ifip SEC2021
OSLO NORWAY

Revitalizing Self-Organizing Map:
Anomaly Detection using Forecasting Error Patterns

IFIP SEC 2021, 24 June 2021

Young Geun Kim[1] Jeong-Han Yun[3] Siho Han[2]
Hyoung Chun Kim[3] Simon Woo[2]

[1]Department of Statistics, Sungkyunkwan University, Seoul, South Korea

[2]Department of Applied Data Science, College of Computing and Informatics, Sungkyunkwan University, Suwon, South Korea

[3]The Affiliated Institute of ETRI, Daejeon, South Korea

Attacks on CPSs

- Cyber-Physical Systems (CPSs) are susceptible to various types of anomalies
  1. Attacks on controllers, networks, or cyber-physical elements
  2. Hardware failures, operator errors, and software misconfigurations
- Anomaly detection in CPS
  1. Actual anomalies
  2. Glitches

Setting

- Two periods of data
  1. Training dataset (normal): Since CPS hardly collect anomalous observation, we only train on the normal pattern data
  2. Test dataset (normal + anomaly): observations will be updated in real-time
- After training a model using training set,
- our goal is to detect contextual anomalies in the test period using the trained model in real time

Out-of-Limit (OOL) Threshold

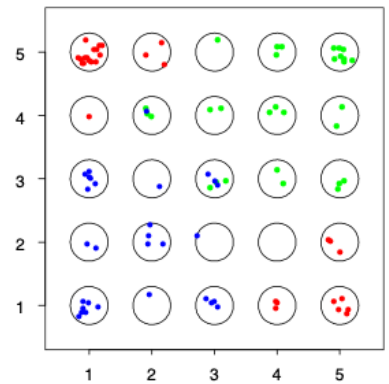
1. Rule-based or statistical machine learning-based forecasting model from the given training data [Giraldo et al., 2018, Filonov et al., 2017, Kim et al., 2019]
2. Anomaly score is computed from the forecasting error (FE)
3. The observation is considered anomalous if the score exceeds the anomaly score (OOL threshold)
   - Static threshold: p-norm [Filonov et al., 2016, Filonov et al., 2017, Kim et al., 2019]
   - Cumulative sum (CUSUM) method [Goh et al., 2017]: divide the time series into the fixed window intervals and computes the sum of the p-norm

Contribution

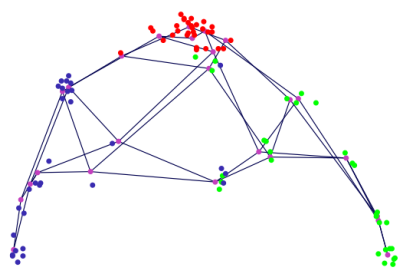
- Novel Self-Organizing Map-based anomaly detection framework
- Detect well unseen anomalies in high-dimensional CPS data in real-time
- Conduct experiments on benchmark CPS datasets: SWaT [Goh et al., 2016] and HAI [Shin et al., 2020]
- Experiments show average 36% increase in the time series-aware F_1 score compared to those of baseline approaches (static threshold and CUSUM method)

Self-Organizing Maps (SOM)

- By [Kohonen, 1982]
- Artificial Neural Network structure that only needs computing distances
- SOM maps observations to topological maps with finite number of prototypes called Kohonen neurons (SOM grids)
- Each grid has its own vector called the codebook in the input space



(a) Points projected to each prototype [Hastie et al., 2009]



(b) Wiremesh representation [Hastie et al., 2009]

