中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

中国科学院网络数据科学与技术重点实验室
Key Laboratory of Network Data Science & Technology, CAS

IJCAI 2019 MACAO

BeatGAN:
Anomalous Rhythm Detection using Adversarially Generated Time Series

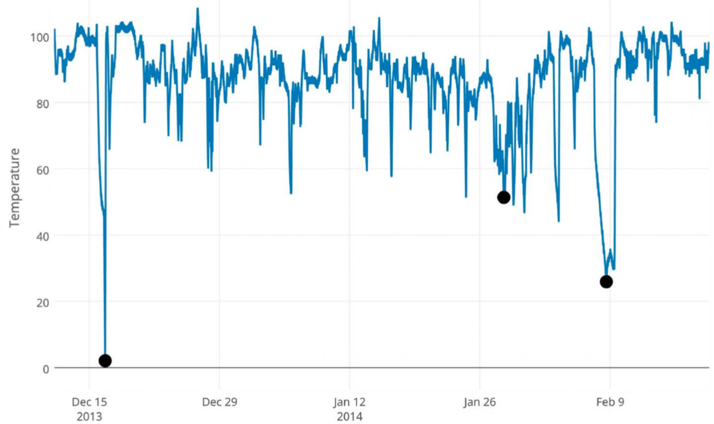
B. Zhou, S Liu, B Hooi, X. Cheng, J. Yang

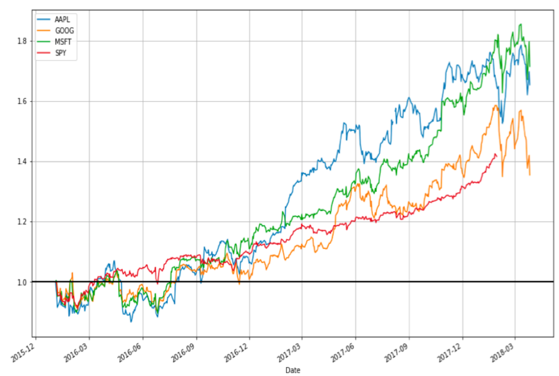
Institute of Computing Technology ICT, CAS

University of Chinese Academy of Sciences, UCAS

Background

- Time series anomaly detection is important in many application





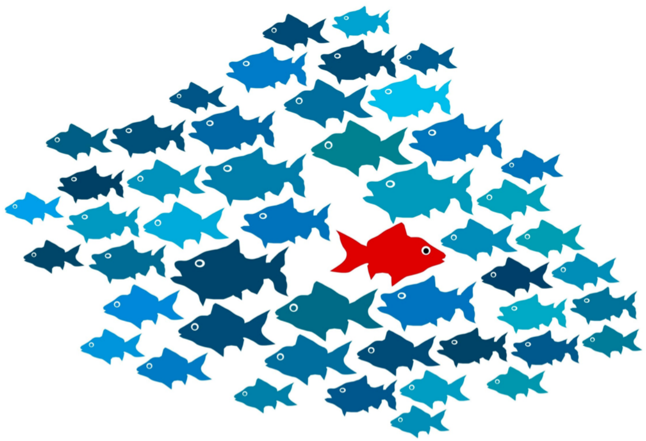
Medical Industry Finance

Background

- **Problem**: What is an anomaly ?

"Observation which deviates so much from other observations as to arouse suspicion it was generated by a different mechanism"—Hawkins(1980)

- Outlier
- Unbalance
- Unknown category



Background

Previous studies focus on

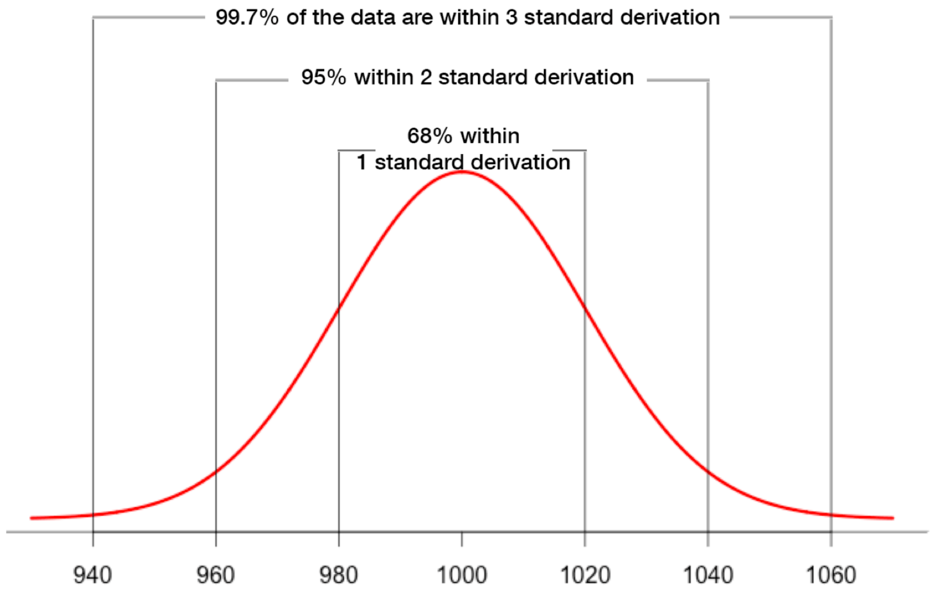
- Classification-based methods, which needs large amount of labels.
- Vocabulary-based methods, which is not accurate for anomaly detection.

