

Unsupervised Anomaly Detection in Sequences Using Long Short Term Memory Recurrent Neural Networks

Majid S. alDosari

April 20, 2016

George Mason University

Contents I

Introduction

The Challenge of Anomaly Detection in Sequences

Procedure

1. Sample Extraction
2. Transformation
3. Detection Technique

Proximity

Effects on Point Distribution

Data Classification

Nearest Neighbor

Clustering

Contents II

Models

Problems with Established Techniques

Recurrent Neural Networks (RNNs)

Using RNNs for Anomaly Detection

Conclusions

Reproducibility

Discussion Time

Introduction

Modern technology facilitates the capture, storage, and processing of sequential data at scale

- Data capture
 - physiological signals
 - network traffic
 - industrial processes
 - automobiles
 - website navigation
 - environment
- Data storage
 - Hadoop
 - MongoDB
- Ubiquitous computing
 - cloud/cluster
 - desktop
 - at point of capture

Problem: Finding anomalous data is challenging

- large
- varied
- domain knowledge required

Solution: Use recurrent neural networks to generically find anomalous data

- This work:
 1. Background: Anomaly detection in sequences
 2. Sequence Modeler: Recurrent neural network (RNN)
 3. Experiments: RNNs for anomaly detection
- Prior: Malhotra¹ but no emphasis on process

¹Malhotra et al., “Long Short Term Memory Networks for Anomaly Detection in Time Series”.

The Challenge of Anomaly Detection in Sequences

Anomaly detection work is fragmented

- variety of solutions in communication networks, biology, economics, biology, ...etc.
- different settings
- no comparison between application domains
- technical basis in computer science vs. statistics
- not much review literature: Cheboli² and Gupta³

²Cheboli, “Anomaly Detection of Time Series”.

³Gupta et al., “Outlier Detection for Temporal Data : A Survey”.

Define the problem to focus on the right solution

Sequence \mathbf{x}

$$\mathbf{x} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(T)}\}$$
$$\mathbf{x}^{(t)} \in \mathbb{R}^d$$

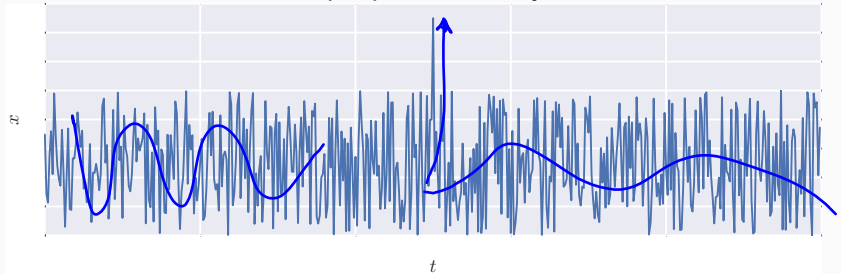
Assumption: Anomalies are a small part of the data.

Solution must answer the following:

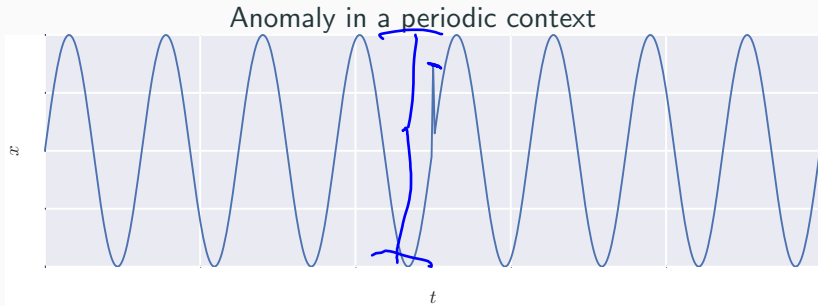
1. What is normal (as an anomaly is defined as what is *not* normal)?
2. What measure is used to indicate how anomalous point(s) are?
3. How is the measure tested to decide if it is anomalous?

Solution must address different anomaly types I

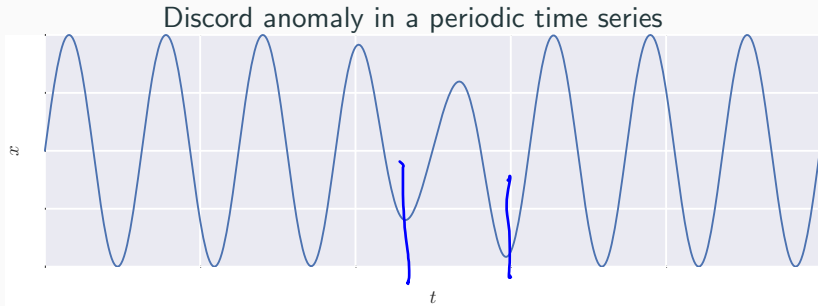
Simple point anomaly



Solution must address different anomaly types II

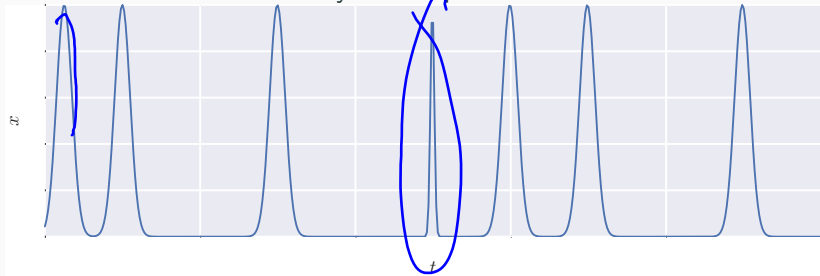


Solution must address different anomaly types III



Solution must address different anomaly types IV

Discord anomaly in an aperiodic time series



Solution must address different anomaly types V

Multivariate: (a)synchronous and (a)periodic

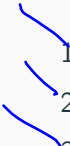
Procedure

Description of anomaly detection procedure is straightforward

1. Compute an anomaly score for an observation
2. Aggregate the anomaly scores for many observations.
3. Use the anomaly scores to determine whether an observation can be considered anomalous

