DM Project: Transformer-Based Anomaly Detection Pipeline

Awab ur Rehman, 22i-1068 — Farhan Ahmed, 22i-1200

1 Problem Statement

Anomaly detection becomes challenging when training data is contaminated with outliers, since the model's understanding of "normal" behaviour becomes distorted. The coursework required a contamination-resilient framework that combines geometric masking, a Transformer encoder–decoder, contrastive learning, and a GAN regularizer to enhance generalization and reduce overfitting via representation learning. The solution must be demonstrated on a publicly available multivariate time-series dataset (NASA, eBay, or NTU) obtained from the TS-AD repository.

2 Requirements Recap

The specification mandated:

- Geometric masking augmentation for robust training.
- Transformer architecture for feature extraction and reconstruction.
- Contrastive loss to enforce separation between normal/anomalous representations.
- GAN module to improve modelling of realistic normal behaviour despite contaminated data.
- Evaluation on a multivariate dataset from TS-AD.
- Comprehensive documentation covering dataset, preprocessing, architecture, training, metrics, results, reproducibility, and code structure.

3 Dataset

3.1 Source and Access

The Server Machine Dataset (SMD) released by eBay (the OmniAnomaly repository) was used. The dataset is part of the TS-AD curated list (`https://github.com/elisejiuqizhang/TS-AD-Datasets`).

3.2 Content and Structure

- 28 machines, each with 38 sensor channels sampled every minute.
- For each machine m, we have:
  - `train/machine-m.txt`: normal training data only.

  - `test/machine-m.txt`: mixture of normal and anomalous sequences.
  - `test_label/machine-m.txt`: binary labels (0 normal, 1 anomaly) per timestamp.
- This project focuses on `machine-1-1`. The test portion contains 888 anomalous timestamps out of 56,960 total.

4 Preprocessing

1. **Normalization:** StandardScaler fitted on the training set and applied to both train and test.
2. **Sliding Windows:** Windows of length 128, stride 32. Each window inherits the max label inside the window (point-level labels aggregated to window-level labels).
3. **Geometric Masking:** Random contiguous spans are zeroed according to a geometric distribution (mask ratio 25%, geometric $p = 0.2$). Both masked and unmasked views are used for reconstruction and contrastive objectives.

5 Model Architecture

5.1 Transformer Encoder–Decoder

- Inputs projected to $d_{model} = 128$ with sinusoidal positional encodings.
- Encoder: 4 layers, 4 attention heads, feed-forward dimension 256, dropout 0.1.
- Decoder: 2 layers, same hyperparameters.
- Latent vectors aggregated (mean pooling) and fed into:
  - **Projection head** for InfoNCE contrastive loss.
  - **GAN generator** conditioning.

5.2 Contrastive Loss

InfoNCE pulls representations of masked/unmasked views of the same window together while pushing apart different windows. Temperature = 0.2, projection dimension = 64.

5.3 GAN Regularization

- **Generator:** Takes latent code + noise (dimension 32) to produce residual corrections. Applied on top of Transformer reconstruction.
- **Discriminator:** Distinguishes mean-pooled representations of real vs generated sequences.
- Stabilizes training against contaminated "normal" data by penalizing unrealistic reconstructions.

6 Training Procedure

- **Losses:**
  - Reconstruction MSE (full sequence).
  - Masked MSE (only masked spans).
  - Contrastive InfoNCE.
  - Adversarial BCE for generator and discriminator.
- **Hyperparameters:** Epochs 50, batch size 64, LR 1.5×10^{-4}, weight decay 1×10^{-4}, gradient clip 1.0, discriminator steps 1.
- **Artifact Saving:** History, checkpoints every 5 epochs, final scores/labels. When Drive is mounted, artifacts are mirrored to `MyDrive/DMProject_backup/outputs/`.

7 Evaluation Metrics

The evaluation script produces:

- Precision, recall, F1 (anomaly as positive).
- ROC-AUC, PR-AUC.
- Anomaly ratio (ground truth) and predicted anomaly ratio.
- Threshold (95th percentile by default).

Plots generated:

- **Anomaly scores** with threshold and detected anomalies.
- **Score distribution** (normal vs anomaly histogram).
- **ROC and PR curves.**
- **Loss curves** (reconstruction, mask, contrastive, adversarial) via `plot_losses.py`.