Question:
Given a collection of multivariate time series with most normal class.
How can we detect anomalies effectively by an unsupervised method?

Basic Idea

Anomaly(outlier) is the data point that deviates significantly from the normal.

- If we know the data distribution of the normal
- That rare samples can be view as anomalies.
  - 3σ rule of thumb in normal distribution.



Reconstruction-based methods

- The model learns the feature of the normal.
- The model learns the distribution of the normal.
- Then anomaly score of test sample x can be defined as:

$$A(x) = ||x - G(x)||_2$$

$G(x)$ is the reconstructed data of input x

Reconstruction-based methods

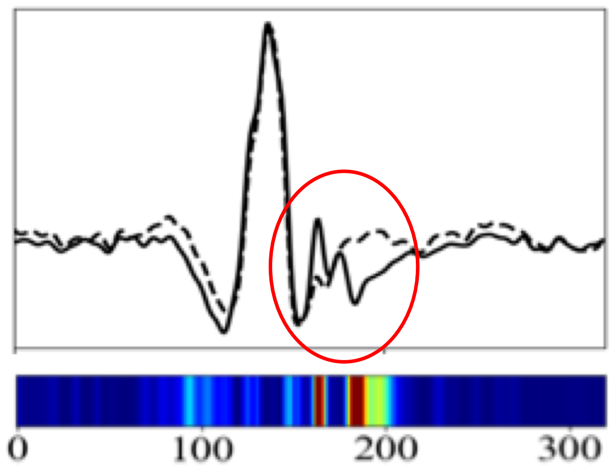
- Existing reconstruction-based models:
- SVD [Shah *et al.*, 2014] :
  - Apply linear functions for reconstruction, which is not suitable for real-world data.
- AutoEncoder [An and Cho, 2015] :
  - Lack proper regularization for the model which leads to overfitting and low accuracy.
- AnoGAN [Schlegl *et al.*, 2017] :
  - The generator can learn the data distribution well but lacks the encoder which is time consuming in test phase.

Road Map

- Background
- Proposed Model: **BeatGAN**
  - Problem definition <<
  - Proposed method
- Experiments
- Conclusion

Problem definition

- ***Given*** *a collection of multivariate time series beats* $X = \{x_i, i = 1,2, \ldots\}$ *with most of beats in the normal class*
  - ***Detect*** *anomalous beats* x *in a collection of unseen time series,*
  - ***Such that*** *they deviate significantly from the recon-structed time series, and can pinpoint anomalous time ticks in x for explanation.*



Proposed model

- General reconstruct-based Framework
  - **Input**: sample x
  - **Output**: reconstructed sample $x' = G(x)$
  - **Loss function**: R(G) is the regularization terms for model G

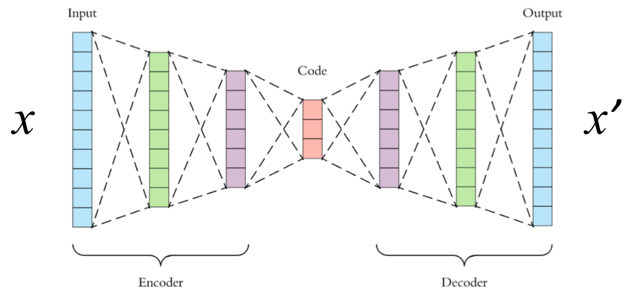
$$
\begin{aligned}
L &= ||X - G(X)||_2 + R(G) \\
&= \sum ||x - G(x)||_2 + R(G)
\end{aligned}
$$

  - Anomalousness score

$$
A(x) = \|x - x'\|
$$

Proposed model

- Model the normal data by reconstruct-based method
  - AutoEncoder to reconstruct the input data
    - Compress the data to capture the true feature

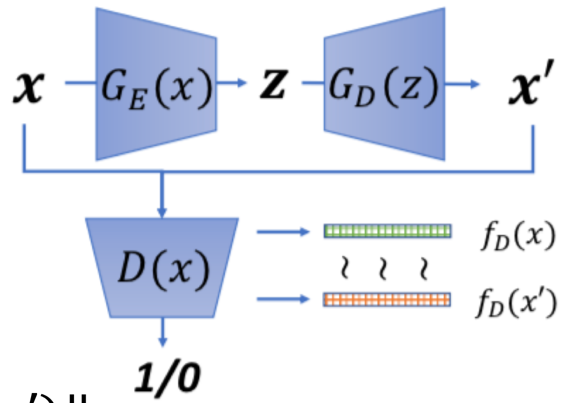


  - Adversarial training to learn the data distribution
    - Guide the Generator(AE) to produce the data that satisfy the normal data's distribution.

