Anomaly detection in radar data using VAE

PAMI Practice Summer Semester 2025
Prof. Dr. Visvanathan Ramesh
Supervisor: Konrad Wartke
By: Mahmoud Alhag Ali
20.10.2025

Agenda

1. Introduction
2. Solution Idea
3. Implementation
4. Result

Introduction

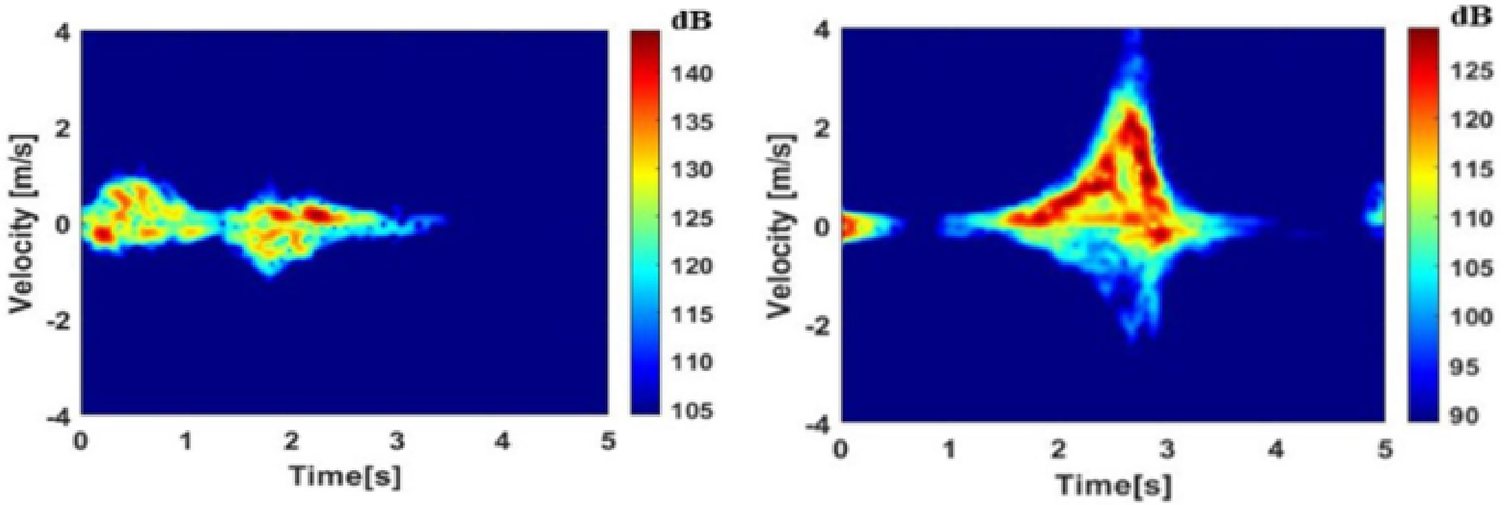
The anomaly detection system can help elderly patients by detecting them falling down. We have a variety of activities captured by radar sensors and we are trying to fall events in processed radar data. Using Variational Auto Encoders we can learn a model of "normal activity" and classify activities as anomalies according to the reconstruction error. The encoder / decoder can use prior knowledge on the expected data structure for better kernel design.