Characterizing normal behavior is involved

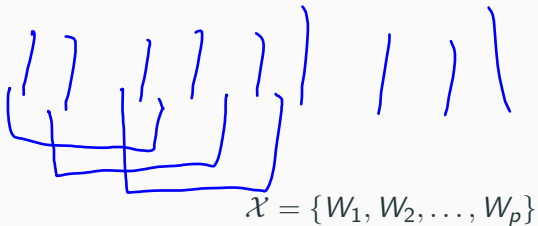
$$\mathbf{x} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(T)}\}$$

- 
1. Extract Samples
 2. Transform Samples
 3. Apply Detection Technique

Procedure

1. Sample Extraction

Use sliding windows to obtain samples



• hop, h

• window, w

Problem: Hops can skip over anomalies

sequence: *abcabcabc*

hop (<i>h</i>)	Ordered Windows
1	<i>abc, bcc, cca, cab, abc, bca, cab, abc</i>
2	<i>abc, cca, abc, cab</i>
3	<i>abc, cab, cab</i>
4	<i>abc, abc</i>

Problem: Window size must be large enough to contain anomaly

sequence: aaabbbccc_aabbbcccaaabbbccc

Window width must be at least 4.

Problem: Treating window width as a dimension size ignores temporal nature

$$\mathcal{X} = \{W_1, W_2, \dots, W_p\}$$

$$W \in \mathcal{R}^{1 \times \underbrace{w}}$$

Procedure

2. Transformation

Transformation can help reveal anomalies

- Haar transform
- Symbolic Aggregate approXimation (SAX)⁴

⁴Lin et al., “Experiencing SAX: A novel symbolic representation of time series”.

Transformation is not general

- Choice of representation must be compatible with data characteristics
- normal:anomaly as transform(normal):transform(anomaly)

Study⁵ suggests generally little difference among representations.

⁵Wang et al., “Experimental comparison of representation methods and distance measures for time series data”.

Procedure

3. Detection Technique

Anomaly detection techniques and their application domains are varied

Based on

- Segmentation
- Information Theory
- Proximity
- Modeling

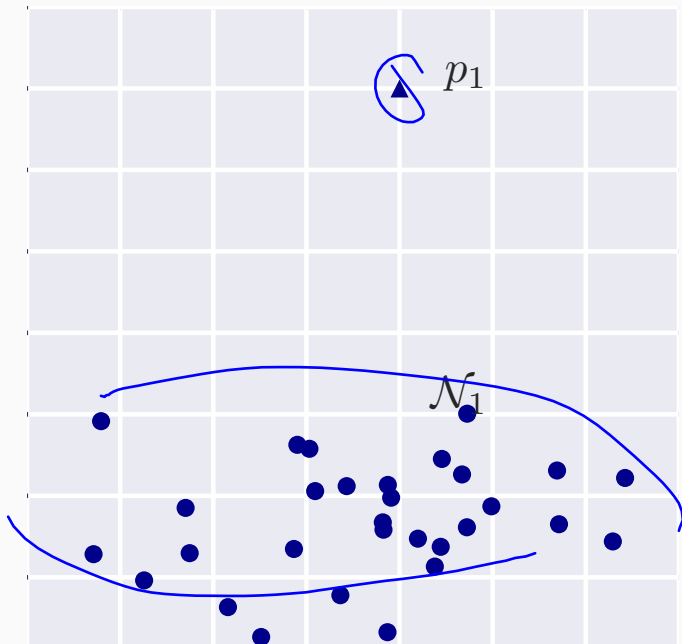
Model and proximity-based techniques are most developed

Based on

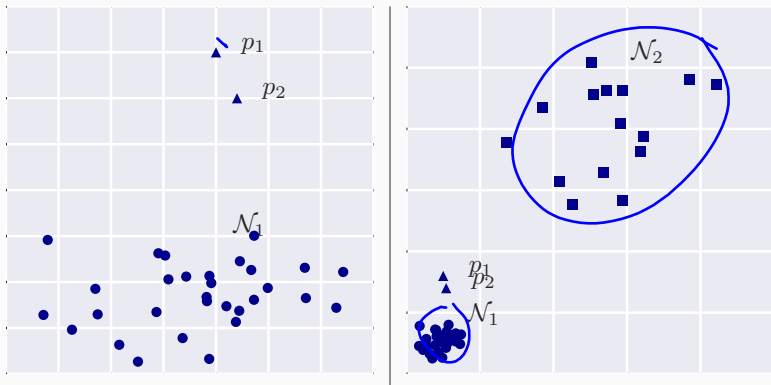
- Segmentation: requires homogeneous segments
- Information Theory: requires finding sensitive information-theoretic measure
- Proximity
- Modeling

Proximity

Idealization never occurs



Practically, distributions are complicated

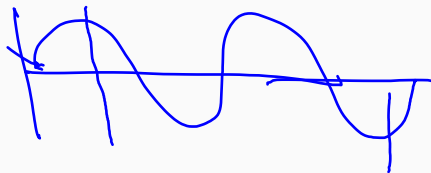


Proximity

Effects on Point Distribution

Distance measure should be invariant to:

- length
- translation
- (skew)
- (amplitude)



Study⁶: Not much difference in similarity measures

⁶Wang et al., “Experimental comparison of representation methods and distance measures for time series data”.