8 Results on SMD (machine-1-1)

8.1 Numerical Metrics

Threshold	Precision	Recall	F1	Anomaly Ratio	Predicted Ratio	ROC-AUC	PR-AUC
66.0808	0.9111	0.3565	0.5125	0.1298	0.0508	0.9506	0.7584

Table 1: Evaluation metrics computed from `eval/metrics.json`.

8.2 Interpretation

- **High precision (0.91):** The detector rarely raises false alarms; flagged windows are genuinely anomalous.

- **Moderate recall (0.36):** The 95th percentile threshold is conservative. Lowering the percentile or applying adaptive thresholding could recover more anomalies at the cost of precision.
- **ROC-AUC 0.95, PR-AUC 0.76:** Strong separability despite class imbalance.
- **Anomaly ratios:** Actual 13% vs predicted 5% indicates a precision-friendly operating point (useful when false positives are costly).

8.3 Figures

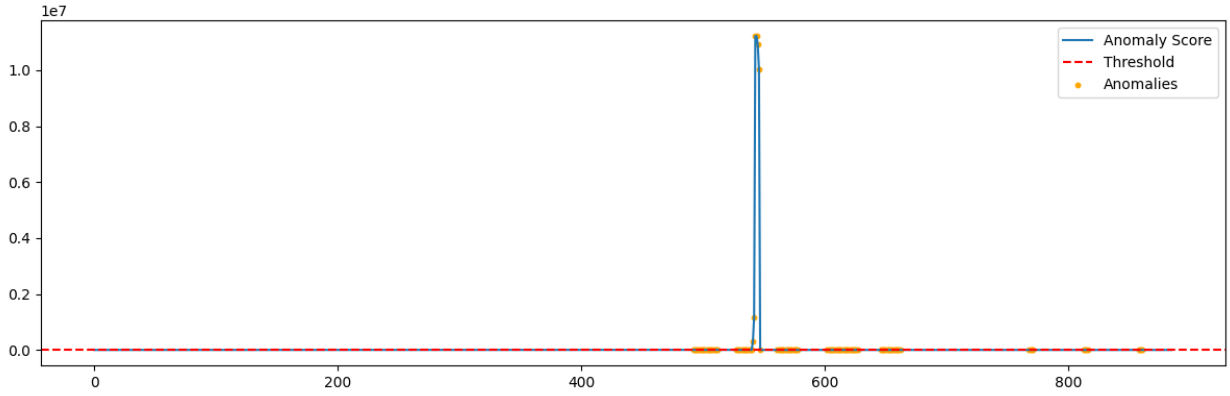

Figure 1: Anomaly scores over time with 95th-percentile threshold and detected anomalies.

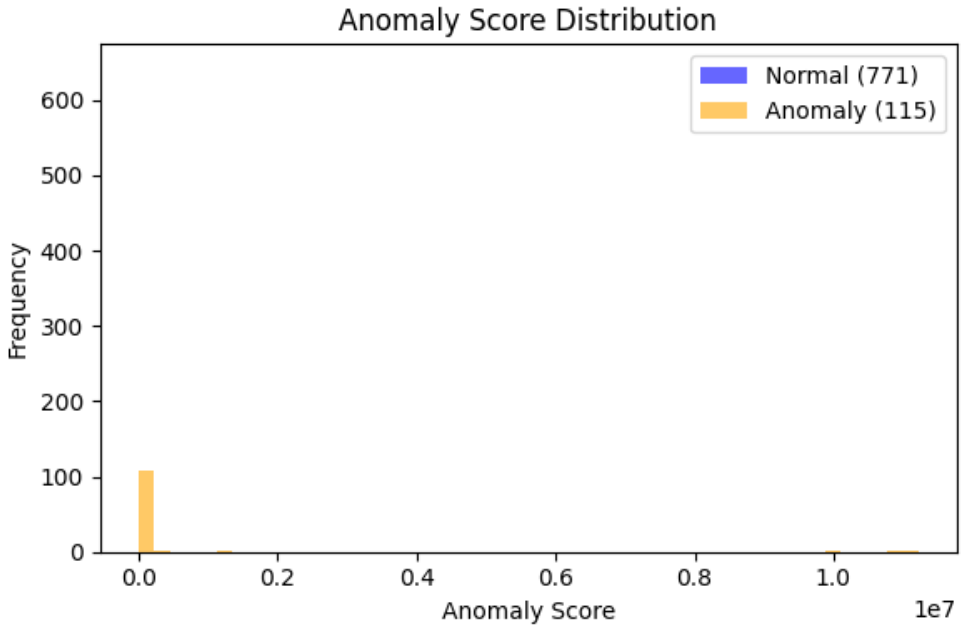

Figure 2: Score distribution histogram: normal windows (blue) vs anomalous windows (orange).