Lindeberg-Feller Theorem

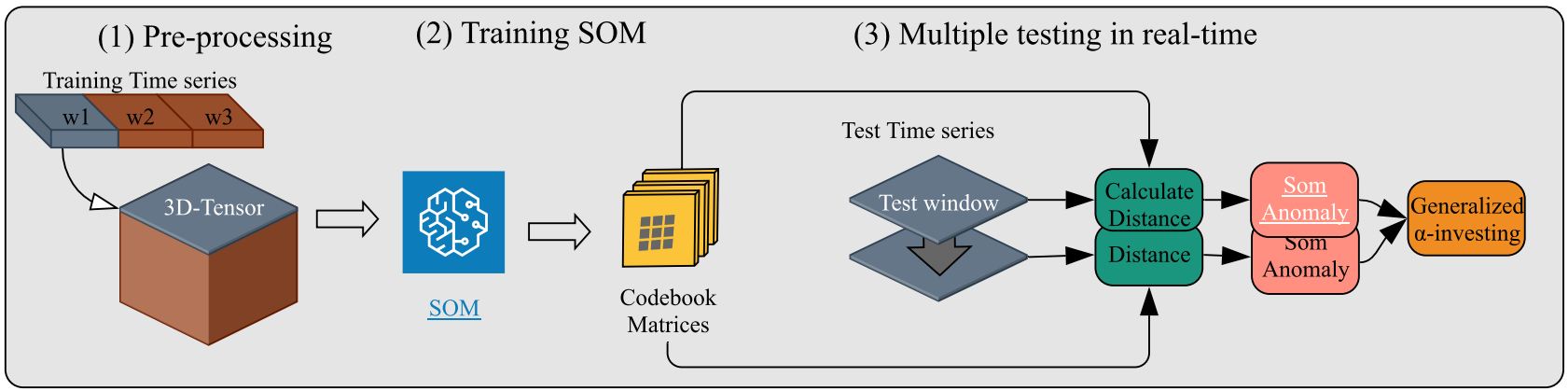
- Apply Central Limit Theorem (CLT) to provide a statistical foundation for setting the anomaly threshold
- Triangular array of random variables $\{X_{nj}\}_{j=1}^{n}$
  - X_{nj} independent for each n
  - $E[X_{nj}] = 0$, $Var[X_{nj}] = \sigma_{nj}^2 < \infty$
  - Let $Z_n = \sum_{j=1}^{n} X_{nj}$ and $B_n^2 = \sum_{j=1}^{n} \sigma_{nj}^2$
- Lindeberg-Feller theorem [Lindeberg, 1922, Ferguson, 1996]
  - Generalization of CLT
  - Lindeberg condition: for every $\epsilon > 0$,

$$\frac{1}{B_n^2} \sum_{j=1}^{n} E\left[X_{nj}^2 I\left(|X_{nj}| \geqslant \epsilon B_n\right)\right] \to 0$$

  - Lindeberg-Feller theorem: weakly convergence

$$\frac{Z_n}{B_n} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1)$$

Overall Pipeline

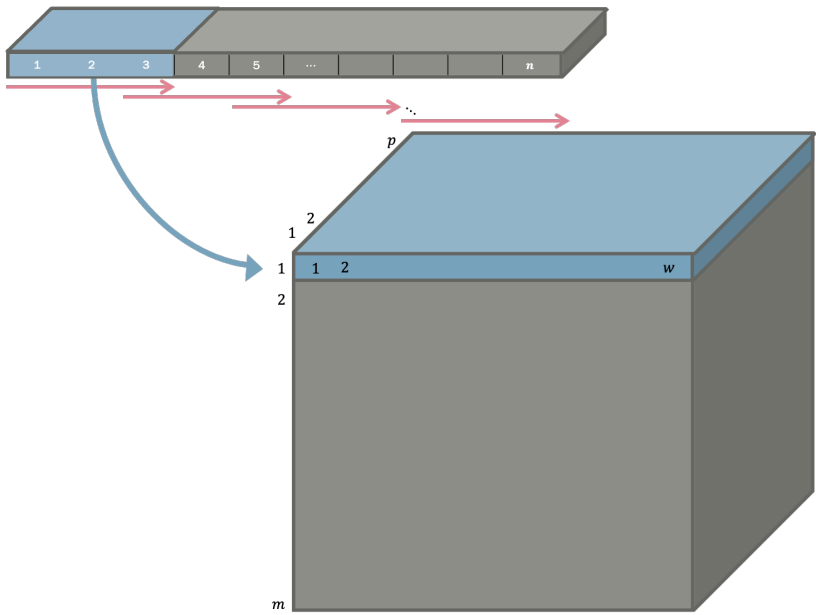


1. Pre-processing
2. Training the SOM
3. SomAnomaly statistic for multiple testing

3D Tensor Input

Given Error data (p-dimensional time series with sample size n)

1. Slide the window of size w with a shift size s
2. Combine the windows into a 3D tensor of size $m \times w \times p$, where $m = \frac{n-w}{s} + 1$



