ANOMALY DETECTION ALGORITHMS

나영석

2020 7월 데이터팀 인턴

목표

- 시계열 데이터의 이상치 감지 알고리즘 조사
- 알고리즘의 장단점

개요

1. 용어 정리
2. 이상치 감지 순서
3. 알고리즘 소개 및 비교
   - 통계학적 알고리즘
   - 신경망구조를 활용한 알고리즘
4. 결론
5. 제시되지 않은 방법을 이용한 예측모델 예시
   - GluonTS using Simple Feedforward Estimator

용어

- Time Series Decomposition (시계열 분해)
- NAB (Numenta Anomaly Benchmark) data
- Precision (정밀도), Recall (검출율), F1-score (F 점수)
- MAE (Mean Absolute Error) (평균 절대 오차)
  - MAPE (Mean Absolute Percentage Error) (평균 절대 백분율 편차)
- CI (Confidence Interval) (신뢰구간)

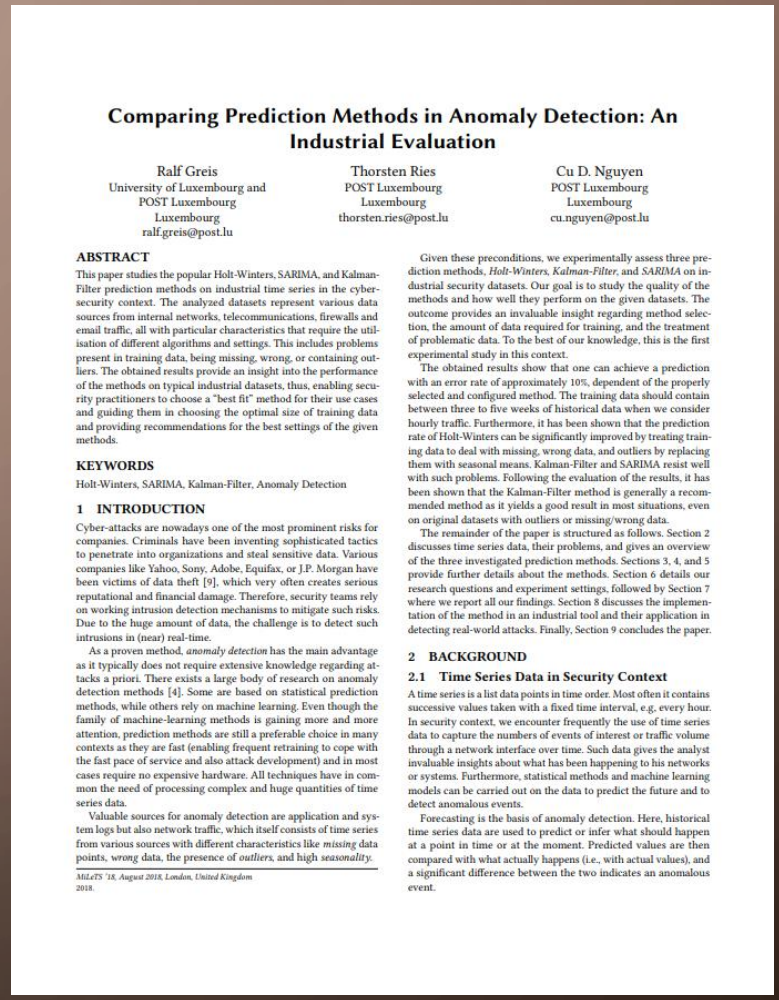
이상치 감지 순서

1. 시계열 분해 및 분석
2. 알고리즘을 통한 예측치 도출
   - 평균 혹은 중앙값
3. 실제 데이터와 비교 후 이상치 감지
   - 상한가 하한가 사용
   - 95% CI

HOLT-WINTERS VS SARIMA VS KALMAN-FILTER

- 2018년 8월 자료
- Holt-Winters
- SARIMA (p, d, q) x (P, D, Q, s)
  - <Seasonal Autoregressive Integrated Moving Average>
- Kalman-Filter

Source: https://milets18.github.io/papers/milets18_paper_19.pdf



Comparing Prediction Methods in Anomaly Detection: An Industrial Evaluation

Ralf Greis
University of Luxembourg and POST Luxembourg
Luxembourg
ralf.greis@post.lu

Thorsten Ries
POST Luxembourg
Luxembourg
thorsten.ries@post.lu

Cu D. Nguyen
POST Luxembourg
Luxembourg
cu.nguyen@post.lu

ABSTRACT

This paper studies the popular Holt-Winters, SARIMA, and Kalman-Filter prediction methods on industrial time series in the cyber-security context. The analyzed datasets represent various data sources from internal networks, telecommunications, firewalls and email traffic, all with particular characteristics that require the utilisation of different algorithms and settings. This includes problems present in training data, being missing, wrong, or containing outliers. The obtained results provide an insight into the performance of the methods on typical industrial datasets, thus, enabling security practitioners to choose a "best fit" method for their use cases and guiding them in choosing the optimal size of training data and providing recommendations for the best settings of the given methods.

KEYWORDS

Holt-Winters, SARIMA, Kalman-Filter, Anomaly Detection

1 INTRODUCTION

Cyber-attacks are nowadays one of the most prominent risks for companies. Criminals have been inventing sophisticated tactics to penetrate into organizations and steal sensitive data. Various companies like Yahoo, Sony, Adobe, Equifax, or J.P. Morgan have been victims of data theft [9], which very often creates serious reputational and financial damage. Therefore, security teams rely on working intrusion detection mechanisms to mitigate such risks. Due to the huge amount of data, the challenge is to detect such intrusions in (near) real-time.

As a proven method, *anomaly detection* has the main advantage as it typically does not require extensive knowledge regarding attacks a priori. There exists a large body of research on anomaly detection methods [4]. Some are based on statistical prediction methods, while others rely on machine learning. Even though the family of machine-learning methods is gaining more and more attention, prediction methods are still a preferable choice in many contexts as they are fast (enabling frequent retraining to cope with the fast pace of service and also attack development) and in most cases require no expensive hardware. All techniques have in common the need of processing complex and huge quantities of time series data.

Valuable sources for anomaly detection are application and system logs but also network traffic, which itself consists of time series from various sources with different characteristics like *missing* data points, *wrong* data, the presence of *outliers*, and high *seasonality*.

MiLeTS '18, August 2018, London, United Kingdom
2018.

Given these preconditions, we experimentally assess three prediction methods, *Holt-Winters*, *Kalman-Filter*, and *SARIMA* on industrial security datasets. Our goal is to study the quality of the methods and how well they perform on the given datasets. The outcome provides an invaluable insight regarding method selection, the amount of data required for training, and the treatment of problematic data. To the best of our knowledge, this is the first experimental study in this context.

The obtained results show that one can achieve a prediction with an error rate of approximately 10%, dependent of the properly selected and configured method. The training data should contain between three to five weeks of historical data when we consider hourly traffic. Furthermore, it has been shown that the prediction rate of Holt-Winters can be significantly improved by treating training data to deal with missing, wrong data, and outliers by replacing them with seasonal means. Kalman-Filter and SARIMA resist well with such problems. Following the evaluation of the results, it has been shown that the Kalman-Filter method is generally a recommended method as it yields a good result in most situations, even on original datasets with outliers or missing/wrong data.

The remainder of the paper is structured as follows. Section 2 discusses time series data, their problems, and gives an overview of the three investigated prediction methods. Sections 3, 4, and 5 provide further details about the methods. Section 6 details our research questions and experiment settings, followed by Section 7 where we report all our findings. Section 8 discusses the implementation of the method in an industrial tool and their application in detecting real-world attacks. Finally, Section 9 concludes the paper.

2 BACKGROUND

2.1 Time Series Data in Security Context

A time series is a list data points in time order. Most often it contains successive values taken with a fixed time interval, e.g, every hour. In security context, we encounter frequently the use of time series data to capture the numbers of events of interest or traffic volume through a network interface over time. Such data gives the analyst invaluable insights about what has been happening to his networks or systems. Furthermore, statistical methods and machine learning models can be carried out on the data to predict the future and to detect anomalous events.

Forecasting is the basis of anomaly detection. Here, historical time series data are used to predict or infer what should happen at a point in time or at the moment. Predicted values are then compared with what actually happens (i.e., with actual values), and a significant difference between the two indicates an anomalous event.

Table 4: Results for interval length equal to 1h. One needs at least 1 week as training set to use the Holt-Winters and the Kalman-Filter method. Size(w) stands for the training size in weeks.

Dataset	Size(w)	Method	Time(s)	MAE	MAPE(%)
DNS	10	HW	2	12,348	14.13
	5	HW	2	**11,426**	**12.95**
	3	HW	1	13,493	17.28
	1	HW	1	23,193	31.35
DNS	10	KF	1	10,315	12.40
	5	KF	1	12,519	15.38
	3	KF	1	10,402	11.59
	1	KF	1	**9,819**	**10.77**
Diameter	13	HW (addi.)	2	58,487	64.17
	10	HW (addi.)	2	68,788	78.08
	5	HW (addi.)	1	**42,087**	**50.76**
	3	HW (addi.)	1	46,934	56.60
	1	HW (addi.)	1	60,453	64.27
Diameter	13	KF	1	10,961	12.15
	10	KF	1	14,346	14.48
	5	KF	1	**9,070**	**10.63**
	3	KF	1	12,152	12.60
	1	KF	1	12,513	12.98
Firewall	13	HW	6	13,576	11.96
	10	HW	2	13,567	12.09
	5	HW	2	13,138	12.78
	3	HW	2	**8,823**	**8.97**
	1	HW	1	21,249	20.39
Firewall	13	KF	1	15,676	13.45
	10	KF	1	15,640	13.40
	5	KF	1	15,029	12.46
	3	KF	1	**13,966**	**11.75**
	1	KF	1	15,490	13.13
Email	10	HW	4	**861**	**27.75**
	5	HW	2	1,008	34.30
	3	HW	1	1,079	33.19
	1	HW	1	1,345	39.79
Email	10	KF	1	794	22.65
	5	KF	1	**736**	**20.57**
	3	KF	1	780	21.90
	1	KF	1	808	21.21

Table 5: Results for interval length equal to 4 h. SARIMA reported an error when running with one and three-weeks training sets. Size(w) stands for the training size in weeks.

Dataset	Size(w)	Method	Time(s)	MAE	MAPE(%)
DNS	10	HW	2	45,479	12.42
	5	HW	1	**39.873**	**10.50**
	3	HW	1	40,063	11.64
	1	HW	1	112,580	35.20
DNS	10	KF	1	36,662	10.15
	5	KF	1	47,642	13.88
	3	KF	1	37,193	9.73
	1	KF	1	**32,591**	**8.34**
DNS	10	SARIMA	28	34,075	9.17
	5	SARIMA	68	**31,962**	**8.19**
Diameter	13	HW (addi.)	1	63,702	18.07
	10	HW (addi.)	1	69,734	20.84
	5	HW (addi.)	1	**56,146**	**16.74**
	3	HW (addi.)	1	109,999	31.51
	1	HW (addi.)	1	119,429	31.23
Diameter	13	KF	1	41,654	11.76
	10	KF	1	53,246	14.19
	5	KF	1	**34,689**	**10.77**
	3	KF	1	47,680	12.69
	1	KF	1	49,438	13.36
Diameter	13	SARIMA	33	**40,559**	**11.67**
	10	SARIMA	25	41,194	12.28
	5	SARIMA	68	47,349	12.76
Firewall	13	HW	1	52,301	11.96
	10	HW	1	52,214	11.97
	5	HW	1	48,436	11.71
	3	HW	1	**31,230**	**8.36**
	1	HW	1	188,178	41.70
Firewall	13	KF	1	62,936	14.06
	10	KF	1	63,582	14.14
	5	KF	1	58,138	12.83
	3	KF	1	**54,242**	**12.14**
	1	KF	1	58,021	12.98
Firewall	13	SARIMA	28	48,073	10.93
	10	SARIMA	28	**48,070**	**10.93**
	5	SARIMA	5	51,705	12.05
Email	10	HW	2	**3,106**	**25.30**
	5	HW	1	3,583	29.74
	3	HW	1	3,703	30.40
	1	HW	1	4,767	31.96
Email	10	KF	1	2,967	20.97
	5	KF	1	**2,734**	**18.63**
	3	KF	1	2,928	20.47
	1	KF	1	3,093	19.86
Email	10	SARIMA	22	**2,268**	**15.09**
	5	SARIMA	60	257,119	2,099.73

