OUR APPROACH FOR ANOMALY DETECTION

PREPROCESSING

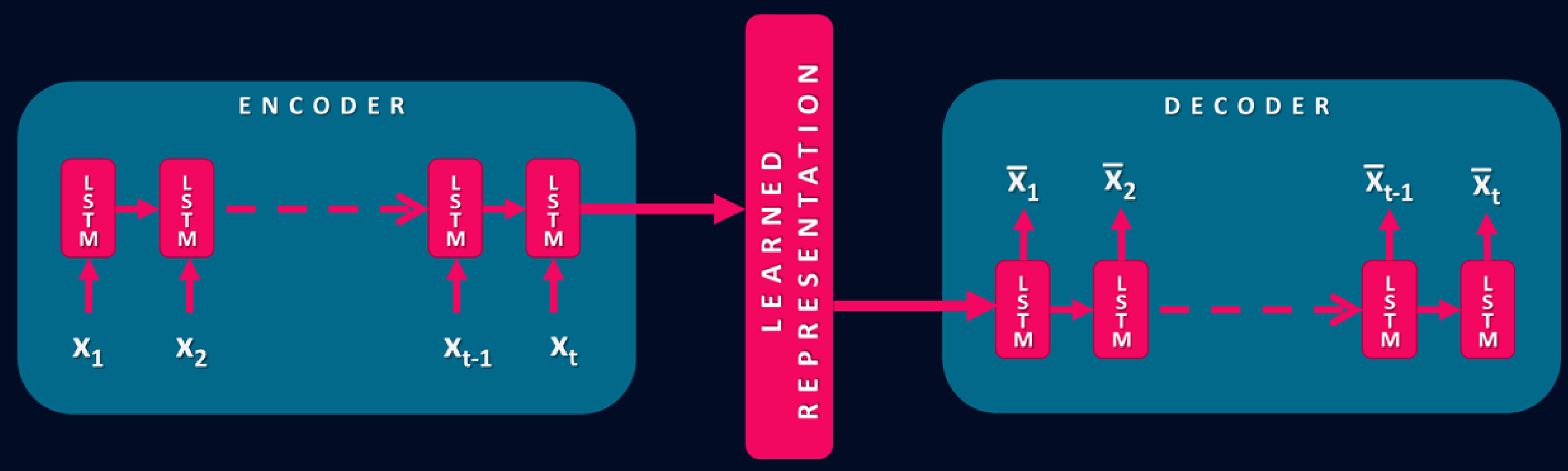
We applied pretty classical preprocessing such as one-hot encoding, MinMaxScaling and PCA.

- One-Hot encoding on categorical variables
- XGBoost to predict ‘p35’
- Normalized data.
- PCA to reduce dimensionality from 42 to 12 (99% of the variance)

1 RECONSTRUCTION ERROR

We Trained autoencoders to reconstruct normal data well and analyzed reconstruction errors on anomalous data.

We built an LSTM Autoencoder structure that was trained to reconstruct normal data expecting that the reconstruction error would increase on anomalous data, whish was unfortunately not the case...