Pre-processing Multivariate Time Series

SOM for the Matrix

- The error pattern data is extended from a vector to a matrix
- For matrix computation, we consider the Frobenius norm
- and corresponding distance function between $A = (\alpha_{jk}), B = (\beta_{jk}) \in \mathbb{R}^{w \times p}$

$$d(A, B) = \left(\sum_{j,k} (\alpha_{jk} - \beta_{jk})^2 \right)^{1/2}$$

- Replace distance function in the incremental SOM algorithm

Incremental SOM training algorithm using 3D tensor

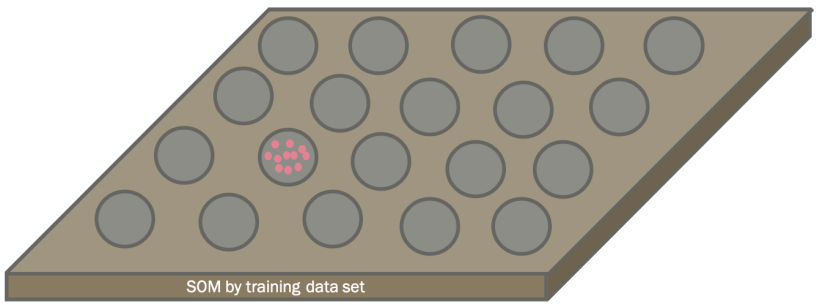
Data: 3D tensor for error $[X_1, \ldots, X_m] \in \mathbb{R}^{m \times w \times p}$
Input: SOM parameters
Initialize learning rate and radius
Initialize codebook matrices
Compute the distance $r_c - r_i$ between nodes c and i in the SOM space
for $j \leftarrow 1$ **to** N **do**
    Randomly choose an input observation
    for $j \leftarrow 1$ **to** N **do**
        if $r_c - r_j \leqslant \sigma(t)$ **then**
            Update the neighboring node of BMU by

$$W_j(t+1) = W_j(t) + \alpha(t)h(r_c - r_j)[X(t) - W_j(t)]$$

        end
        Decay $\alpha(t)$ and $\sigma(t)$
    end
end
Output: $W_j(u), j = 1, 2, \ldots, N$

Output of SOM

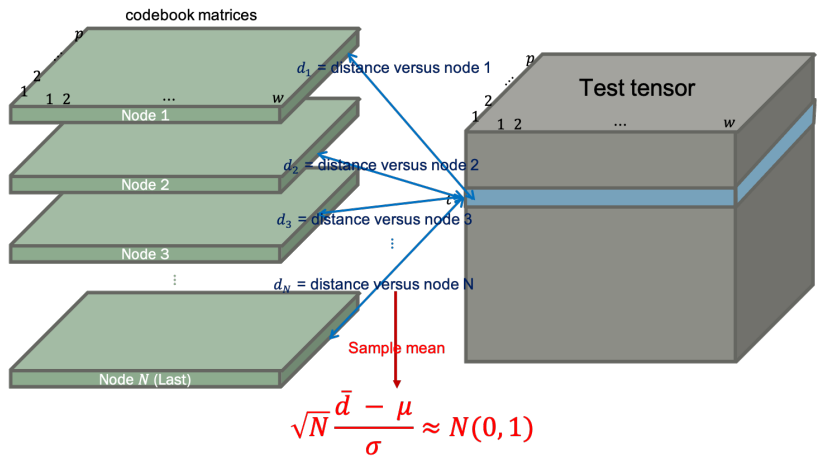
- Training process of SOM maps each normal pattern window onto the SOM grids by finding the closest corresponding codebook matrix
- The number of grids is finite
  - Normal pattern is discretized
  - Training error window maps onto finite prototypes, each of which has its own codebook matrix
  - Normal error patterns are discretized by the patterns represented by the codebook matrices



Discretized Pattern in SOM Grid

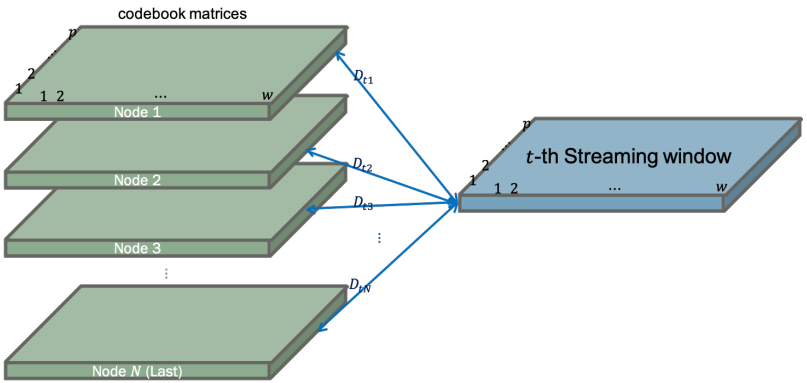
Motivation

- Test dataset to detect anomalies
  - Online dataset
  - Construct a window whenever a new set of samples of size w is available (streaming window)
- If the distance between codebook matrices and the test error pattern is large, then that pattern can be anomaly
  - In what criterion?
  - Hypothesis testing