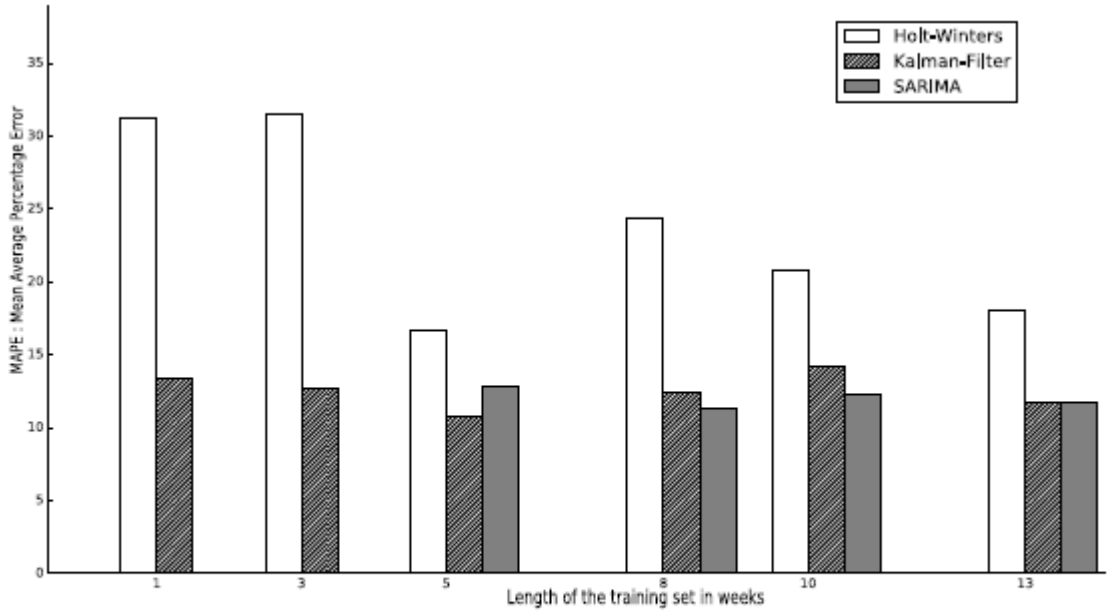


Figure 2: Barplot showing the MAPE of the Diameter dataset with 4 h interval for different lengths of the training data.

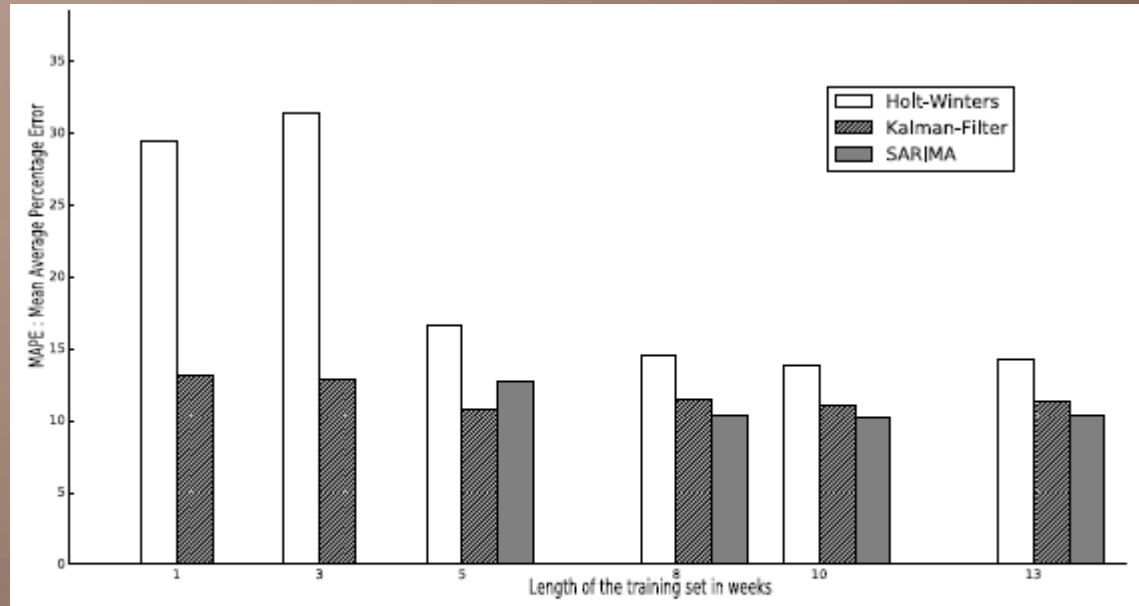


Figure 3: Barplot showing the MAPE of the treated Diameter dataset with 4 h interval for different lengths of the training data.

결과

- 모델 생성시 3-5주 정도의 시간별 데이터가 적합하다.
- 연산속도: SARIMA > Holt-Winters, Kalman-Filter
- 예측치: SARIMA > Kalman-Filter >> Holt-Winters
- 비정상 데이터에 대한 취약성: Holt-Winters > Kalman-Filter, SARIMA
- Kalman-Filter 가 가장 적합하다.

DEEP LEARNING VS STATISTICAL METHODS FOR AD

- 2018년 7월 자료
- STL을 활용한 시계열 데이터 분석
  - <Seasonal-Trend Decomposition Procedure Based on Loess>
- DBN (Deep Belief Network)을 더 정교하게 하기 위해 EKF (Extended Kalman Filter) 사용.
- 통계학적 방식은 상한가 하한가 계산.

Source: https://ojs.library.queensu.ca/index.php/wdsa-ccw/article/view/12495

1st International WDSA / CCWI 2018 Joint Conference, Kingston, Ontario, Canada – July 23-25, 2018

COMPARING DEEP LEARNING WITH STATISTICAL CONTROL METHODS FOR ANOMALY DETECTION

Zheng Yi Wu[1], Yekun He[2] and Qiao Li[3]
[1,2]Bentley Systems, Incorporated, Watertown, CT, USA
[3]Department of Electrical Engineering, Denver University, Denver, CO, USA
Zheng.wu@bentley.com

ABSTRACT

In this paper, we present and compare two approaches for anomaly detection. The first approach is based on Deep Belief Neural network integrated with Extended Kalman Filter (DBN-EKF), which has been developed by the authors. It allows us to construct a good predictive model with the upper and lower bounds of the predicted or expected attribute. The upper and lower limits enable the detection of possible outliers in a system. The outliers can be further processed and categorized as likely system anomaly events. The second approach is to apply the well-established Statistical Control (SC) methods, including X bar chart, exponentially weighted moving average chart, cumulative sum and seasonal hybrid extreme studentized deviate. Both DEB-EKF and SC approaches are applied to and tested on the dataset of a real water system. This comparison study helps us to understand the effectiveness of different methods for anomaly detection.

Keywords: Deep learning, statistical control, anomaly detection

1 Introduction

With increasingly adoption of smart meters and data loggers deployed for water distribution system management, the system status is continuously monitored and embedded in the data. It is essential to enable engineers to gain the insights in the large dataset of flows and pressures collected throughout a system. One such an insightful and useful outcome of analyzing the time series data is to detect when an anomaly event occurs in a system, so that operators can be alerted to further localize and pinpoint whereabout the anomaly hotspots by other means, and consequently take proper actions to mitigate the impact of event.

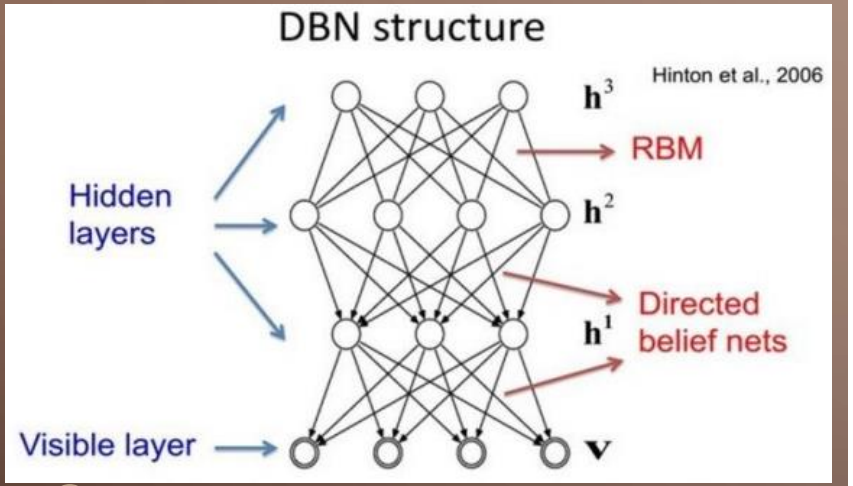
There are many methods developed for analyzing the time series data including flows and pressures for a water distribution system. In general, there are two types of methodologies [1] – [4], one is based on classic statistical control while the other is based on machine learning techniques. The former is to establish some sorts of upper and lower bounds, so-called control limits, to determine if a data point is outlier, and the latter usually requires for training a model to predict the time series and establish the expected prediction envelope, served as the control limit, to identify the outlier when the actual data is falling out of the expected envelop.

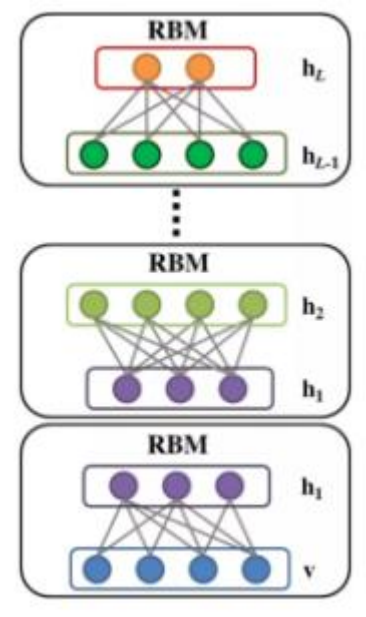
Both types of methods have been investigated and applied for anomaly event detection of water distribution systems. The comparison study, as elaborated in this paper, was undertaken to improve our understanding on the merits and shortcomings of both approaches for water distribution systems.

DBN

나. 심층신뢰망의 개발 배경

- 피드포워드 신경망에서 신경망의 층 수가 늘어나면서 발생하는 경사감소 소멸(Gradient Descent Vanishing) 문제 해결 필요





- 여러층의 RBM (Restricted Boltzmann Machine) 계산 후 튜닝 과정을 통해 가장 가까운 최종 가중치 계산
- 비지도 학습
- 학습데이터가 불충분할때 유용

Sources:
https://address83.tistory.com/38
http://blog.skby.net/%EC%8B%AC%EC%B8%B5%EC%8B%A0%EB%A2%B0%EB%A7%9D-dbn-deep-belief-network/

X-BAR

$$UCL = \mu + 3\sigma$$

$$LCL = \mu - 3\sigma$$

EWMA

$$EWMA_i = \lambda X_i + (I - \lambda) Z_{i-1}$$

$$UCL_i = \mu_0 + 3\alpha \sqrt{\frac{\lambda}{(2-\lambda)}[1-(1-\lambda)^{2i}]}$$

$$LCL_i = \mu_0 - 3\alpha \sqrt{\frac{\lambda}{(2-\lambda)}[1-(1-\lambda)^{2i}]}$$