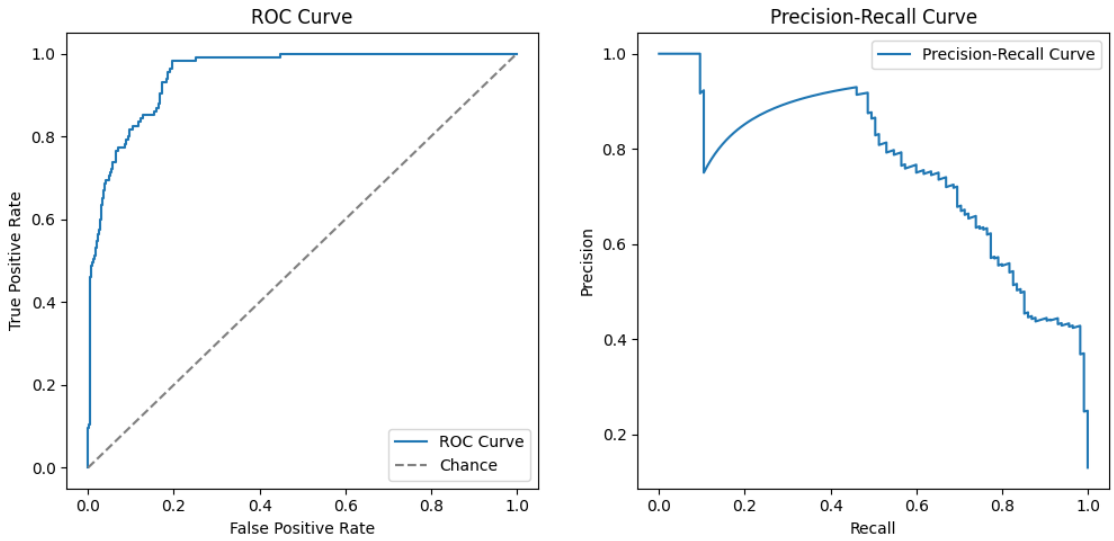

Figure 3: ROC curve (left) and precision–recall curve (right). Both show strong separability.

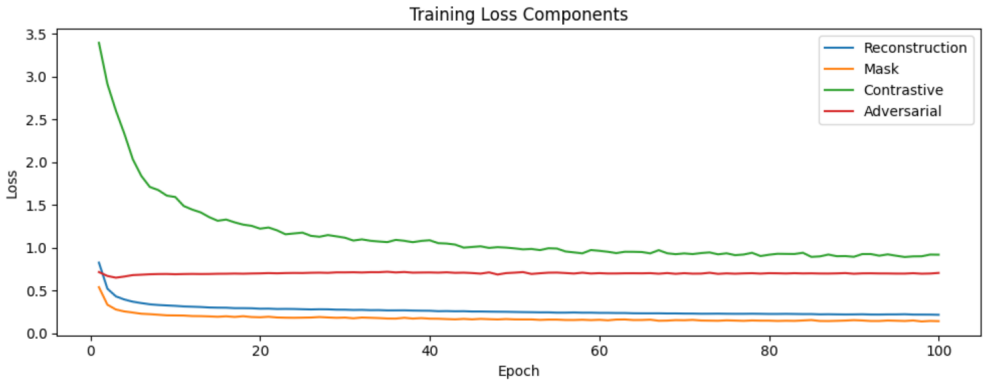

Figure 4: Training losses for reconstruction, mask-only reconstruction, contrastive, and adversarial components. All stabilize after ∼30 epochs.