Proposed model

- Combine Autoencoder and GAN

$$L = \sum \|x - G_D(G_E(x))\| + R(G)$$



- Two regularization terms
  - adversarial (model level)

$$R_1(G) = \sum_x D\big(x, G_D(z)\big) = \lambda \sum_x \|f_D(x) - f_D(x')\|$$

  - time warping (data level)

$$R_2(G) = \sum_{\hat{x}} \|\hat{x} - G_D(G_E(\hat{x}))\| \ , \hat{x} = DTW(x)$$

DTW: dynamic time waring to augment data

Proposed model

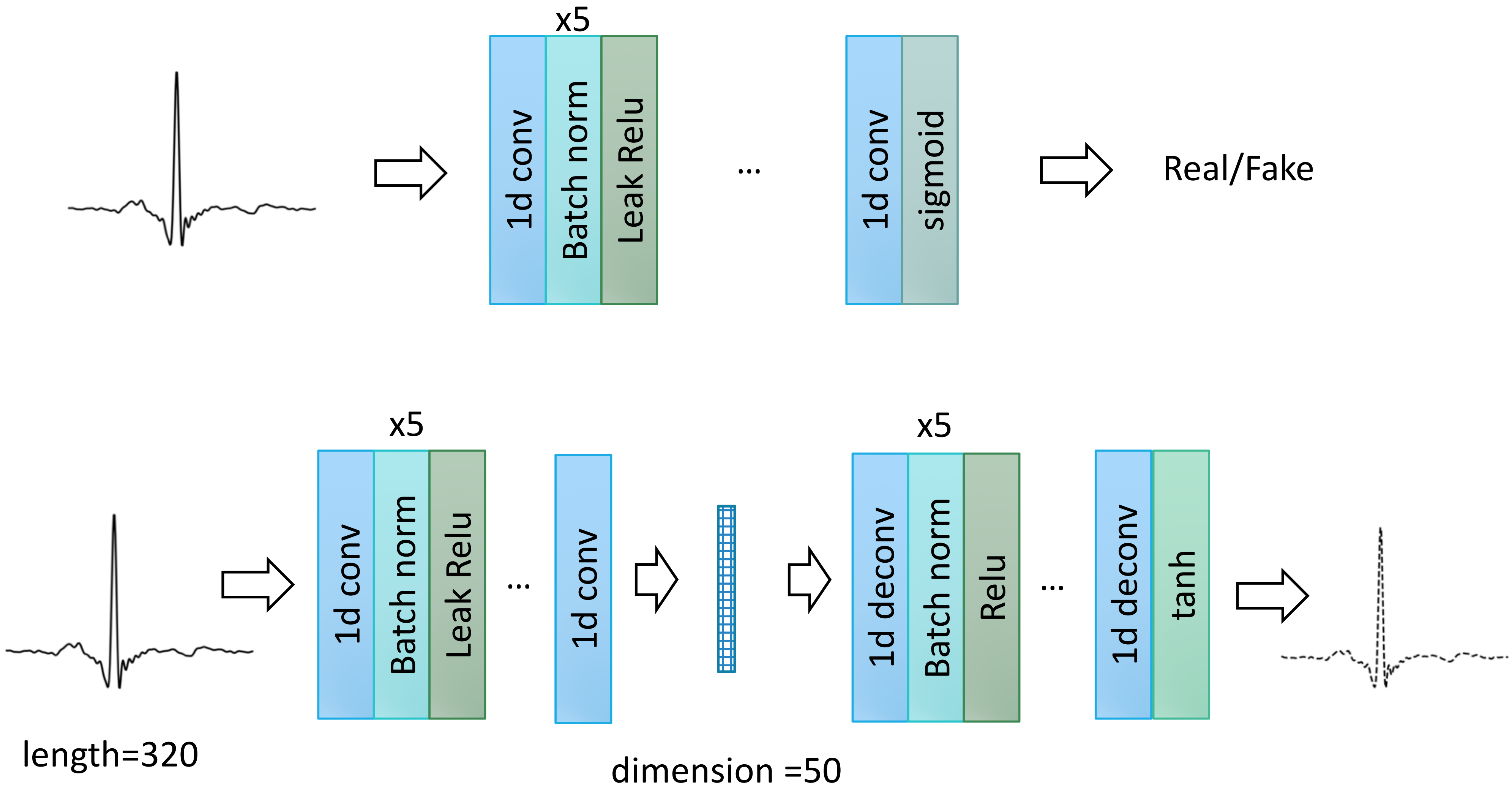
- Loss function for D
  - BCE (Binary Cross Entropy):
    - distinguish the generated data from real data

$$L_D = \frac{1}{N}\sum_{\mathrm{i}} \log D(x_i) + \log(1 - D(x_i{}'))$$

- Loss function for G
  - Reconstruct error
  - Feature match error

$$L_G = \frac{1}{N}\sum_{i} \|x_i - x_i'\|_2 + \lambda \|f_D(x_i) - f_D(x_i')\|_2$$

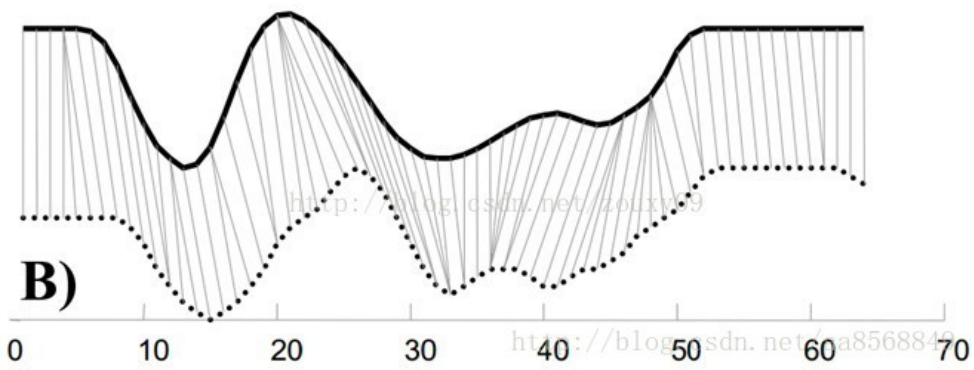
Model structure



Why time warping regularization?