CUSUM

$$SH(0) = SL(0) = 0$$

$$SH(i) = Max[0, SH(i-1) + X_i - \mu - 0.5\sigma]$$

$$SL(i) = Min[0, SL(i-1) + X_i - \mu + 0.5\sigma]$$

S-H-ESD

$$\lambda_k = \frac{(n-k)t_{p,n-k-1}}{\sqrt{(n-k-1+t_{p,n-k-1}^2)(n-k+1)}}$$

$$C_k = median_k(|X_k - median(X)|)$$

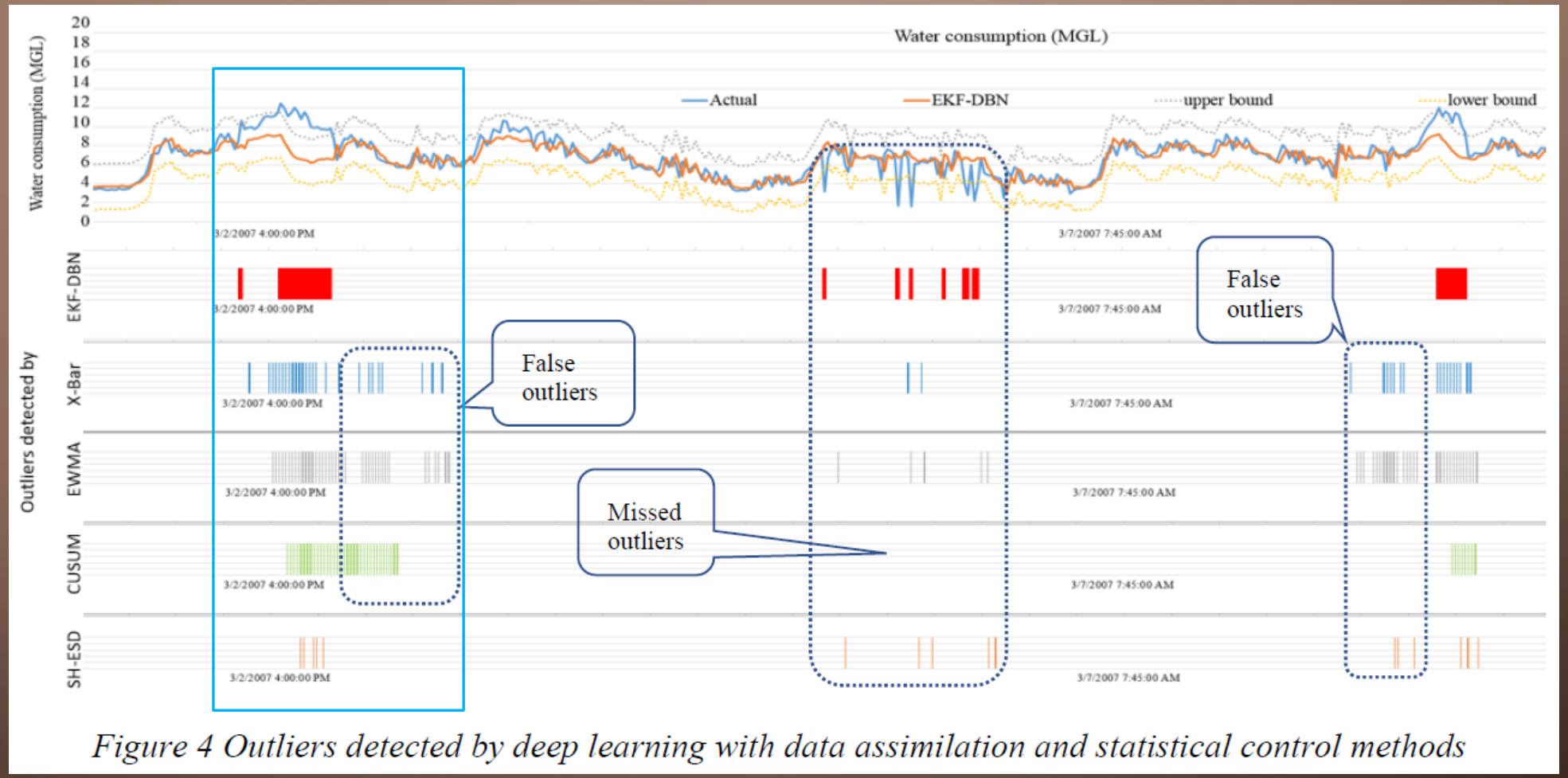


Figure 4 Outliers detected by deep learning with data assimilation and statistical control methods


/2/2007 4:00:00 PM
/2/2007 4:00:00 PM
3/2/2007 4:00:00 PM
3/2/2007 4:00:00 PM
3/2/2007 4:00:00 PM
3/2/2007 4:00:00 PM

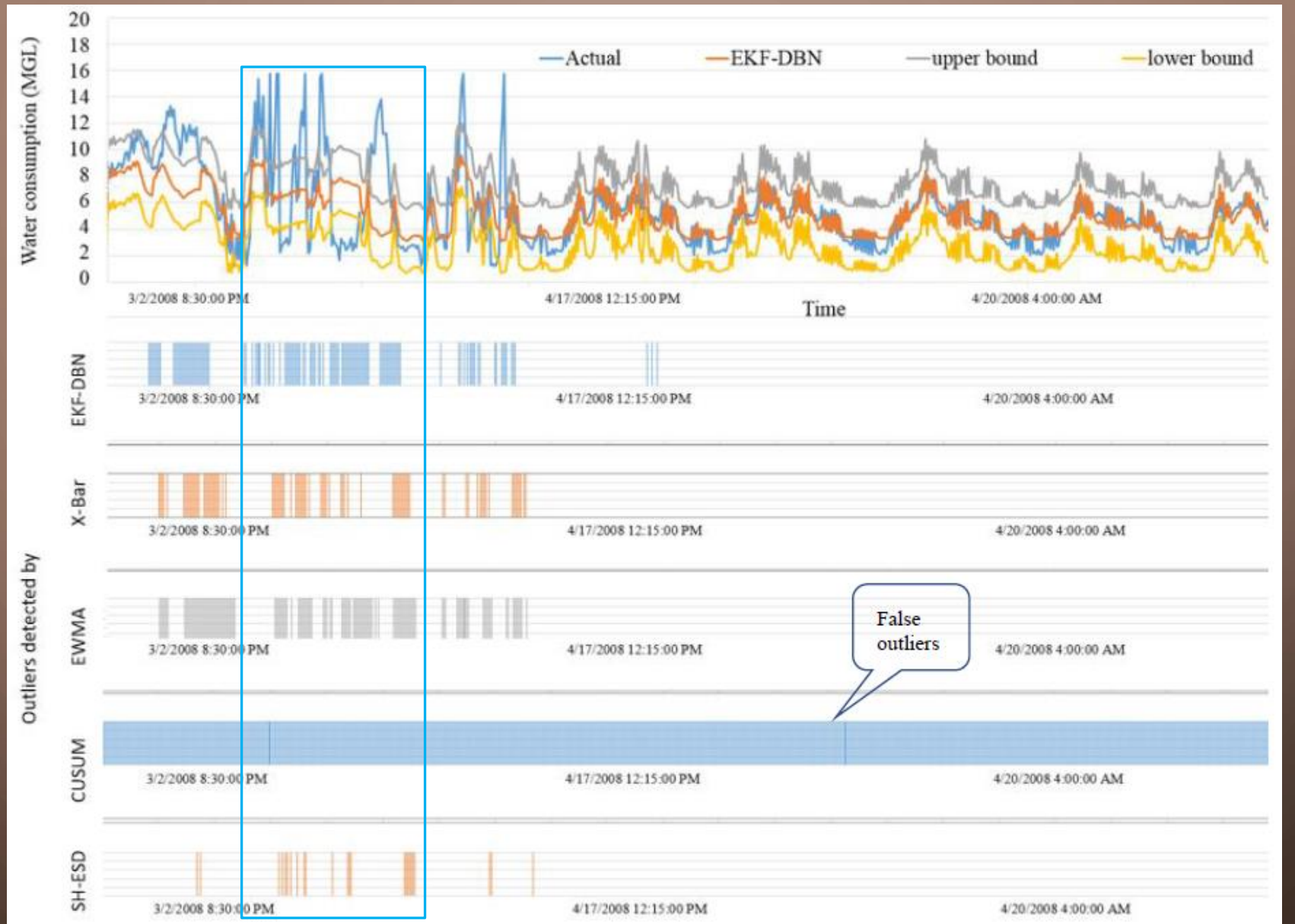
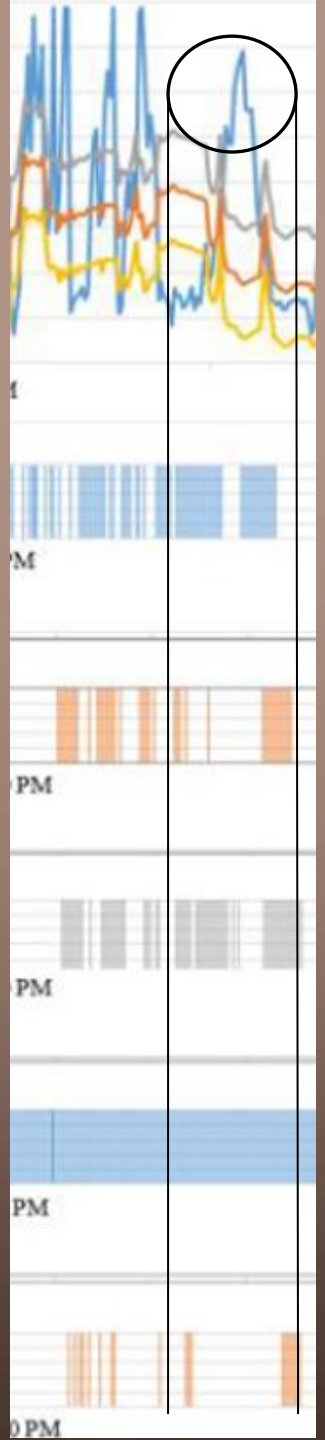


Figure 5 More comparison of outliers detected by deep learning with data assimilation and statistical control methods



ONLINE ANOMALY DETECTION METHODS

- 2019년 7월 자료
- SARIMA
- Facebook Prophet
- Multi-step forecasting RNN
- Twitter's Anomaly Detection
- Donut

Source: Freeman, Cynthia & Merriman, Jonathan & Beaver, Ian & Mueen, Abdullah. (2019). Experimental Comparison of Online Anomaly Detection Algorithms.

The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

Experimental Comparison of Online Anomaly Detection Algorithms

Cynthia Freeman,[1 2] Jonathan Merriman,[1] Ian Beaver,[1] Abdullah Mueen[2]
[1]Verint Intelligent Self-Service, [2]University of New Mexico
cynthia.freeman@verint.com, jonathan.merriman@verint.com, ian.beaver@verint.com, mueen@cs.unm.edu

Abstract

Anomaly detection methods abound and are used extensively in streaming settings in a wide variety of domains. But a strength can also be a weakness; given the vast number of methods, how can one select the best method for their application? Unfortunately, there is no one best way for all domains. Existing literature is focused on creating new anomaly detection methods or creating large frameworks for experimenting with multiple methods at the same time. As the literature continues to grow, extensive evaluation of every available anomaly detection method is not feasible. To reduce this evaluation burden, in this paper we present a framework to intelligently choose the optimal anomaly detection methods based on the characteristics the time series displays. We provide a comprehensive experimental validation of multiple anomaly detection methods over different time series characteristics to form guidelines. Applying our framework can save time and effort by surfacing the most promising anomaly detection methods instead of experimenting extensively with a rapidly expanding library of anomaly detection methods.

Introduction

An anomaly in a time series is a pattern that does not conform to past patterns of behavior in the series. It is used in a wide variety of fields such as intrusion and fraud detection, tracking KPIs, and medical sensor technologies. Early detection of anomalies is vital for ensuring undisrupted business and efficient troubleshooting.

Time series anomaly detection is a difficult problem for a multitude of reasons: **(1)** What is defined as anomalous may differ based on application. There is no one-size-fits-all method (Kejariwal 2015; Laptev, Amizadeh, and Flint 2015). **(2)** Anomaly detection often must be done on real-world streaming applications. Strictly speaking, an *online* anomaly detection method must determine anomalies and update all relevant models before occurrence of the next time step (Saurav et al. 2018). Depending on the needs of the user, it may be acceptable to detect anomalies periodically. Regardless, computational efficiency is vital which presents a challenge. **(3)** Given the application-specific nature of anomaly detection, it is unlikely that anomaly detection systems will have access to large numbers of tagged datasets. Therefore, these systems will likely encounter behavior that was not present in the training data. **(4)** As an anomaly is a pattern that does not conform to past patterns of behavior, non-anomalous data tends to occur in significantly larger quantities than anomalous data. This guarantees the imbalanced class problem for a machine learning classifier approach to anomaly detection. **(5)** It is important to detect as many anomalies as accurately and efficiently as possible, but minimizing false positives is also desirable to avoid alarm fatigue. This demands the selected anomaly detection method be optimal to the application for success. **(6)** There is a massive wealth of anomaly detection methods to choose from.

Because of these difficulties inherent in time series anomaly detection, we present a framework for automating the classification of time series and choice of anomaly detection method based on the characteristics the time series possesses. For example, if the time series data in a user's application exhibits concept drift, the user may want to consider a forecasting RNN and not Twitter AnomalyDetection. If the time series exhibits trend, Donut's variational auto-encoder may be a good choice. After a discussion of time series characteristics and datasets, we proceed with a description of all anomaly detection methods we experiment with, parameters used, and how we evaluate the performance of the methods. After obtaining the results, we derive several guidelines on method selection by time series characteristics and outline areas for future work.

Preliminaries

We discuss the time series characteristics under consideration, how to detect them, and the evaluation datasets.

Time Series Characteristics

A time series is **stationary** if the mean, variance, and autocorrelation structure are constant for all time (Natrella 2013). A typical first step for determining stationarity is conducting an Augmented Dickey-Fuller test, a statistical unit root test which can determine how strongly a time series is defined by a trend (Brownlee 2016). The Levene test is a test for nonconstant variance (Pardoe 2018). Although other such tests exist, the Levene test does not require terms to be drawn from a normal distribution.