Pipline

1.radar
2.pre-processing
3.probabilistic-modeling
4.decision
5.threshold
6.alert

The Training Data

- Convert the radar data to Images

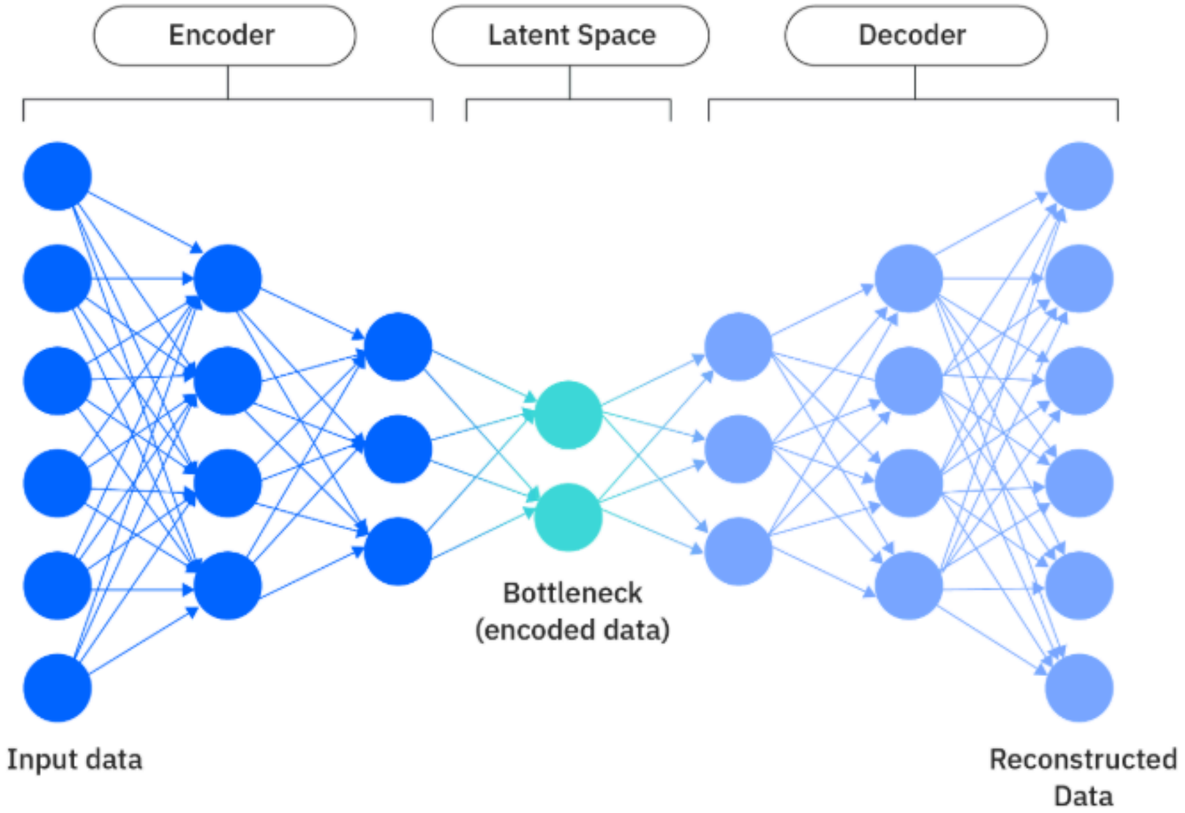


Time-velocity maps showing a single human sitting down (left) and falling (right) (from Li 2023)