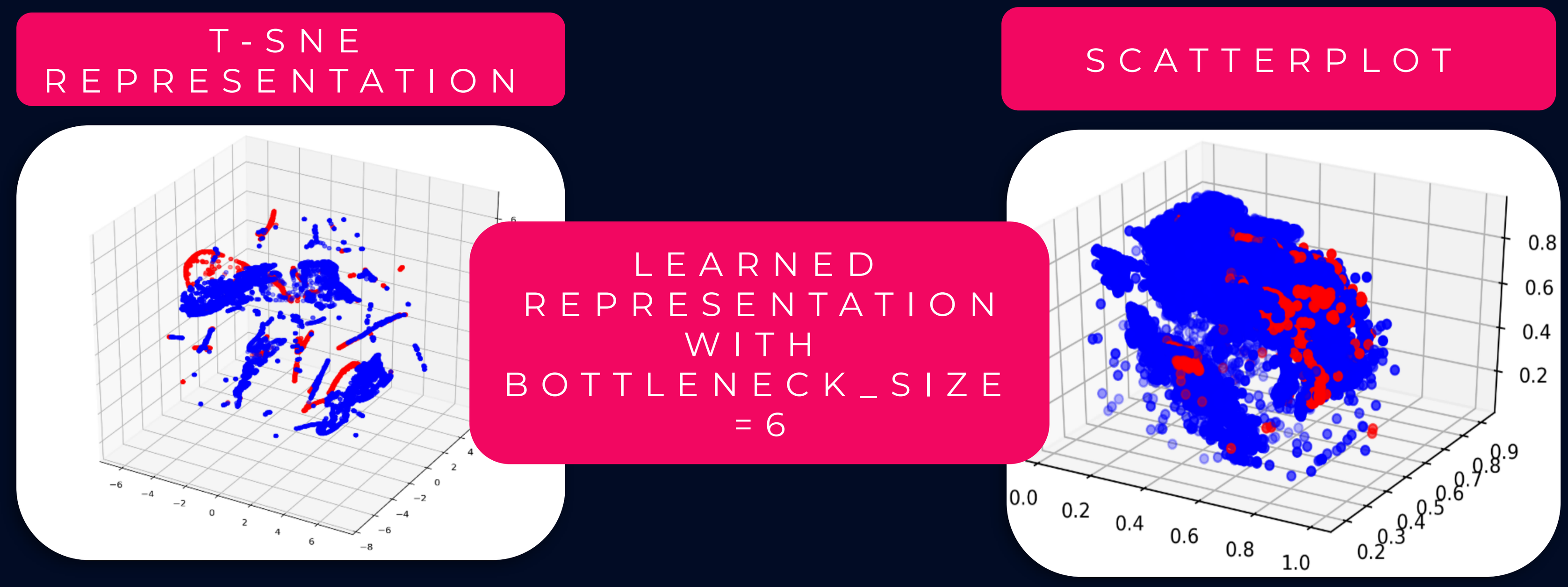


Reconstruction error is defined as:

2 LATENT FEATURES ANALYSIS

We analyzed the autoencoders’ bottleneck features and tried to find how to separate between normal data and anomalies

We used the learned latent representation to try to discriminate anomalies from normal data



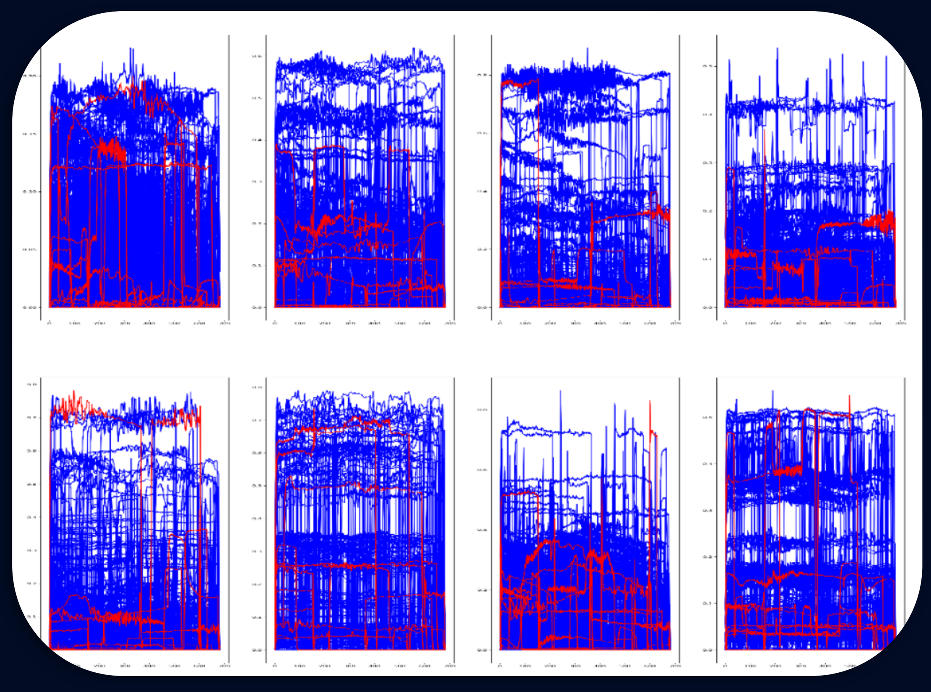
3 DISTANCE VARIATIONS ALONG TIMESTEPS

We looked at the distance variations between latent feature points from t to t+1 to see if anomalies could be detected.