	SARIMA				Prophet				RNN				Twitter				Donut			
	Threshold	P	R	F-Score	Threshold	P	R	F-Score	Threshold	P	R	F-Score	Threshold	P	R	F-Score	Threshold	P	R	F-Score
Seasonality	1	1	.22	.36	.9999	.6	.67	**.63**	.9999	.33	.78	.47	.0001	1	.22	.36	1	.33	.44	.38
Trend	1	0	0	0	1	1	.17	.29	.9999	.14	.33	.2	.0001	1	.17	.29	1	.43	.5	**.46**
Concept Drift	1	1	.43	**.6**	1	1	.29	.44	.9835	.46	.86	**.6**	.0001	.67	.29	.4	1	.5	.57	.53

Table 2: Results on behavior corpora. Each method returns an anomaly score between 0 and 1. The anomaly threshold that minimizes the number of window mistakes across *all* datasets that comprise the corpus is shown, where false positives are weighted .5 and false negatives are weighted 1. For example, a threshold of .9835 for the RNN returns the minimum number of weighted mistakes across all three concept drift datasets. The precision (P), recall (R), and F-score across all datasets making up the corpus is displayed in the table, where the best F-score for a behavior is bolded in its respective row.

결과

- RNN 방식과 SARIMA: 제일 높은 F-점수
- 정밀도(precision)와 검출율(recall), 그리고 F-점수에 의거:
  1. “Facebook Prophet performs well on seasonality, suggesting that decomposition based methods with components specifically for seasonality aid in performance on this characteristic.”
  2. “Although all methods suffer on trend, Donut, the variational auto-encoder, appears to perform the best on this behavior, suggesting that VAEs may be a worthwhile choice for trend.”
  3. “Despite the prevalence of false positives, the RNN with GRU units adapts well to concept drifts whereas Twitter AD suffers.”
- 비정상 시계열 데이터는 테스트 안됨.
  - Donut can innately handle missing time steps but Twitter AD cannot.

ONLINE ANOMALY DETECTION WITH SGP-Q

- 2019년 5월 자료
- SPG-Q (Sparse Gaussian Process with Q-function) 제시
- 다른 AD 방식들
  - GPR-AD
  - GPR-ADAM
  - GPR-IADAM

Source: https://arxiv.org/abs/1905.05761

Online Anomaly Detection with Sparse Gaussian Processes

Jingjing Fei , Shiliang Sun *

Department of Computer Science and Technology, East China Normal University, 3663 North Zhongshan Road, Shanghai 200241, P. R. China

Abstract

Online anomaly detection of time-series data is an important and challenging task in machine learning. Gaussian processes (GPs) are powerful and flexible models for modeling time-series data. However, the high time complexity of GPs limits their applications in online anomaly detection. Attributed to some internal or external changes, concept drift usually occurs in time-series data, where the characteristics of data and meanings of abnormal behaviors alter over time. Online anomaly detection methods should have the ability to adapt to concept drift. Motivated by the above facts, this paper proposes the method of sparse Gaussian processes with Q-function (SGP-Q). The SGP-Q employs sparse Gaussian processes (SGPs) whose time complexity is lower than that of GPs, thus significantly speeding up online anomaly detection. By using Q-function properly, the SGP-Q can adapt to concept drift well. Moreover, the SGP-Q makes use of few abnormal data in the training data by its strategy of updating training data, resulting in more accurate sparse Gaussian process regression models and better anomaly detection results. We evaluate the SGP-Q on various artificial and real-world datasets. Experimental results validate the effectiveness of the SGP-Q.

Key words: Online Anomaly Detection; Gaussian Processes; Sparse Gaussian Processes; Q-Function; Concept Drift

1 Introduction

The arrival of the Internet of Things (IoT) [1] has inspired companies to install more sensors on their machines. These sensors can produce vast amounts of

* Corresponding author. Tel.: +86-21-62233507;
E-mail address: shiliangsun@gmail.com, slsun@cs.ecnu.edu.cn

Given a new input $\mathbf{x}^*$, the prediction distribution is still a Gaussian distribution,

$$p(y^*|X, \mathbf{y}, \mathbf{x}^*) = N(\mu^*, \Sigma_y^*), \tag{5}$$

where $\mu^* = k(\mathbf{x}^*, X)[K_{NN} + \sigma^2 I]^{-1}\mathbf{y}$ and $\Sigma_y^* = k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, X)(K_{NN} + \sigma^2 I)^{-1}k(X, \mathbf{x}^*) + \sigma^2$. The time complexity of the GP is $O(N^3)$, where N is the number of training data.

Given a test input $\mathbf{x}_*$, the prediction distribution of the SGP with variational inducing variables is still a Gaussian distribution as follows [23],

$$p(y^*|X, Z, \mathbf{y}, \mathbf{x}^*) = N(\mu^*, \Sigma_y^*), \tag{11}$$

where $\mu^* = k(\mathbf{x}^*, Z)K_{MM}^{-1}\tilde{\boldsymbol{\mu}}$, $\Sigma_y^* = k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, Z)K_{MM}^{-1}k(Z, \mathbf{x}_*) + k(\mathbf{x}^*, Z)B$ $k(Z, \mathbf{x}_*) + \sigma^2$, $\tilde{\boldsymbol{\mu}} = \sigma^{-2}K_{MM}\Sigma K_{MN}\mathbf{y}$, $A = K_{MM}\Sigma K_{MM}$, $\Sigma = (K_{MM} + \sigma^{-2}K_{MN}K_{NM})^{-1}$, and $B = K_{MM}^{-1}AK_{MM}^{-1}$. The time complexity of the SGP with variational inducing variables is $O(NM^2)$.

Algorithm 2 GPR-AD

Input: current time m, $D_T = \{t_i, y_i\}_{i=m-q+1}^{m}$ and the size of sliding window q

Initialize: select an appropriate covariance function (the mean function is generally set to 0), and initialize parameters of the covariance functions

1: **repeat**
2: train the GPR model using data in the sliding window D_T
3: predict mean μ_{m+1} and variance σ_{m+1} at time t_{m+1} by Equation (5)
4: compute 95% confidence interval $[\mu_{m+1} - 1.96\sigma_{m+1}, \mu_{m+1} + 1.96\sigma_{m+1}]$
5: **if** data point y_{m+1} in 95% confidence interval **then**
6: y_{m+1} is a normal data point
7: **else**
8: y_{m+1} is an abnormal data point
9: **end if**
10: add $\{t_{m+1}, y_{m+1}\}$ into D_T
11: remove the earliest data point and its time from D_T
12: $m = m + 1$
13: **until** all test data points have been detected

Algorithm 3 GPR-ADAM

Input: current time m, $D_T = \{t_i, y_i\}_{i=m-q+1}^{m}$ and the size of sliding window q

Initialize: select an appropriate covariance function (the mean function is generally set to 0), and initialize parameters of the covariance function

repeat
  train the GPR model using data in sliding window D_T
  predict mean μ_{m+1} and variance σ_{m+1} at time t_{m+1} by Equation (5)
  compute 95% confidence interval $[\mu_{m+1} - 1.96\sigma_{m+1}, \mu_{m+1} + 1.96\sigma_{m+1}]$
  if data point y_{m+1} in 95% confidence interval **then**
    y_{m+1} is a normal data point
    add $\{t_{m+1}, y_{m+1}\}$ into D_T
  else
    y_{m+1} is an abnormal data point
    add $\{t_{m+1}, \mu_{m+1}\}$ into D_T
  end if
  remove the earliest data point and its time from D_T
  $m = m + 1$.
until all test data points have been detected

Algorithm 1 GPR-IADAM

Input: current time m, $D_T = \{t_i, y_i\}_{i=m-q+1}^{m}$, the size of sliding window q and threshold β_{max}

Initialize: select an appropriate covariance function (the mean function is generally set to 0), and initialize parameters of the covariance function

repeat
  train the GPR model using data in the sliding window D_T
  predict mean μ_{m+1} and variance σ^2_{m+1} at time t_{m+1} by Equation (5)
  compute 95% confidence interval $[\mu_{m+1} - 1.96\sigma_{m+1}, \mu_{m+1} + 1.96\sigma_{m+1}]$
  if data point y_{m+1} in 95% confidence interval **then**
    y_{m+1} is a normal data point
    add $\{t_{m+1}, y_{m+1}\}$ into D_T
  else
    y_{m+1} is an abnormal data point
    compute the value of $\beta(y_{m+1})$ by Equation (6)
    if $\beta(y_{m+1}) \leq \beta_{max}$ **then**
      add $\{t_{m+1}, \mu_{m+1}\}$ into D_T
    else
      add $\{t_{m+1}, y_{m+1}\}$ into D_T
    end if
  end if
  remove the earliest data point and its time from D_T
  $m = m + 1$
until all test data points have been detected

If the test data point y_{m+1} is within the 95 percent confidence interval, it is considered as a normal data point, and the data point y_{m+1} and its time t_{m+1} are added into the sliding window D_T. If the test data point y_{m+1} is not in the 95 percent confidence interval, it is marked as an abnormal data point. The value of $\beta(y_{m+1})$ needs to be calculated according to the Equation (6).

$$\beta(y_{m+1}) = P(z < 1.96 - \frac{|\mu_{m+1} - y_{m+1}|}{\sigma_{m+1}}), \tag{6}$$

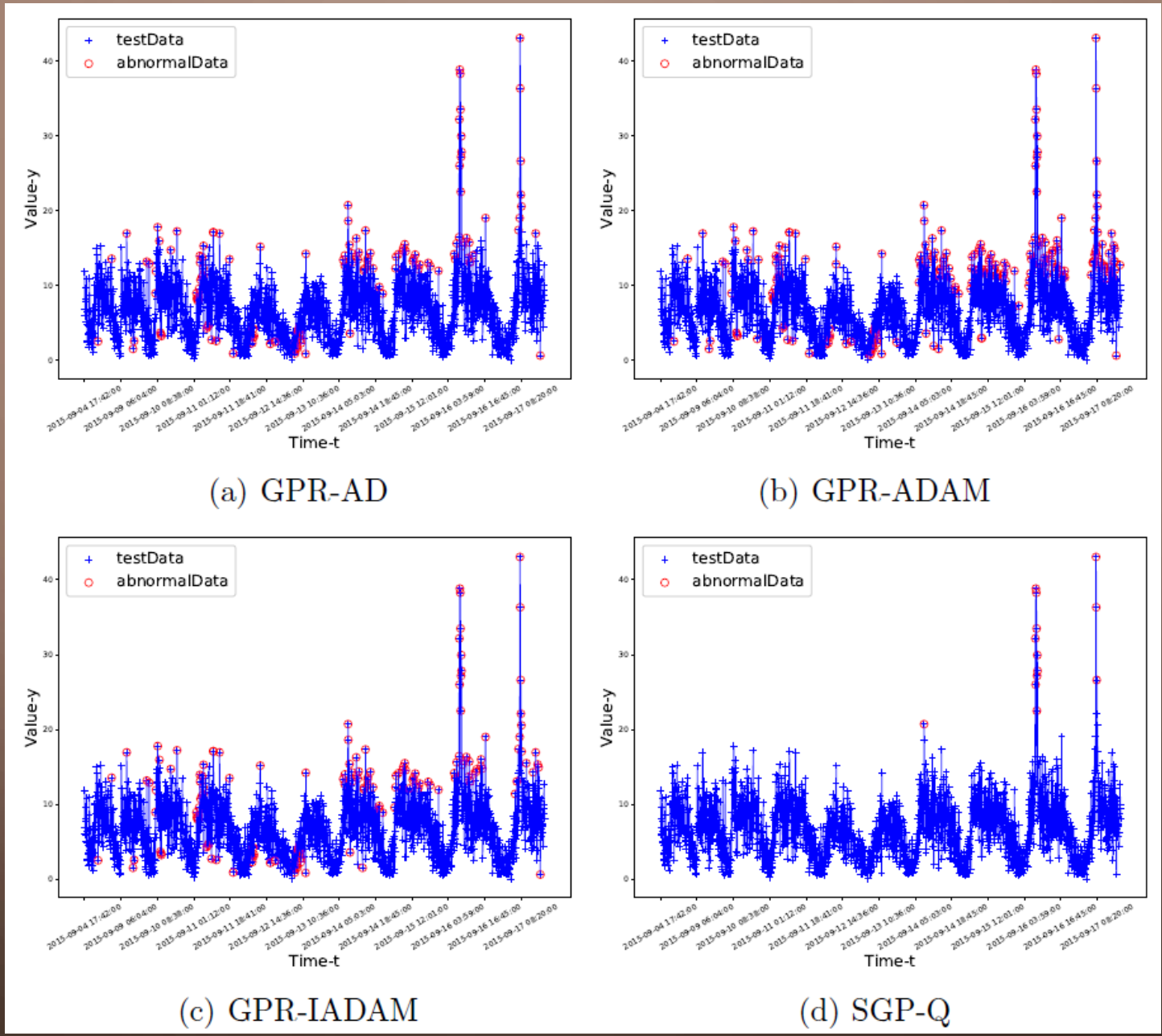
where z obeys the standard Gaussian distribution. The value of $\beta(y_{m+1})$ is used to measure the deviation between the data point y_{m+1} and prediction mean μ_{m+1}. The smaller the value of β is, the greater the deviation between the data point y_{m+1} and prediction mean μ_{m+1} is. The larger the value of β is, the smaller the deviation between the data point y_{m+1} and prediction mean μ_{m+1} is. Comparing the value of β and β_{max}, if β is less than or equal to