Solution Idea

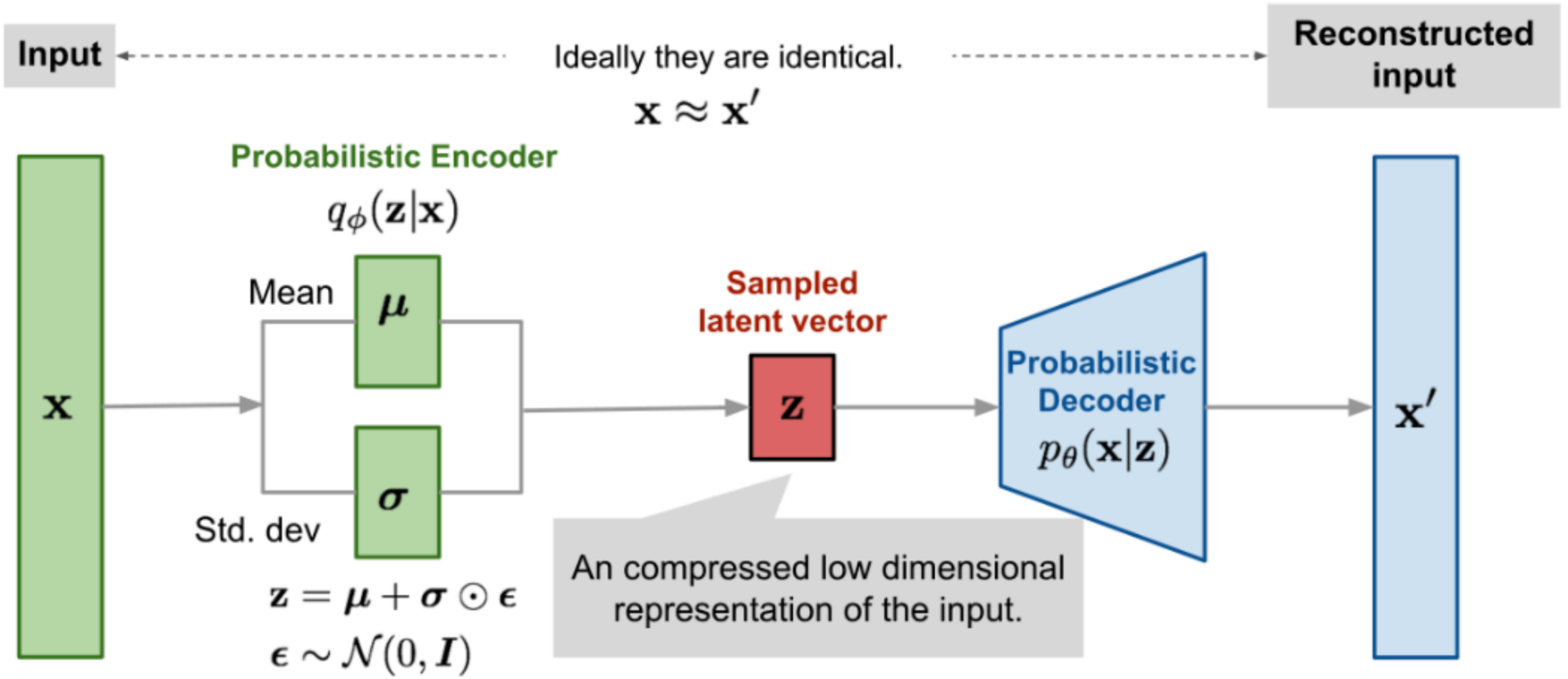
- Convert the radar data to Images
- Use VAE for anomaly detection

Autoencoder



Source:https://www.ibm.com/think/topics/variational-autoencoder

Variational Autoencoder



Source:https://lilianweng.github.io/posts/2018-08-12-vae/

Implementation

- An, J., & Cho, S. (2015). Variational Autoencoder based Anomaly Detection using Reconstruction Probability.
- Its Implementation from Michedev (https://github.com/Michedev)

Implementation

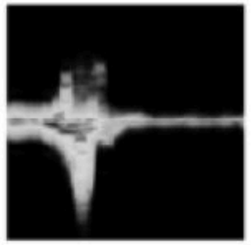
- Frameworks/Libraries: Python: PyTorch, NumPy, Seaborn
- Model architecture: 5 layers for encoder and decoder
- Training dataset: 5844
- Testing dataset: 1125
- Batchsize: 32
- Epoch: 15
- Lernrate: 1e-3

Implementation

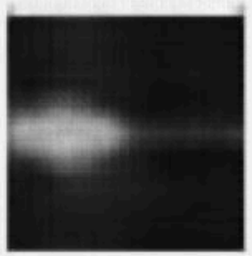
```python
def make_encoder(self, input_channels: int, latent_size: int, image_size: int):
    # Convolutional Encoder architecture
    final_spatial_dim = image_size // (2**4) # = 8 for image_size=128
    final_conv_output_channels = 128 # Output channels for the last conv layer before flattening

    return nn.Sequential(
        # Input: (batch_size, input_channels, image_size, image_size) e.g., (B, 1, 128, 128)
        nn.Conv2d(input_channels, 16, kernel_size=4, stride=2, padding=1), # -> (B, 16, 64, 64)
        nn.ReLU(),
        nn.Conv2d(16, 32, kernel_size=4, stride=2, padding=1), # -> (B, 32, 32, 32)
        nn.ReLU(),
        nn.Conv2d(32, 64, kernel_size=4, stride=2, padding=1), # -> (B, 64, 16, 16)
        nn.ReLU(),
        nn.Conv2d(64, final_conv_output_channels, kernel_size=4, stride=2, padding=1), # -> (B, 128, 8, 8)
        nn.ReLU(),
        nn.Flatten(), # -> (B, 128*8*8) = (B, 8192)
        nn.Linear(final_conv_output_channels * final_spatial_dim * final_spatial_dim, latent_size * 2) # -> (B, latent_size * 2)
    )
```