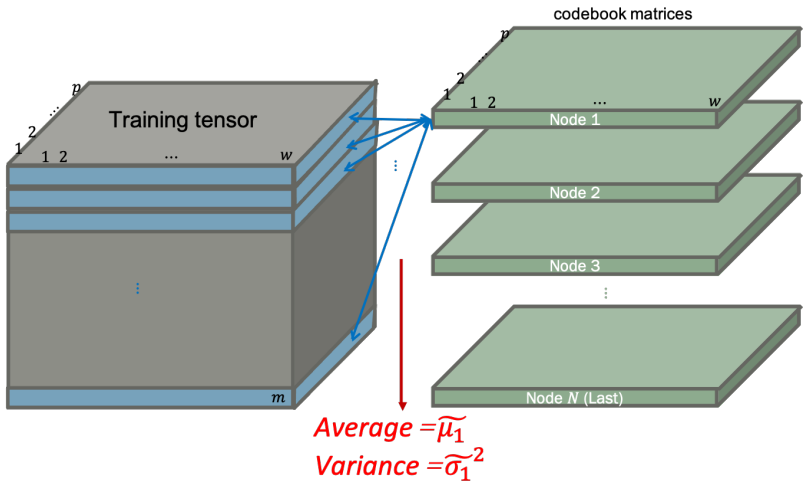


Motivation

Notation



$\{D_{ti}; t = 1, \ldots, i = 1, \ldots, N\}$: distance



- mutually independent for each t
- μ_i, σ_i^2: True mean and variance of each node i
  - Need to know μ_i and σ_i^2 to build test statistic

- Since the training set consists only of normal observations, we treat the training set as a pseudo-population
- $\tilde{\mu}_1, \ldots, \tilde{\mu}_N$ and $\tilde{\sigma}_1^2, \ldots \tilde{\sigma}_N^2$

Hypothesis Testing

- Pseudo-mean and variance:

$$\tilde{\mu} = \frac{1}{N}\sum_{i=1}^{N}\tilde{\mu}_i, \quad \tilde{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}\tilde{\sigma}_i^2$$

- For $t = 1, 2, \ldots,$

$$H_{0t} : \frac{1}{N}\sum_{i=1}^{N}\mu_i = \tilde{\mu} \quad \textit{vs.} \quad H_{1t} : \frac{1}{N}\sum_{i=1}^{N}\mu_i > \tilde{\mu}$$

- Rejecting the t-th null hypothesis H_{0t} corresponds to marking the t-th window as anomalous

SomAnomaly Statistic

- Sample mean of $\{D_{ti}\}_{i=1}^{N}$: $\overline{D}_t = \frac{1}{N}\sum_{i=1}^{N} D_{ti}$
- Based on the mutual independence assumption of $\{D_{ti}\}_{i=1}^{N}$, employ the Lindeberg-Feller CLT [Lindeberg, 1922]

Definition (SomAnomaly Statistic)

$$S_t = \frac{1}{B_N}\sum_{i=1}^{N}(D_{ti} - \tilde{\mu}_i) = \frac{N(\overline{D}_t - \tilde{\mu})}{B_N}$$

where $B_N^2 = \sum_{i=1}^{N} \sigma_i^2$, for each t-th test.

Multiple Testing

- Under some assumptions, Linderberg-Feller CLT [Lindeberg, 1922]
- SomAnomaly S_t weakly converges to standard nomral distribution under the corresponding null hypothesis
- p-value for each t-th test:

$$P_t = Pr(Z \geqslant s_t), \quad Z \sim \mathcal{N}(0, 1)$$

- We can reject the null if P_t is smaller than the significance level α (e.g. 0.05)
- If we compare P_t with usual α for every t, type I error or false discovery rate [Benjamini and Hochberg, 1995] may increase

Online Multiple Testing

- Since we have infinitely many multiple tests, we apply one of many online muliple testing methods
  - Generalized α-investing (GAI) [Aharoni and Rosset, 2014]
  - It controls the marginal false discovery rate (mFDR) under the significance level α [Foster and Stine, 2008]

