Different Facets in Multivariate Time-series Interpretability

Anika Tabassum

Data Seminar

04/09/2021

Outline

- Deep reconstruction of strange attractors from timeseries. Wiliam Gilphin. Proceedings of the NeuRIPS 2020.
- Interpretable, Multidimensional, Multimodal Anomaly Detection with Negative Sampling for Detection of Device Failure. John Sipple. Proceedings of the 37th ICML, 2020
- Benchmarking Deep Learning Interpretability in Time Series Predictions. Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, Soheil Feizi. Proceedings of the NeuRIPS 2020.

Outline

- **Deep reconstruction of strange attractors from timeseries**. Wiliam Gilphin. Proceedings of the NeuRIPS 2020.
- Interpretable, Multidimensional, Multimodal Anomaly Detection with Negative Sampling for Detection of Device Failure. John Sipple. Proceedings of the 37th ICML, 2020
- Benchmarking Deep Learning Interpretability in Time Series Predictions. Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, Soheil Feizi. Proceedings of the NeuRIPS 2020.

Motivation & Idea

- Inverse problem: Given a single, time-resolved measurement of a complex dynamical system, is it possible to reconstruct the higher-dimensional process driving the dynamics?
- Introduce state-space reconstruction method: reconstruct the d-dimensional attractor of an unknown dynamical system, given only a univariate measurement time series.

Background

Suppose that a d-dimensional dynamical system $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, t)$ occupies an attractor A. The time-evolving state variable $\mathbf{y}$ may be represented abstractly by composition with a flow operator, $\mathbf{y}(t) = \mathcal{F} \circ \mathbf{y}(t_0)$. At any given instant in time, a measurement $\mathbf{x}(t)$ corresponds to composition with the operator, $\mathcal{M}$, such that $\mathbf{x}(t) = \mathcal{M} \circ \mathbf{y}(t) = \mathcal{M} \circ (\mathcal{F} \circ \mathbf{y}(t_0))$, where $d_m \equiv \dim \mathbf{x}_t$. We define the data matrix $X = \left[\mathbf{x}_1^\top \mathbf{x}_2^\top \cdots \mathbf{x}_N^\top\right]^\top$

- N : #measurements
- $X \in R^{N\ X\ T}$ has Hankel structure along its diagonals converted from y

We seek a parametric similarity transformation $\hat{\mathbf{y}} = \mathbf{g}(\mathbf{x})$ such that $\hat{Y} \sim Y$, where $Y \in \mathbb{R}^{N \times d}$ and $\hat{Y} \in \mathbb{R}^{N \times d_E}$. The point set $Y = \left[\mathbf{y}_1^\top \mathbf{y}_2^\top \cdots \mathbf{y}_N^\top\right]^\top$ corresponds to a finite-duration sample from the true attractor A, and the point set $\hat{Y} = \left[\hat{\mathbf{y}}_1^\top \hat{\mathbf{y}}_2^\top \cdots \hat{\mathbf{y}}_N^\top\right]^\top$ refers to the embedding of $\mathbf{x}$ at the same timepoints.

Approach

- Train an AutoEncoder:

Encoder: attractor $\bar{Y} = g(X)$

Decoder:

reconstructed input $\bar{X} = g'(\bar{Y})$

AE: $\bar{X} = g'(g(X))$

Loss function:

$$\mathcal{L}(X, \hat{X}, \hat{Y}) = \| X - \hat{X} \|^2 + \lambda \mathcal{L}_{\text{FNN}}(\hat{Y})$$

Reconstruction loss

Novel sparsity promoting loss (false nearest neighbor loss)
AE learns an effective embedding dimension d_E because of the regularizer

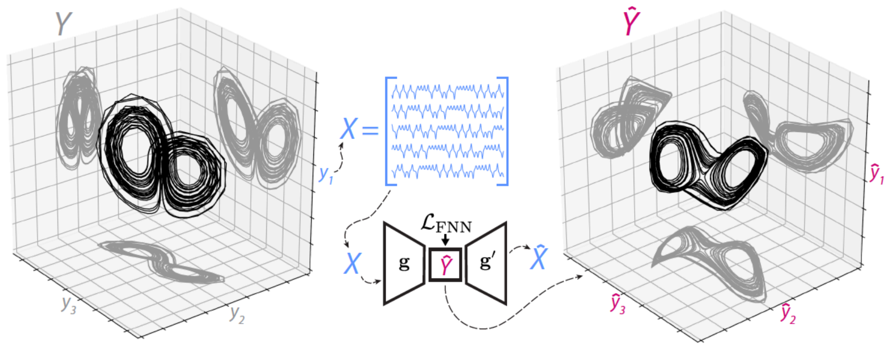


Figure 1: Overview of problem and approach. A univariate time series $y_1(t)$ is observed from a multivariate attractor $Y = [y_1(t)\ y_2(t)\ y_3(t)]$. This signal is converted into a time-lagged Hankel matrix X, which is used to train an autoencoder with the false-nearest-neighbor loss $\mathcal{L}_{\text{FNN}}$. The latent variables reconstruct the original coordinates.