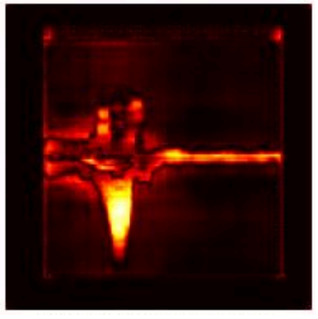
Original (Anomalie)
Batch 69, Sample 12



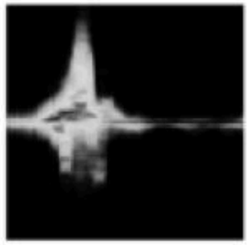
Rekonstruktion
Prob: 12.309555



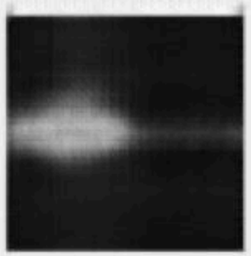
Fehler (|Original - Rekon|)
Mean: 0.087180



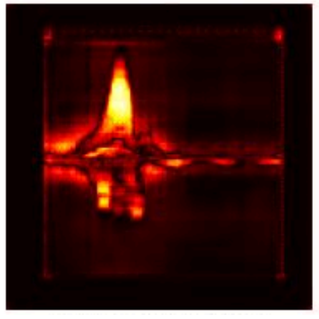
Original (Anomalie)
Batch 69, Sample 13



Rekonstruktion
Prob: 12.250659

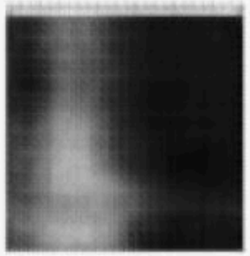


Fehler (|Original - Rekon|)
Mean: 0.085880

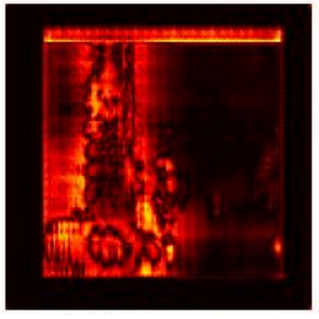