GAI using SomAnomaly

Data: Trained SOM on the normal tensor input data
Input: Window size, shift size, α, η, ρ
Initialize $W(0) = \alpha\eta$
for $t = 1, 2, \ldots$ **do**
  Compute *SomAnomaly* and its *p*-value P_t for the streaming window

$$\phi_t = \frac{1}{10} W(t-1)$$

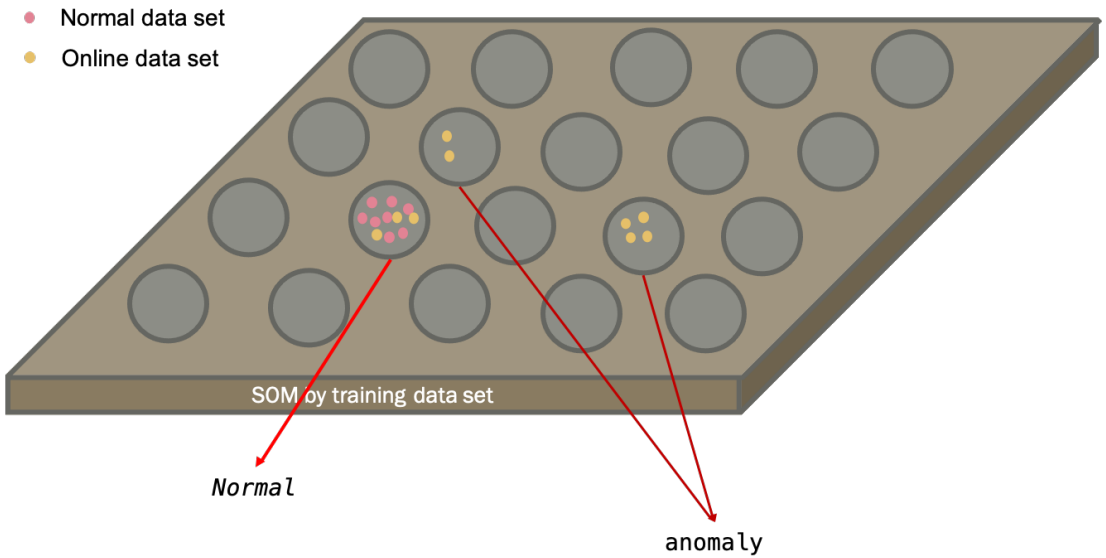
  Set α_t such that $\frac{\phi_t}{\rho} = \frac{\phi_t}{\alpha_t} - 1$
  Test t-th hypothesis as follows: $R_t = \begin{cases} 1 & P_t \leqslant \alpha_t \\ 0 & \text{otherwise} \end{cases}$

$$\psi_t = \min\left(\frac{\phi_t}{\rho} + \alpha, \frac{\phi_t}{\alpha_t} + \alpha - 1\right)$$

$$W(t+1) = W(t) - \phi_t + R_t\psi_t$$

end
Output: Results of the tests $\{R_1, R_2, \ldots\}$

Modification of SomAnomaly



Discretized Pattern of Test Dataset

- Empirically, SOM maps streaming windows onto very small number of grids due to its similar pattern
- Compute SomAnomaly for the non-empty grids

Optimized SomAnomaly (Modification of SomAnomaly)

Definition (Optimized SomAnomaly Statistic)

Let v be the index of mapped nodes and $B_v^2 = \sum_{i \in v} \sigma_i^2$.

$$S_t^* = \frac{1}{B_v} \sum_{i \in v} (D_{it} - \tilde{\mu}_i)$$

- Experimentally, S_t^* seems better than S_t
- We refer to SomAnomaly as S_t^*

CPS Datasets

1. Two benchmark datasets
   - Secure water treatment (SWaT) [Goh et al., 2016]
   - HIL-based augmented ICS (HAI) [Shin et al., 2020]
2. Three NN models
   - Apply Sequence-to-Sequence (seq2seq) [Sutskever et al., 2014] to SWaT, which was proposed by [Kim et al., 2019]
   - Apply Mixture Density Networks (MDN) [Bishop, 1994] to SWaT
   - Apply Recurrent Neural Networks (RNNs) [Rumelhart et al., 1986] to SWaT and HAI
3. Compute the error.