False Nearest Neighbor Loss

- $L_{FNN} \rightarrow$ input: hidden layer from AE $h \in R^{B\ X\ L}$, B: batch size L: latent dimension
- Compute pairwise Euclidean distance of m points of L and sort by column: $D \in R^{B\ X\ L\ X\ B}$
- Select k nearest neighbor for each i $\in$ m

$$\mathcal{L}_{\mathrm{FNN}} = \sum_{m=2}^{L} \underbrace{(1 - \bar{F}_m)}_{\text{Batch-averaged fraction of false neighbors not in k for each latent index}} \bar{h}_m^2 \longrightarrow \text{Batch averaged activity in mth latent unit}$$

Models & Datasets

Models:

- LSTM with L_{FNN}
- MLP with L_{FNN} (L=10)

Baselines:

- MLP with L=1
- AE without L_{FNN}
- time-lagged independent component analysis (tICA)
- Eigen-time delay coordinates (ETD)

Datasets (chaotic or quasiperiodic systems):

-Lorenz (3d)

-Rossler (3d)

-Lotka-Volterra ecosystem (10d)

-Torus (3d)

-Pendulum (4d)

Metrics (compare Y, $\bar{Y}$)

- Point-wise comparison: Euclidean, DTW
- Forecasting: reconstructed $\bar{Y}$ are the predicted future values of Y
- Local-neighborhood: compare k neighbors of Y,$\bar{Y}$
- Attractor dimensionality:
- Topological feature: quantify degree to which $\bar{Y}$ retain same feature as Y
- Fractal dimension: quantify similarity of correlation of fractal dimensions

Results (Evaluating Reconstruction)

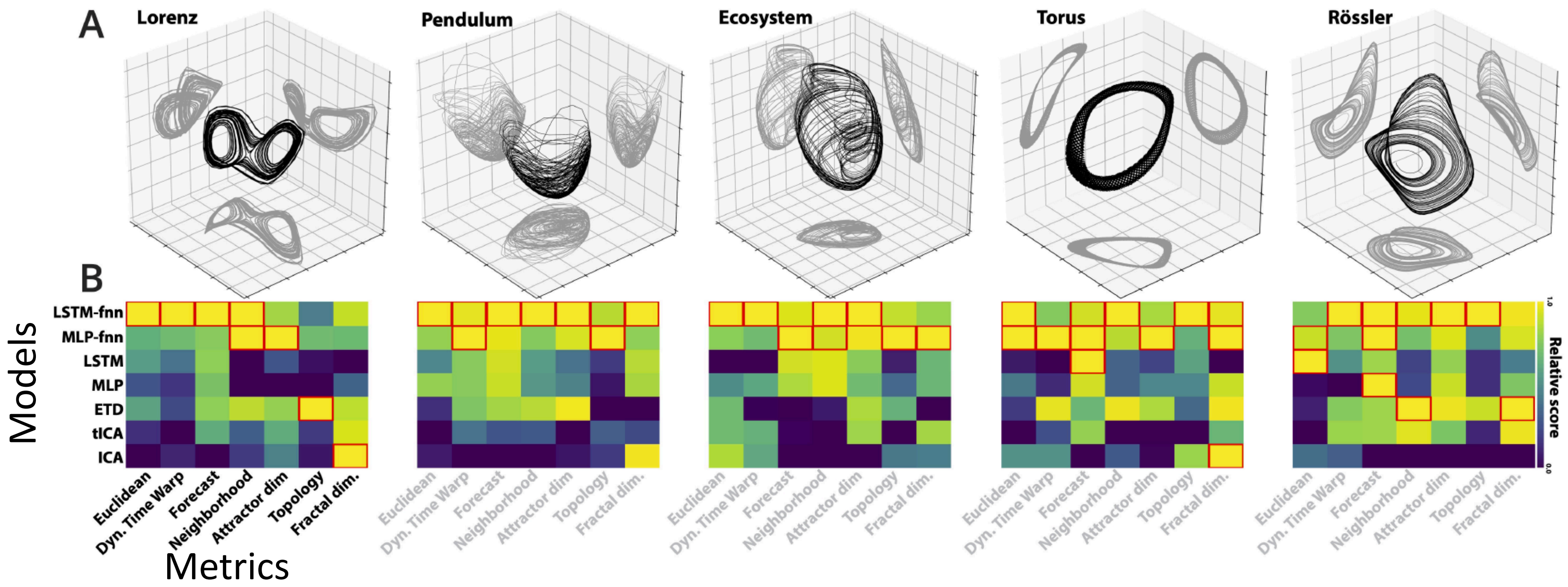


Figure 3: (A) Embeddings produced by the autoencoder with $\mathcal{L}_{\mathrm{FNN}}$, trained on only the first coordinate of each system. (B) For each system, a variety of baseline embeddings are compared to the original attractor via multiple similarity measures. Hue indicates mean across 5 replicates scaled by column range, with red boxes indicating column maximum, or values falling within one standard deviation of it. Because distinct similarity metrics have different dynamic ranges, each column has been normalized separately to accentuate differences across models (see appendix for tabular values).

