An intelligent Alarm System for time series database based on Machine Learning Techniques

Dec 2, 2015

HANBAT NATIONAL UNIVERSITY

SINCE 1921

Mai Tuan Hai

Advisor: Kim Young Chan

Contents

- Introduction
- Anomaly Detection
- Deep Learning
- Intelligent Alarm System
- Evaluation
- Conclusion

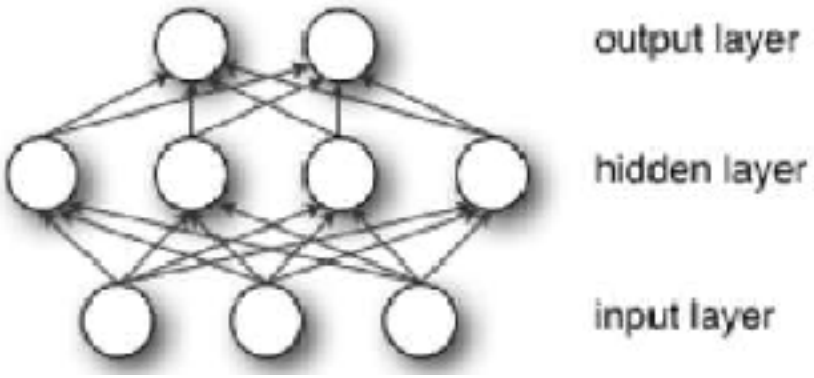
Introduction

- In machine learning, time series data anomaly detection is one of major challenges.
- There are two major approaches:
  - Shallow architecture: it extends the complexity of presentation by “horizontal way”.
  - Deep architecture: it extends the complexity of the presentation by “vertical way”.
- Classification based on deep learning is called “human classification”.

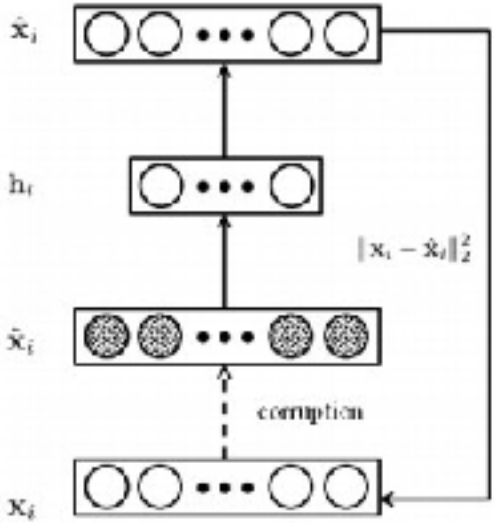
Shallow vs. Deep Learning

- Most AI has been shallow architectures:
  - 1->3 layers of transformation
- Deep architectures just do more:
  - 4->7 layers (or more) of transformation

Multi Layer Perceptron



Denoising Autoencoders



Contribution

- We propose an intelligent alarm system for anomaly detection in time series database.

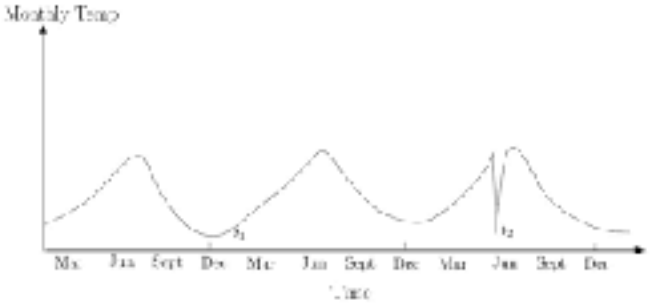
- This system uses:
  - Threshold Concept in Anomaly Detection.
  - Auto-Encoders in Deep Learning.
  - Fast computation of classification.

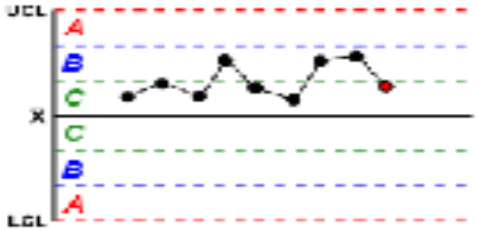
An intelligent Alarm System for time series database based on Machine Learning Techniques

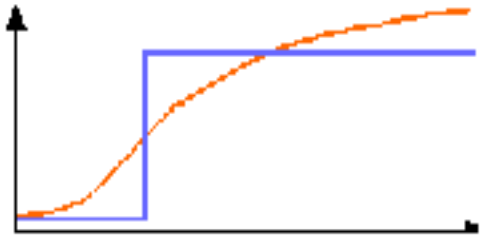
ANOMALY DETECTION

What is anomaly?

- Something that deviates from what is standard, normal, or expected..
- In time-series:
  - Detect contextual anomalies.
  - Detect anomalous subsequence.
  - Detect anomalous time series based on another time series.