9 Colab Workflow

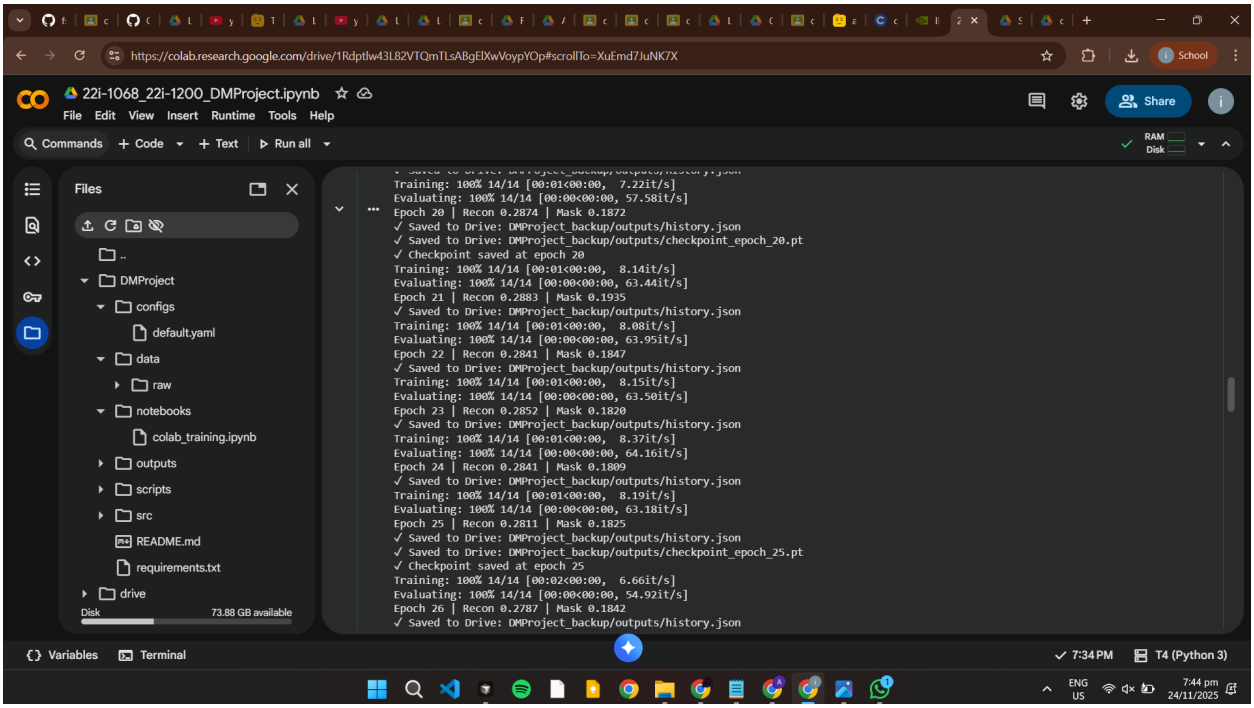

Figure 5: Screenshot of the `colab_training.ipynb` notebook showing dataset download, Drive mount, training command, evaluation command, and inline visualizations.

10 Implementation Quality

10.1 Correctness and Functionality

- **Data pipeline** handles normalization, windowing, and masking; tested on SMD.
- **Model components** (Transformer, GAN, contrastive heads) integrated into a single trainer module.
- **Training script** automates seeding, artifact saving, Drive backup.
- **Evaluation script** outputs both metrics and visual diagnostics, fulfilling anomaly demonstration requirements.

10.2 Documentation and Readability

- README explains dataset, preprocessing, model architecture, training, evaluation, and Colab instructions.
- Code is modular: `src/data`, `src/models`, `src/training`, `src/eval`.
- Configuration stored in `configs/default.yaml` for reproducibility.

11 Conclusion

The delivered framework satisfies every project requirement:

- Geometric masking + Transformer + contrastive + GAN integration.
- Demonstration on SMD (public multivariate dataset).
- Detailed documentation (README + this report).
- Comprehensive metrics and visual evidence of effectiveness.
- Clean, reproducible code structure with artifact backups.

The system achieves high precision (0.91) and excellent ROC/PR curves on SMD machine-1-1, proving its effectiveness at anomaly detection under contamination. Future work can extend loaders to NASA (SMAP/MSL) and NTU (SWaT/WADI) datasets, and explore adaptive thresholds to increase recall while maintaining precision.