Window width needs to be chosen on the scale of expected anomaly

If width is too large:

- anomalous points not distinguished

- data becomes equidistant in high-dimensional space

Sliding windows challenge anomaly detection assumptions

- anomalous points are not necessarily in sparse space while repeated patterns are not necessarily in dense space⁷
- “Clustering of Time Series Subsequences is Meaningless”⁸

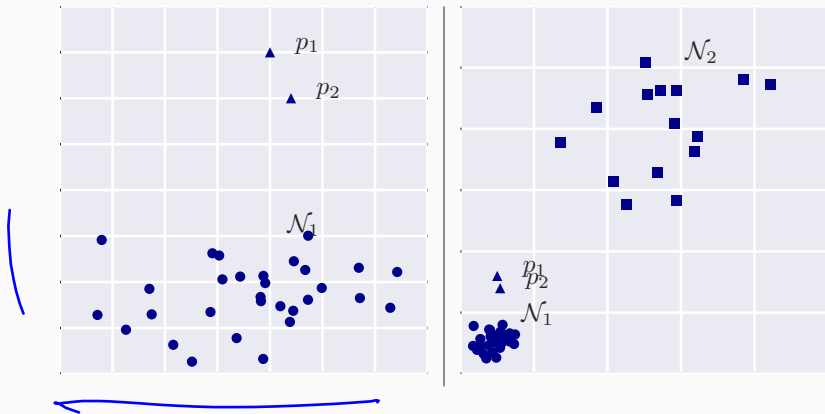
⁷Keogh, Lin, and Fu, “HOT SAX: Efficiently finding the most unusual time series subsequence”.

⁸Keogh et al., “Clustering of Time Series Subsequences is Meaningless:”

Proximity

Data Classification

Global vs Local: Local techniques use neighborhood data



Proximity

Nearest Neighbor

Overlapping windows distort data similarity

abcabcXXXabababc

<i>h = 1</i>	<i>h = 3</i>
<i>abc</i> <i>bca</i> <i>cab</i>	<i>abc</i>
<i>abc</i> <i>bcX</i> <i>cXX</i>	<i>abc</i>
<i>XXX</i> <i>XXa</i> <i>Xab</i>	<i>XXX</i>
<i>abc</i> <i>bca</i> <i>cab</i>	<i>abc</i>
<i>aba</i> <i>bab</i> <i>abc</i>	<i>aba</i>

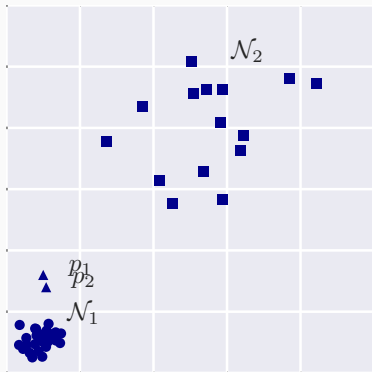
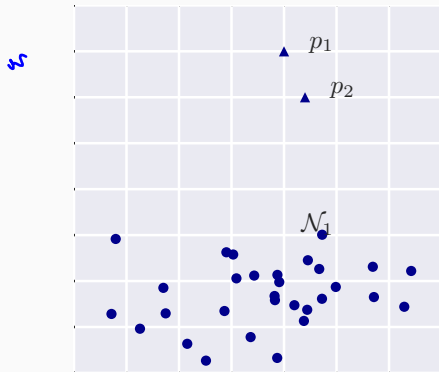
Solution: Use non-self matches⁹

$h = 1$	$h = 3$
<i>abc</i> <i>bca</i> <i>cab</i>	<i>abc</i>
<i>abc</i> <i>bcX</i> <i>cXX</i>	<i>abc</i>
<i>XXX</i> <i>XXa</i> <i>Xab</i>	<i>XXX</i>
<i>abc</i> <i>bca</i> <i>cab</i>	<i>abc</i>
<i>aba</i> <i>bab</i> <i>abc</i>	<i>aba</i>

⁹Keogh, Lin, and Fu, "HOT SAX: Efficiently finding the most unusual time

kNN uses no local information

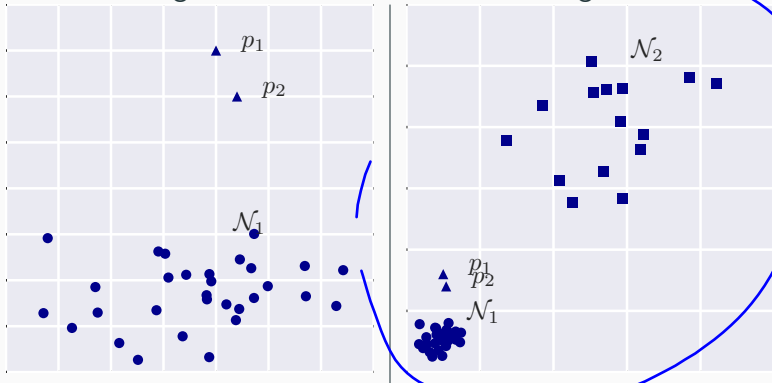
$k=1 \times$



\times

Local Outlier Factor¹⁰ uses local density information

A point is likely to be an anomaly if its neighbors are in dense regions while it is in a less dense region



(Didn't you say anomalies may not be in less dense regions?!)

¹⁰Breunig et al., "Optics-of: Identifying local outliers".

Proximity

Clustering

Clustering algorithms are usually not designed to find anomalies

Assumptions: anomalous points

- do not belong to a cluster (DBSCAN¹¹)
- are far from a cluster centroid
- are in less dense clusters

¹¹Ester et al., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise".