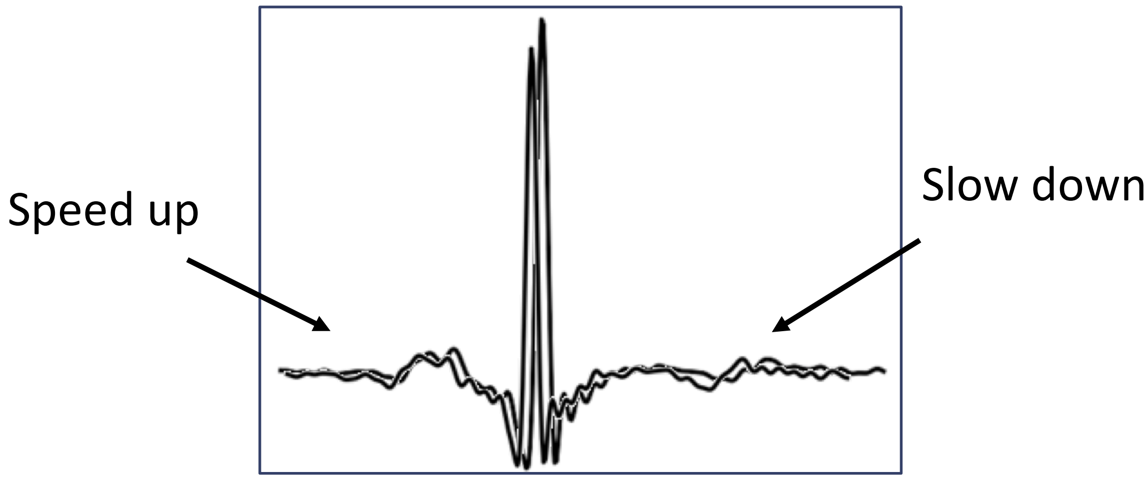
- ECG naturally and slightly speed up or slow down, similarity by ED (Euclidean Distance) is not accurate.

- DTW (dynamic time warping)
  - Better than ED
  - $O(n^2)$ complexity
  - not differentiable