Original (Anomalie)
Batch 69, Sample 14

Rekonstruktion
Prob: 11.355322

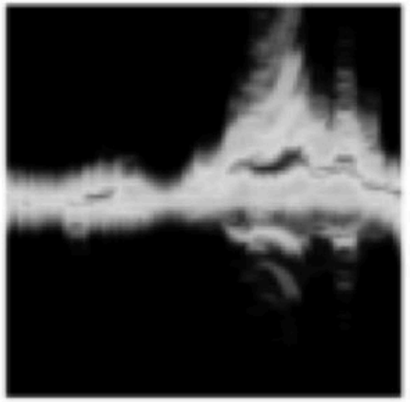


Fehler (|Original - Rekon|)
Mean: 0.121943

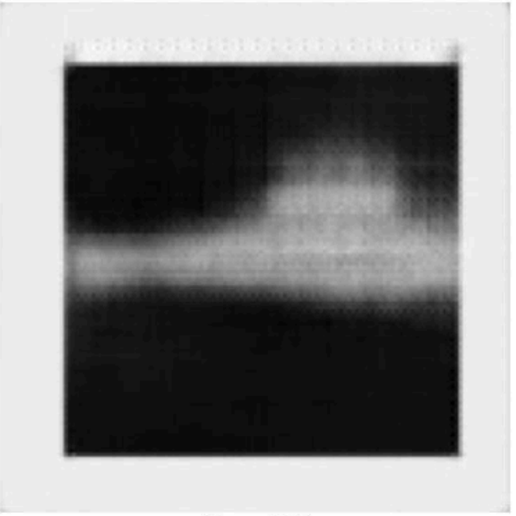


VAE Anomaly Detection: Normal vs Anomaly Comparison

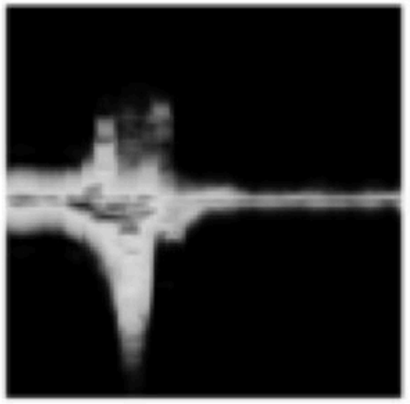
Normal Activity
(Original)



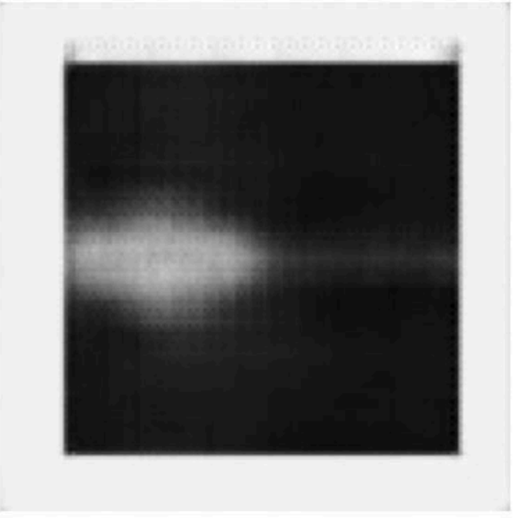
Normal Activity
(Reconstruction)



Fall Activity
(Original)



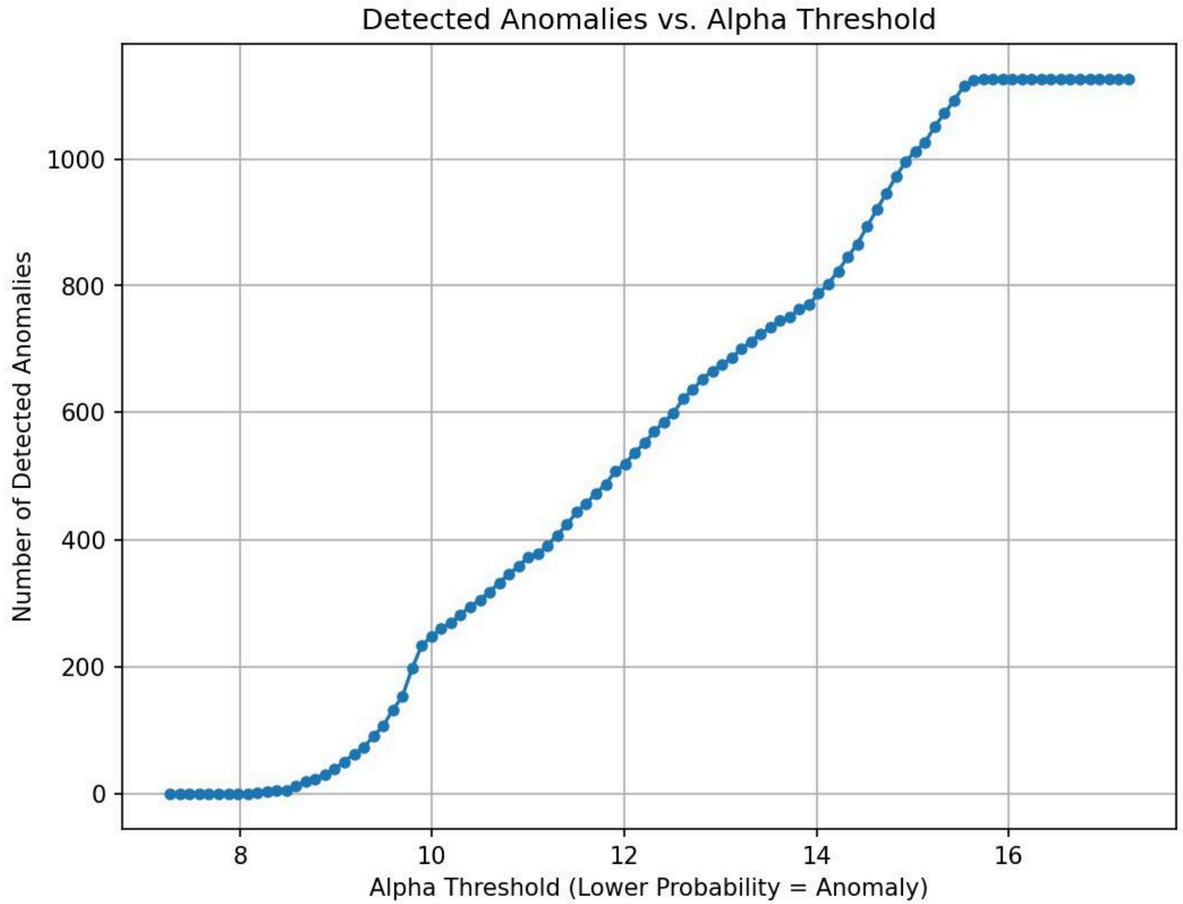
Fall Activity
(Reconstruction)