Models

Hidden Markov Models (HMMs) are general advanced sequence modelers

Restrictions:

- fixed length sequences
- Markovian process

Problems with Established Techniques

How to determine a priori what the best algorithm is?

review papers only give subjective assessments

Proximity-based techniques need a lot of decisions


Choose:

- similarity measure
- sliding window size
- sliding window hop
- compatible classification technique

Solution: Use a model-based technique


- characterize normal
- restriction: use when data *can* be modeled

Ideally:

- 
- model arbitrary time series
 - minimize effect of window length
 - requires as few parameters as possible

Recurrent Neural Networks (RNNs)

RNNs are powerful

- 
- speech recognition
 - handwriting recognition
 - music generation
 - text generation
 - ~~handwriting generation~~
 - translation
 - identifying non-verbal cues from speech
 - image caption generation
 - video to text description
 - generating talking heads

RNNs are more flexible and efficient than HMMs

state:

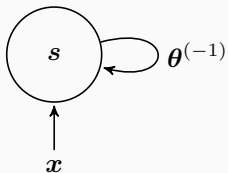
- HMM: hidden state depends only on previous state
- RNN: shared state

generality:

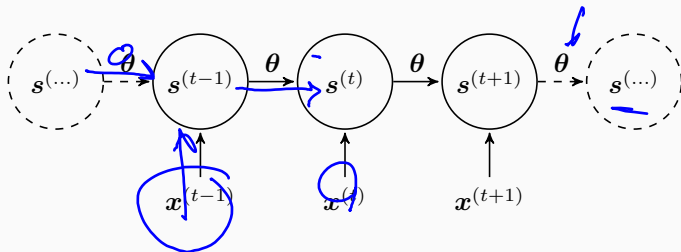
- HMM: Markovian
- RNN: general computation device

Recurrence explains the efficiency of RNN encoding

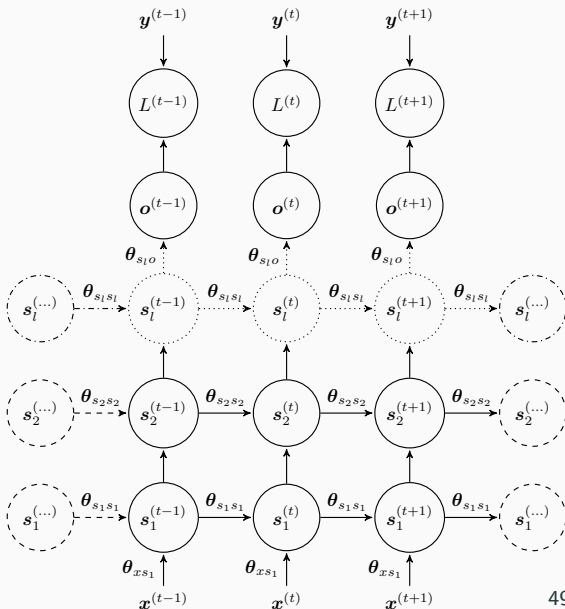
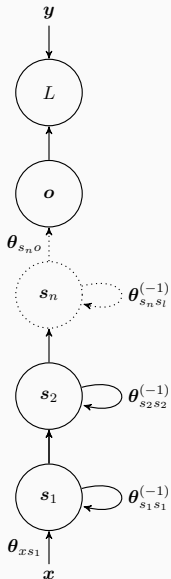
cyclic view



acyclic view



RNN computation is elaborate



Training RNNs is difficult

$$L(\mathbf{o}, \mathbf{y}) = \frac{1}{TV} \sum_t \sum_v (\mathbf{o}_v^{(t)} - \mathbf{y}_v^{(t)})^2$$

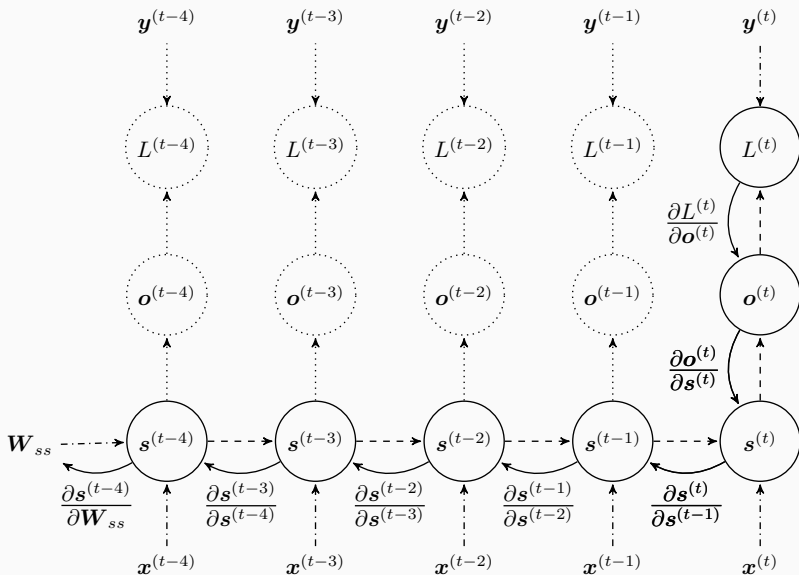
- but mini-batch SGD-flavor training still works

$$\Delta \theta = -\alpha \frac{1}{|M|} \sum_{(\mathbf{x}_m, \mathbf{y}_m) \in M} \frac{\partial L(\mathbf{o}, \mathbf{y}_m)}{\partial \theta} \quad |$$

