2022 Summer ELG 5142 Ubiquitous Sensing and Smart City Project

Professor: Dr. Burak Kantarci

TA: Zhiyan Chen, zchen241@uottawa.ca

Sarah Hossam Abdelhameed Elmowafy, selmo028@uottawa.ca

Submission Deadline: Friday, July 22, 2022 11:59pm EDT Ottawa/Toronto/New York Time

Background

Fake task attack is critical for Mobile Crowdsensing system (MCS) that aim to clog the sensing servers in the MCS platform and drain more energy from participants' smart devices. Typically, fake tasks are created by empirical model such as CrowdSenSim tool. Recently, cyber criminals deploy more intelligent mechanisms to create attacks. Generative Adversarial Network (GAN) is one of the most powerful techniques to generate synthetic samples. GAN considers the entire data in the training dataset to create similar samples. This project aims to use GAN to create fake tasks and verify fake task detection performance.

Introduction

The objective of this project is to generate intelligent fake tasks by GAN. Meanwhile, you need to evaluate detection accuracy for original fake tasks and GAN-generated fake tasks. Detection mechanisms contains classic machine learning (ML) models (e.g., Random Forest (RF) and Adaboost) and a GAN-based cascade detection framework. Another objective is to compare traditional MLs and the cascade detection framework detection performance.

Traditional ML models (RF and Adaboost)-based detection methods are the same as assignment 2 which are considered as benchmarks. This project focuses on the GAN-based cascade framework which is demonstrated in Figure 1. The cascade framework is proposed that implements a two-level cascaded classifier architecture to predict the generated attack samples and original (empirically designed) attack samples and filter them before distributing them to MCS participants.

The first level is formed by the GAN discriminator whereas the second level is a binary classifier. According to Figure 1, the training dataset is utilized to train binary classifiers (e.g., RF, Adaboost) and GAN. The training dataset includes original fake tasks. GANs take input samples of noise, and output adversarial (i.e., synthetic) samples. GAN consists of a Generator neural network and a Discriminator neural network. The role of the generator is to create synthetic samples as similar as the real ones whereas a Discriminator designs to distinguish real samples from the synthetic ones. The motive for the competition is to reduce the gap between generated samples for Generator and increase detection accuracy for Discriminator.

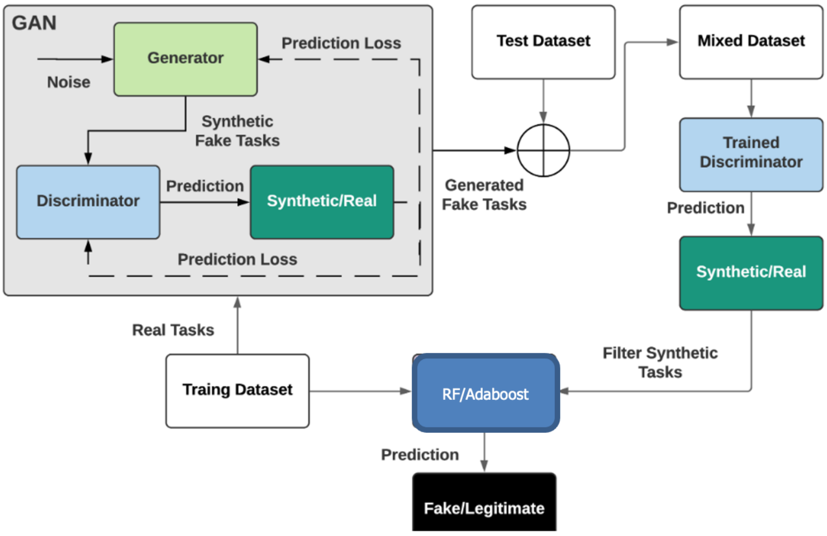

Figure 1 Cascade detection framework overview [1]

This project has the following objectives:

1. Understand GAN model working procedure
2. Implement GAN model (e.g., Conditional GAN (CGAN))
3. Generate fake tasks via CGAN
4. Verify RF and Adaboost fake task detection performance, including original fake tasks and GAN-based fake tasks
5. Verify the cascade framework detection performance for original fake tasks and GAN-based fake tasks

Main Test Steps

Test procedures are summarized in below steps:

1. Download MCS dataset which is used in assignment 2 (Dataset can be downloaded via this link http://nextconlab.academy/MCSData/MCS-FakeTaskDetection.html)
2. Split the dataset into training dataset (80%) and test dataset (20%)
3. Implement classic classifiers (Adaboost and RF)
4. Train Adaboost and RF via training dataset
5. Verify detection performance using test dataset and present results comparison in bar chart
6. Implement a CGAN model [2]
7. Apply the provided training dataset to CGAN (the training dataset can be the same as you used in assignment 2)
8. Generate synthetic fake tasks via Generator network in CGAN after the training procedure
9. Mix the generated fake tasks with the original test dataset to obtain a new test dataset
10. Obtain Adaboost and RF detection performance using the new test dataset and present results in bar chart (This step doesn’t consider Discriminator for filtering synthetic samples). Please refer Figure 2 as test framework structure with test step index.
11. According to the cascade detection framework, as shown in Figure 1, verify the cascade framework

performance and show results in bar chart. Consider the Discriminator to as the first level classifier and RF/Adaboost as the second level classifier. Please refer Figure 3 as test framework structure with test step index.

Tips: a) Step 1 to step 5 are the same as assignment 2. You can reuse the code and results in assignment 2 and focus on the remaining steps; b) in step 10, Adaboost and RF are trained using training dataset which is the same as assignment 2; The difference here (in step 10) is to use the mixed test dataset for testing instead of original test dataset.

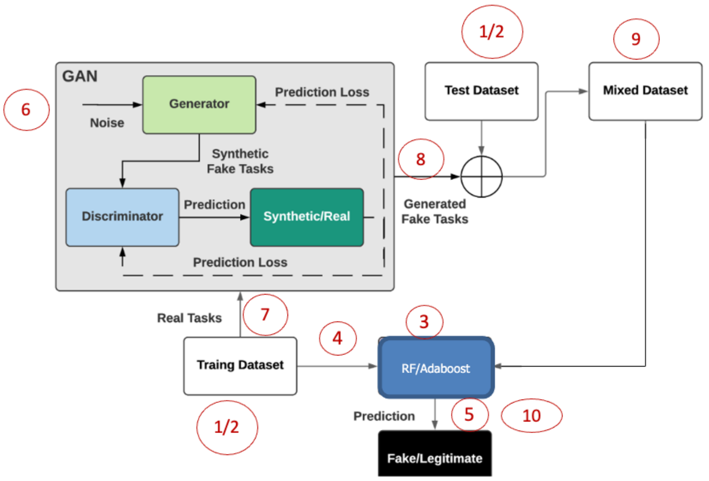

Figure 2 Test framework for step 1 to 10.

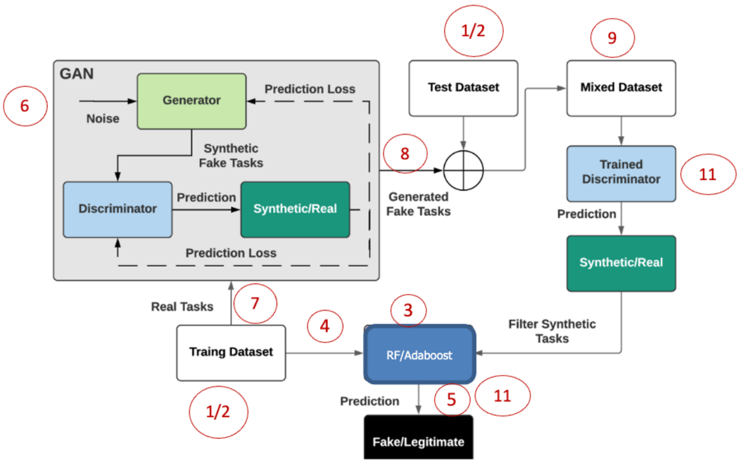

Figure 3 Test framework for step 1 to 11

Required Sections in the assignment:

Points you need to include in your assignment report:

1. In the report, please include code, test results and accuracy comparison figure for three parts:
   a. Under original test dataset for RF and Adaboost, which is the same assignment 2
   b. Under mixed test dataset for RF and Adaboost (without the Discriminator to as first classifier) (35 points)
   c. Under mixed test dataset for the RF-based cascade framework and Adaboost-based cascade framework (35 points)
2. Comparison figure with the format as follows. Please note this figure is just illustrative; the accuracy values are not actual representation of the classifier performances (15 points):

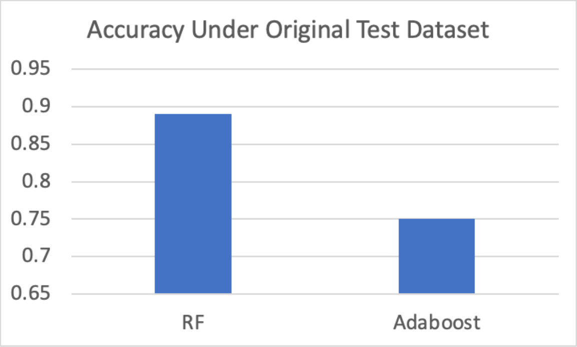

Figure 4 Accuracy comparison using original test dataset using RF and Adaboost

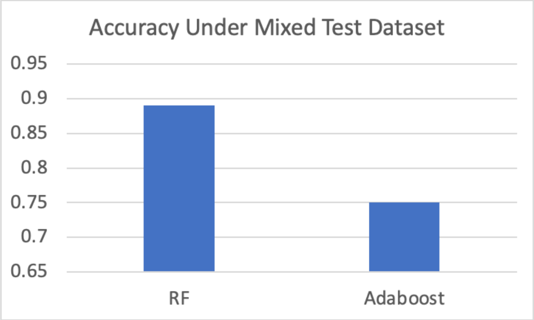

Figure 5 Accuracy comparison using mixed test dataset using RF and Adaboost

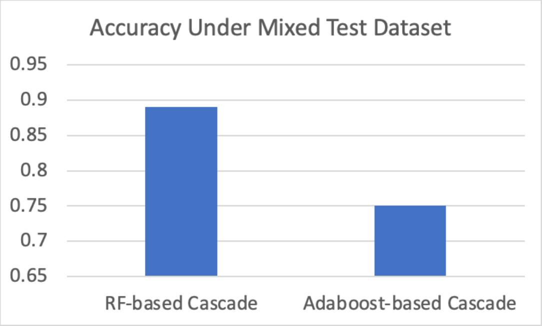

Figure 6 Accuracy comparison using mixed test dataset using cascade framework

3. Analyze test results among the cascade framework and classic ML algorithms. Compare and contrast on the performance of the cascaded approach versus ML-based detection under adversarial ML-generated fake tasks (15 points).

Submission requirement

1. Please submit code to show CGAN implementation, RF/Adaboost implementation, and the cascade framework

2. Summary all in a report, including code snippet, comparison figures, and analysis results.

Resources

3. GAN implementation via keras https://keras.io/examples/generative/conditional_gan/

4. GAN implementation via tensorflow https://www.tensorflow.org/tutorials/generative/dcgan

5. GAN tutorial https://towardsdatascience.com/generative-adversarial-network-gan-for-dummies-a-step-by-step-tutorial-fdefff170391

6. Lab 5 materials

Reference

[1] Chen, Zhiyan, and Burak Kantarci. "Generative Adversarial Network-Driven Detection of Adversarial

Tasks in Mobile Crowdsensing." arXiv preprint arXiv:2202.07802 (2022).

[2] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." arXiv preprint arXiv:1411.1784 (2014).