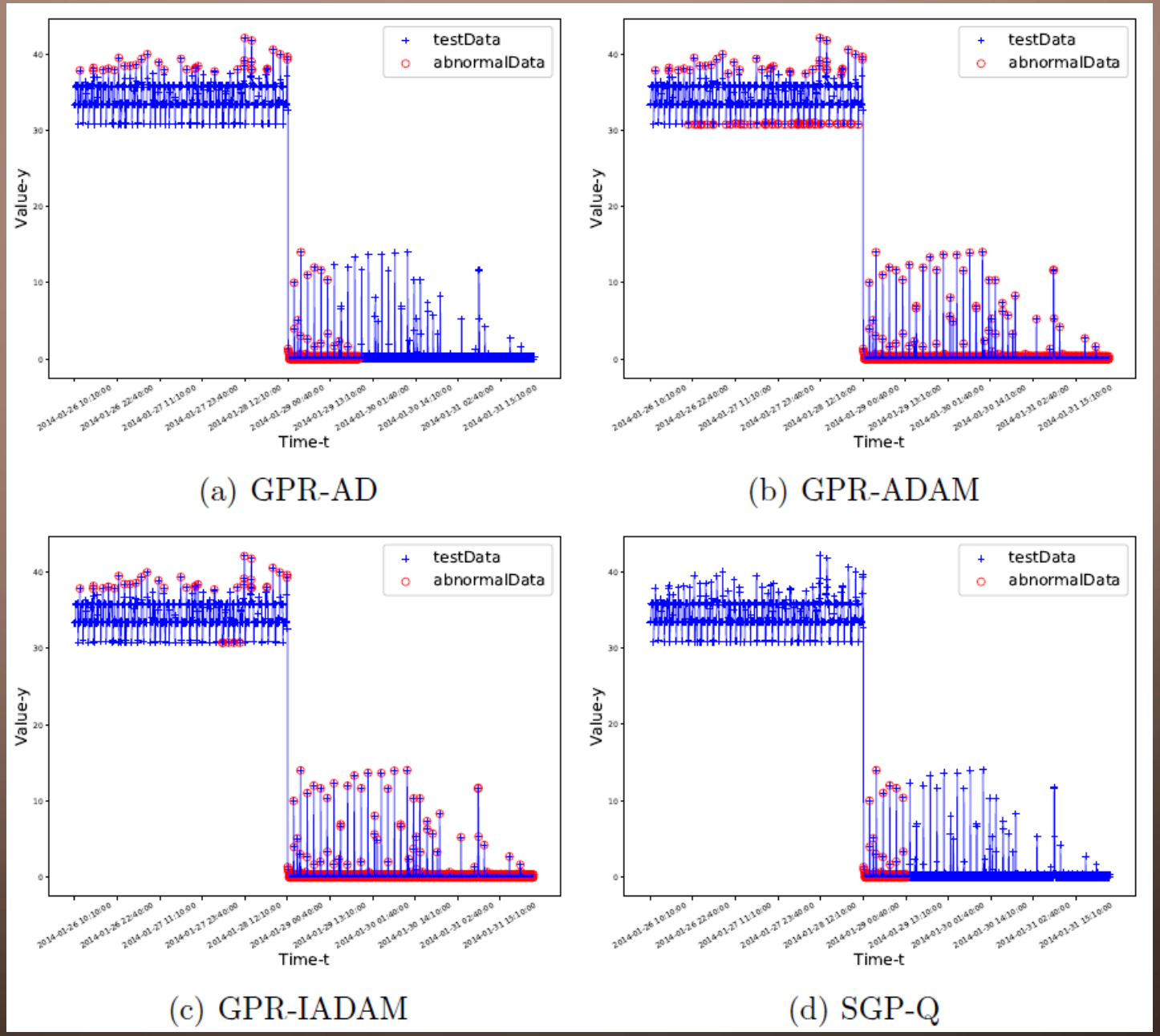
Algorithm 4 SGP-Q

Input: current time m, $D_T = \{t_i, y_i\}_{i=m-q+1}^{m}$, the size of the window q, W, W', threshold ϵ_e, ϵ_l, and the size of inducing points M

Initialize: select an appropriate covariance function (the mean function is generally set to 0), and initialize the parameters of the covariance function

repeat
  train the SGPR model using data in sliding window D_T
  predict mean μ_{m+1} and variance σ^2_{m+1} at time t_{m+1} by Equation (11)
  compute the likelihood of data point $p(y_{m+1})$ by Equation (12)
  select the threshold which performs best in the validation set according to F_1 score as ϵ_p
  if $p(y_{m+1}) > \epsilon_p$ **then**
    y_{m+1} is a normal data point
    add $\{t_{m+1}, y_{m+1}\}$ into D_T
  else
    y_{m+1} is an abnormal data point
    compute the value of QE_{m+1} and QL_{m+1} by Equation (15)
    if $QE_{m+1} \leq \epsilon_e$ or $QL_{m+1} \leq \epsilon_l$ **then**
      add $\{t_{m+1}, \mu_{m+1}\}$ into D_T
    else
      add $\{t_{m+1}, y_{m+1}\}$ into D_T
    end if
  end if
  remove the earliest data point and its time from D_T
  $m = m + 1$
until all test data points have been detected


testData
abnormalData
Value-y
Time-t
(a) GPR-AD
(b) GPR-ADAM
(c) GPR-IADAM
(d) SGP-Q


testData
abnormalData
Value-y
Time-t
(a) GPR-AD
(b) GPR-ADAM
(c) GPR-IADAM
(d) SGP-Q

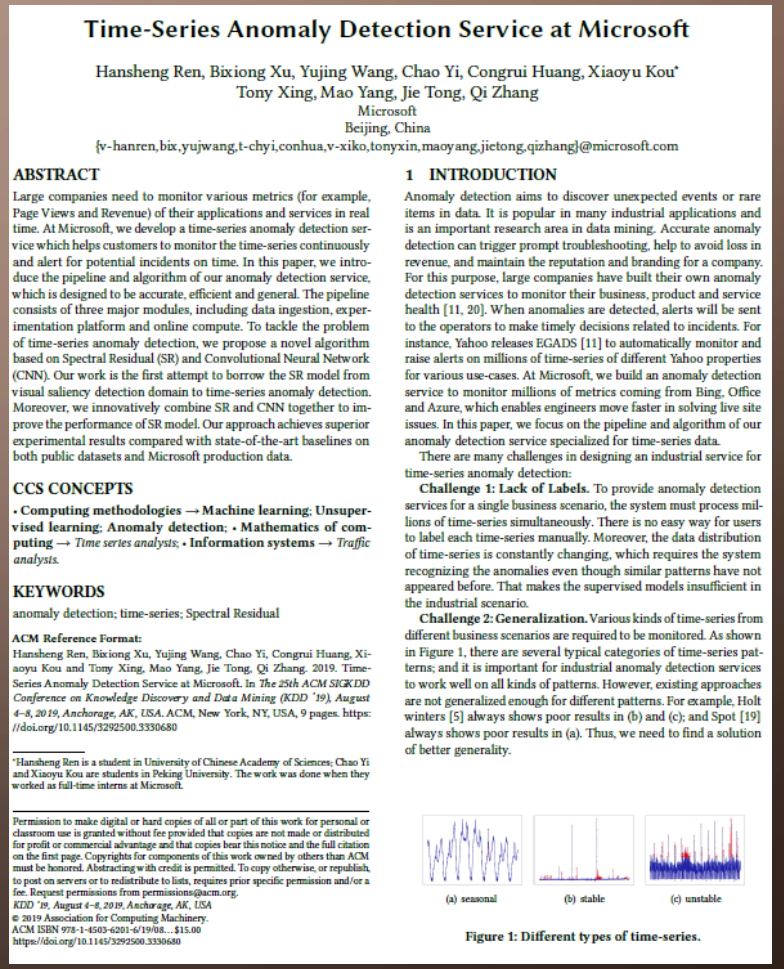
Dataset	GPR-AD	GPR-ADAM	GPR-IADAM	SGP-Q
art_daily_jumpsup	82.20±0.40	91.40±0.80	92.80±0.40	**99.20±0.40**
art_daily_flatmiddle	74.20±0.45	93.20±0.45	93.20±0.45	**96.40±1.09**
ec2_cpu_utilization_24ae8d	98.90±0.74	98.70±0.45	98.70±0.45	**99.00±0.82**
ec2_cpu_utilization_825cc2	93.60±0.55	44.40±0.89	86.40±0.55	**99.60±0.55**
ec2_cpu_utilization_ac20cd	95.40±0.55	29.40±0.55	30.20±0.45	**96.20±1.17**
grok_asg_anomaly	85.20±0.84	52.00±0.00	55.00±0.00	**90.00±1.41**
occupancy_t4013	80.00±0.00	80.40±0.90	80.00±0.00	**83.00±1.22**
speed_t4013	81.40±0.89	77.60±0.54	81.60±0.55	**84.00±0.00**

Table 1
The $F_1(\%)$ score of four methods on eight datasets. The best results are in bold.

MICROSOFT TIME-SERIES ANOMALY DETECTION

- 2019년 6월 자료
- SR (Spectral Residual)
- SR-CNN
- DNN
- SR+DNN

Source: https://arxiv.org/pdf/1906.03821.pdf



Time-Series Anomaly Detection Service at Microsoft

Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou*
Tony Xing, Mao Yang, Jie Tong, Qi Zhang
Microsoft
Beijing, China
{v-hanren,bix,yujwang,t-chyi,conhua,v-xiko,tonyxin,maoyang,jietong,qizhang}@microsoft.com

ABSTRACT

Large companies need to monitor various metrics (for example, Page Views and Revenue) of their applications and services in real time. At Microsoft, we develop a time-series anomaly detection service which helps customers to monitor the time-series continuously and alert for potential incidents on time. In this paper, we introduce the pipeline and algorithm of our anomaly detection service, which is designed to be accurate, efficient and general. The pipeline consists of three major modules, including data ingestion, experimentation platform and online compute. To tackle the problem of time-series anomaly detection, we propose a novel algorithm based on Spectral Residual (SR) and Convolutional Neural Network (CNN). Our work is the first attempt to borrow the SR model from visual saliency detection domain to time-series anomaly detection. Moreover, we innovatively combine SR and CNN together to improve the performance of SR model. Our approach achieves superior experimental results compared with state-of-the-art baselines on both public datasets and Microsoft production data.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Unsupervised learning**; **Anomaly detection**; • **Mathematics of computing** → *Time series analysis*; • **Information systems** → *Traffic analysis*.

KEYWORDS

anomaly detection; time-series; Spectral Residual

ACM Reference Format:

Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou and Tony Xing, Mao Yang, Jie Tong, Qi Zhang. 2019. Time-Series Anomaly Detection Service at Microsoft. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330680

*Hansheng Ren is a student in University of Chinese Academy of Sciences; Chao Yi and Xiaoyu Kou are students in Peking University. The work was done when they worked as full-time interns at Microsoft.

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330680

Figure 1: Different types of time-series.

1 INTRODUCTION

Anomaly detection aims to discover unexpected events or rare items in data. It is popular in many industrial applications and is an important research area in data mining. Accurate anomaly detection can trigger prompt troubleshooting, help to avoid loss in revenue, and maintain the reputation and branding for a company. For this purpose, large companies have built their own anomaly detection services to monitor their business, product and service health [11, 20]. When anomalies are detected, alerts will be sent to the operators to make timely decisions related to incidents. For instance, Yahoo releases EGADS [11] to automatically monitor and raise alerts on millions of time-series of different Yahoo properties for various use-cases. At Microsoft, we build an anomaly detection service to monitor millions of metrics coming from Bing, Office and Azure, which enables engineers move faster in solving live site issues. In this paper, we focus on the pipeline and algorithm of our anomaly detection service specialized for time-series data.