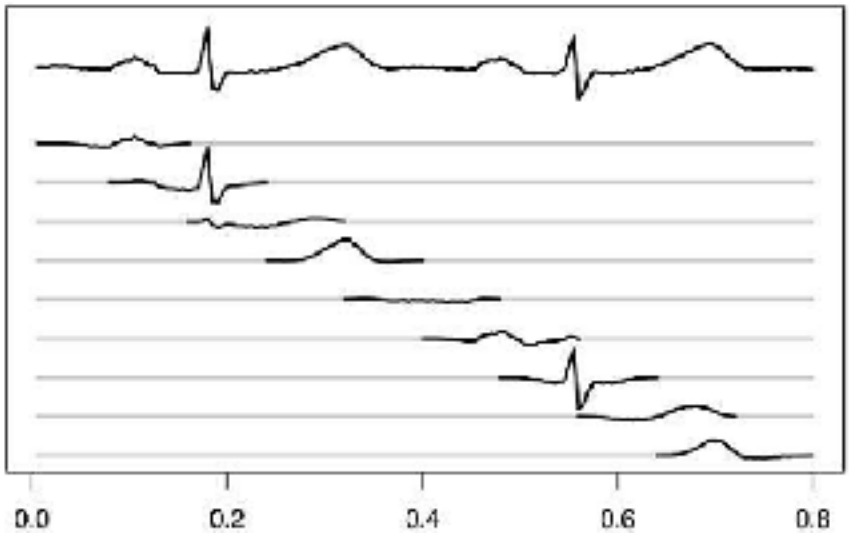


Anomaly Score

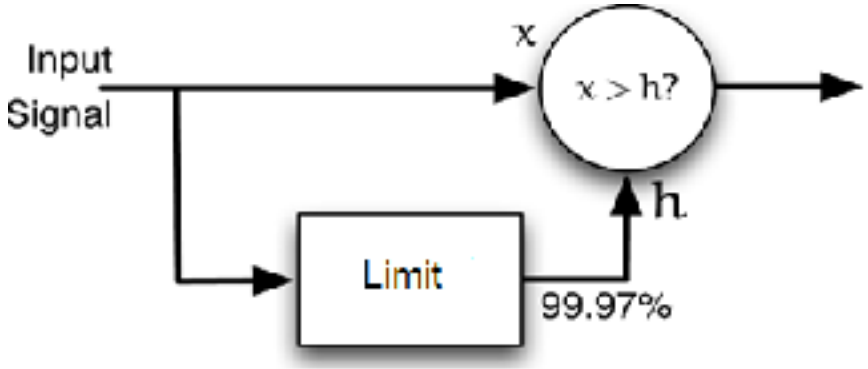
- Anomaly score is to identify how things are "far“ from normal. The bigger scores, the father things are.
- There are two major methods:
  - Popular Anomaly Score: Aggregation of anomaly scores from individual points or subsequences of time-series
  - Advanced Anomaly Score: Decompose a given time-series data into many short segments. The aggregation is much more complex than the first way.

Threshold Models

Decomposition of a time-series



Threshold of Anomaly Scores

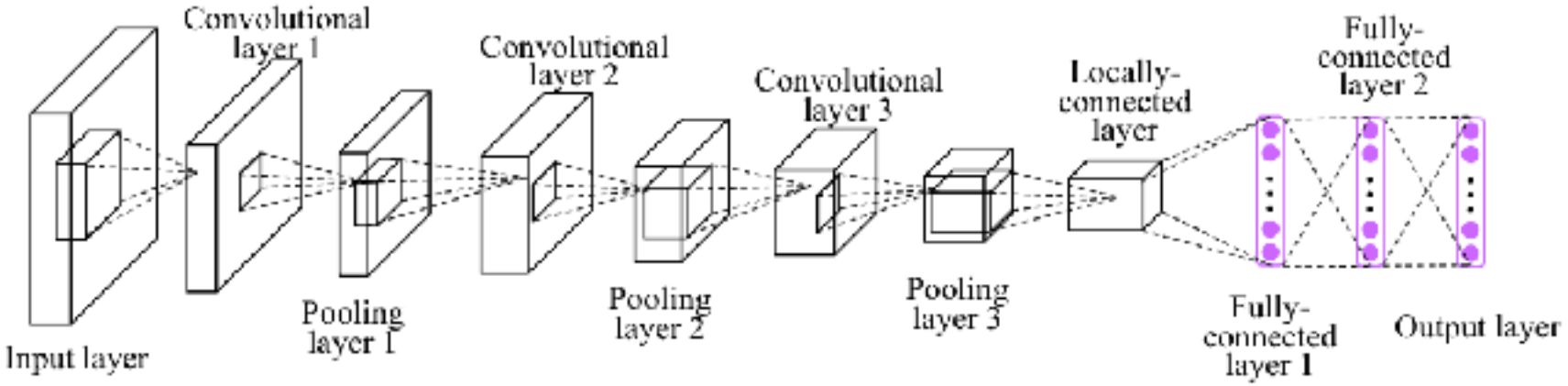


An intelligent Alarm System for time series database based on Machine Learning Techniques

DEEP LEARNING

What is Deep Learning?

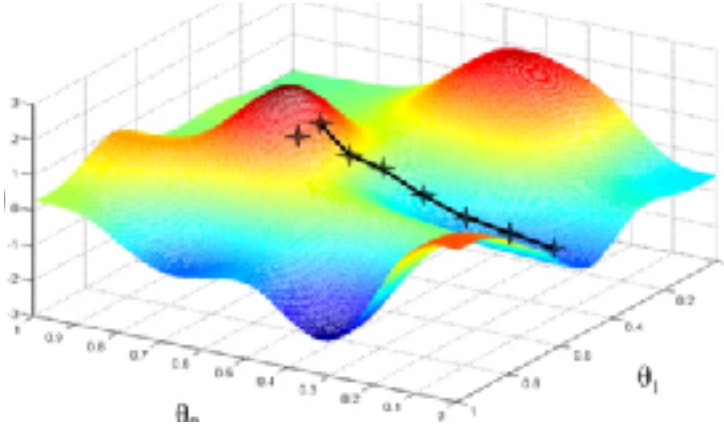
- Deep learning is a set of algorithm in machine learning that attempt to learn in multiple levels, corresponding to different levels of abstraction. It typically uses artificial neural networks.
- The levels in these learned statistical models correspond to distinct levels of concepts, where higher-level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts



Preliminaries

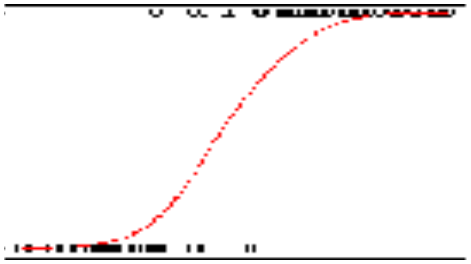
- Regularization
  - After training our model does really well, but with new examples that our model has never seen then it performs really bad.
  - That is over-fitting, to combat with it, common way is regularization which can be done by L1/L2 regularization and Early-Stopping

