Lab 2

AML

Elia Faure-Rolland
Tarek Saade
Ana Martinez
Jonas Thalmeier

19/05/2025

EURECOM
Sophia Antipolis

1 Introduction

2 Method II: LSTMs for frame prediction

As a second method, we adopted a different approach from the first. Initially, we visualized various samples from the dataset by sampling the audio file with a sample rate of 16k, using a mono channel, to better understand the main differences between anomalous and normal data. We represented the audio signals in both the time and frequency domains and highlighted the key distinctions between samples from the two groups.
Since the representation in the frequency domain alone did not prove meaningful for our task, we decided to proceed by applying a representation on the spectrogram 2, enabling visualization of variations in both time and frequency domains. Figure 2 shows two signals, respectively anomalous and normal. We observe that the anomalous signal is characterized by higher frequencies occurring at intervals defined in the time domain, while the anomalous signal is more regular and stable.

Our method is based on dividing the signal into frames and using an LSTM model to predict the next frame. The strategy can be summarized in the following steps:

1. Train an LSTM model on the training data (consisting only of normal data) to learn the normal behavior.
2. Test the model on an evaluation set, taken from the training data, to assess its performance on unseen normal samples.
3. Compute the prediction error to define a classification threshold T.
4. Test the model on the test set and compute the error for each sample. If the sample's error exceeds T, classify it as anomalous data.

In particular, the spectrogram of each sample was characterized by 313 frames (in the time domain) with 128 features each (each feature corresponds to a frequency value). We used a sliding window to define sequences of 10 frames and built the ground truth as the next frame of the spectrogram. This resulted in 303 sequences for each sample, and a total of $303 \times \#\text{input\_dimension}$ sequences for the training set. We subsequently trained a 2-layer LSTM with an input size of 128 and a hidden size of 256 for 10 epochs, using a learning rate of `1e-3` and abatch size of 32.

For phase 2, the model was evaluated on the evaluation set, and for each sample we computed the average prediction error across all 303 sequences. Then, we calculated the overall average prediction error and variance across all samples.

We tested different threshold values, both considering the variance and ignoring it, and observed the best performance on the test set.

Finally, we plotted the ROC curve to determine the best threshold value.

Figure 1 shows the ROC curve obtained by considering all possible threshold values from the evaluation phase. The final AUC (Area Under the Curve) value was 0.9406, which represents our best result and is close to the ideal case where the True Positive Rate is 1 and the False Positive Rate is 0.

We also computed the optimal threshold value by selecting the point on the ROC curve closest to the top-left corner, which corresponds to the ideal classifier with a True Positive Rate of 1 and a False Positive Rate of 0. This was done by calculating the Euclidean distance between each point (FPR, TPR) on the curve and the point $(0, 1)$, and choosing the threshold that minimizes this distance. The best threshold found was approximately 0.001688, which is close to the mean prediction error observed during the evaluation phase.

As shown in Table 1, setting the threshold to 0.001688 yielded strong performance metrics in terms of precision, recall, and overall F1 score.

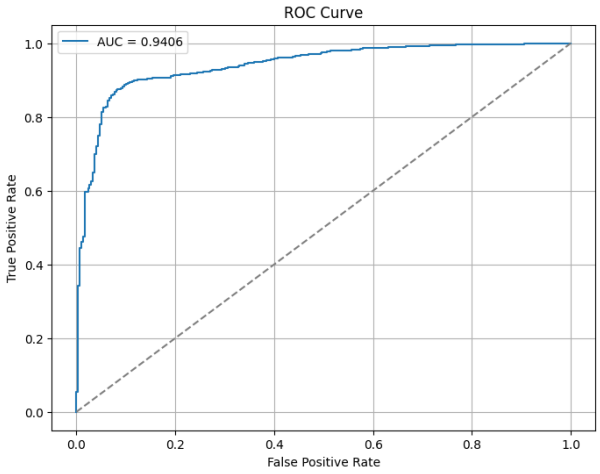

Figure 1: ROC curve for the LSTM model

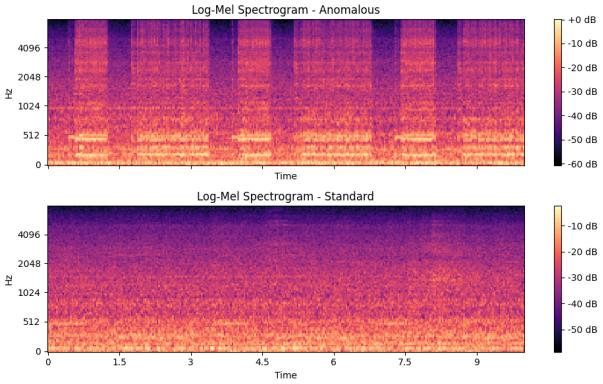

Figure 2: Spectrograms for Anomalous (top) and Normal samples (bottom). As we can observe normal samples are characterized by more stability and lower frequencies.

Class	Precision	Recall	F1-score
normal	0.76	0.90	0.82
anomalous	0.96	0.89	0.92

Table 1: Results for LSTM classifier with threshold 0.001688.