There are many challenges in designing an industrial service for time-series anomaly detection:

Challenge 1: Lack of Labels. To provide anomaly detection services for a single business scenario, the system must process millions of time-series simultaneously. There is no easy way for users to label each time-series manually. Moreover, the data distribution of time-series is constantly changing, which requires the system recognizing the anomalies even though similar patterns have not appeared before. That makes the supervised models insufficient in the industrial scenario.

Challenge 2: Generalization. Various kinds of time-series from different business scenarios are required to be monitored. As shown in Figure 1, there are several typical categories of time-series patterns; and it is important for industrial anomaly detection services to work well on all kinds of patterns. However, existing approaches are not generalized enough for different patterns. For example, Holt winters [5] always shows poor results in (b) and (c); and Spot [19] always shows poor results in (a). Thus, we need to find a solution of better generality.

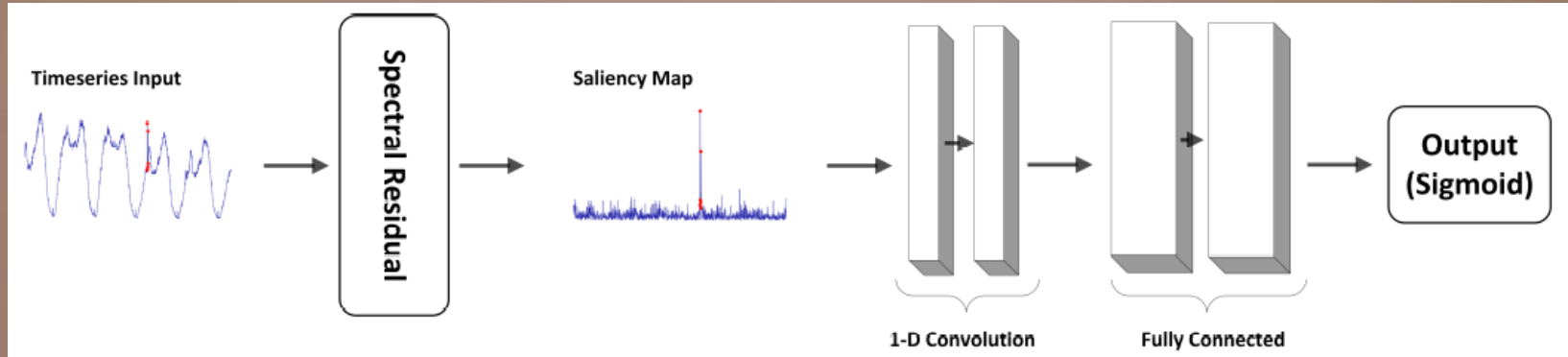


Figure 5: SR-CNN architecture

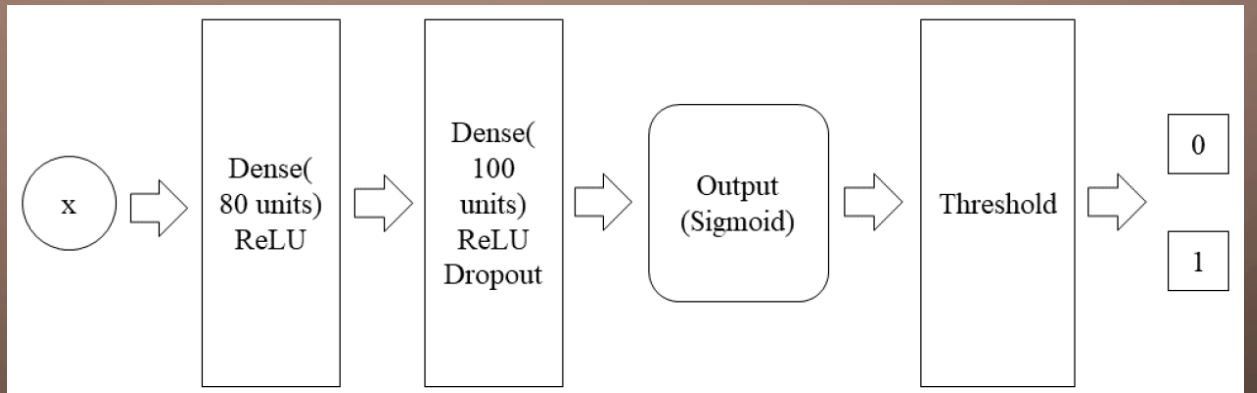


Figure 7: DNN architecture

Table 2: Result comparison of cold-start

	KPI				Yahoo				Microsoft			
Model	F_1-score	*Precision*	*Recall*	*Time*(s)	F_1-score	*Precision*	*Recall*	*Time*(s)	F_1-score	*Precision*	*Recall*	*Time*(s)
FFT	**0.538**	0.478	0.615	3756.63	0.291	0.202	0.517	356.56	0.349	0.812	0.218	8.38
Twitter-AD	0.330	0.411	0.276	523232.0	0.245	0.166	0.462	301601.50	0.347	0.716	0.229	6698.80
Luminol	0.417	0.306	0.650	14244.92	**0.388**	0.254	0.818	1071.25	**0.443**	0.776	0.310	16.26
SR	0.666	0.637	0.697	1427.08	0.529	0.404	0.765	43.59	0.484	0.878	0.334	2.45
SR-CNN	**0.732**	0.811	0.667	6805.13	**0.655**	0.786	0.561	279.97	**0.537**	0.468	0.630	25.26

Table 3: Result comparison on test data

	KPI				Yahoo				Microsoft			
Model	F_1-score	*Precision*	*Recall*	*Time*(s)	F_1-score	*Precision*	*Recall*	*Time*(s)	F_1-score	*Precision*	*Recall*	*Time*(s)
SPOT	0.217	0.786	0.126	9097.85	**0.338**	0.269	0.454	2893.08	0.244	0.702	0.147	9.43
DSPOT	**0.521**	0.623	0.447	1634.41	0.316	0.241	0.458	339.62	0.190	0.394	0.125	1.37
DONUT	0.347	0.371	0.326	24248.13	0.026	0.013	0.825	2572.76	**0.323**	0.241	0.490	288.36
SR	0.622	0.647	0.598	724.02	0.563	0.451	0.747	22.71	0.440	0.814	0.301	1.55
SR-CNN	**0.771**	0.797	0.747	2724.33	**0.652**	0.816	0.542	125.37	**0.507**	0.441	0.595	16.13

Table 6: Train and test split of KPI dataset

DataSet	Total points	Anomaly points
Train	3004066	79554/2.65%
Test	2918847	54560/1.87%

Table 7: Supervised results on KPI dataset

Model	F_1-score	Precision	Recall
DNN	0.798	0.849	0.753
SR+DNN	**0.811**	0.915	0.728

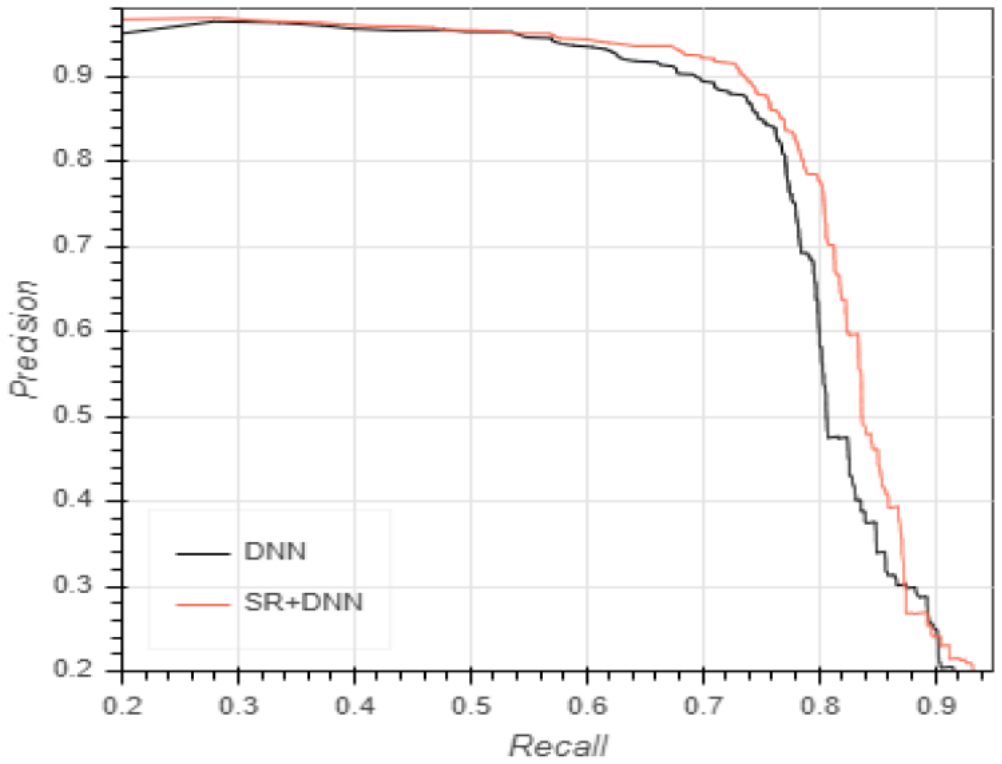


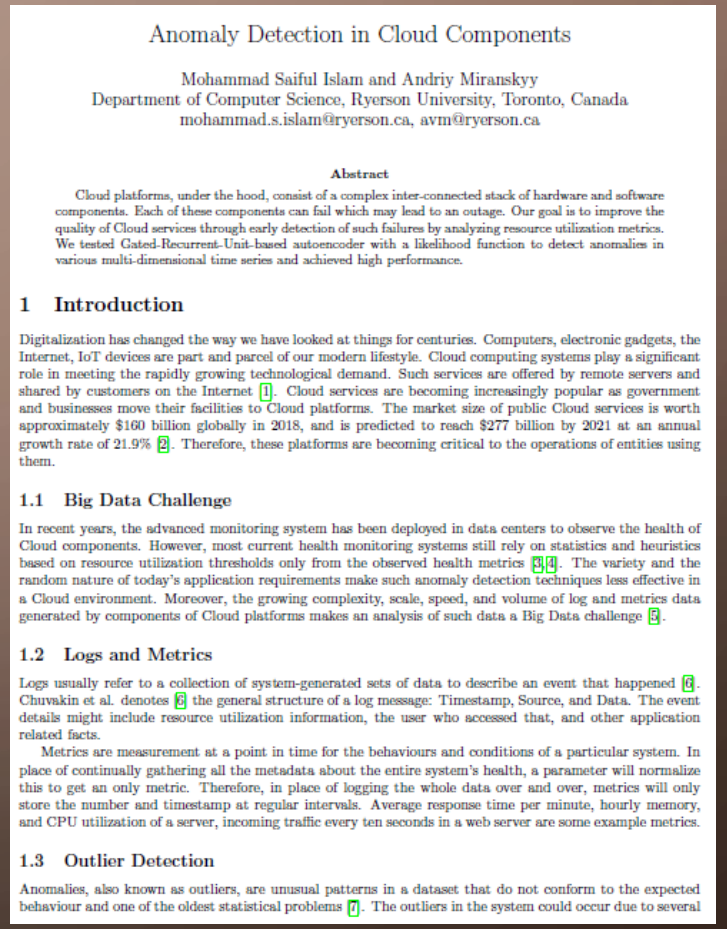
Figure 8: P-R curves of SR+DNN and DNN methods

결과

- SR: 가장 빠른 CPU 런타임; SR-CNN: 높은 정확성 (조금 느린 런타임)
- CNN을 붙였을때 일반화 하기에 더 유용함
- P-R 커브에서 보이듯이 SR을 DNN에 더했을때 DNN만 썼을때의 F-score 보다 1.6% 높음

ANOMALY DETECTION IN CLOUD

- 클라우드 환경의 복잡성때문에 보편적으로 쓰이고있는 시스템 모니터링 방식은 한계가 있어 쉽지 않다
  - 예: ARIMA
- Neural Network Model이 더 적합하다.
- GRU (Gated Recurrent Unit) 을 사용한 모델 의견 제시.
  - 조경현 2014년
  - LSTM Cell 의 간소화 버전 - https://arxiv.org/pdf/1406.1078v3.pdf



Anomaly Detection in Cloud Components

Mohammad Saiful Islam and Andriy Miranskyy
Department of Computer Science, Ryerson University, Toronto, Canada
mohammad.s.islam@ryerson.ca, avm@ryerson.ca

Abstract

Cloud platforms, under the hood, consist of a complex inter-connected stack of hardware and software components. Each of these components can fail which may lead to an outage. Our goal is to improve the quality of Cloud services through early detection of such failures by analyzing resource utilization metrics. We tested Gated-Recurrent-Unit-based autoencoder with a likelihood function to detect anomalies in various multi-dimensional time series and achieved high performance.