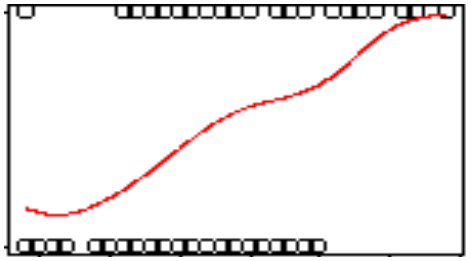
- Batch-Based Learning
  - Gradient Descent (GD)
  - Stochastic Gradient Descent (SGD)
  - Mini-batch SGD



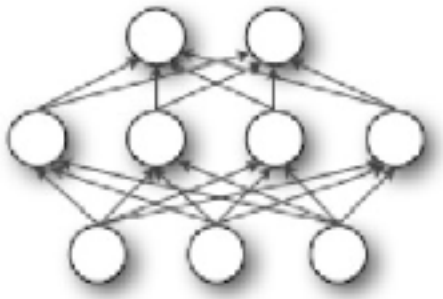
Deep Learning Classifier

- Logistic Regression
  - Linear regression
  - Log Presentation
  - Multi Logistic Regression



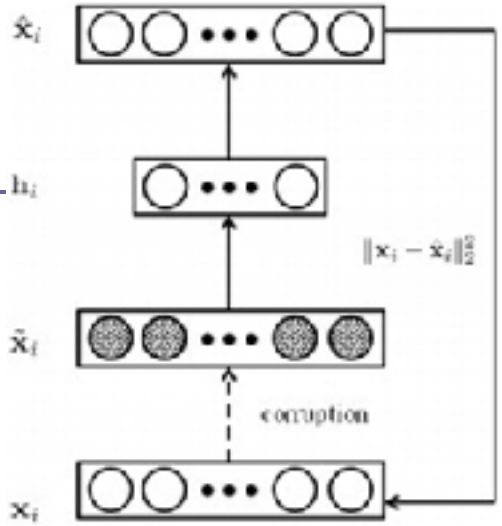


- Multi-Layer Perceptron
  - An MLP can be viewed as a logistic regression classifier where the input is first transformed using a learnt non-linear transformation . This transformation projects the input data into a space where it becomes linearly separable. This intermediate layer is referred to as a hidden layer

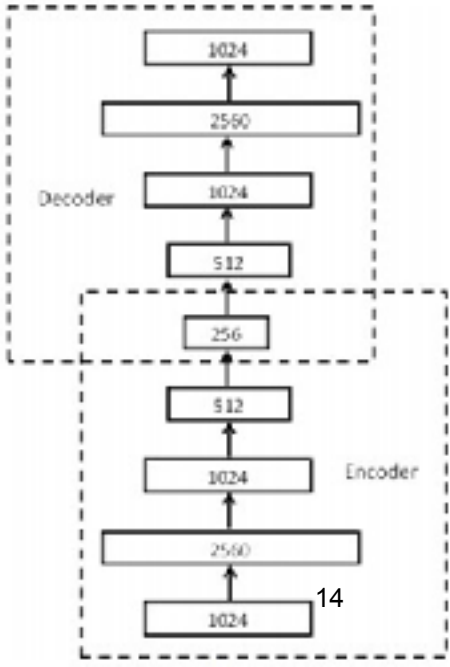


Deep Learning Classifier

- Denoising Autoencoders(dA)
  - To force the hidden layer to discover more robust features and prevent from learning identity, we will train the auto-encoder to reconstruct the input from a corrupted version of it.



- Stacked Denoising Autoencoders(SdA)
  - .
    Denoising auto-encoder can be stacked to develop a deep network by feeding latent representation of the denoising auto-encoder that stays above the layer below as input to the current layer



An intelligent Alarm System for time series database based on Machine Learning Techniques

INTELLIGENT ALARM SYSTEM

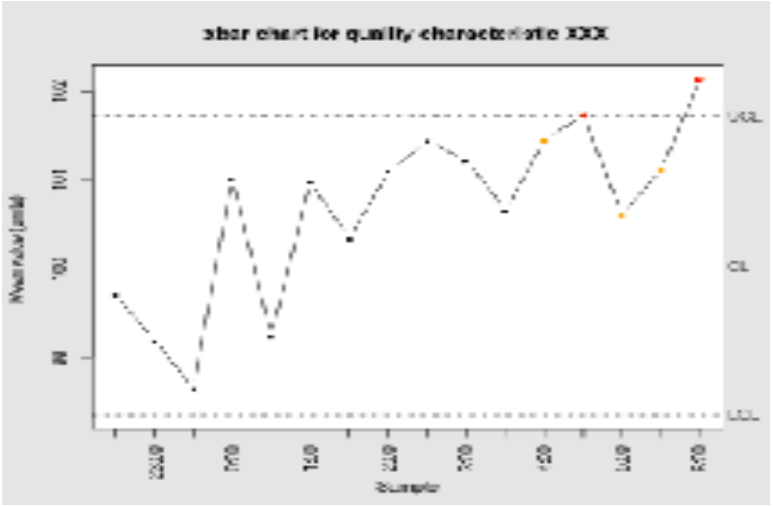
Alarm System

- Definition
  - An alarm system will help operator not only monitor and control plant but also protect people from any fault of system or disasters

- Issues of alarm system
  - Alarm floods
  - False alarms
  - Complex settings

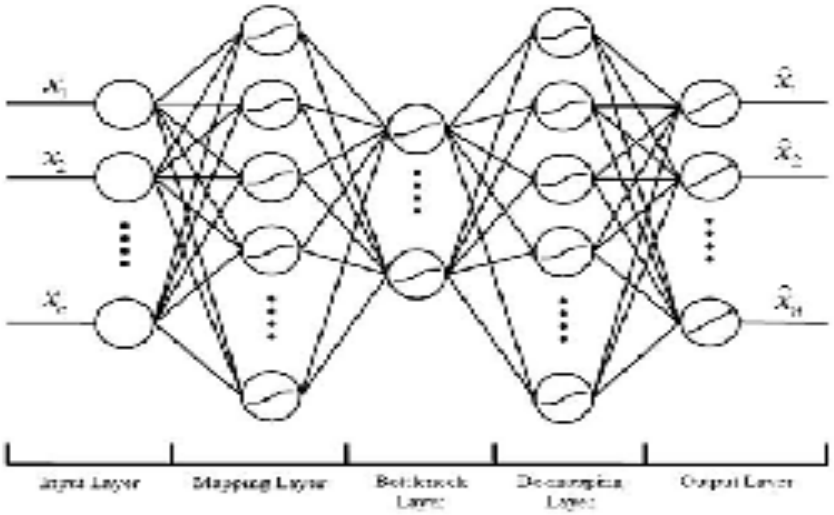
Control chart

- Definition
  - Statistical Process Control (SPC) charts also known as Shewhart Charts have been used widely in many processes.