Yellow is the higher, LSTM-fnn performs best

Results (Robustness to noise)

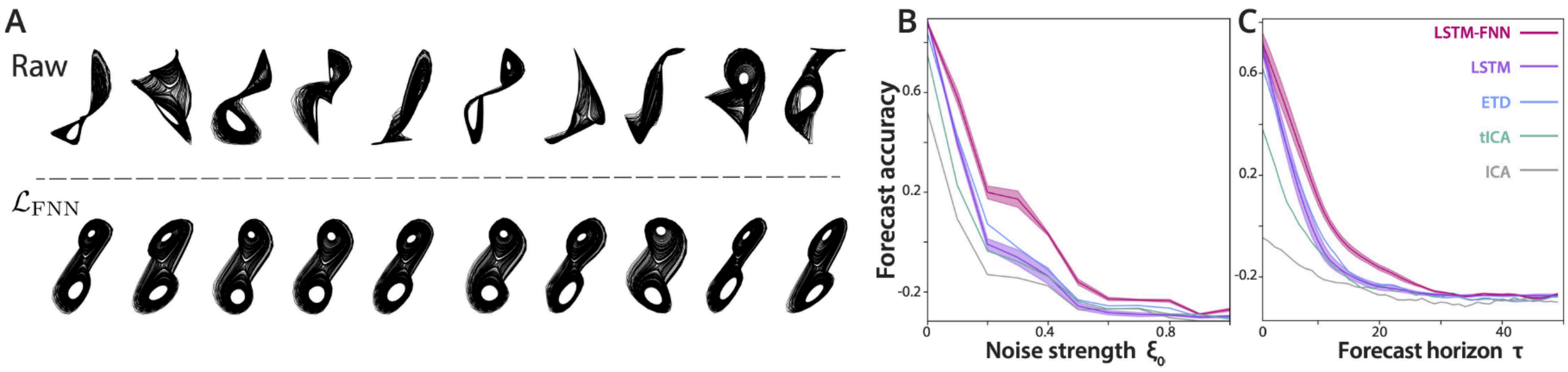
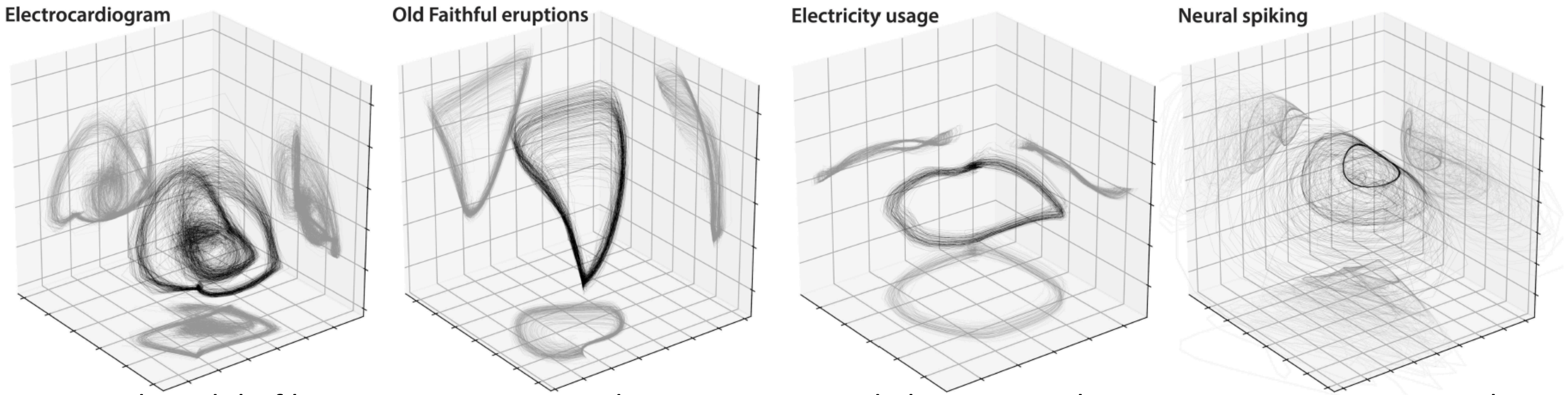


Figure 4: (A) Embeddings of the stochastic Lorenz dataset with and without the false-nearest-neighbors regularizer. Replicates correspond to different random initializations of the Brownian noise force and initial network weights. (B) The cross-mapping forecast accuracy as a function of noise strength ξ_0 (with constant $\tau = 20$). (C) The cross-mapping forecast accuracy versus forecasting horizon τ (with constant $\xi_0 = 0.5$). Standard errors span 5 replicates.

Results (Case-studies)



created model of heart　quasi periodic attractor　daily usage cycle　Mouse neuron spike

Figure 6: Embeddings of an electrocardiogram (160 heartbeats), temperature measurements of the erupting "Old Faithful" geyser in Yellowstone National Park (200 eruptions), average electricity usage by 321 households (200 days), and neural spiking in a mouse thalamus.

Outline

- Deep reconstruction of strange attractors from timeseries. Wiliam Gilphin. Proceedings of the NeuRIPS 2020.
- **Interpretable, Multidimensional, Multimodal Anomaly Detection with Negative Sampling for Detection of Device Failure**. John Sipple. Proceedings of the 37th ICML, 2020
- Benchmarking Deep Learning Interpretability in Time Series Predictions. Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, Soheil Feizi. Proceedings of the NeuRIPS 2020.

Motivation & Idea

- Complex IoT devices have large multidimensional observations (e.g., power control in buildings, electrical components in power plants)
- Anomaly detection refers to finding pattern in this data that do not conform to expected behavior (e.g., device failure)
- Factors to consider in multidimensional anomaly detection:

 - Noise: anomaly is subset of observations, and masked in noise dimensions

 - Correlation: there may or may not have correlation among the features

 - Multimodal: a process can operate in multiple mode (e.g., zone-vacant, zone comfort-mode)

 - Interpretable: understand which observations are contributing to anomaly score

Notations

- X: x(1), x(2), …. : a sequential stream of multidimensional data points
- x(i) : ith data point D- dimensional vector $\mathrm{x} = \{x_1,\ x_2, \ldots x_D\}$
- Objectives:

  - estimate $P(x \in Normal)$

  - attribute anomaly score to each x_d in x

Definition 1. *An* ***anomaly*** *is any data point x with a near zero probability that it was generated by the Normal process:* $P(x \in Normal) \approx 0$.