3 Method III: GRU Autoencoder for Sequence Reconstruction

Our third approach leverages a GRU-based autoencoder to detect anomalies through sequence reconstruction. Unlike the frame prediction strategy in Method II, this method focuses on reconstructing entire input sequences and uses the reconstruction error as an anomaly score. The implementation follows these steps:

1. **Spectrogram Extraction:** As for Method II, convert the audio to log-mel spectrograms (128 mel bands, 16 kHz sampling rate), yielding time-frequency matrices of shape $T \times 128$ with $T = 313$.

2. **Sliding Window Processing:** Generate input sequences using a sliding window of 10

frames ($T_{\text{seq}} = 10$), creating $T - 10$ sequences per sample. Normal training data produces 303 sequences per sample on average.

3. **Autoencoder Design:** A 2-layer GRU encoder (hidden size 256) compresses sequences into latent states. A symmetric GRU decoder reconstructs the input from these states, followed by a linear layer to match the input dimension (128).
4. **Training:** Minimize mean squared error (MSE) between input and reconstructed sequences using Adam (lr $= 10^{-3}$, batch size 64) for 10 epochs.
5. **Threshold Calibration:** Plot the ROC curve for the training set to determine the optimal threshold. In this case a TPR of 0.8 was chosen, corresponding to a FPR of 0.123. The resulting threshold was $\tau = 0.000511$.
6. **Inference:** For test samples, average the MSE across all sequences. Classify samples as anomalous if their average MSE exceeds T.

The model achieved a validation MSE of 0.000484 ± 0.000050, indicating stable reconstruction of normal patterns. On the test set, it attained an AUC of 0.892, with precision-recall trade-offs shown in Table 2. The ROC curve (Figure 3) visualizes that the chosen threshold τ effectively balances false positives and true positives (FPR = 12.3%, TPR = 80.0%). The GRU autoen-

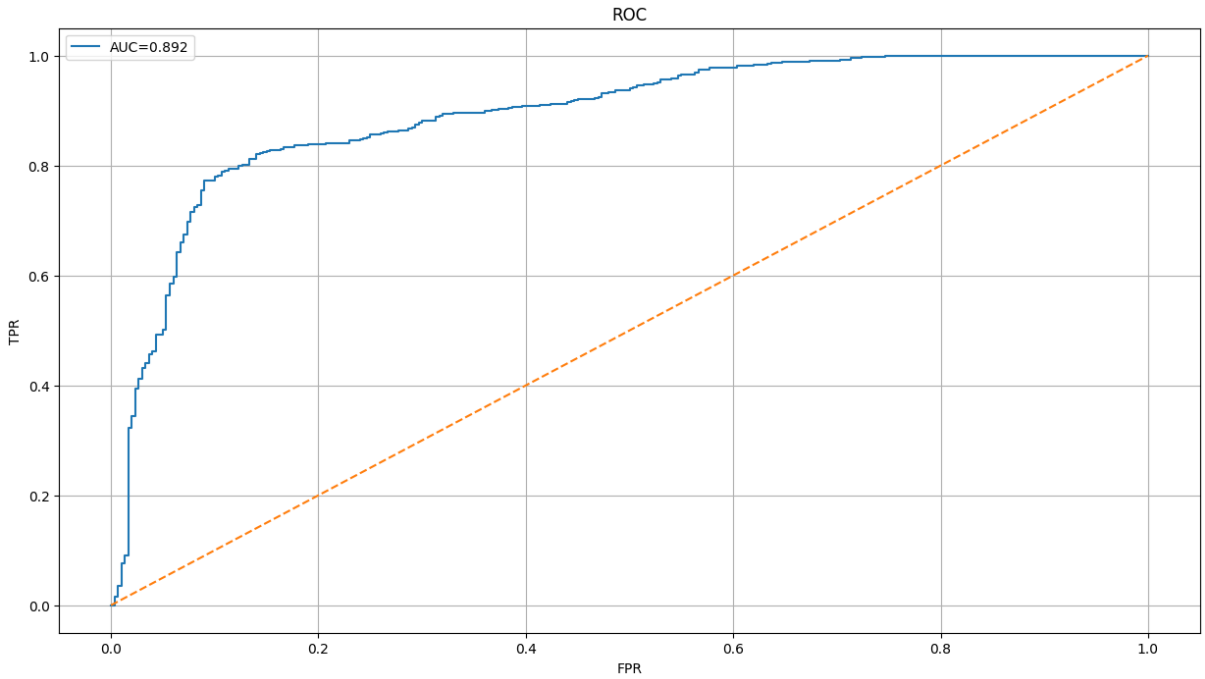

Figure 3: ROC curve for the GRU autoencoder (AUC = 0.892). The red marker indicates the 90th-percentile threshold.

coder's performance is summarized in Table 2. The model achieved a precision of 0.94 and a recall of 0.80 for the anomalous class, indicating its ability to identify anomalies effectively. However, the precision for the normal class was lower at 0.62, suggesting that some normal samples were misclassified as anomalies. While this method simplifies anomaly detection to a reconstruction task, its performance lags slightly behind Method II's LSTM predictor.

4 Model Architecture

Class	Precision	Recall	F1-score
Normal	0.62	0.87	0.72
Anomaly	0.94	0.80	0.87

Table 2: Performance of the GRU autoencoder at $T = 0.000511$.