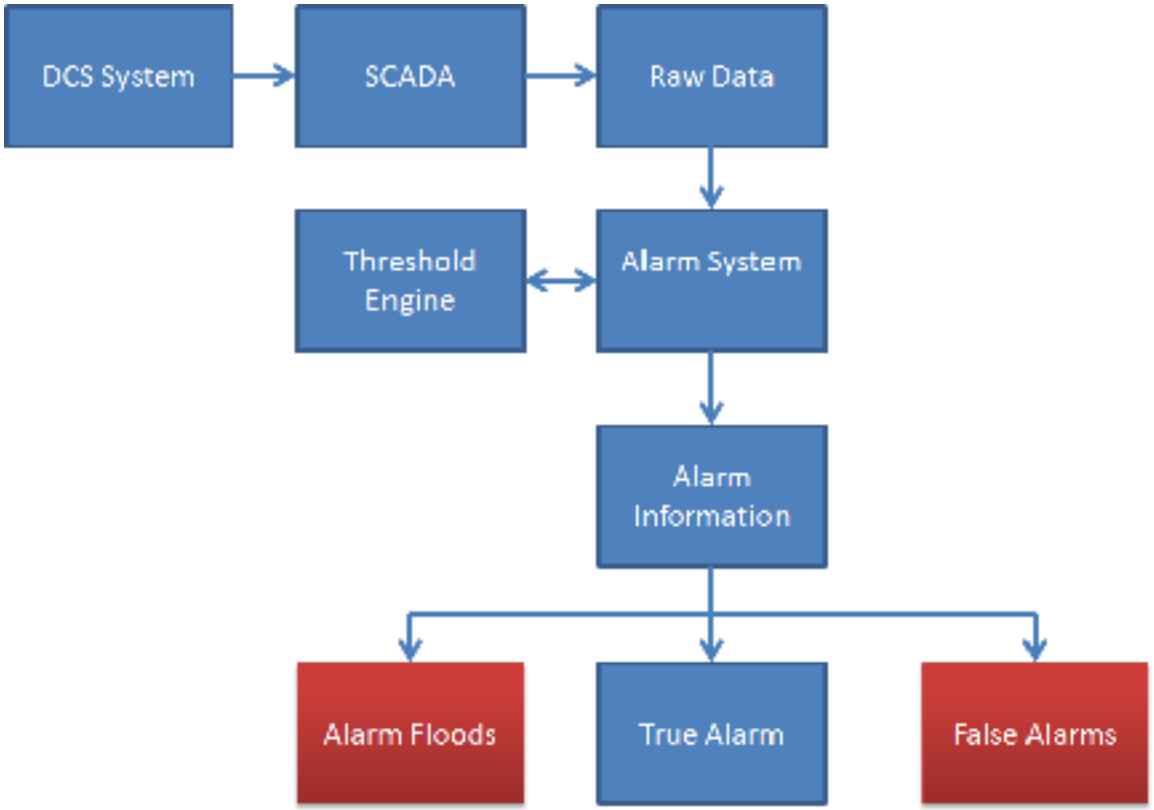
Previous works

- Common alarm based on threshold processing
- Advanced Alarm system based on:
  - Support Vector Machine
  - Auto-Associative Kernel Regression
  - Auto-Associative Neural Network