- Normal process occupies one or more discrete volumes of unknown shape

Framework: Detecting anomaly with negative sampling

- Define 2 class samples:

 - Positive class samples: $\mathrm{U} = \{u(1),\ u(2), \ldots u(M)\}$ → M D-dimensional data points observed from x, may include small number of actual anomalies

- Negative class sample: $\mathrm{V} = \{v(1),\ v(2), \ldots v(N)\}$ → N D-dimensional data points

- Train a classifier to distinguish between 2 classes

$$F : \Re^D \to [0, 1]$$

Assumptions + Prepositions

- Assumptions: U is representative of Normal process and essential to sample enough to reflect all normal modes of observations
- Propose uniform i.i.d for generating negative samples

Proposition 1. (Uniform Negative Sampling): *For each dimension $d \leq D$, let $lim_d = [min\,(U_d) - \delta, max\,(U_d) + \delta]$ be a range bounded by the extrema of the positive sample U extended by a conservative positive length δ that extends lim_d beyond the normal space. We assume that the sample size of U is sufficiently large to bound the Normal region. Choose a negative sample V, by selecting N points uniformly i.i.d. bounded by lim_d for each $d \leq D$. In high dimension, $D \rightarrow \infty$, false negative sampling error decays exponentially to zero, regardless of the shape of the Normal region.*

Proposition 2. (Labeled Training Set for Anomaly Detection): *Given a sufficiently sampled, high-dimensional dataset from a target process and uniform negative sampling, we can generate a labeled two-class dataset to train a classifier F for detecting anomalies.*

Framework: Interpreting anomalies with integrated gradients

- Integrated Gradient:

- used to show what pixels contribute most to an image classification

- computes and integrates gradients for each dimension from a baseline point to the observed point

- key step is to select a good baseline ($U^* \subset U$)

Proposition 3. (Baseline Set for Anomaly Detection) *Points from the positive sample used to train the anomaly detection classifier with high Normal class confidence scores, $U^* \subset U : \forall_{u \in U^*} F(x) \geq 1 - \epsilon$ are a sufficient baseline set.*

Compute Integrated Gradients

- Choose nearest point from U* to anomalous point x (an approximation for the closest point of Normal)
- Choose *baseline point* from U* with minimum Euclidean distance

$$u^* = \mathbf{argmin}_{u \in U^*}\{\mathrm{dist}(x, u)\}$$

- Apply integrated gradient eq along dth dimension:

Gradient of
the classifier F

$$B_d(x) \equiv (u_d^* - x_d) \times \int_{\alpha=0}^{1} \frac{\partial F(x + \alpha \times (u^* - x))}{\partial x_d} d\alpha$$

Path variable

Datasets + Baselines

- Baselines:
  - One class SVM (OC-SVM): kernel (linear, polynomial, RBF, sigmoid)
  - Isolation Forest (ISO): ensemble based
  - Deep-SVDD (DSVDD): deep learning adaptation of anomaly detector (replaced CNN with dense and dropout layers)
  - Extended Isolation Forest (EIF): reduce false positive regions in ISO for multimodal X
- Datasets:

Table 1. Summary of Anomaly Detection Datasets.

DATA SET	SIZE	DIM	ANOMALY
FOREST COVER (FC)	286,048	10	2,747 (0.9%)
SHUTTLE (SH)	49,097	9	3,511 (7%)
MAMMOGRAPHY (MM)	11,183	6	260 (2.3%)
MULCROSS (MC)	262,144	4	26,214 (10%)
SATELLITE (SA)	6,435	36	2,036 (32%)
SMART BUILDINGS (SB)	60,425	7	1,921 (3.2%)

Experiments: Anomaly Detection

- Anomaly detection Classifiers with negative sampling:

- Random Forest (NS-RF)
- Neural Network (NS-NN)
  - i. drop-out layer
  - ii. RELU

- 5 fold Cross validation
- Vaidation set: 20%

Table 2. Mean and Standard Deviations of AUC values as % for benchmark datasets and the Smart Buildings dataset. Highlighted values are the top-scoring detectors based on a 5% significance threshold.

	OCSVM	DSVDD	ISO	EIF	NSRF	NSNN
FC	53±20	69±7	85±4	**93±1**	80±2	86±4
SH	93±0	88±9	**96±1**	91±1	**93±7**	**96±5**
MM	71±7	78±6	77±2	**86±2**	**85±4**	84±2
MC	90±0	54±17	88±0	66±4	94±1	**99±1**
SA	51±1	62±3	67±2	**71±3**	65±4	**73±3**
SB	76±1	60±7	71±7	80±4	**95±1**	93±1