Forecasting Error Data

Dataset/NN	Forecasting model and CPS dataset
SWaT/seq2seq	seq2seq for each station in SWaT [Kim et al., 2019]
SWaT/MDN	MDN for each station in SWaT
SWaT/RNN	RNN for 14 correlation groups in SWaT
HAI/RNN	RNN for 14 correlation groups in HAI

Names of the Error Sets

Time Series Performance Evaluation

- They give precision and recall as traditional evaluation
  - Rather than comparing point-to-point,
  - range-based evaluation
  - Recall anomalies (or attack) in CPSs is range-based and our goal is contextual anomaly
- Metrics
  - TaPR [Hwang et al., 2019][1]
    - Detection scoring parameter: 0.001
    - Weight for the detection score: 0.8
    - Subsequent scoring parameter: 60
  - TSAD [Tatbul et al., 2018][2]
    - Default setting in the Github repository

[1] https://github.com/saurf4ng/TaPR
[2] https://github.com/IntelLabs/TSAD-Evaluator

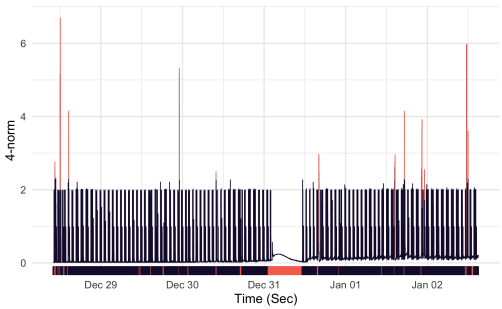
TaPR-based recall (Re), precision (Pr), and F_1 score

Method	SWaT/seq2seq			SWaT/MDN			SWaT/RNN			HAI/RNN		
	Re	Pr	F_1	Re	Pr	F_1	Re	Pr	F_1	Re	Pr	F_1
Static	0.44	0.45	0.45	0.63	0.40	0.49	0.78	0.64	0.70	0.87	0.76	0.81
CUSUM	0.58	0.70	0.63	0.64	0.56	0.59	**0.79**	0.59	0.67	0.71	0.52	0.60
SOMAD	**0.65**	**0.94**	**0.77**	**0.94**	**0.81**	**0.87**	0.76	**0.93**	**0.84**	**0.88**	**0.79**	**0.83**

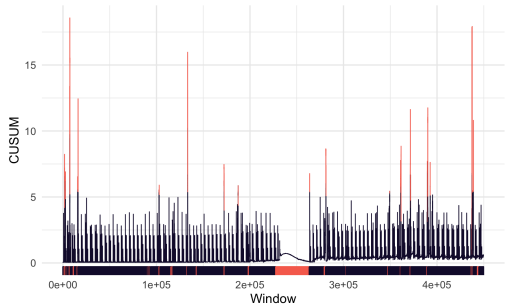
TSAD-based recall (Re), precision (Pr), and F_1 score

Method	SWaT/seq2seq			SWaT/MDN			SWaT/RNN			HAI/RNN		
	Re	Pr	F_1	Re	Pr	F_1	Re	Pr	F_1	Re	Pr	F_1
Static	0.25	0.41	0.31	0.34	0.35	0.35	0.33	**0.55**	0.42	0.20	0.71	0.31
CUSUM	0.30	**0.62**	0.40	0.38	0.39	0.38	0.37	0.45	0.41	0.36	0.44	0.39
SOMAD	**0.61**	0.60	**0.61**	**0.92**	**0.58**	**0.71**	**0.59**	0.54	**0.57**	**0.65**	**0.79**	**0.71**