1 Introduction

Digitalization has changed the way we have looked at things for centuries. Computers, electronic gadgets, the Internet, IoT devices are part and parcel of our modern lifestyle. Cloud computing systems play a significant role in meeting the rapidly growing technological demand. Such services are offered by remote servers and shared by customers on the Internet [1]. Cloud services are becoming increasingly popular as government and businesses move their facilities to Cloud platforms. The market size of public Cloud services is worth approximately $160 billion globally in 2018, and is predicted to reach $277 billion by 2021 at an annual growth rate of 21.9% [2]. Therefore, these platforms are becoming critical to the operations of entities using them.

1.1 Big Data Challenge

In recent years, the advanced monitoring system has been deployed in data centers to observe the health of Cloud components. However, most current health monitoring systems still rely on statistics and heuristics based on resource utilization thresholds only from the observed health metrics [3,4]. The variety and the random nature of today's application requirements make such anomaly detection techniques less effective in a Cloud environment. Moreover, the growing complexity, scale, speed, and volume of log and metrics data generated by components of Cloud platforms makes an analysis of such data a Big Data challenge [5].

1.2 Logs and Metrics

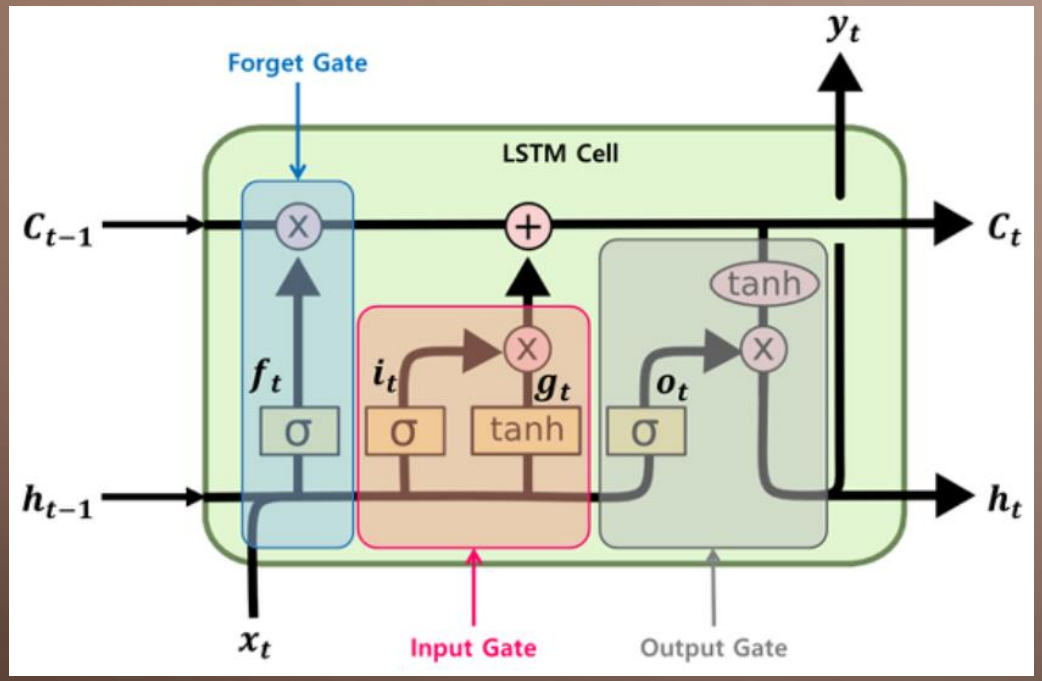
Logs usually refer to a collection of system-generated sets of data to describe an event that happened [6]. Chuvakin et al. denotes [6] the general structure of a log message: Timestamp, Source, and Data. The event details might include resource utilization information, the user who accessed that, and other application related facts.

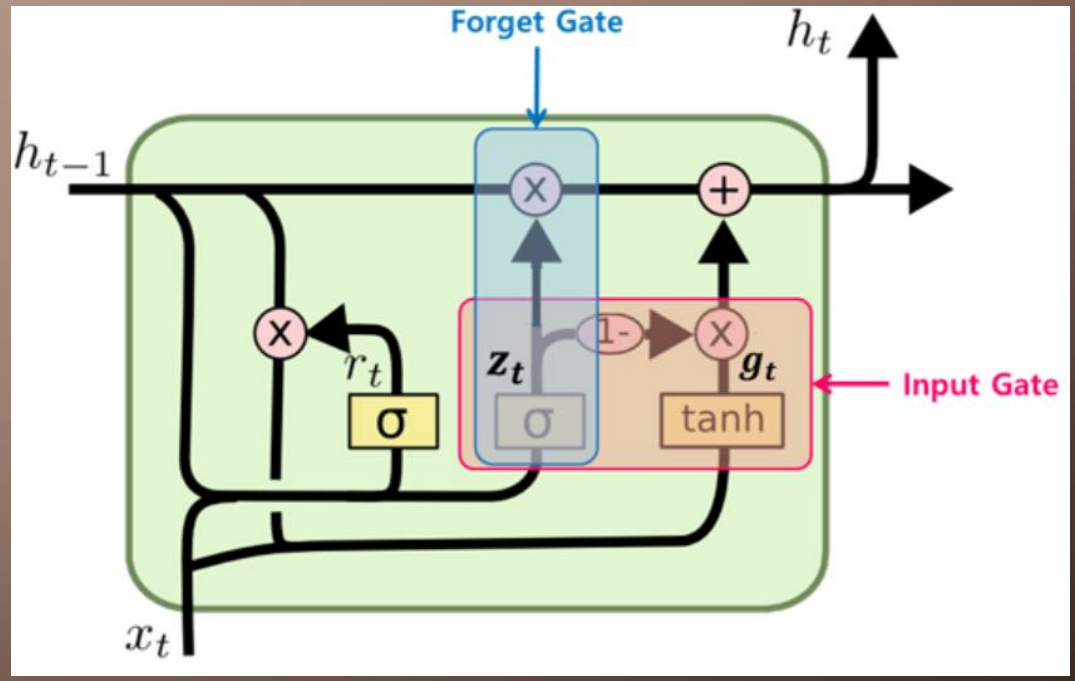
Metrics are measurement at a point in time for the behaviours and conditions of a particular system. In place of continually gathering all the metadata about the entire system's health, a parameter will normalize this to get an only metric. Therefore, in place of logging the whole data over and over, metrics will only store the number and timestamp at regular intervals. Average response time per minute, hourly memory, and CPU utilization of a server, incoming traffic every ten seconds in a web server are some example metrics.

1.3 Outlier Detection

Anomalies, also known as outliers, are unusual patterns in a dataset that do not conform to the expected behaviour and one of the oldest statistical problems [7]. The outliers in the system could occur due to several

LSTM vs GRU





Source: https://excelsior-cjh.tistory.com/185

ONLINE-OFFLINE HYBRID

- 2019 8월 자료
- Offline: Rad-NN (Radius Nearest Neighbor)
- Online: One-class SVM (Support Vector Machine)

A Hybrid Online Offline System for Network Anomaly Detection

Murugaraj Odiathevar
Engineering and Computer Science
Victoria University of Wellington
New Zealand
murugaraj.odiathevar@ecs.vuw.ac.nz

Winston K.G. Seah
Engineering and Computer Science
Victoria University of Wellington
New Zealand
winston.seah@ecs.vuw.ac.nz

Marcus Frean
Engineering and Computer Science
Victoria University of Wellington
New Zealand
marcus.frean@ecs.vuw.ac.nz

***Abstract*—With the advancement in technology, normal network traffic is becoming more heterogeneous. In this scenario, the problem of detecting anomalies is intensified. In the literature, offline methods see more data and can be optimised to achieve lower false positive rates. However, they cannot readily adapt to changing network conditions or capture *concept-drift*. This necessitates an incremental online learning model. On the other hand, online training is easily affected by noise. In this paper, we propose a hybrid Online Offline system in which the Offline model retains general characteristics of network traffic while the Online model continuously learns. The Offline model acts as a bias for the Online model to select new data to learn from. The Online model retains its knowledge and adapts to the changing ground truth. They are put to work together to detect anomalies. We implement this idea with an Online Support Vector Machine (SVM) which retains its support vectors and shifts its decision boundary guided by an Offline Radius Nearest Neighbor (Rad-NN). The method is evaluated on the NSL-KDD 2009 dataset. This relatively simple model achieves over 95% accuracy on known anomalies and over 60% detection rate on most of the unknown anomalies.**

***Index Terms*—Anomaly detection, Concept-drift, Incremental learning, Support Vector Machine, Radius Nearest Neighbour**

I. Introduction

One of the main challenges in anomaly detection is the notion of *concept-drift*. Concept-drift refers to a change in the underlying distribution of the data. With the advancement in technology, new applications are being created almost daily around the world. Also with the recent advances in the Internet of Things (IoT), normal traffic profiles display large variations and heterogeneity. Many network anomaly detection techniques miss this aspect of the network and focus on algorithms. As such, there is a need for systems to be able to adapt to the changing ground truth. And our proposed method attempts to fill this gap.

Traditionally, machine learning models are trained from a dataset and then deployed in the real-world. Whenever the model's efficiency withers, it needs to be retrained with new data. In the supervised learning context, obtaining new data and labelling them manually is painful. Network traffic changes faster than one can label and retrain a model. Unsupervised learning method for anomaly detection mainly consist of outlier detection. Unfortunately, new normal traffic may also be outliers at a different point in time such as *Flash-Crowd* traffic. To combat the weaknesses of each model, hybrid models are used. There have been hybrids between supervised and unsupervised and between misuse-based and behaviour-based intrusion detection systems [1]. Combination between offline and online methods in the literature usually involve an offline step and then online learning. Offline learners are the traditional machine learning models. They can see more data and pick out patterns that are not easily recognisable by humans. Though they are optimised, they suffer from lengthy training times. Online learners are in the form of incremental learners where they learn in time windows as new data arrives. They are more suitable in tracking concept-drift. However, they are required to be quick and thus are difficult to optimise in real time. Furthermore, they suffer from noisy or irrelevant data points during training. Most of the online methods in the literature are generally unsupervised methods. Our approach is a form of semi-supervised learning where the Online model is trained using labelled data with a confidence score. Other semi-supervised learning methods include the use of data from only the normal class such as the one-class Support Vector Machine (SVM) [2] or the use of a few labelled data points [3]. This paper presents a hybrid Online Offline system to address the inadequacies of each individual model. The main challenge is enabling an Online and an Offline model to work together in a logical and effective manner to detect anomalies because these methods are classifier dependent. We address this issue with the use of confidence scores.

Another aspect of machine learning models is that be it supervised, unsupervised or hybrid, they have a bias-variance trade-off. Models with high bias are over simplified and may not have captured the regularities of the data while models with high variance are over-fit to training data and do not generalise well to unseen data. In the network anomaly context, both false positive and false negatives can be costly and this is one of the reasons why machine learning for network anomaly detection is not readily implemented in the real world [4].