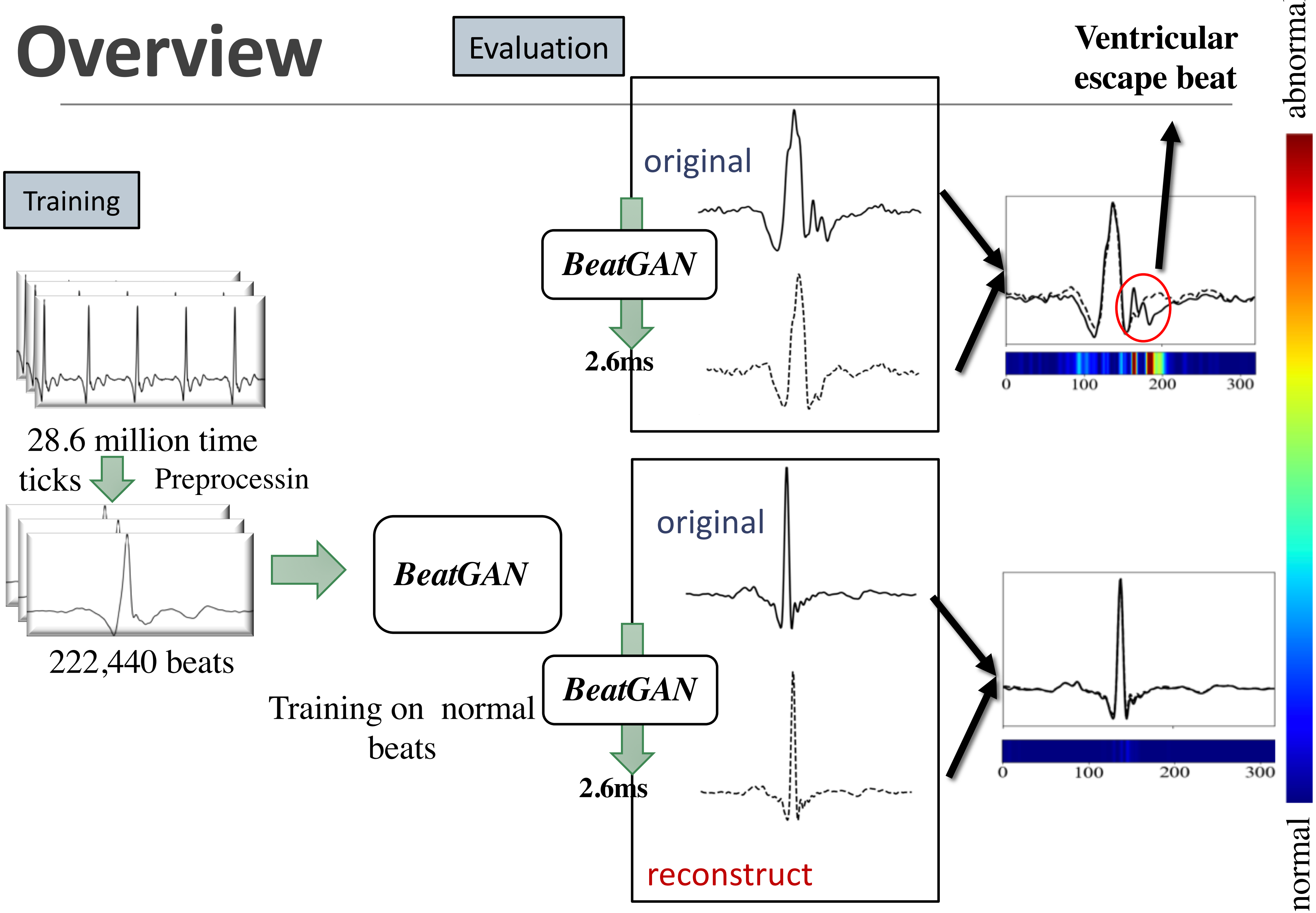


Augmentation method

- Invariant transform for data augmentation
  - randomly pick a small number *k*
  - Pick *k* time ticks to "speed up" (delete values)
  - Pick *k* time ticks to "slow down" (insert values)



Overview


Evaluation
Training
28.6 million time ticks
Preprocessin
222,440 beats
BeatGAN
Training on normal beats
original
BeatGAN
2.6ms
original
BeatGAN
2.6ms
reconstruct
Ventricular escape beat
0
100
200
300
0
100
200
300
abnormal
normal

Road Map

- Background
- Proposed Method: BeatGAN
- Experiments <<
- Conclusion

Dataset

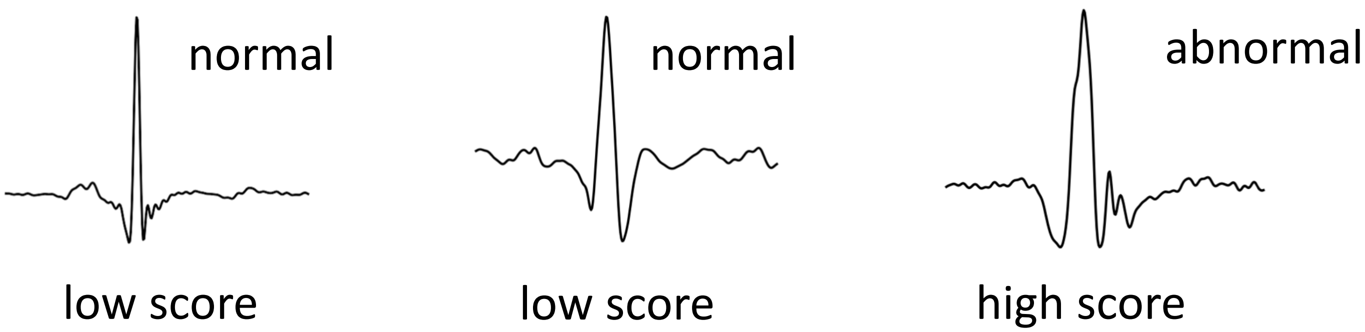
- MIT-BIH dataset:
  - Almost 100,000 beats
  - 28.6 million time ticks
  - 90% normal beats/ 10% abnormal beats

normal normal abnormal

- CMU motion capture
  - time series of walking, running, jumping, hopping
  - Total 10,000 ticks

Evaluation Metric

- Output anomalousness score for each test sample



  - AUC (area under roc curve)
  - AP (area under prc curve)

Results

Data: MIT-BIH nearly 100,000 beats,
28.6 million time ticks
90% normal beats beats

Method	AUC	AP
PCA	0.8164 ± 0.0037	0.6522 ± 0.0061
OCSVM	0.7917 ± 0.0018	0.7588 ± 0.0027
AE	0.8944 ± 0.0128	0.8415 ± 0.0163
VAE	0.8316 ± 0.0025	0.7882 ± 0.0024
AnoGAN	0.8642 ± 0.0100	0.8035 ± 0.0069
Ganomaly	0.9083 ± 0.0122	0.8701 ± 0.0141
BeatGAN	0.9447 ± 0.0053	0.9108 ± 0.0049
BeatGAN$_{aug-3\times}$	$\mathbf{0.9475 \pm 0.0037}$	$\mathbf{0.9143 \pm 0.0047}$
BeatGAN$^{0.1\%}_{aug-3\times}$	0.9425 ± 0.0022	0.8973 ± 0.0042

Table 3: BeatGAN performs the best for anomalous rhythm detection in ECG data. In BeatGAN$_{aug-3\times}$, we augment the training data size to $3\times$ with time warping. In BeatGAN$^{0.1\%}_{aug-3\times}$, we add the 0.1% anomalous time series to the training data for evaluating robustness. 5-fold cross-validations are run, and averaged metrics and std are given.