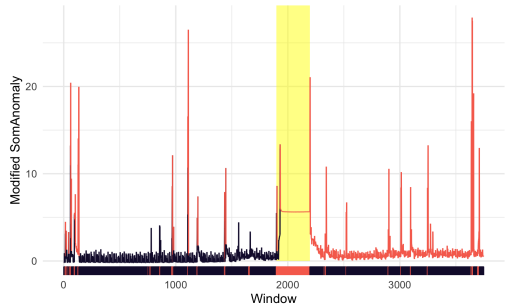
Detection Plots for the First Station



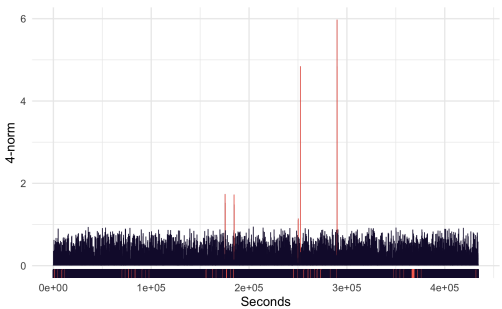
(a) Static - SWaT/RNN



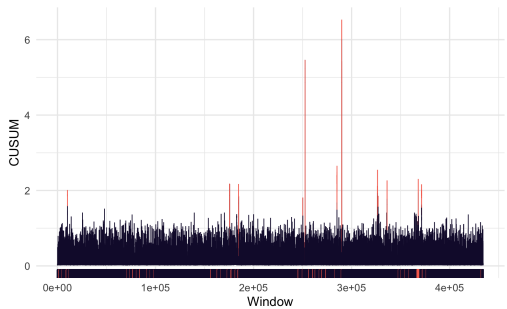
(b) CUSUM - SWaT/RNN



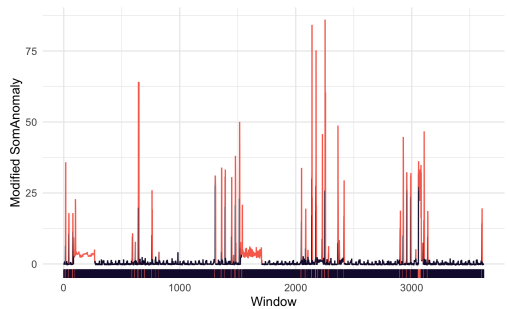
(c) SOMAD - SWaT/RNN



(d) Static - HAI/RNN



(e) CUSUM - HAI/RNN

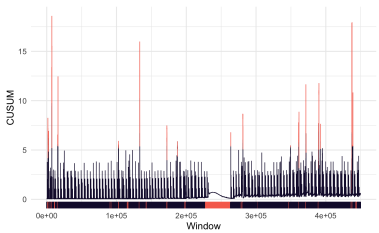


(f) SOMAD - HAI/RNN

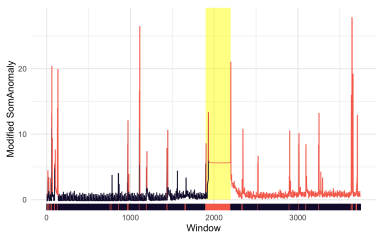
S_t^* worked: The size of SomAnomaly was similar to whether the window is anomaly

Locality Property of SOM

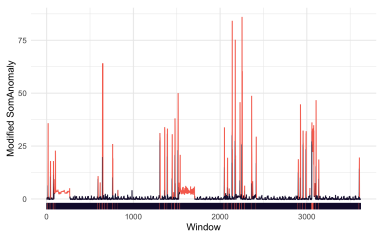
- SOMAD vs baseline approaches
  - SWaT and HAI datasets contain multiple clusters of consecutive anomaly samples over time
  - SOMAD is capable of detecting clustered anomalies
- How?
  - A time series prediction based on SOM is characterized by locality [Barreto, 2007]
  - SOM step exerts clustering effect
  - Lead to reduce false alarm rates and consequently to enhanced detection power



(a) CUSUM - SWaT/RNN



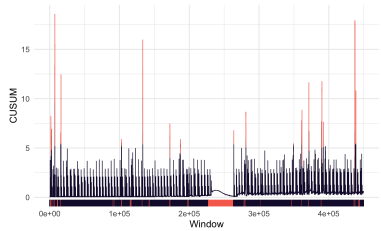
(b) SOMAD - SWaT/RNN



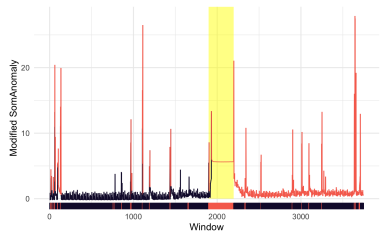
(c) SOMAD - HAI/RNN

False Positivities

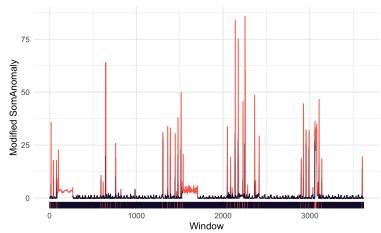
- SOM's locality property is readily reflected in our anomaly detection task as well
- However, SOMAD incorrectly classifies normal samples (black in strip) as anomalies (red line): after highlight
- This kind of performance loss occurs due to long-term dependency issue of forecasting model