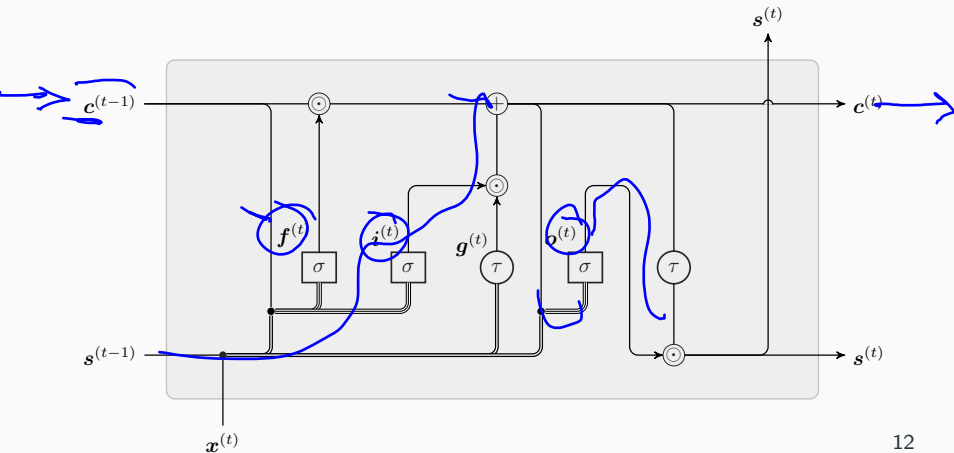
- acute vanishing gradient problem

$$\frac{\partial L^{(t)}}{\partial \mathbf{w}_{ss}} = \sum_{i=0}^T \frac{\partial L^{(t)}}{\partial \mathbf{o}^{(t)}} \frac{\partial \mathbf{o}^{(t)}}{\partial \mathbf{s}^{(t)}} \left(\prod_{j=i+1}^T \frac{\partial \mathbf{s}^{(j)}}{\partial \mathbf{s}^{(j-1)}} \right) \frac{\partial \mathbf{s}^{(i)}}{\partial \mathbf{w}_{ss}}$$

Understand vanishing gradient problem through computational graph for $T = 4$



Long Short Term Memory (LSTM) 'cells' store information but are more complicated than vanilla RNNs' *tanh*



12

¹²Colah, *Understanding LSTM Networks*.

Using RNNs for Anomaly Detection

Use same procedure for test time series to test generality

1. sample
2. setup RNN autoencoder
3. train
4. optimize
5. evaluate anomaly scores

1. Sample with sliding windows of varying length to test versatility

1. spikes

2. sine

3. power demand

4. electrocardiogram (ECG)

5. polysomnography ECG (PSG-ECG)

2. Setup RNN autoencoder

- Set target to (uncorrupted) input

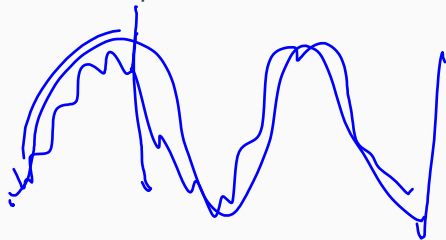
$$\mathbf{y} = \mathbf{x}$$

- Add noise to input

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathcal{N}(0, (0.75\sigma_{\text{std}}(\mathbf{x}))^2)$$

3a. Train: RMSprop is appropriate algorithm

- works with ~~mini batch~~ learning as data is highly redundant
- similar in results to second order methods with less computational cost



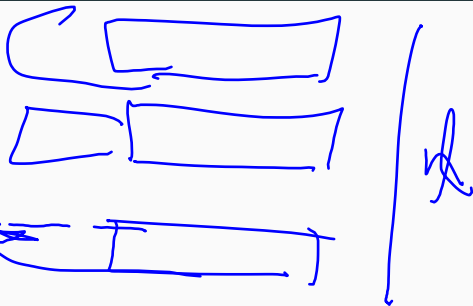
3b. Optimize RNN hyperparameters to find best RNN configuration

Optimize

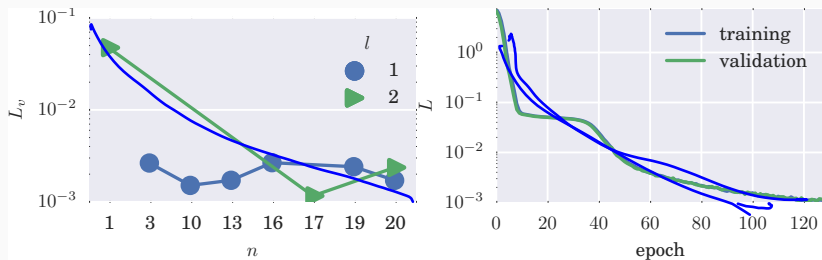
- number of layers, l
- 'size' of each layer, n

using Bayesian optimization

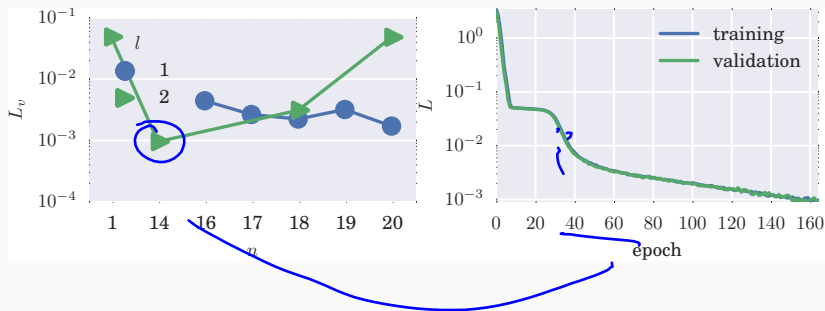
- minimize expensive objective function calls
- considers stochasticity of function