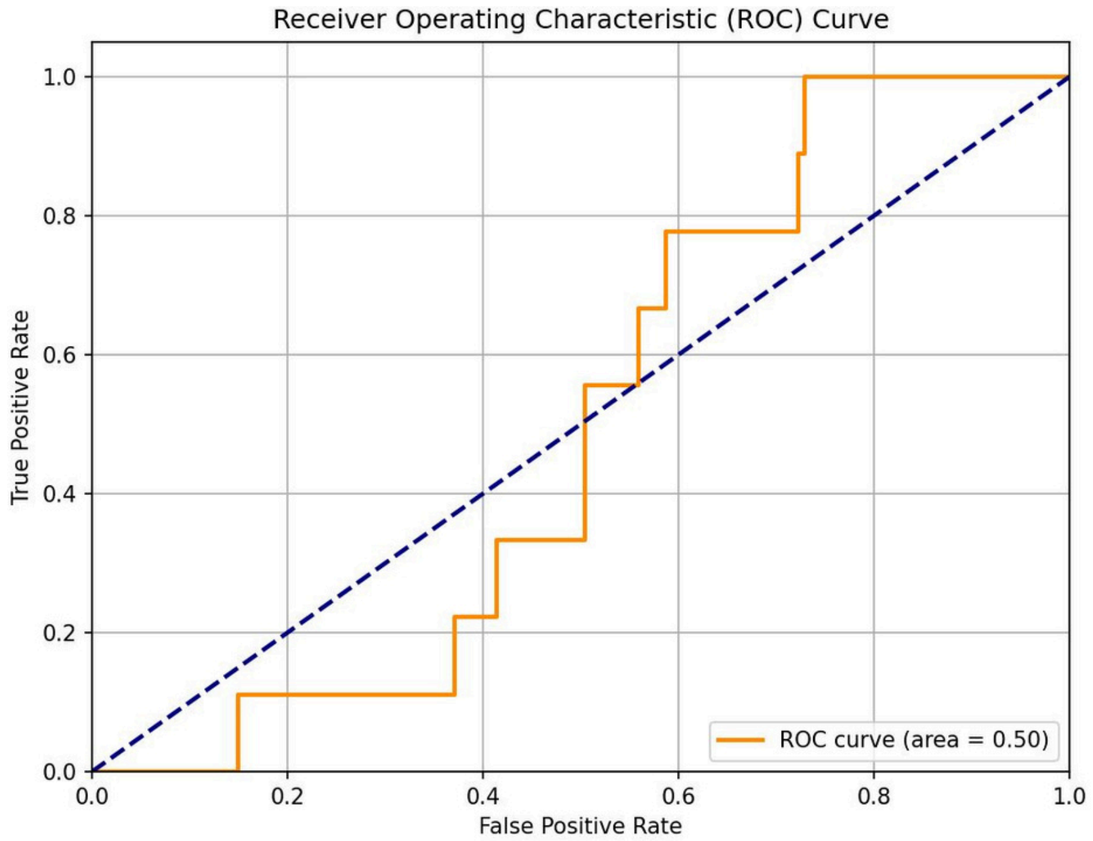
Normal Error: 0.1

Anomaly Error: 0.1

Results

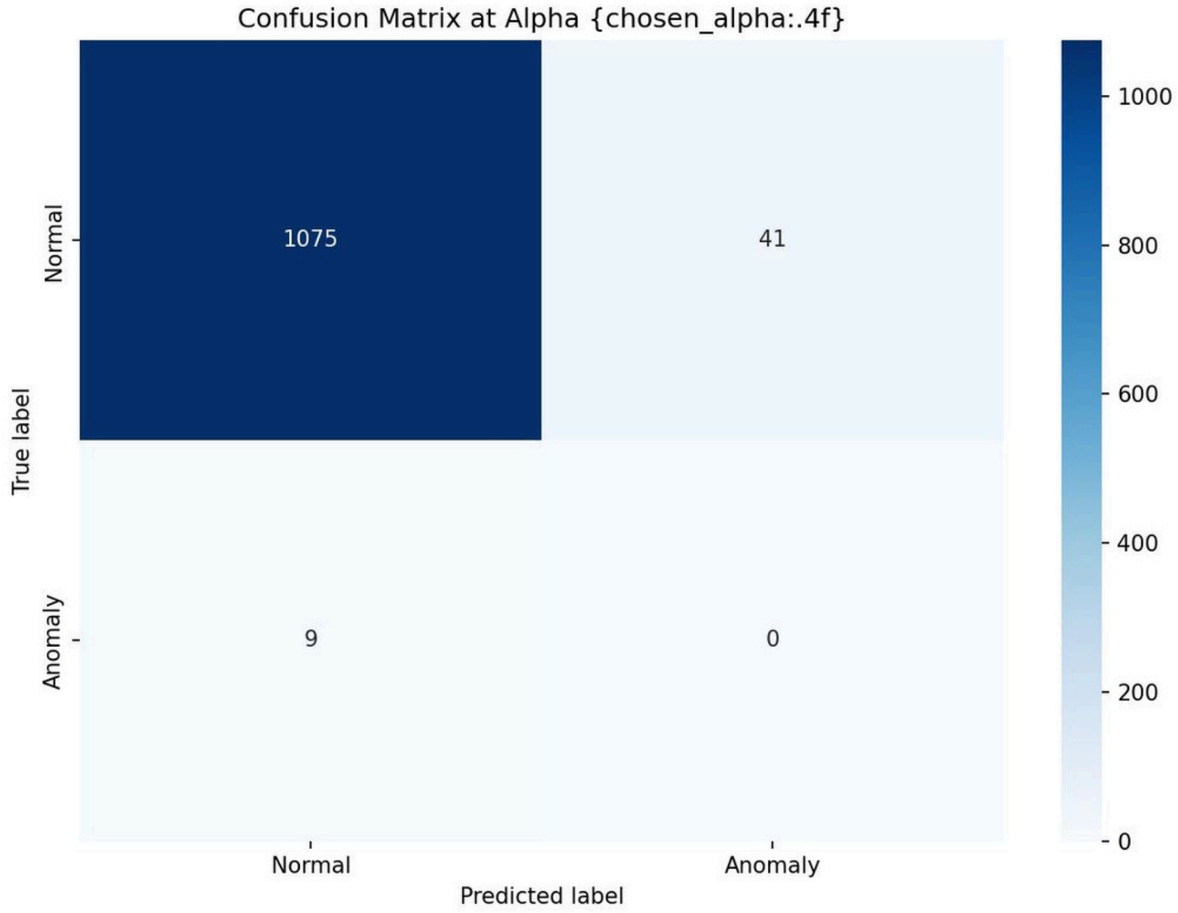

Detected Anomalies vs. Alpha Threshold
Number of Detected Anomalies
0
200
400
600
800
1000
8
10
12
14
16
Alpha Threshold (Lower Probability = Anomaly)