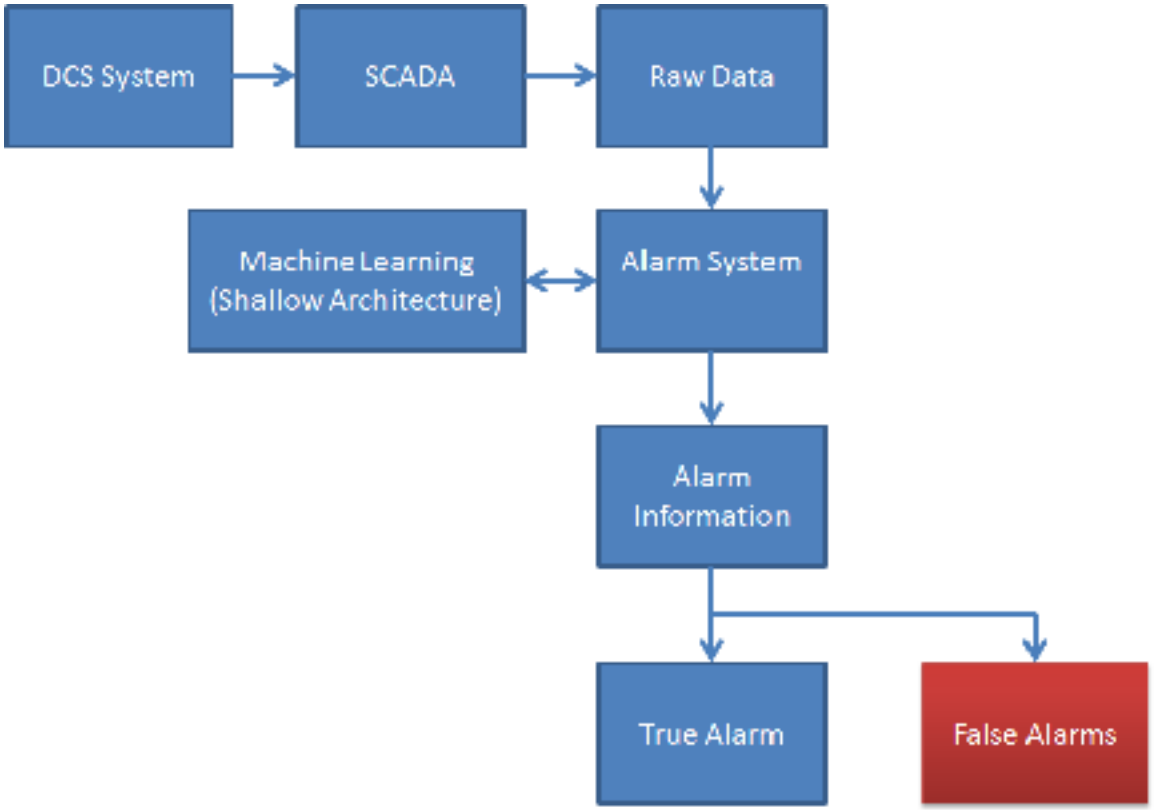
Common Alarm System

- Common alarm based on threshold processing



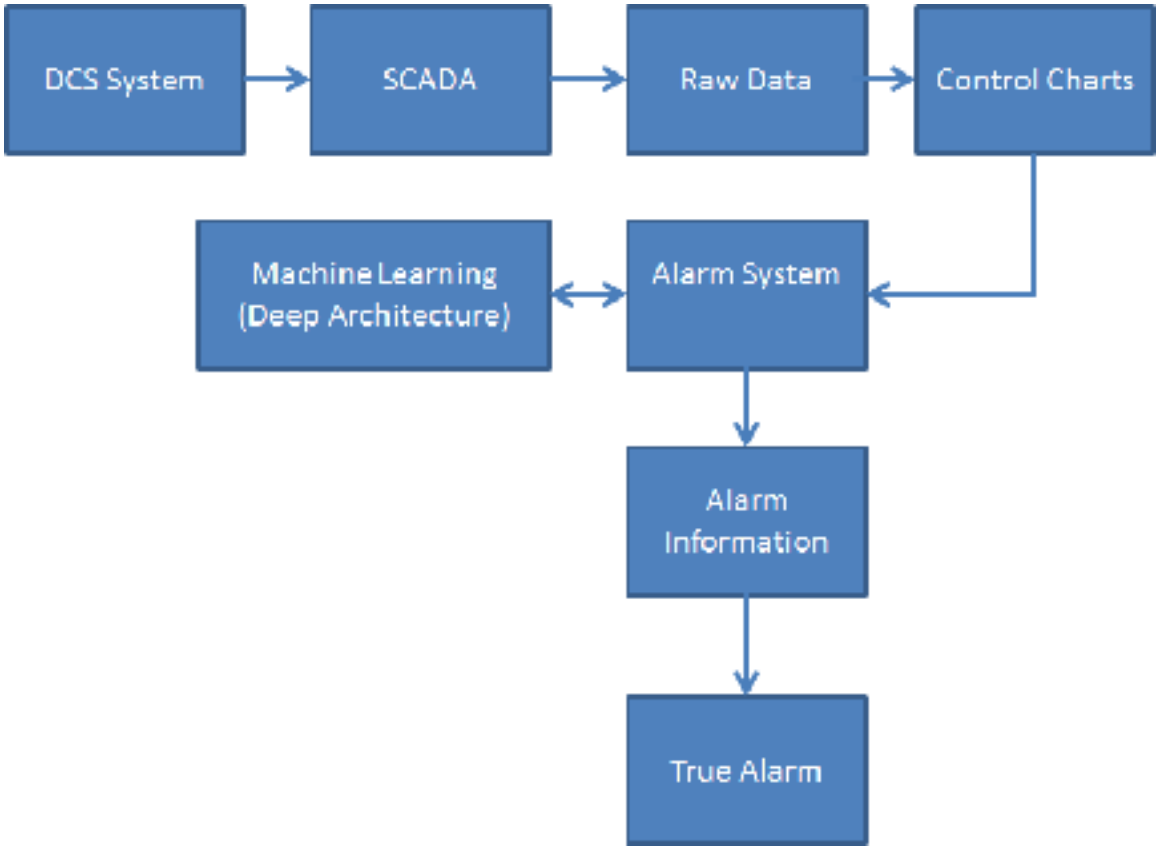
Advanced Alarm System

- Based on shallow architecture



Proposed Alarm System (1)

- Based on deep architecture



한밭대학교
HANBAT NATIONAL UNIVERSITY

Proposed Alarm System (2)

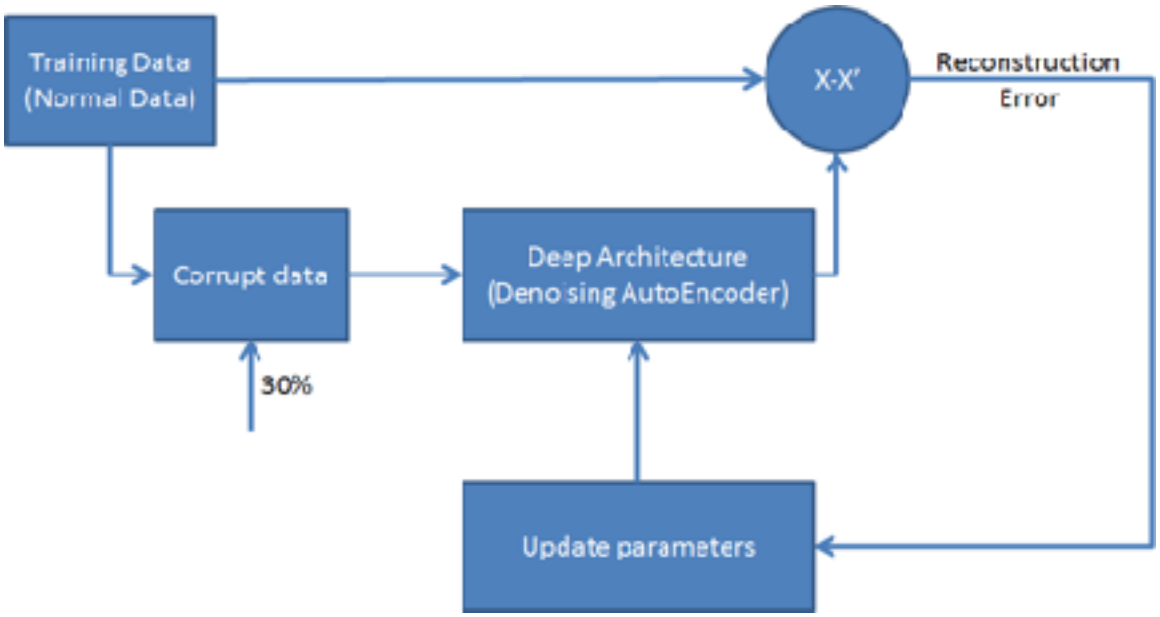
- Using Feature
  - Control Chart is built based on subgroup average.