Experiments: Anomaly Interpretation

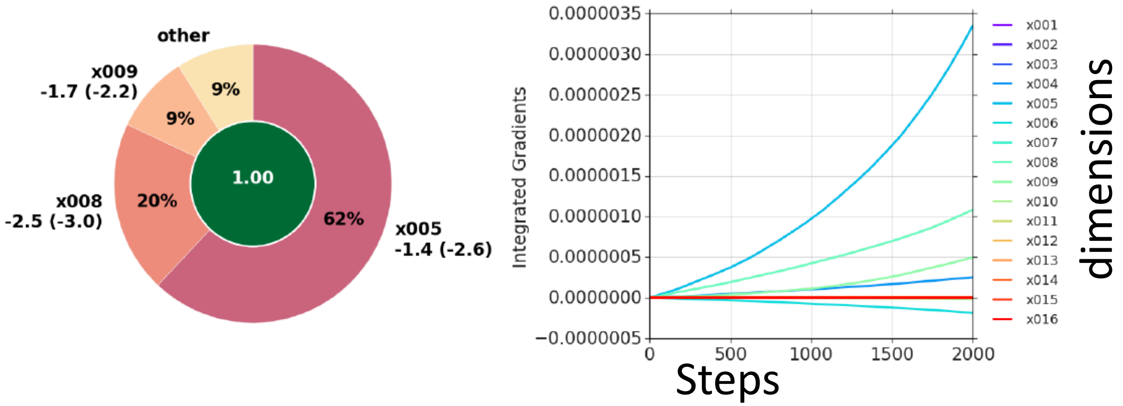


Figure 1. Anomaly Interpretation of a *Normal* point x. The left image shows $F(x) = 1$ in the center green circle, and the proportional blame B_d against dimensions $x005$, $x008$, and $x009$ as exterior wedges. The right chart displays the stepwise integrated gradients from x at $k = 0$ to the nearest baseline u^* at $k = 2,000$. Since the point is normal, the gradients are very small, with $\sum B_d \approx 0$.

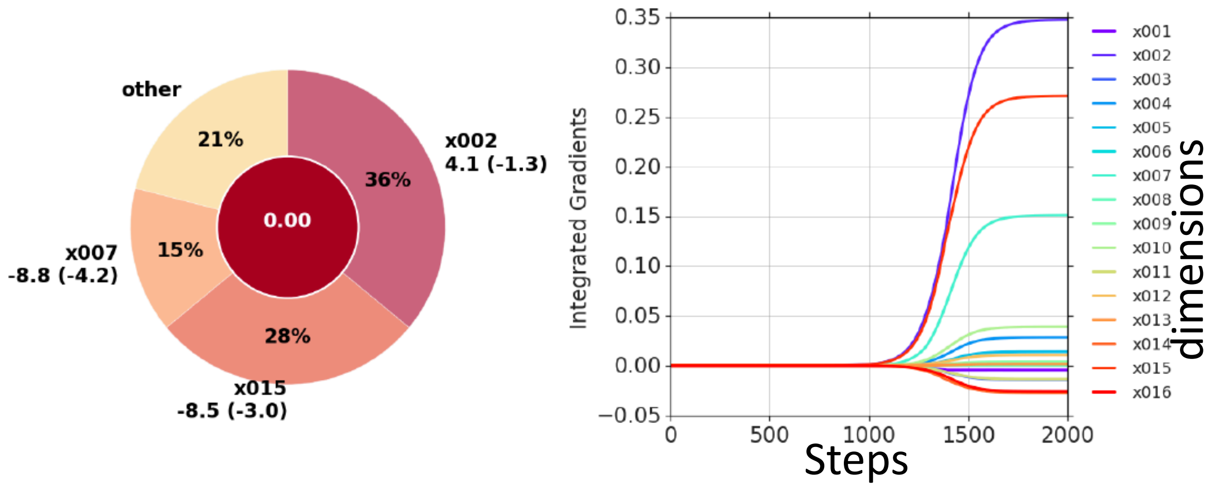


Figure 2. Anomaly Interpretation of an *Anomalous* point x with $F(x) = 0$, Three dimensions ($x002$, $x015$, and $x007$) assigned most of the blame, $\sum B_d \approx 1$. The observed and expected normal values, x_d (u^*_d), are displayed next to each wedge.

Synthetic dataset: positive sample-> 2500 data points, D-> 16, anomaly-> additional 125 points

Outline

- Deep reconstruction of strange attractors from timeseries. Wiliam Gilphin. Proceedings of the NeuRIPS 2020.
- Interpretable, Multidimensional, Multimodal Anomaly Detection with Negative Sampling for Detection of Device Failure. John Sipple. Proceedings of the 37th ICML, 2020
- **Benchmarking Deep Learning Interpretability in Time Series Predictions**. Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, Soheil Feizi. Proceedings of the NeuRIPS 2020.

Motivation & Idea

- Estimating feature importance for multivariate time-series data is challenging
- Saliency maps are faithful visualizing interpretation method
- The authors compare performance over multiple:

  - interpretability methods (gradient-based, perturbation-based)

  - neural architectures (RNN, TCN, Transfmormers)

  - synthetic datasets to capture different spatio-temporal aspects

- Propose Two-step Temporal Saliency Rescaling approach (TSR)

Problem Definition

- Input: A multivariate time-series $\mathrm{X} = \{x_1,\ x_2, \ldots x_T\} \in R^{N\ X\ T}$
- Model produces an output $\mathrm{S(X)} = \{S_1(X),\ S_2(X), \ldots S_C(X)\}$, C: #classes
- Output: For a target class c, saliency method finds relevance $\mathrm{R(X)} \in R^{N\ X\ T}$which assign relevance score of each input feature i at time t

Saliency Methods:

1. Gradient-based (Integrated Grad, Smooth Grad, DeepLift)
2. Perturbation-based (feature occlusion, feature pertubation)
3. Shapley Value Sampling (SVS): approximate shapley value that involves random permutation of input features

Temporal Saliency Rescaling (TSR)

Algorithm 1: Temporal Saliency Rescaling (TSR)