This paper attempts to overcome the above mentioned limitations and fill in the gap of adapting to normal network traffic by a novel Hybrid Online Offline System. The proposed contributions of this paper include

(i) A method to enable an Offline and an Online model to work together

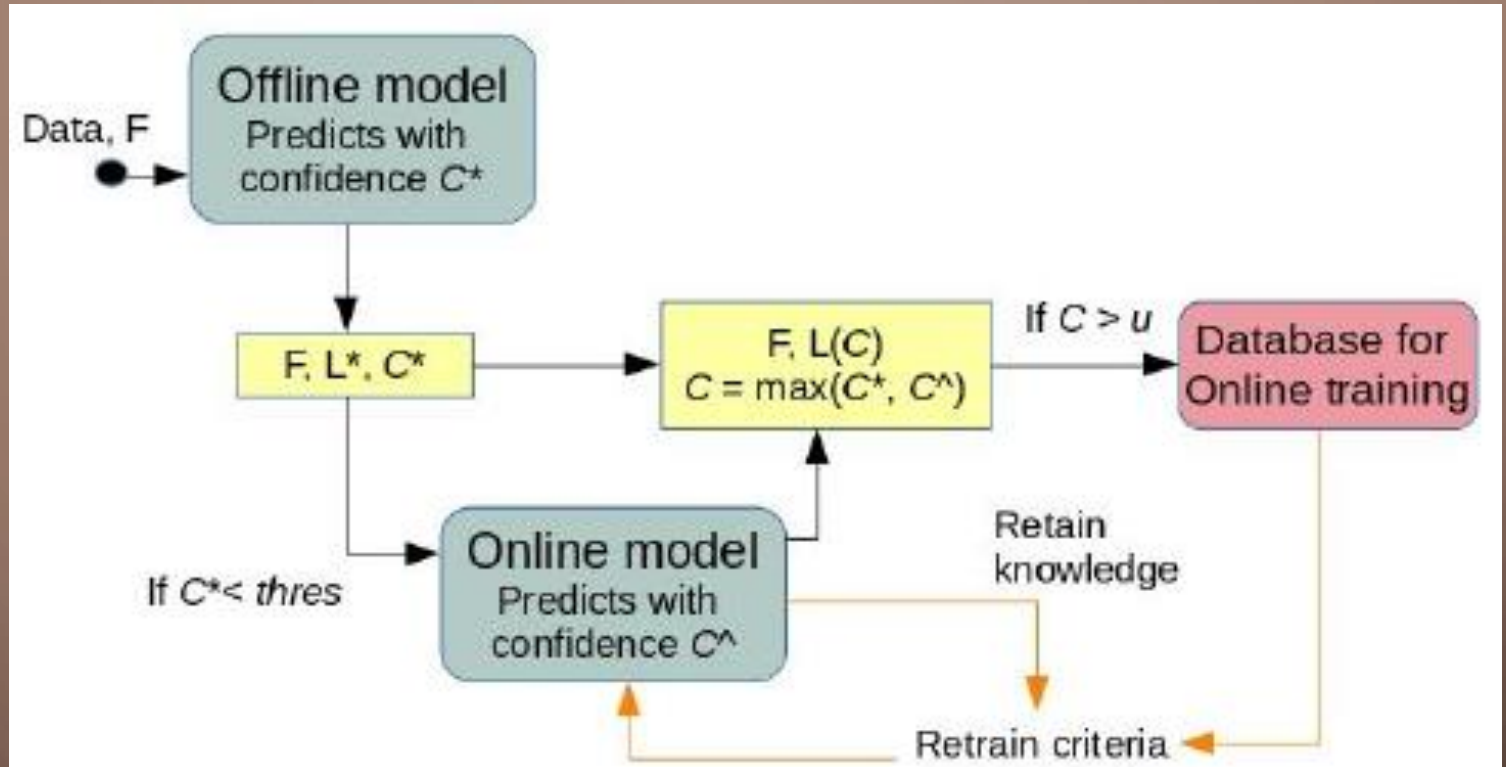


Fig. 1. System overview

TABLE III

COMPARISON WITH OFFLINE METHODS

Model	FPR (%)	TPR (%)	Acc (%)
Our System	2.96	94.42	**95.55**
SVM (C=100, γ=0.1)	8.04	71.01	80.02
Rad-NN	**2.13**	59.33	75.90
Decision Tree	3.37	70.12	81.51
Gaussian Naive Bayes	3.26	62.04	76.95
k-NN (k=5,equal-weighted votes)	2.88	61.33	76.71
Artificial Neural Networks [7]	3.23	81.20	81.16
Deep Recurrent Neural Networks [8]	3.07	72.95	83.28
Self taught learning [9]	21.60[a]	**95.95**	88.39
Stochastically improved Denoising Autoencoder [11]	4.01[a]	83.08	88.65

[a]Derived from the confusion matrix of the published results

TABLE IV

COMPARISON TO OTHER ONLINE METHODS

Model	**FPR (%)**	**TPR (%)**	**Acc (%)**
Our System	**2.96**	**94.42**	**95.55**
SOM & Neural Network [14]	14.06	94.40	90.00
2 layer GMM clustering [13]	7.00	85.00	88.44[a]
Deep Belief Network with Adaptive Linear Function [16]	4.47[a]	95.2	95.34[a]

[a]Derived from the confusion matrix of the published results

기타

- Luminol (LinkedIn AD)
- Elastic X-pack
- Support Vector Machine (SVM)
- Intuitionistic Fuzzy Time Series (IFTS) Graph Mining
- Random Cut Forest
- Streaming Peak-Over-Threshold (SPOT)
  - SPOT with Drift (DSPOT)

마치며

- 다양한 / 조합적 모델 실험 (한 논문은 STL-ARMA-KF)
  - No absolute correct algorithm exists
  - Article “Experimental Comparison of Online AD Algorithm”
    - … If the time series data in a user’s application exhibits concept drift, the user may want to consider a RNN with GRU units and not Twitter AnomalyDetection. Instead of doing an extensive literature review and trying every anomaly detection method in a rapidly expanding library, one could just observe characteristics present in the data and narrow the choice down to a smaller class of promising anomaly detection methods.
- 실시간 → 머신러닝

마치며

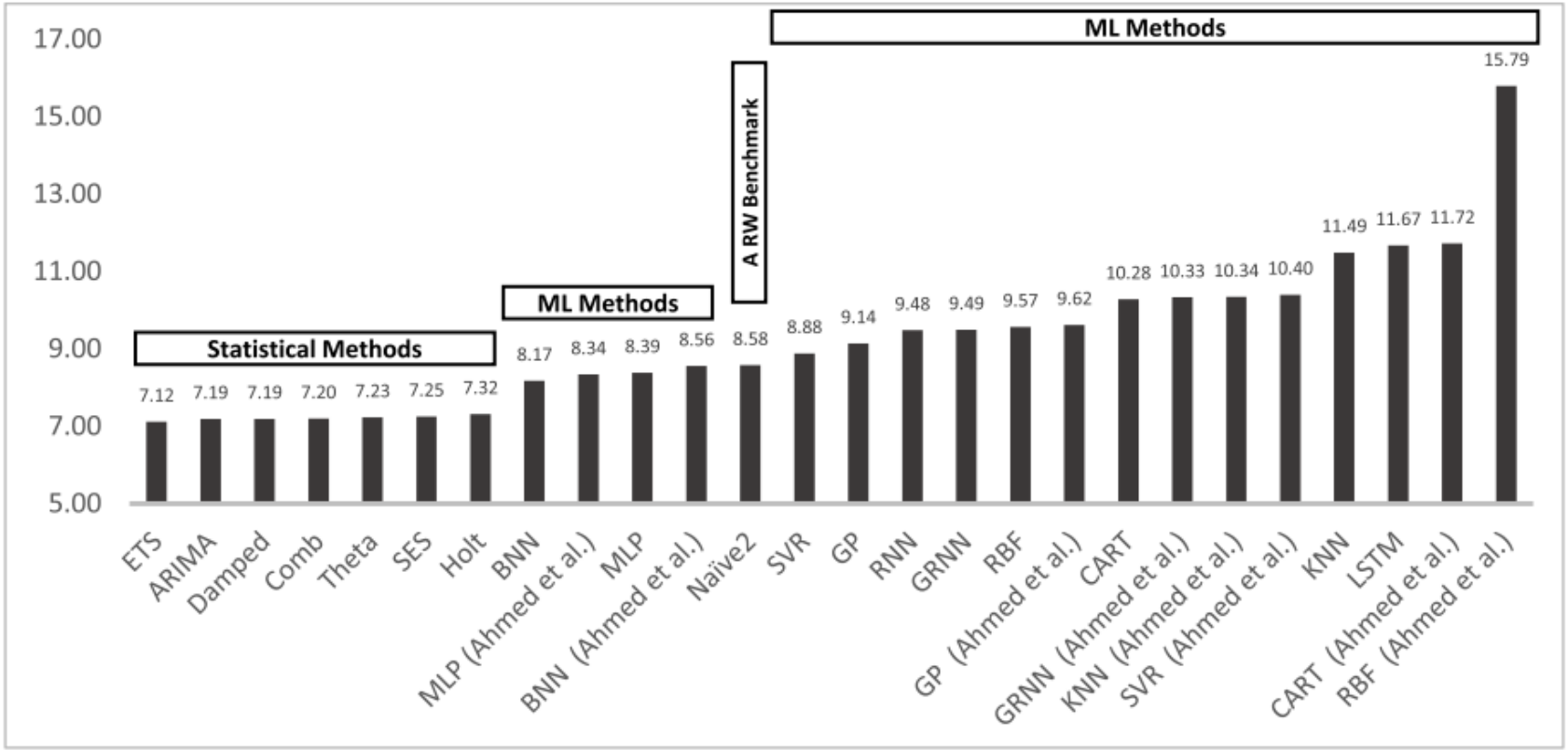
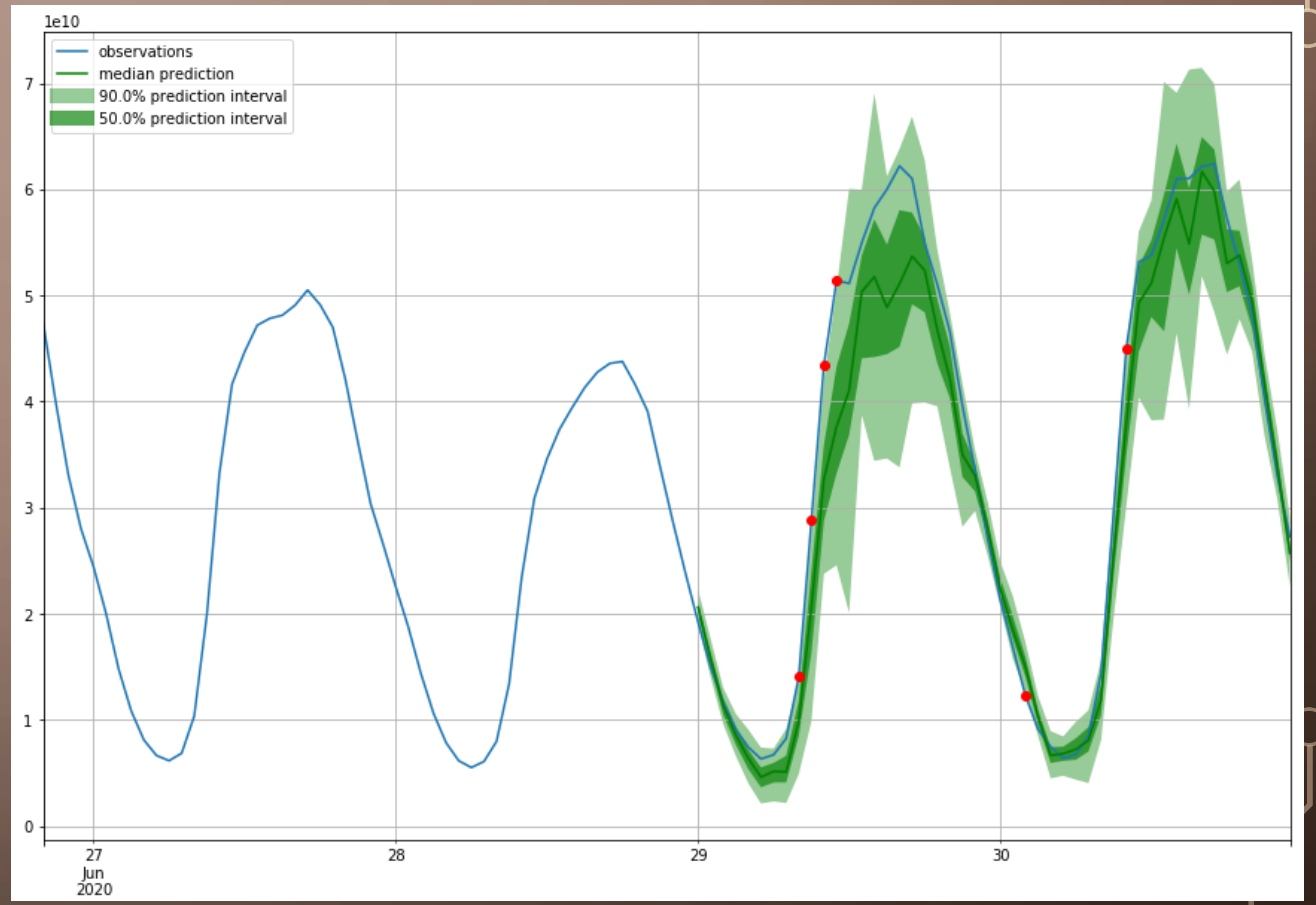


Fig 2. Forecasting performance (sMAPE) of the ML and statistical methods included in the study. The results are reported for one-step-ahead forecasts having applied the most appropriate preprocessing alternative.

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0194889

예시

- 2019/06/12
- GluonTS
- Feedforward ANN (MLP)
- 90% 신뢰구간
- 메모리: < 300MB



Q&A