  => System will be much more stable

- Deep Architecture
  - Can capture the characteristics of signal completely.
  - Deep learning has been widely used in research area but no in process industry.

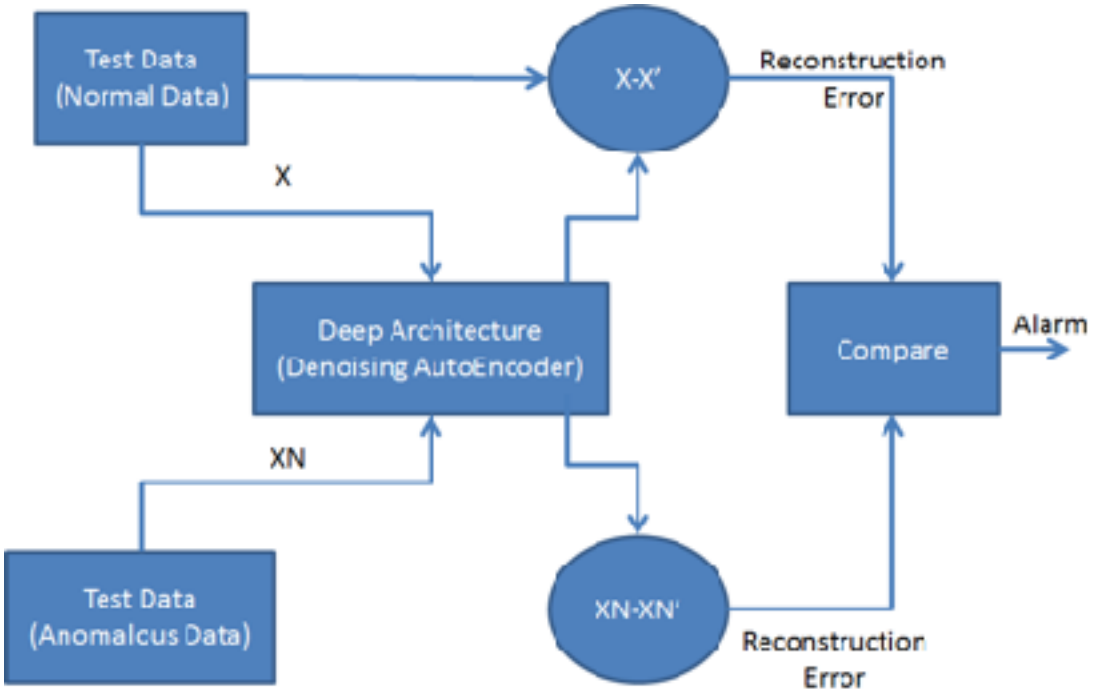
Proposed Alarm System (3)

- Denoising Architecture
  - Corruption level is 30%, 30% input will be set to 0 randomly.
  - Visible layer neuron: 60, hidden layer 100
    => “over complete”



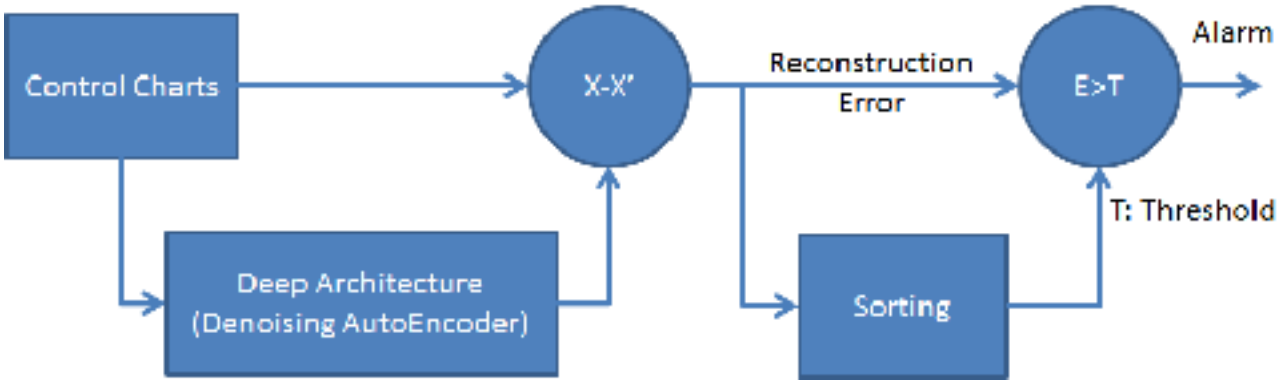
Proposed Alarm System (4)

- Anomaly Detection
  - Reconstruction error is used as a main feature to detect anomaly.
  - Test Model will not update parameters.

