Chair of IT Security
Brandenburg University of Technology

Asya Mitseva, M.Sc.
Prof. Dr.-Ing. Andriy Panchenko

Winter Semester 2024/2025

Part V: Comparison of Detection Methods (Part 2) **Deadline: 22.01.2025**

Study Project: Adversarial Machine Learning Anomaly Detection in 5G Network Data

Introduction

The fifth-generation (5G) technology standard for cellular networks offers unprecedented data rates, low latency, and the ability to support a wide range of interconnected devices. However, 5G comes with an increased complexity that expands the attack surface, requiring robust security measures to ensure the integrity and reliability of these networks. Intrusion Detection Systems (IDSs) have long been a critical component in protecting network infrastructures by identifying malicious activities. In the context of 5G, the vast amounts of data generated, the heterogeneity of traffic, and the widespread distribution of network nodes present a significant challenge to centralized detection methods. Moreover, privacy concerns arise when dealing with sensitive data, particularly in sectors such as healthcare and finance, where regulatory restrictions further complicate data sharing. As a result, there is a need to propose decentralized approaches capable of handling the scale and complexity of 5G networks, while preserving data privacy. Federated Learning (FL) offers a solution to address these challenges. By enabling decentralized model training, FL allows individual nodes within the network to develop local intrusion detection models on their own data. Only the trained model updates are shared with a central server, preserving the privacy of the raw data while contributing to a global model. Each node—whether a smartphone, Internet of Things (IoT) device or base station—trains a local model on its own data, which is then aggregated into a global model. This approach not only preserves the privacy of sensitive information but also improves the detection of sophisticated attacks by leveraging knowledge from a wide range of network segments.

The goal of this study project is to implement and evaluate different methods for centralized and distributed (e.g., federated learning) anomaly detection in 5G network traffic. In the previous practical sheet, we implemented different detection methods and started with their evaluation in different scenarios. In the last practical part of the study project, we continue with the evaluation of the different classifiers and the analysis of their classification errors.

Task 1: Execution of Experiments

The goal of this task is to continue the evaluation of your classifiers (those implemented in the previous practical sheet, your federated learning method, and your deep learning based method) by using the features created in the previous practical sheet or the raw data for your local autoencoders for each of the evaluation scenarios, defined in Task 3 of the Practical sheet 3, and by using the corresponding training and testing sets. In particular, you should execute the following experiments:

a) For the first evaluation scenario, you should execute one experiment for each of the traditional ML classifiers (with which this execution would be possible) and your federated learning method and using the corresponding training and testing set. After splitting the real traffic flows into training and testing sets, you should replace the flows in the corresponding training sets with synthetic flows, generated from your synthetic method

implemented in the first practical sheet. The corresponding number of synthetic flows should be generated using the real training flows for a given class. Make sure that for each classifier you identify and use the optimal grid parameters and compute the *precision* and *recall* for each of the experiments.

b) For the second evaluation scenario, you should execute one experiment for each of the traditional ML classifiers (with which this execution would be possible) and your deep learning based method (with $M = 5000$ and including the 5 manual features) using the corresponding training and testing set. After splitting the real traffic flows into training and testing sets, you should replace the flows in the corresponding training sets with synthetic flows, generated from your synthetic method implemented in the first practical sheet. The corresponding number of synthetic flows should be generated using the real training flows for a given class. Make sure that for each classifier you identify and use the optimal grid parameters and compute the *precision* and *recall* for each of the experiments.

c) For the third evaluation scenario, you should execute one experiment for each of the traditional ML classifiers (with which this execution would be possible) and your deep learning based method (with $M = 5000$ and including the 5 manual features) using the corresponding training and testing set. After splitting the real traffic flows into training and testing sets, you should replace the flows in the corresponding training sets with synthetic flows, generated from your synthetic method implemented in the first practical sheet. The corresponding number of synthetic flows should be generated using the real training flows for a given class. Make sure that for each classifier you identify and use the optimal grid parameters and compute the *precision* and *recall* for each of the experiments.

d) For each of the second and third evaluation scenarios, you should execute one experiment using your deep learning based method for $M = 2500$ and including the 5 manual features, one experiment using your deep learning based method for $M = 5000$ and excluding the 5 manual features, one experiment using your deep learning based method for $M = 2500$ and excluding the 5 manual features, using the corresponding training and testing set. After splitting the real traffic flows into training and testing sets, you should replace the flows in the corresponding training sets with synthetic flows, generated from your synthetic method implemented in the first practical sheet. The corresponding number of synthetic flows should be generated using the real training flows for a given class. Make sure that for each classifier you identify and use the optimal grid parameters and compute the *precision* and *recall* for each of the experiments.

e) For the first evaluation scenario, you should execute one experiment for your n-gram based detection method implemented in Pracrical sheet 3.

Plot all obtained classification results in an appropriate way and submit your plots and the obtained classification results. Describe in details all takeaways that can be concluded after executing the experiments.

Task 2: Feature Importance and Analysis of Prediction Errors

The goal of this task is to establish the importance of the manually generated features, implemented in the previous practical sheet, and to take a closer look at the classification errors produced by each of the classifiers from the experiments, executed in the previous practical sheet. In this task, you should write a piece of code that executes the following analysis:

a) For each of the implemented traditional ML classifiers and for each of the evaluated scenarios, you should establish the importance score of each of your manually generated features and plot the obtained results, always starting with the most important feature.

b) For each of the implemented traditional ML classifiers and for each of the evaluated scenarios, you should measure the effect of using a subset of features. Based on your calculated

feature importance score for the corresponding classifier, you should execute several experiments where you train that classifier with only subsets of the most informative features in batches of six by increasing the number of features for each subsequent experiment and compute the precision and recall for each of the experiments, i.e., first you train your classifier with the first six most important features only, next you train your classifier with the first 12 most important features only, etc.

c) For the set of experiments in the items a)–d) in Task 3 from the previous practical sheet, you should implement and plot a Venn diagram of classification errors for each of the evaluation scenarios. Each circle should represent the set of prediction errors for one of your classifiers. The circles should intersect when the same testing instance is incorrectly classified by the different classifiers.

Plot all obtained evaluation results in an appropriate way and submit your plots. Describe in details all takeaways that can be concluded after executing the experiments.

Task 3: Extension and Finalization of Your Toolbox

In Practical sheet 2, you created a toolbox that combined your piece of code prepared in Practical Sheet 1 and Practical Sheet 2. The goal of this task is to extend this toolbox by adding all remaining parts of your project implemented in the subsequent practical sheets in this toolbox. Your toolbox should provide several options that are read from the command line. The parameters that were already implemented in the previous practical sheet should be kept. In addition, the support of the parameter *-s <keyword>* should be extended to support the execution of all other parts of the project. Any already existing command line parameters can be kept and used in the toolbox as well.

Preparation for Q&A Session

Prepare yourself for a Q&A session of 30 minutes. During the Q&A session, you need to give an overview of libraries, scripts or already existing tools that you have used. Furthermore, you need to make a demo of your piece of code for every task, explain its operation in detail, and show your classification results and plots. Last but not least, you should be ready to answer spontaneous questions asked by the reviewer.