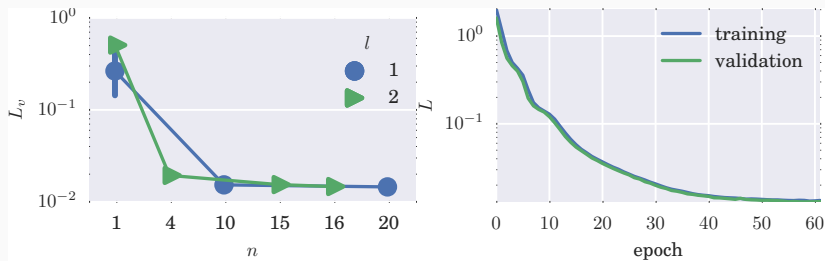
Optimization of spike-1



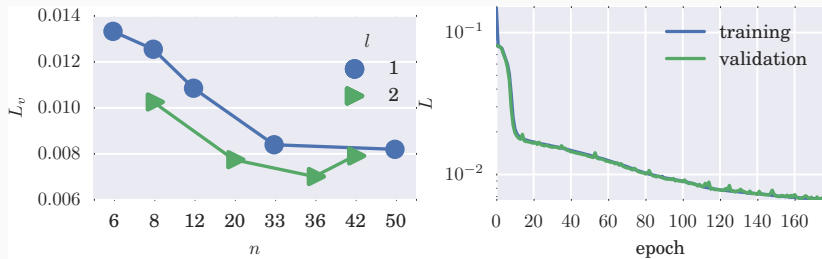
Optimization of spike-2



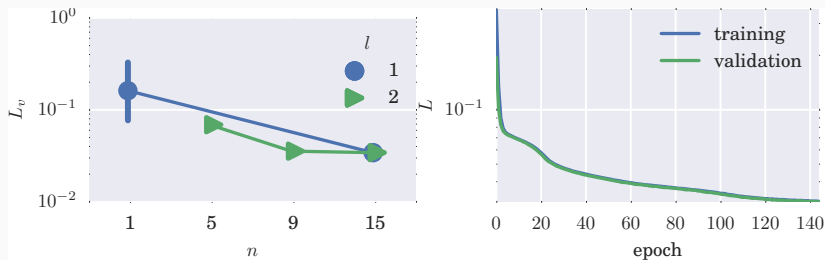
Optimization of sine



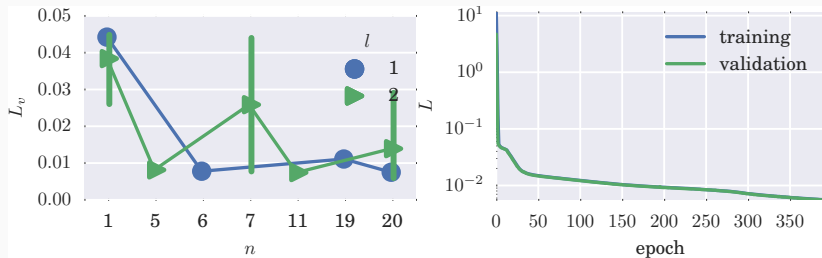
Optimization of power



Optimization of ECG



Optimization of PSG-ECG

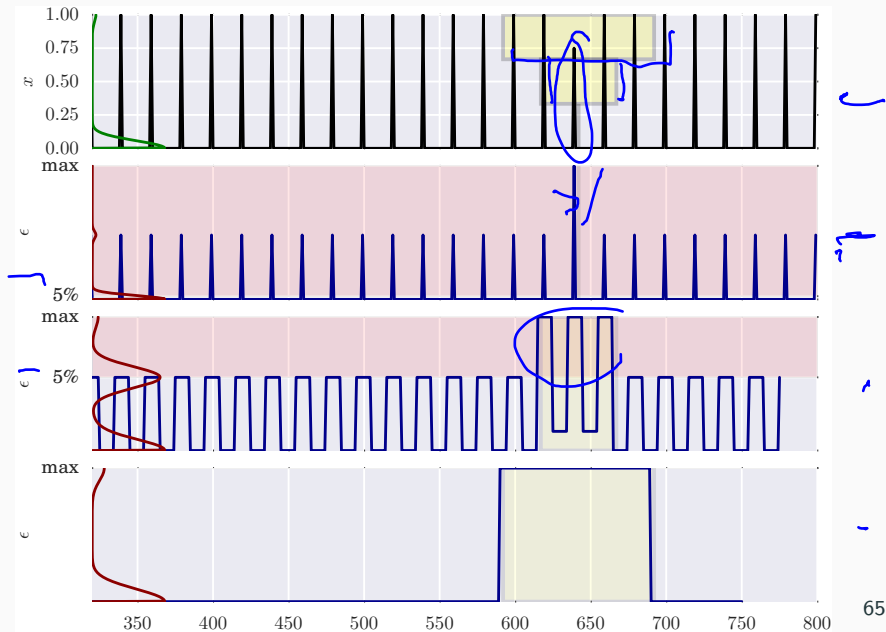


4. Use squared error as an anomaly score

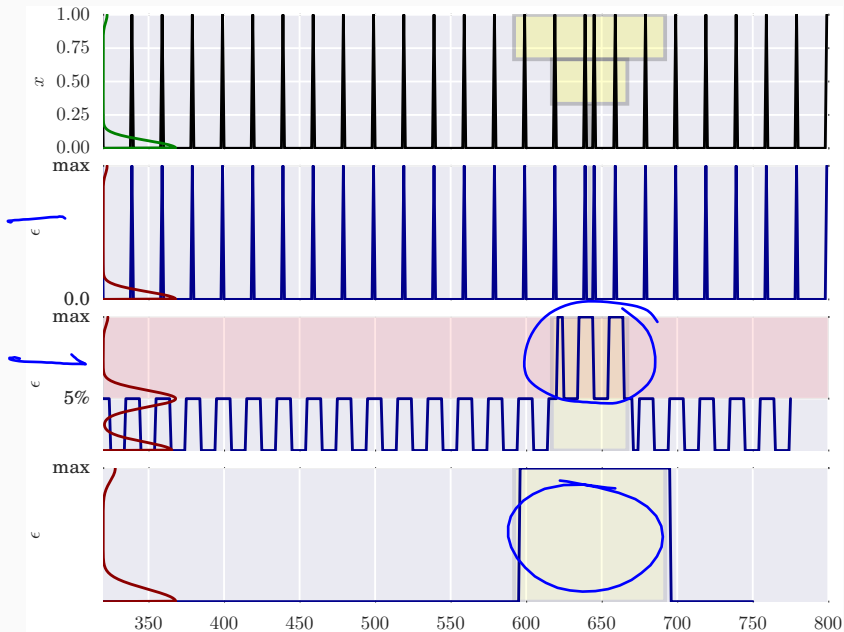
Reconstruction Error

- individual squared error
- mean squared error of a window

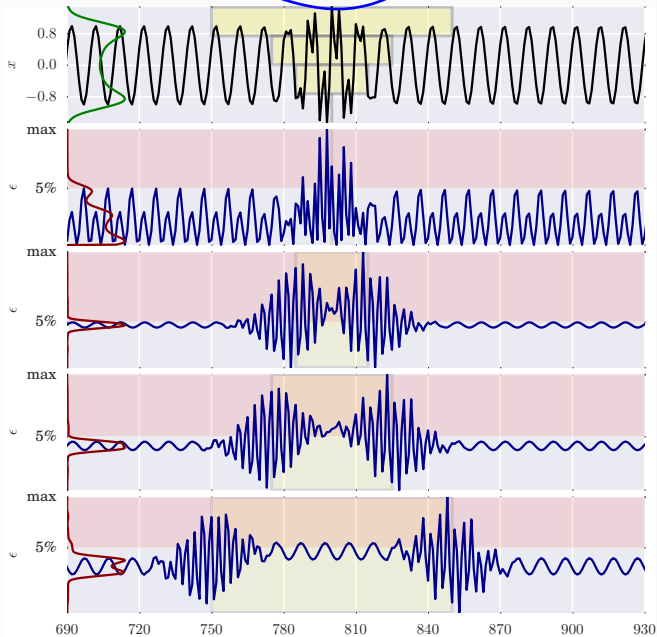
spike-1: atypical value detected



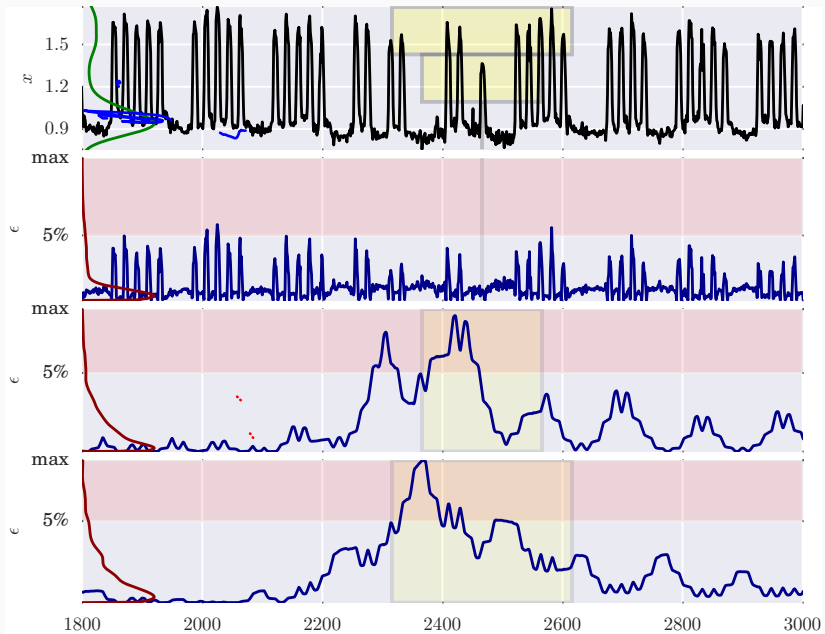
spike-2: irregularity detected



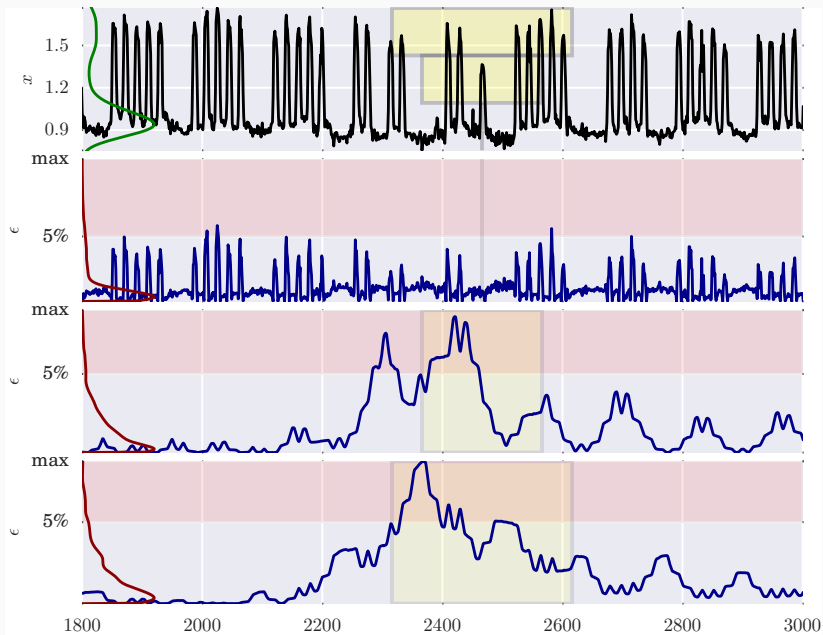
sine: discord detection inconclusive



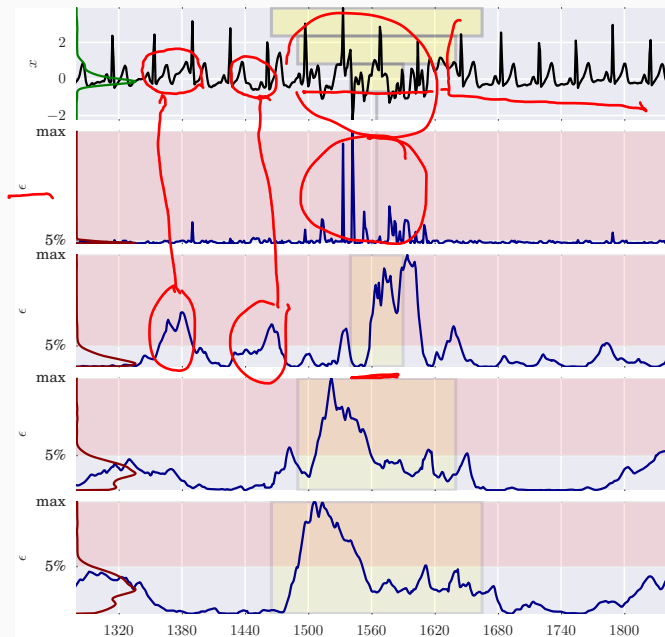
power: discord detected



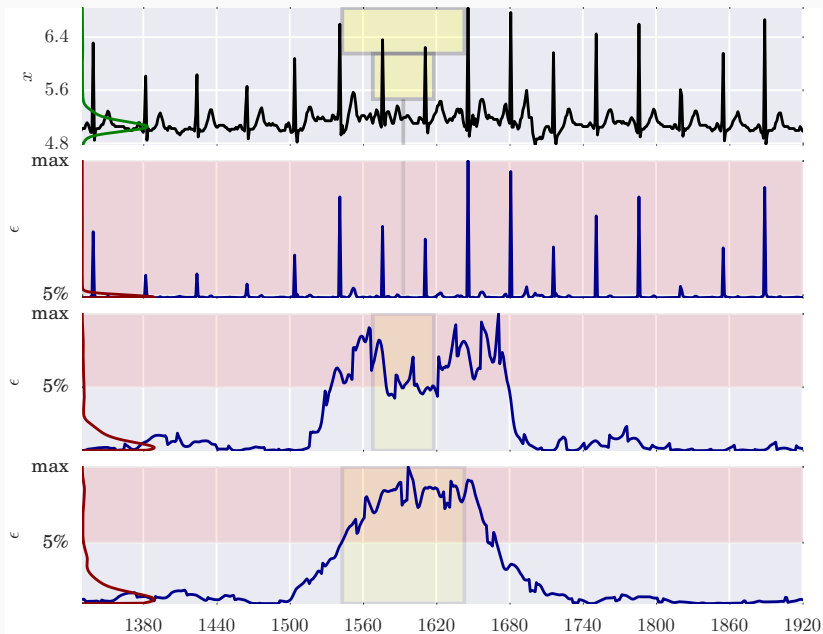
power: discord detected



ECG: discord detected



PSG-ECG: discord detected



Experiment conclusion: squared reconstruction error of AE-RNNs can be used to detect anomalies