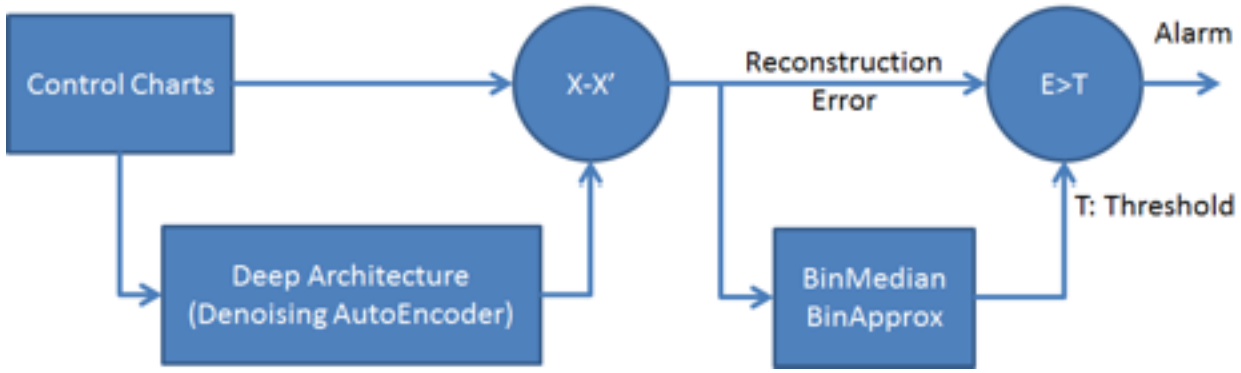


Proposed Alarm System (5)

- Normal Threshold Mechanism



- Threshold with BinMedian/BinApprox

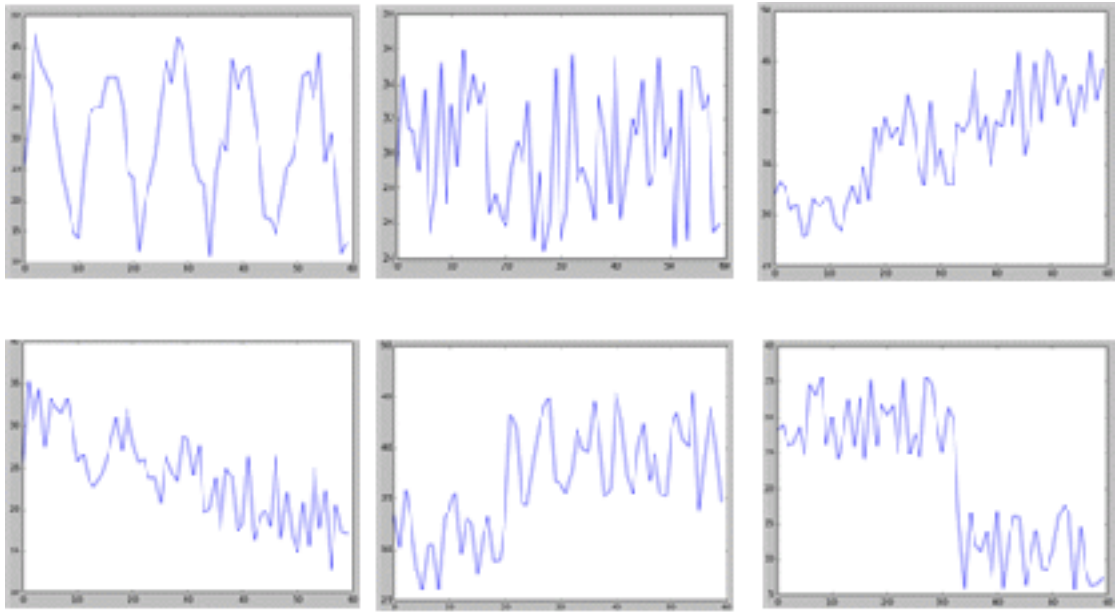


Evaluation (1)

- Control chart dataset

Training dataset	
Sample Data	**Size**
Normal Dataset	49,900

Testing dataset	
Sample Data	**Size**
Normal Pattern	100
Cyclic Pattern	100
Upward Trend Pattern	100
Downward Trend Pattern	100
Upward Shift Pattern	100
Downward Shift Pattern	100



Evaluation (2)

- Anomaly Scored based on Auto-Encoders
  - Formula

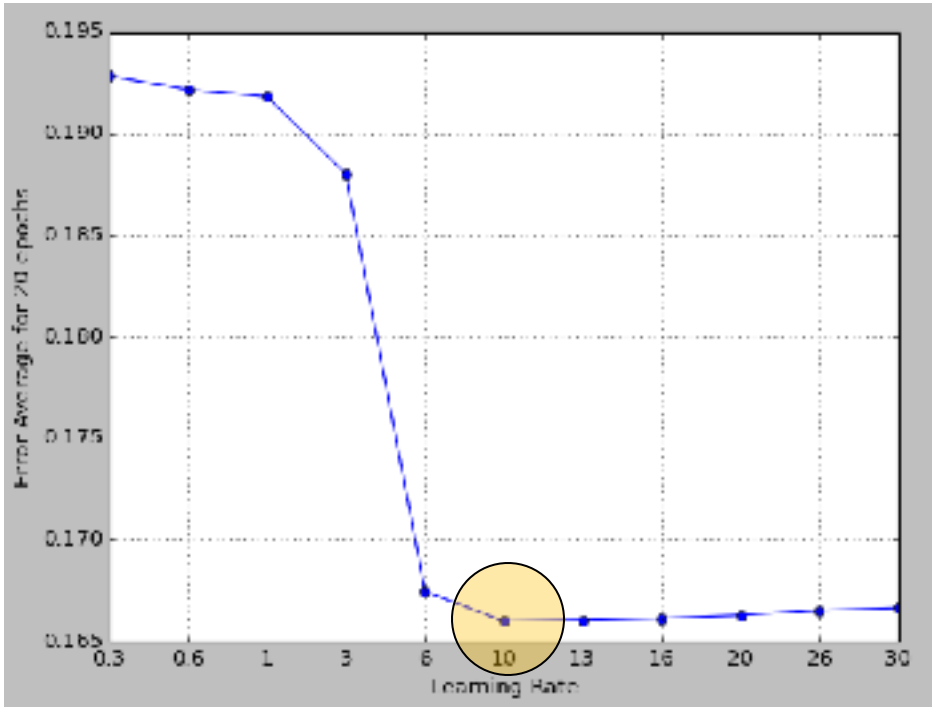
$$\text{cost} = \sqrt{\frac{\sum_{1}^{n}(x - x')^2}{n}}$$