Results

Data: MIT-BIH nearly 100,000 beats,
28.6 million time ticks
90% normal beats beats

Improved by adversarial regularization

Method	AUC	AP
PCA	0.8164 ± 0.0037	0.6522 ± 0.0061
OCSVM	0.7917 ± 0.0018	0.7588 ± 0.0027
AE	0.8944 ± 0.0128	0.8415 ± 0.0163
VAE	0.8316 ± 0.0025	0.7882 ± 0.0024
AnoGAN	0.8642 ± 0.0100	0.8035 ± 0.0069
Ganomaly	0.9083 ± 0.0122	0.8701 ± 0.0141
BeatGAN	0.9447 ± 0.0053	0.9108 ± 0.0049
$\text{BeatGAN}_{aug-3\times}$	$\mathbf{0.9475 \pm 0.0037}$	$\mathbf{0.9143 \pm 0.0047}$
$\text{BeatGAN}^{0.1\%}_{aug-3\times}$	0.9425 ± 0.0022	0.8973 ± 0.0042

Table 3: BeatGAN performs the best for anomalous rhythm detection in ECG data. In $\text{BeatGAN}_{aug-3\times}$, we augment the training data size to $3\times$ with time warping. In $\text{BeatGAN}^{0.1\%}_{aug-3\times}$, we add the 0.1% anomalous time series to the training data for evaluating robustness. 5-fold cross-validations are run, and averaged metrics and std are given.

Results

Data: MIT-BIH nearly 100,000 beats,
28.6 million time ticks
90% normal beats beats

Method	AUC	AP
PCA	0.8164 ± 0.0037	0.6522 ± 0.0061
OCSVM	0.7917 ± 0.0018	0.7588 ± 0.0027
AE	0.8944 ± 0.0128	0.8415 ± 0.0163
VAE	0.8316 ± 0.0025	0.7882 ± 0.0024
AnoGAN	0.8642 ± 0.0100	0.8035 ± 0.0069
Ganomaly	0.9083 ± 0.0122	0.8701 ± 0.0141
BeatGAN	0.9447 ± 0.0053	0.9108 ± 0.0049
BeatGAN$_{aug-3\times}$	$\mathbf{0.9475 \pm 0.0037}$	$\mathbf{0.9143 \pm 0.0047}$
BeatGAN$^{0.1\%}_{aug-3\times}$	0.9425 ± 0.0022	0.8973 ± 0.0042

Table 3: BeatGAN performs the best for anomalous rhythm detection in ECG data. In BeatGAN$_{aug-3\times}$, we augment the training data size to $3\times$ with time warping. In BeatGAN$^{0.1\%}_{aug-3\times}$, we add the 0.1% anomalous time series to the training data for evaluating robustness. 5-fold cross-validations are run, and averaged metrics and std are given.

Improved by data augmentation

Results

Data: MIT-BIH nearly 100,000 beats,
28.6 million time ticks
90% normal beats beats

Method	AUC	AP
PCA	0.8164 ± 0.0037	0.6522 ± 0.0061
OCSVM	0.7917 ± 0.0018	0.7588 ± 0.0027
AE	0.8944 ± 0.0128	0.8415 ± 0.0163
VAE	0.8316 ± 0.0025	0.7882 ± 0.0024
AnoGAN	0.8642 ± 0.0100	0.8035 ± 0.0069
Ganomaly	0.9083 ± 0.0122	0.8701 ± 0.0141
BeatGAN	0.9447 ± 0.0053	0.9108 ± 0.0049
BeatGAN$_{aug-3\times}$	$\mathbf{0.9475 \pm 0.0037}$	$\mathbf{0.9143 \pm 0.0047}$
BeatGAN$^{0.1\%}_{aug-3\times}$	0.9425 ± 0.0022	0.8973 ± 0.0042

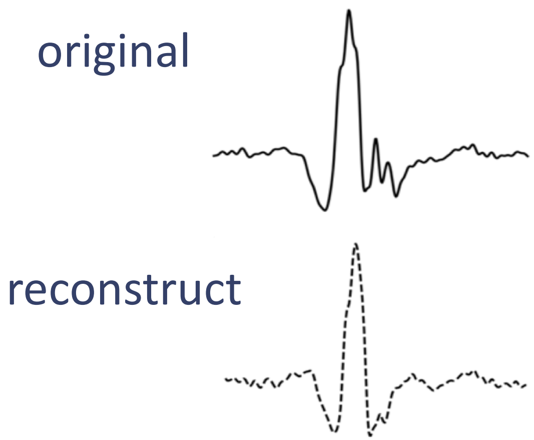
Table 3: BeatGAN performs the best for anomalous rhythm detection in ECG data. In BeatGAN$_{aug-3\times}$, we augment the training data size to $3\times$ with time warping. In BeatGAN$^{0.1\%}_{aug-3\times}$, we add the 0.1% anomalous time series to the training data for evaluating robustness. 5-fold cross-validations are run, and averaged metrics and std are given.

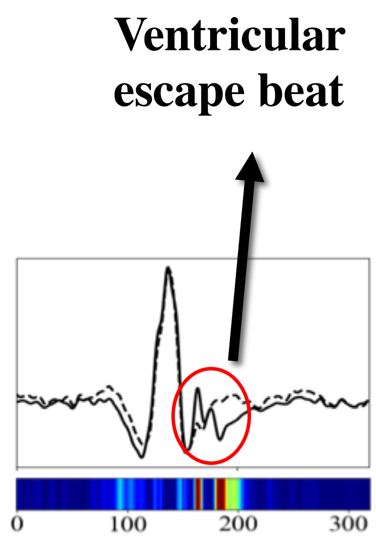
The model is robust even add 0.1% abnormal time series in training set

