capability for safeguarding enterprise networks against unauthorized access, data breaches, and advanced persistent threats.

By analyzing sequences of authentication events, the models will learn patterns of normal user behavior and flag deviations indicative of malicious activities. The task leverages **sequence modeling, temporal analysis, and attention mechanisms** to address the complexity of real-world cybersecurity data.

This project emphasizes the application of cutting-edge Deep Learning architectures to a high-stakes cybersecurity challenge, using the **Comprehensive, Multi-Source Cyber-Security Events** Dataset from the **Los Alamos National Laboratory (LANL) enterprise network**, a real-world enterprise network dataset, to ensure practical relevance and scalability.

2 Objective

The main objective is to design, implement, and evaluate deep learning models for anomaly detection using user authentication logs.

Sub-Objectives

- Acquire the dataset and prepare the data.
- Preprocess sequential event data into suitable model inputs.
- Implement and train three types of Deep Learning models described in Section 5.
- Evaluate the model performance using relevant metrics.
- Compare the performance of the models.

3 Datasets

For this assignment, the required dataset is the **Comprehensive, Multi-Source Cyber-Security Events** Dataset from the **Los Alamos National Laboratory (LANL) enterprise network** (2017), which is a large, publicly available cybersecurity dataset capturing 58 days of real enterprise network activity, containing billions of events across authentication logs, process executions, network flows, and DNS queries, along with labeled malicious activities performed by a red team simulating insider and external attacks. The dataset is fully timestamped, allowing event correlation across sources, and it preserves enterprise realism while anonymizing sensitive information.

Link to the official dataset page: `https://csr.lanl.gov/data/cyber1/`

Ensure that the dataset is properly preprocessed before training.

4 Data Preprocessing

Load the dataset and split it into training, validation, and test sets. **Preprocess the data to resize the dataset to be suitable for deep learning models on your machines.**

5 Model Architecture

The students must implement the following models:

- **LSTM:** Sequence modeling using unidirectional LSTM layers.
- **BiLSTM + Attention:** Bi-directional LSTM followed by an attention mechanism to weigh important events.
- **Transformer for Anomaly Detection:** A transformer-based model specialized for detecting anomalies in sequences via attention.

Each model will predict whether a given session is **anomalous** or **normal**.

6 Hints

You can:

- Ensure that class imbalance is handled properly (anomalous sessions are rare);
- Use embedding layers instead of one-hot encoding for categorical variables;
- Batch normalization and dropout can improve model convergence.

7 Training and Evaluation

Train the model on the training set using suitable loss functions and optimization techniques. Monitor the training process by evaluating the model's performance on the validation set. Use appropriate evaluation metrics. Fine-tune the model hyperparameters to improve its performance on the validation set. Finally, evaluate the trained model on the test set and report its performance metrics.

8 Model Analysis

Perform an analysis of the model's performance. Identify any challenges or limitations faced by the model. Provide recommendations for further improvements or modifications to enhance the model's accuracy and efficiency.

9 Deliverables

- A well-commented Python code implementation of the models.
- A report documenting the detailed approach taken to solve the problem, including data preprocessing, model architecture, training, and evaluation results.
- Analysis and discussion of the model's performance, challenges faced, and recommendations for improvement.
- A presentation that will be discussed at the time of the Oral Exam.

Notes

You are encouraged to utilize existing deep learning libraries such as TensorFlow or PyTorch for implementing the model. Make sure to properly document your code and provide clear explanations in the report to demonstrate your understanding of the concepts and techniques used.