AI-DAC Runbook

William K.
University of Vienna
Department of Computer Science
a09726537@unet.univie.ac.at

May 21, 2025

Overview

This runbook outlines the procedures, file structure, and scripts necessary to reproduce the AI-DAC experimental environment and validate the Triple Loop Learning (TLL) anomaly detection system on RDBMS logs.

Repository

The codebase is available at: `https://github.com/a09726537/tll-rdbms/`

Included Scripts and Files

- `load_unsw.py`: Converts UNSW-NB15 .csv data into normalized time-series event logs for ingestion.
- `inject_gan_samples.py`: Injects synthetic anomalies generated by MAD-GAN into the training stream.
- `runbook.pdf`: This runbook file (PDF), outlining usage and reproducibility steps.
- `config.yaml`: Defines detection parameters, logging, and batch modes.
- `docker-compose.yml`: Local orchestration of the AI-DAC detector and RAG module.
- `helm-values.yaml`: Kubernetes Helm chart values for production deployment.

Execution Steps

1. Clone the repository:

```
git clone https://github.com/a09726537/tll-rdbms.git
cd tll-rdbms
```

2. Preprocess datasets:

```
python scripts/load_unsw.py
python scripts/inject_gan_samples.py
```

3. Launch system (Docker):

```
docker-compose up
```

4. Launch system (Kubernetes):

```
helm install tll ./helm-chart/ -f helm-
    values.yaml
```

Output Artifacts

- **GAN Logs:** Generated using `inject_gan_samples.py`
- **Normalization:** Use `load_unsw.py` to preprocess and standardize.
- **Explainability Logs:** Stored in `/logs/explanations/` with SHAP + RAG output.

Contact and Attribution

William K.
University of Vienna, Department of Computer Science
a09726537@unet.univie.ac.at