Aggregating learned representation by timesteps, we tried to identify abnormal changes along time compared to normal data

NORMAL DATA DEVIATION FROM NORMAL DATA MEAN FROM t TO t+1

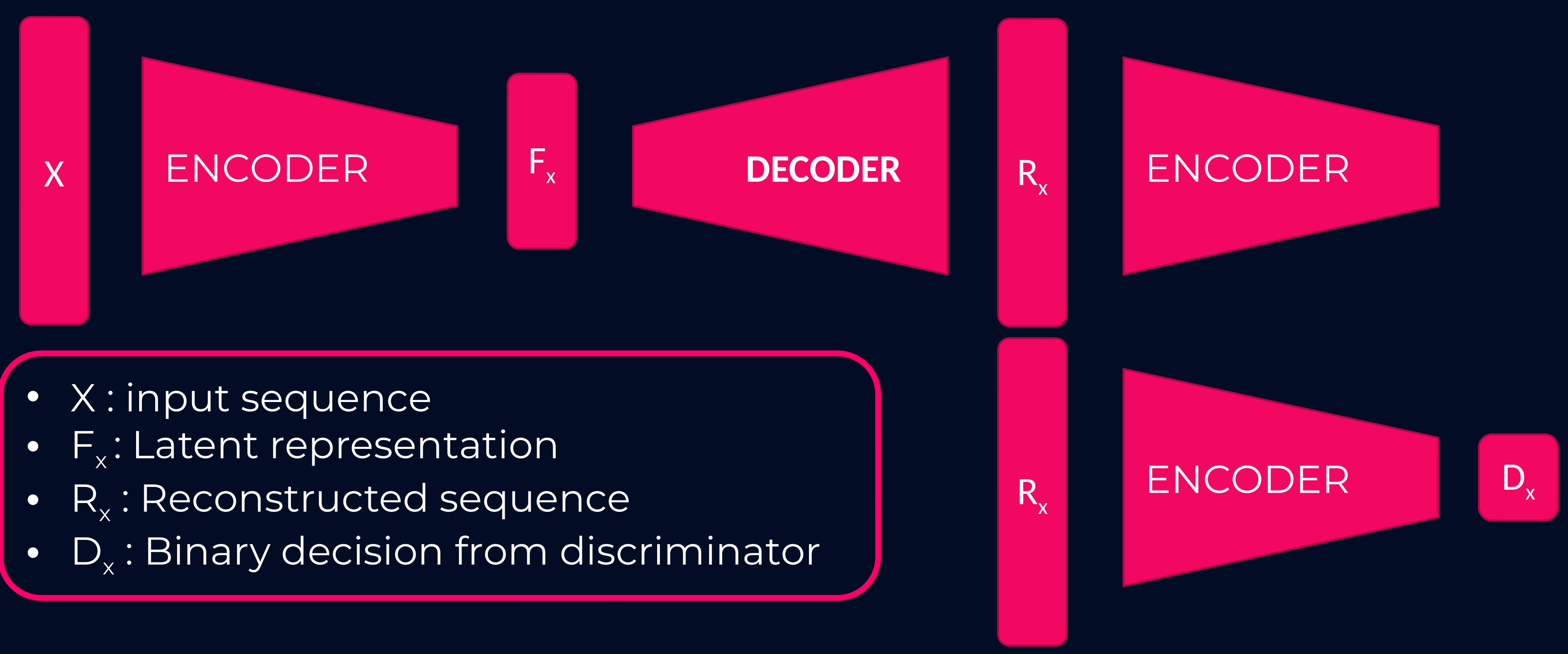
ABNORMAL DATA DEVIATION FROM NORMAL DATA MEAN FROM t TO t+1



4 ADVERSARIAL TRAINING

We have built an encoder-decoder-encoder with an adversarial encoder and trained it on normal data.

As last approach, we built an encoder-decoder-encoder architecture with an adversarial encoder as discriminator:



- X : input sequence
- F_x: Latent representation
- R_x : Reconstructed sequence
- D_x : Binary decision from discriminator