  - In Python

```
def get_cost(self):

    y = self.get hidden values(self.x)
    z = self.get_reconstructed_input(y)

    L = z-self.x
    cost =T.sqrt(T.mean(L**2))

    return cost
```

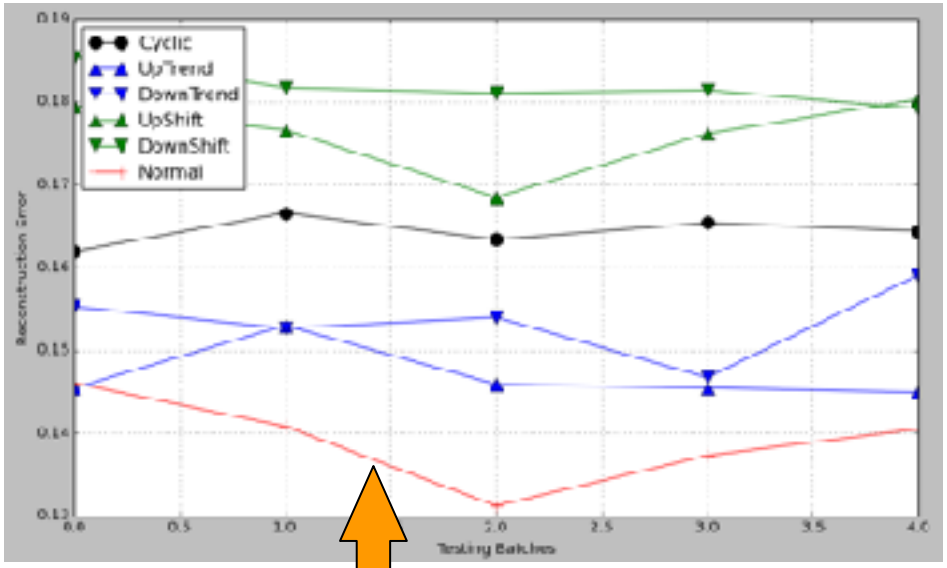
Evaluation (3)

- After some epoch number, we select epochs =20 and batch-size =20 are reasonable number
- Now how to select learning_rate
  - *learning_rate=[0.3,0.6,1,3,6,10,13,16,20,26,30]*
  - *training_epochs = 20*
  - *batch_size =20*

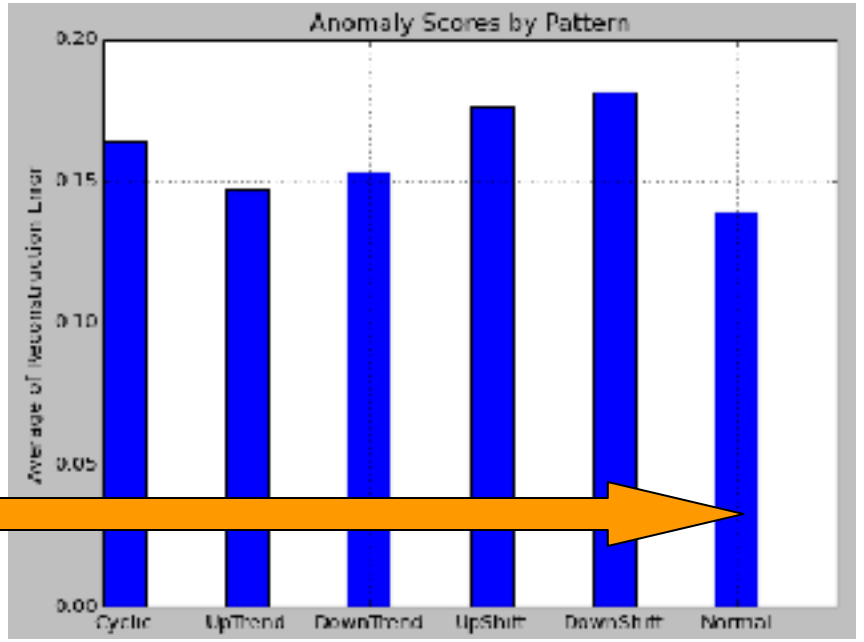


Evaluation (4)

- Anomaly Detection



This is Normal



Conclusion

- Control Chart Pattern Recognition
  - Control Chart Pattern Recognition (CCPR) is very important tool to help process engineers know how well processes work by using the corresponding assignable causes
- Deep learning
  - called "human classification" because it is inspiring from the way human brain process information. Recently, It is proven that deep learning can capture the characteristic which any of machine learning system based on shallow architecture cannot.
- Our Experiment
  - clearly demonstrates that deep learning (denoising autoencoder) can detect anomalous data effectively even with the anomalous patterns which are very similar to normal data. The threshold of system can be set by sorting or BinMedian in case of very big data.


Q
&
A

Proposed Alarm System (etx)

BinMedian Algorithm[17]:

Given data points x1, . . . xn and assuming that n is odd1, the basic strategy of the binmedian algorithm is:

1. Compute the mean μ and standard deviation σ
2. Form B bins across [μ −σ+ σ], map each data point to a bin
3. Find the bin b that contains the median
4. Recurse on the set of points mapped to b

Where B is a fixed constant

BinApprox Algorithm[17]:

Binapprox is a simple approximation algorithm derived from binmedian. It follows the same steps as binmedian, except that it stops once it has computed the median bin b, and just returns the midpoint of this interval.

1. Compute the mean μ and standard deviation σ
2. Form B bins across [μ −σ+ σ], map each data point to a bin
3. Find the bin b that contains the median
4. Return the midpoint of bin b

The median can differ from this midpoint by at most 1/2 the width of the interval, or 1/2 · 2σ= σ. Since we use B = 1000 in practice, binapprox is accurate to within 1/1000th of a standard deviation.