(a) CUSUM - SWaT/RNN



(b) SOMAD - SWaT/RNN



(c) SOMAD - HAI/RNN

Conclusion

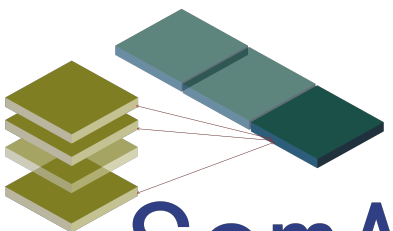
- While most of the prior work focused on improving the base forecasting model itself, our research deals with the statistical method for finding threshold with forecasting error values
- SOMAD inflates the differences between the respective FE patterns of normal and abnormal events
- SOMAD outperforms conventional methods, achieving a high detection rate without compromising precision

Future Study

- Parameter selection
  - In this work, we chose parameteters of SOM and GAI empirically or from preliminary works
  - Rolling window method only in the training dataset: Since there is no anomaly, good detector should detect no anomaly
- Long-term forecasting
  - Even MDN or seq2seq model becomes worse and worse as the time point goes further from the training term
  - Ad hoc solution: train NN model again using the data aggregated with normal-detected
- Another test method
  - e.g. Bayesian inference

Questions and Answers

- Thanks!
- Q & A ✋
- Github repository for our `Python` code: https://github.com/ygeunkim/somanomaly



SomAnomaly

References I

[Aharoni and Rosset, 2014] Aharoni, E. and Rosset, S. (2014). Generalized α-investing: definitions, optimality results and application to public databases. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):771–794.

[Barreto, 2007] Barreto, G. A. (2007). Time series prediction with the self-organizing map: A review. *Perspectives of neural-symbolic integration*, pages 135–158.

[Benjamini and Hochberg, 1995] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

[Bishop, 1994] Bishop, C. M. (1994). Mixture density networks.

[Ferguson, 1996] Ferguson, T. S. (1996). *A Course in Large Sample Theory*. Kogan Page Publishers.

[Filonov et al., 2017] Filonov, P., Kitashov, F., and Lavrentyev, A. (2017). RNN-based Early Cyber-Attack Detection for the Tennessee Eastman Process. *CoRR*.

[Filonov et al., 2016] Filonov, P., Lavrentyev, A., and Vorontsov, A. (2016). Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model. *ArXiv*, abs/1612.06676.

[Foster and Stine, 2008] Foster, D. P. and Stine, R. A. (2008). α-investing: a procedure for sequential control of expected false discoveries. *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 70(2):429–444.

[Giraldo et al., 2018] Giraldo, J., Urbina, D., Cardenas, A., Valente, J., Faisal, M., Ruths, J., Tippenhauer, N. O., Sandberg, H., and Candell, R. (2018). A survey of physics-based attack detection in cyber-physical systems. *ACM Computing Surveys (CSUR)*, 51(4):76.

References II

[Goh et al., 2016] Goh, J., Adepu, S., Junejo, K. N., and Mathur, A. (2016). A Dataset to Support Research in the Design of Secure Water Treatment Systems. In *CRITIS*, pages 88–99, Cham. Springer International Publishing.

[Goh et al., 2017] Goh, J., Adepu, S., Tan, M., and Lee, Z. S. (2017). Anomaly detection in cyber physical systems using recurrent neural networks. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pages 140–145. IEEE.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer.

[Hwang et al., 2019] Hwang, W.-S., Yun, J.-H., Kim, J., and Kim, H. C. (2019). Time-series aware precision and recall for anomaly detection: Considering variety of detection result and addressing ambiguous labeling. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2241–2244. ACM.

[Kim et al., 2019] Kim, J., Yun, J.-H., and Kim, H. C. (2019). Anomaly detection for industrial control systems using sequence-to-sequence neural networks.

[Kohonen, 1982] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69.

[Lindeberg, 1922] Lindeberg, J. W. (1922). Eine neue herleitung des exponentialgesetzes in der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 15(1):211–225.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

References III

[Shin et al., 2020] Shin, H.-K., Lee, W., Yun, J.-H., and Kim, H. (2020). HAI 1.0: Hil-based augmented ICS security dataset. In *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)*. USENIX Association.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

[Tatbul et al., 2018] Tatbul, N., Lee, T. J., Zdonik, S., Alam, M., and Gottschlich, J. (2018). Precision and recall for time series. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 1924–1934, USA. Curran Associates Inc.