Interpretable output

- Pinpoint the time ticks involved in the anomalous patterns.
- The heatmap's value at time t is calculated by

$$residual(t) = \max_{channel}\left(x(t) - x'(t)\right)^2$$





Result

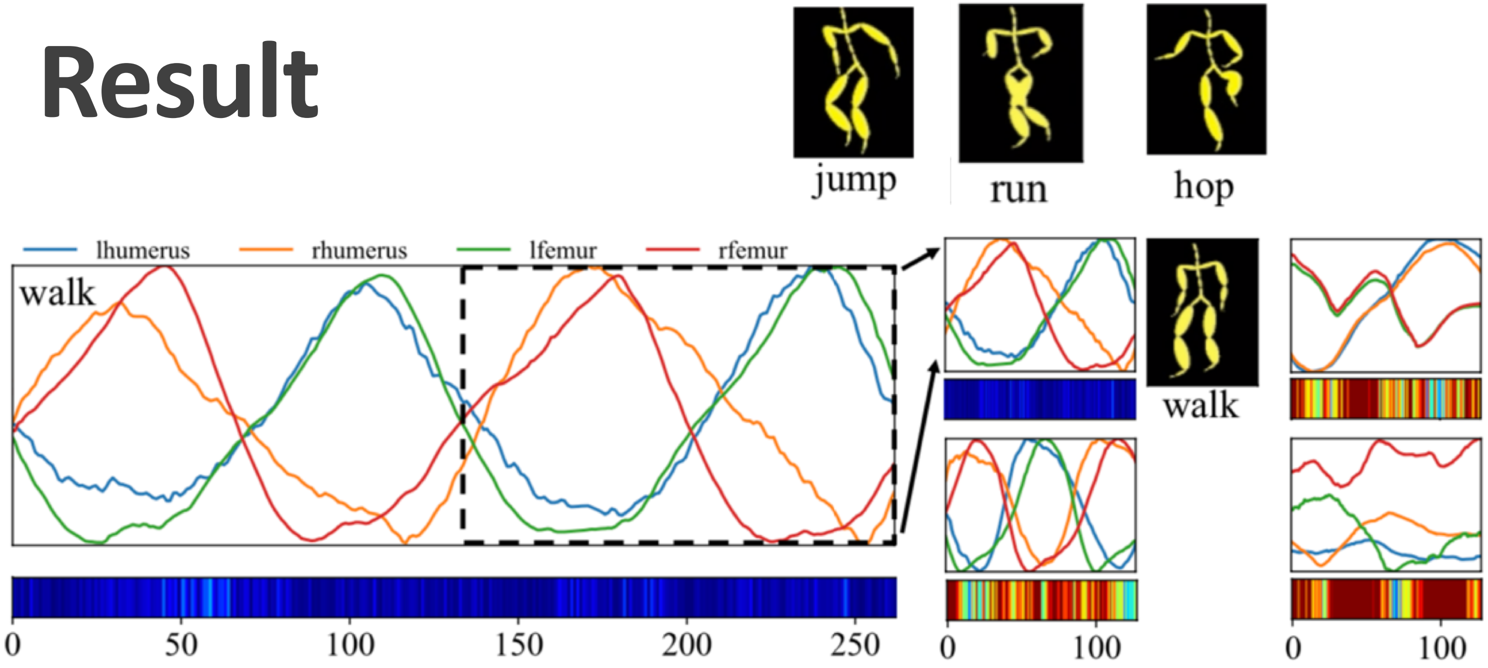
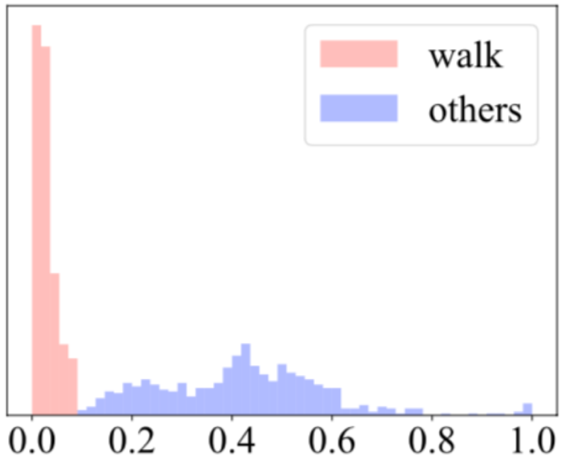


Figure 3: Example of anomaly detection on motion capture time series. The heatmaps show the anomalousness scores for each time tick, showing that we accurately assign high scores to the non-walking activities.

- Regard the walking as normal class
- Separate the walking and non-walking time series



Fast Inference

- Faster than other reconstruct-based deep model
  - AnoGAN needs iterative gradient descent in Inference.
  - Ganomaly has another encoder-network and compute anomalousness score in hidden space.

