Iran University of Science and Technology

School of Computer Engineering

Final Project

ADVANCED DATA MINING, SPRING 2025

DR. MINAEI

Amirhossein Namazi

Project: Implementing SRR for Unsupervised Anomaly Detection on Image and Tabular Data

Project Due Date: 15,Tir,1404

📄 Overview

This project focuses on implementing the **SRR (Self-supervise, Refine, Repeat)** framework for **fully unsupervised anomaly detection (AD)**. The SRR method enhances AD by iteratively refining unlabeled training data—which contains both normal and anomalous samples—using an ensemble of one-class classifiers (OCCs). Self-supervised data representations are then updated using this cleaner data.

You will implement this framework for **both the CIFAR-10 image dataset and the Thyroid tabular dataset**, focusing on the core methodology described in the paper.

🎯 Goal

The primary goal is to implement a robust version of the SRR framework for your chosen dataset (CIFAR-10 or Thyroid), ensuring all components align with the SRR paper's methodology. You will then evaluate its performance, particularly its ability to handle varying anomaly ratios in the training data.

🛠️ Core Implementation Steps

You will follow **both** of the following tracks.

Track 1: Image Anomaly Detection with CIFAR-10

The specific Self-Supervised Learning (SSL) task to be implemented for this track is **Rotation Prediction**.

1. **Data Preparation and Contamination (CIFAR-10):**

   - Load the CIFAR-10 dataset and designate one class as "normal" and another as "anomalous".
   - Inject a specified percentage of "anomalous" samples into the "normal" training data to create your unlabeled training set.
   - The test set should consist of clean "normal" and "anomalous" samples.
2. **Self-Supervised Representation Learning (Rotation Prediction):**
   - Implement a Representation Learner using a **ResNet-18** backbone.
   - The self-supervised task is **Rotation Prediction**. The model is trained to predict the rotation (e.g., 0, 90, 180, 270 degrees) applied to input images.
   - This learner is trained *only* on the refined data from the next step.

Track 2: Tabular Anomaly Detection with Thyroid

The SSL task for this track will be based on the **GOAD** methodology, which uses affine transformations.

1. **Data Preparation and Contamination (Thyroid):**
   - Load the **Thyroid** dataset from the UCI repository (or a similar source).
   - Following the paper, use a portion of normal samples for training and contaminate it by injecting a specified percentage of anomaly samples to create.
   - The rest of the data will be used for testing.
2. **Self-Supervised Representation Learning (Transformation Classification):**
   - Implement a Representation Learner using a simple **MLP (Multi-Layer Perceptron)** backbone suitable for tabular data.
   - The self-supervised pretext task involves applying a set of transformations to the data and training the model to classify which

transformation was applied, similar to the **GOAD** approach described in the paper.

Shared Implementation Steps (Both Tracks)

The following modules are part of the core SRR loop and apply to both tracks, operating on the representations produced by your chosen learner.

1. **Data Refinement Module:**
    - Implement an ensemble of **K One-Class Classifiers (OCCs)**. The paper suggests K=5.
    - The suggested OCC is a **Gaussian Density Estimator (GDE)**.
    - In each update step, train the K OCCs on **disjoint subsets** of the current data representations.
    - Determine a threshold for each OCC based on a percentile (gamma) of its anomaly scores. gamma should be set based on the expected anomaly ratio (e.g., 1-2 times the ratio).
    - **Pseudo-labeling:** A sample is considered pseudo-normal if **all** K OCCs classify it as normal.
2. **Iterative Training Loop (SRR Algorithm 1):**
    - Initialize the Representation Learner and loop for a set number of SRR iterations.
    - **Refine Data:** Use the current learner and OCC ensemble to get a refined dataset.
    - **Update Learner:** Train the Representation Learner on the SSL task using the refined dataset.
    - Update the data refinement OCCs intermittently as suggested in the paper (e.g., at iterations 1, 2, 5, 10...).
    - After the loop converges (e.g., based on SSL loss plateaus), perform one final refinement to get $\widehat{D}$.

3. **Final One-Class Classifier:**
    - Train a final **GDE-OCC** on the representations of the final refined dataset, $\widehat{D}$. This OCC will be used for evaluation.

📊 Evaluation

Your analysis must demonstrate the performance of your implementation with **varying anomaly contamination ratios** in the training data (e.g., 0% to 10%).

- **For CIFAR-10:** Use **AUC (Area Under the Curve)** and **AP (Average Precision)** to quantify performance. Plot these metrics against the anomaly ratio and compare your results to the trends in the SRR paper (e.g., Fig. 5).
- **For Thyroid:** Use the **F1-Score** to quantify performance. Plot the F1-score against the anomaly ratio and compare your results to the trends in the SRR paper (e.g., Fig. 4).

Deliverables

1. **Source Code:** Your finalized, well-commented Python code (e.g., Jupyter Notebook) implementing the SRR framework for your **chosen track (CIFAR-10 or Thyroid)**.
2. **Report:**
    - A brief overview of the SRR framework.
    - Details of your implementation, specifying your chosen track and explaining how your code aligns with the paper's methodology.
    - Description of your data setup, contamination strategy, and evaluation procedure.
    - **Key Results:** Graphs and discussion showing your model's performance (AUC/AP or F1-score) across different training anomaly ratios.
    - Discussion of any challenges encountered and observations.

Notes

- Any instance of cheating, including copying code or reports from other students, will result in a grade of zero.
- Grace time cannot be used for the project.
- Please be advised that the project deadline is firm and will not be changed.
- If you have any questions, feel free to ask. You can ask your questions in the Telegram group.
- Please upload your assignments as a zipped folder with all necessary components. Upload your file in Final Project-YourStudentID-YourName.zip format.