Given: input X, a baseline interpretation method $R(.)$
Output: TSR interpretation method $R^{TSR}(.)$
for $t \leftarrow 0$ **to** T **do**
    Mask all features at time t: $\overline{X}_{:,t} = 0$, otherwise $\overline{X} = X$;
    Compute Time-Relevance Score $\Delta_t^{time} = \sum_{i,t} |R_{i,t}(X) - R_{i,t}(\overline{X})|$;
for $t \leftarrow 0$ **to** T **do**
    for $i \leftarrow 0$ **to** N **do**
        if $\Delta_t^{time} > \alpha$ **then**
            Mask feature i at time t: $\overline{X}_{i,:} = 0$, otherwise $\overline{X} = X$;
            Compute Feature-Relevance Score $\Delta_i^{feature} = \sum_{i,t} |R_{i,t}(X) - R_{i,t}(\overline{X})|$;
        else
            Feature-Relevance Score $\Delta_i^{feature} = 0$;
        Compute (time,feature) importance score $R_{i,t}^{TSR} = \Delta_i^{feature} \times \Delta_t^{time}$;

Datasets & Metrics

- Datsets: 10 time-series datasets, each synthetic dataset generated by 7 different process→ 70 synthetic datasets
- Performance metric:

- Precision (AUP): Are all features identified as salient informative?

- Recall (AUR): Is the saliency method able to identify all Informative features?

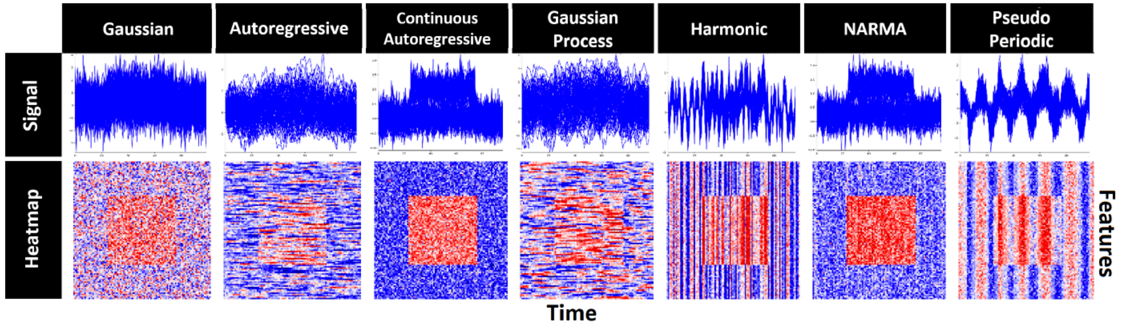


Figure 2: Middle box dataset generated by different time series processes. The first row shows how each feature changes over time when independently sampled from time series processes. The bottom row corresponds to the heatmap of each sample where red represents informative features.

Experiments: Saliency Map Quality

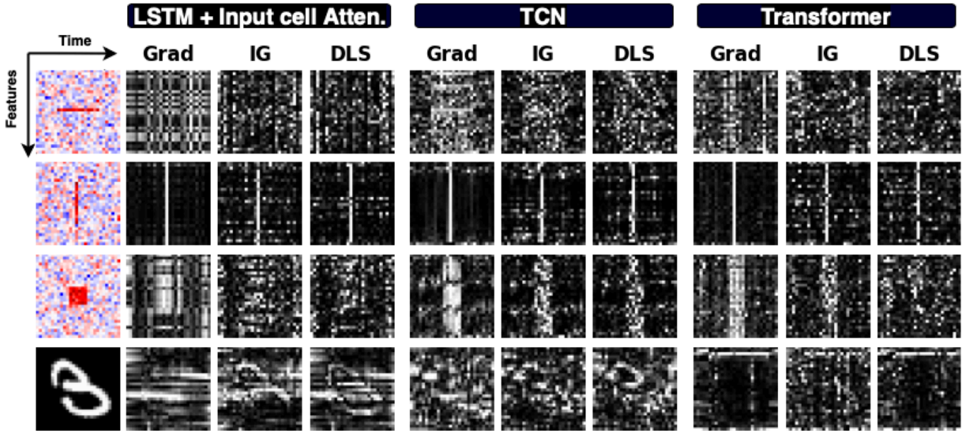


Figure 4: Saliency maps produced by Grad, Integrated Gradients, and DeepSHAP for 3 different models on synthetic data and time series MNIST (white represents high saliency). Saliency seems to highlight the correct time step in some cases but fails to identify informative features in a given time.

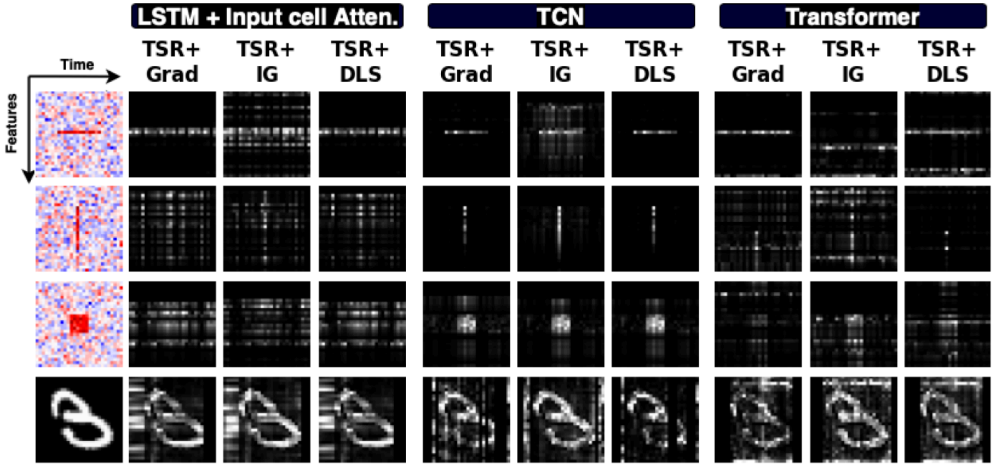


Figure 5: Saliency maps when applying the proposed Temporal Saliency Rescaling (TSR) approach.