- point errors may find extreme values
- windowed errors find anomalies if size is on the order of anomaly
- RNNs were insensitive to translation and length
- RNNs learned normal behaviour despite having some anomalies in the training data
- the same process found anomalies in all tests

Conclusions

RNNs have advantages over advanced techniques

Model-based: HMMs

- more efficient encoding
- varying sequence length

Proximity-based: HOT SAX

- more efficient *after training*
- multivariate
- not forced to find an anomaly

Alternative method checklist

- Is only the test sequence needed to determine how anomalous it is? (Is a summary of the data stored?)
- Is it robust against some window length?
- Is it invariant to translation? (Is it invariant to sliding a window?)
- Is it fundamentally a sequence modeler?
- Can it handle multivariate sequences?
- Can the model prediction be associated with a probability?
- Can it work with unlabeled data? If not, is it robust to anomalous training data?
- Does it require domain knowledge?

Main disadvantage of RNNs: computational cost

- training
- hyperoptimization

Further work is needed to strengthen the case for using RNNs for anomaly detection

- better optimize
- use autocorrelation to determine minimum window length
- accelerate training: normalization, optimum training data size
- use drop out to guard against overfitting
- experiment with RNN architectures: bi-directional RNNs, LSTM alts., more connections
- incorporate uncertainty
- objective comparisons with labelled data
- try multivariate series



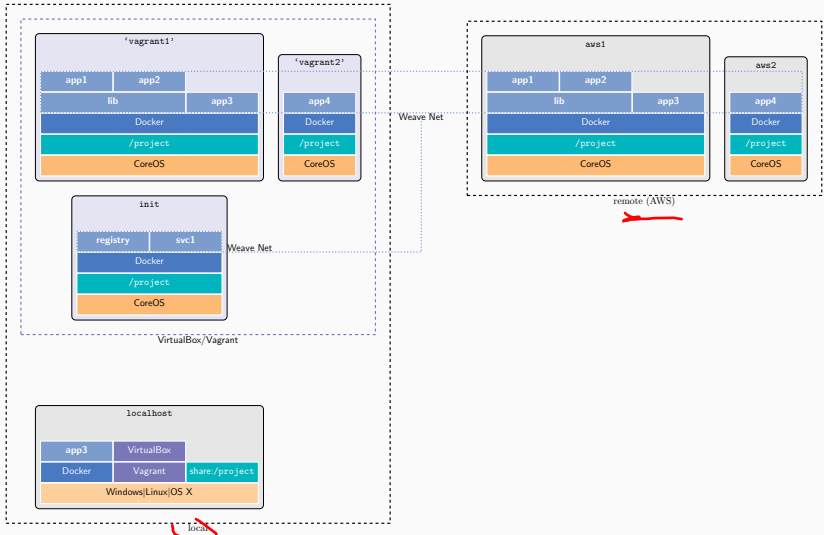
Use RNNs to find anomalies when computational cost can be managed.

Reproducibility

Technology stack enables automation and reproducibility

application
container network	Weave	
app. containerization	Docker	
operating system	CoreOS	
machine	(x64)	x64
hypervisor	VirtualBox	...
hypervisor interface	Vagrant	AWS
host operating sys.	Windows OS X Linux	...
hardware	x64	x64
	local	remote

Reproducibility of technology stack on any machine enables parallel processing



Discussion Time