Results

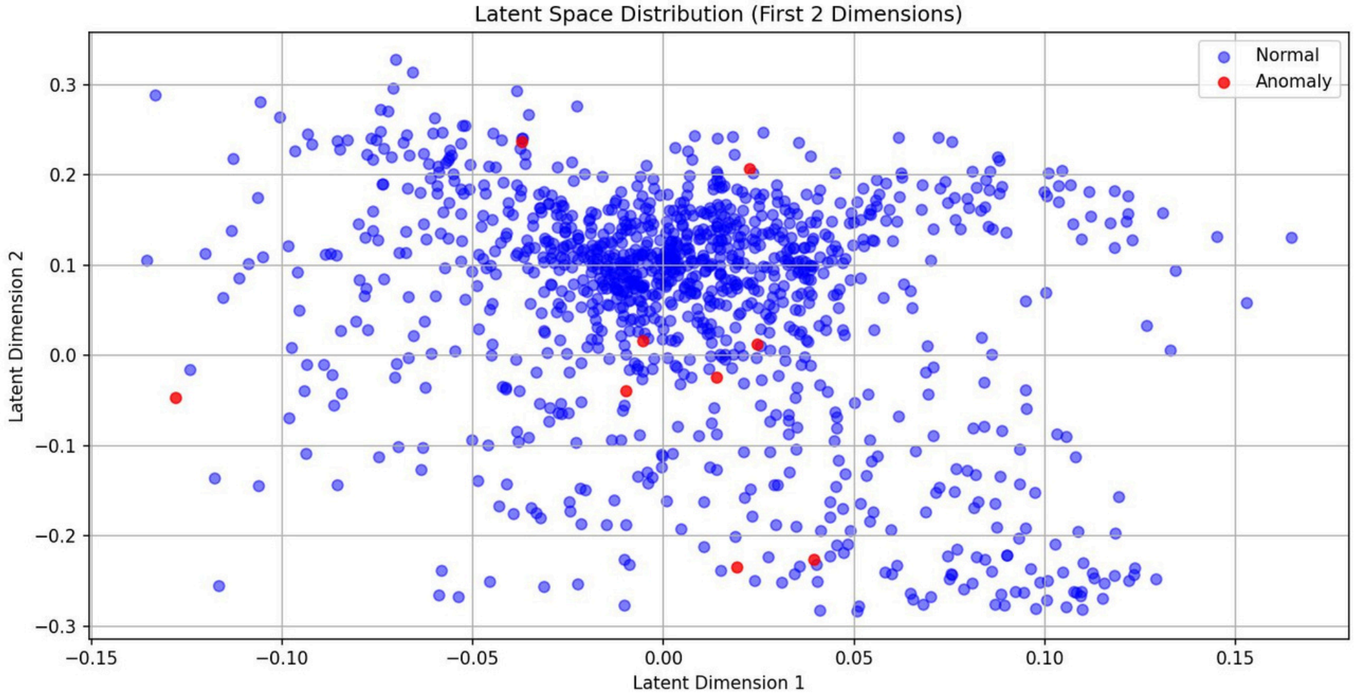

Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.50)

Results

Alpha (Threshold) = 9.0



Results


Latent Space Distribution (First 2 Dimensions)
Normal
Anomaly
Latent Dimension 2
Latent Dimension 1
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15

Results

```
Classification Report:
              precision    recall  f1-score   support

      Normal       0.99      0.96      0.98      1116
     Anomaly       0.00      0.00      0.00         9

    accuracy                           0.96      1125
   macro avg       0.50      0.48      0.49      1125
weighted avg       0.98      0.96      0.97      1125
```

Thank You