TSR able to give good quality saliency than normal saliency methods

Experiments: Saliency vs Random ranking

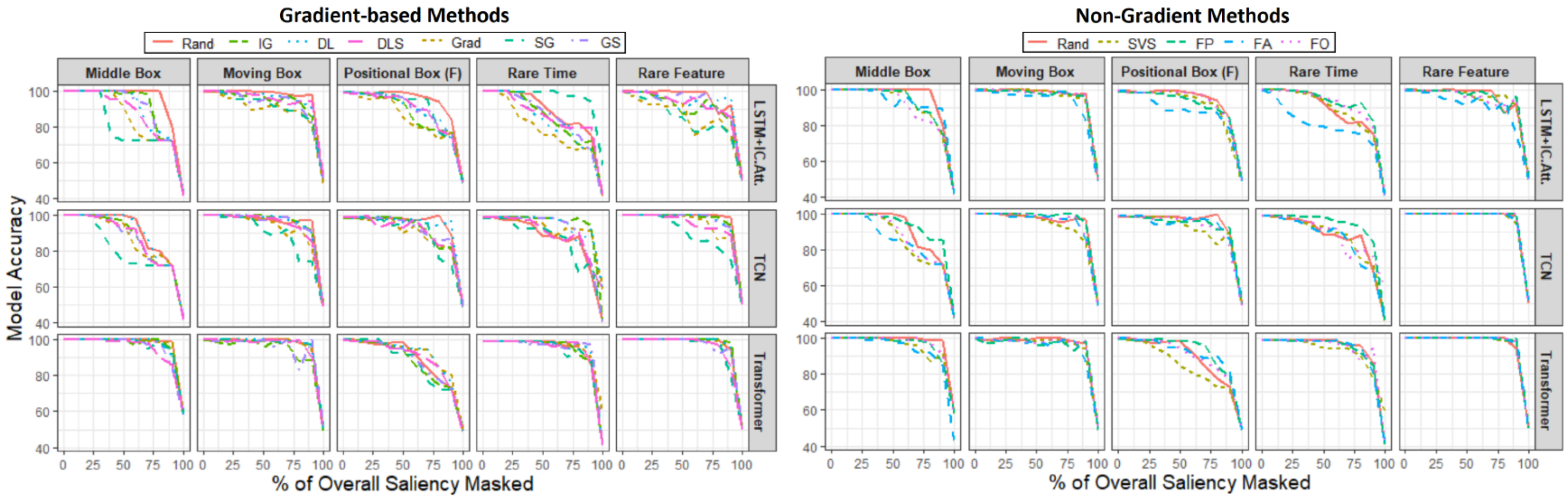


Figure 6: The effect of masking features identified as salient by different methods against a random baseline. Gradient-based and non-gradient based saliency methods are shown in the left and right plots, respectively. The rate of accuracy drop is not consistent; in many cases there is not much improvement over random baseline.