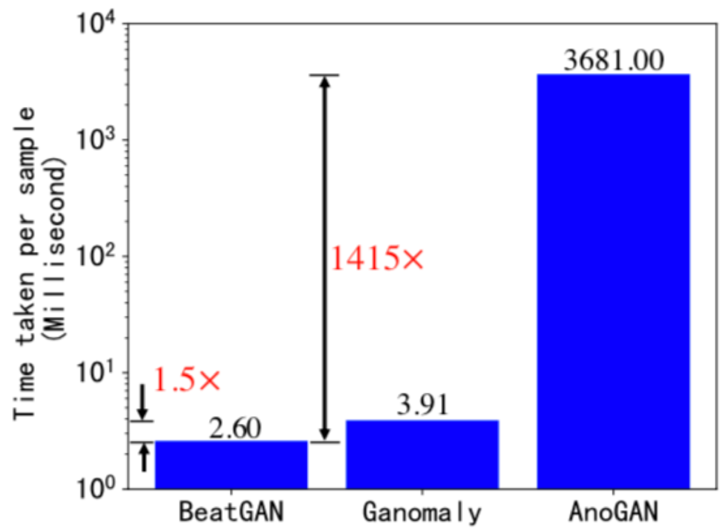
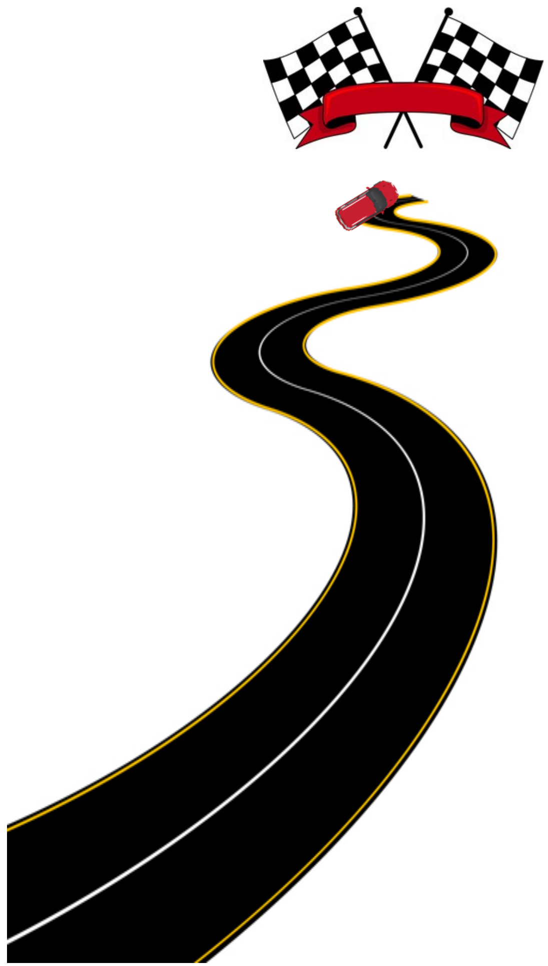


Figure 5: BeatGAN has fast inference (2.6ms) for detecting anomalous beats.

Road Map

- Background
- Proposed Method: BeatGAN
- Experiments
- Conclusion <<



Conclusion

- Anomaly detection from normal time series
- Effectiveness
- Explainability

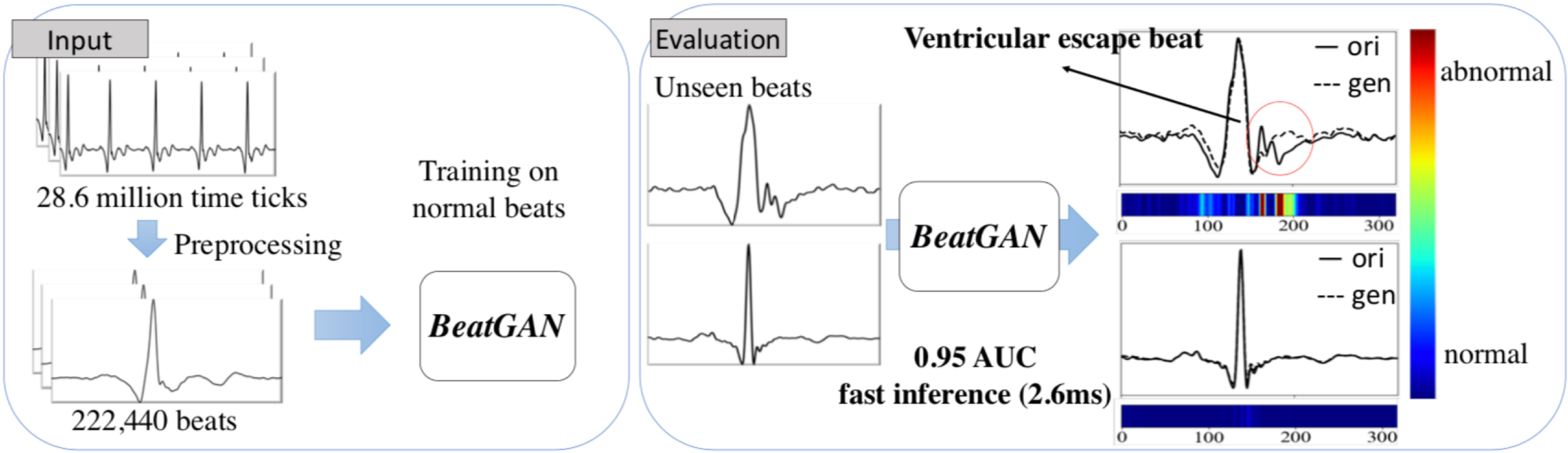


Figure 1: BeatGAN successfully detects anomalous rhythms, and explains the results. The size of training input is 28.6 million time ticks, and inference can be as fast as 2.6 ms per beat. The original beat is shown by solid lines, and the generated beat is shown by dashed lines.

Thank you !

Source codes and datasets used in the paper are available at
https://github.com/BGT-M/spartan2
with tutorial
https://github.com/BGT-M/spartan2-tutorials/blob/master/BeatGAN.ipynb