Accuracy don’t drop every time

Experiments: Saliency vs Random ranking

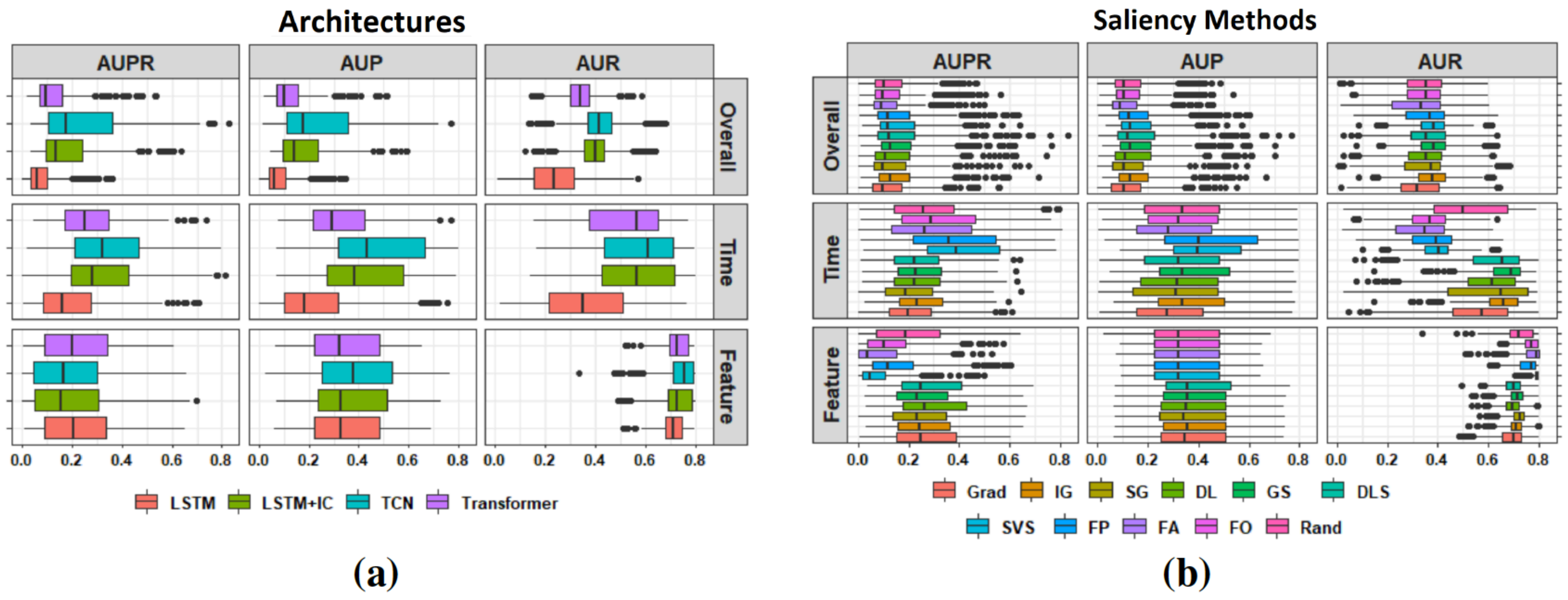


Figure 7: Precision and Recall distribution box plots, the top row represents overall Precision/Recall, while the second two rows show Precision/Recall distribution on time and feature axes (a) Distribution across architectures. (b) Distribution across saliency methods.

1. Model architecture has largest performance over precision and recall
2. Results donot show clear distinction between saliency methods
3. Methods can identify informative time-steps but fail to identify informative features (Time and feature domain)

Saliency Maps: Image over Multivariate Time-

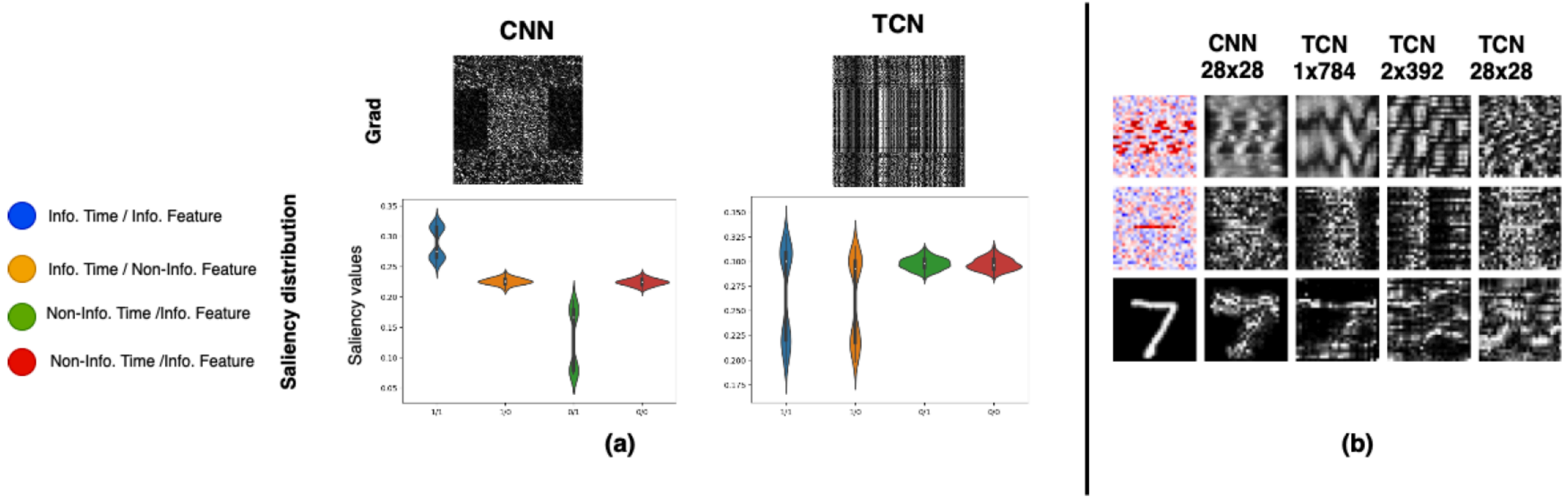


Figure 8: **(a)** Saliency maps and distribution produced by CNN versus TCN for *Middle Box*. **(b)** Saliency Maps for samples treated as image (CNN) vs. uni-, bi- or multi-variate time series (TCN).

Evaluation on TSR

	Middle Box				Moving Box			
Saliency Methods	AUPR	AUP	AUR	AUC	AUPR	AUP	AUR	AUC
Grad	0.331	0.328	0.457	64.90	0.225	0.229	0.394	95.35
DLS	0.344	0.344	0.452	68.30	0.288	0.288	0.435	94.05
SG	0.294	0.300	0.451	64.00	0.241	0.247	0.395	92.90
TSR + Grad	**0.399**	**0.381**	**0.471**	**62.20**	**0.335**	**0.326**	**0.456**	**84.00**

Table 1: Results from TCN on Middle Box and Moving Box synthetic datasets. Higher AUPR, AUP, and AUR values indicate better performance. AUC lower values are better as this indicates that the rate of accuracy drop is higher.

Summary:

1. Commonly used saliency methods fail to produce high quality interpretations for multivariate time-series
2. They can produce good quality saliency if multivariate time-series treated as image or univariate
3. No clear distinction of performance between multiple saliency methods on multiple metrics
4. TSR has substantial improvement over existing saliency methods