Anomaly Detection in Zabbix: Present and Future



Aleksandrs Kalimulins

Product Team



•. •

ZABBIX

SUMMIT

2024

Introduction

\$

What is AI used for in IT infrastructure monitoring?

- AI has become a crucial tool in IT infrastructure monitoring, providing a more proactive and efficient way to manage and optimize systems. Here are some key uses of AI in this area:
 - Anomaly Detection: Al algorithms can quickly identify deviations from normal behavior in IT systems. These anomalies might indicate potential issues like hardware failures, cyberattacks, or configuration errors. By detecting anomalies early, Al helps prevent these issues from escalating into serious problems.





What is an anomaly?



Image credit: Herluf Bidstrup

© 2024 by Zabbix. All rights reserve





What is an anomaly?

anomaly noun

anom·a·ly (ə-ˈnä-mə-lē ◄»)

plural anomalies

Synonyms of *anomaly* >

something different, abnormal, peculiar, or not easily classified : something anomalous

They regarded the test results as an *anomaly*.

2 : deviation from the common rule : IRREGULARITY





What is an anomaly in time series?

An anomaly in a time series is a **rare** or **unexpected** point or sequence occurring over a specified time interval, often considered unusual or undesirable.





Anomaly types

Point-based



© 2024 by Zabbix. All rights reserved





Anomaly types

Short duration
Easy to spot
Easy to detect

Point-based

 sometimes even with stddevsamp()





Anomaly types

Subsequence-based



© 2024 by Zabbix. All rights reserved





© 2024 by Zabbix. All rights reserved

Anomaly types

Subsequence-based

Longer duration

- Harder to spot
- Difficult to detect







How does Zabbix do it?

Trigger functions

- stddevpop(), stddevsamp(), mad()
- varpop(), varsamp()
- baselinedev()
- trendstl()



How does Zabbix do it?

trendstl() - features

- implements STL anomaly detection algorithm
- decomposes data into trend, seasons, residual
- data must have pronounced seasonal pattern



How does Zabbix do it?

trendstl() - shortcomings

7 parameters (need a data science degree)

trendstl(/host/key,100h:now/h-10h,100h,2h,3,"mad",1001)

- careful choice of seasonality
- no support for multiple seasons

subsequence-based anomalies out of scope





More detection methods

Requirements

- Performance and efficiency
- Easy to use (the less knobs and switches the better)
- Easy to understand and interpret results
- Subsequence-based anomalies
- Multiple seasonalities







| RobustPCA [101] Eros-SVMs [74] k-Means [151] XGBoosting [34] KNN [110] | SR[112] DWT-MLEAD[134] H CMC (MCC) |
|--|---|
| NetworkSVM [160] MS-SVDD [149] sequenceMiner [23] AOSVM [48] | I-HMM [127] U-GMM-HMM [68] |
| RUSBoost [54] OC-KFD [114] PhaseSpace-SVM [85] NoveltySVR [86] | Signal AnalysisSmartSifter [152]LaserDBN [100] |
| Random Black Forest [165] Classic ML SLADE-TS [141] PCA [121] S-SVM [11] | Online DWT- FFT[111] GLA [84] Stochastic |
| Hybrid K-Means [140] Random Forest Regressor [165] | EM-HMM [105] Learning EDBN [107] |
| SLADE-MTS [142] PCC [121] Normaliz Hybrid KNN [124] ISTA | ing Flow [116] M. bacad EncDec-AD [88] MultiHMM [78] HSMM [129] CxDBN [137] |
| STOMP[164] HBOS[47] DeepISTM[31] SSA[155] VAF-0 | GAN [98] LAMP [166] FuzzyDNBC [136] |
| Sevies 2 Create [16] | DAE [117] TCN-AE [135] HMAD [49] |
| DeepNAP[72] LSTM-VAE[106] | AD-GAN [77] Omni Anomaly [125] |
| GrammarViz[120] TwoFinger [90] CoalESN [99] Torsk [60] | |
| V 0 0 [100] Left STAMPi[156] STOPN[123] Deen I | AD-LII[148] ConInd [5] |
| KnorrSeq2[102] Left STRAT [130] STORN [125] Donut [150] | DeepAnT[94] 		 S-H-ESD[62] |
| TSBitmap[144] DADS [119] MSCRED [159] | TTM [146] T-lementer [64] LSTM-AD [89] FAST-MCD [115] SH-ESD+ [138] |
| HOT SAX[70] DissimilarityAlgo [6] DADM [40] | IIM[146] Telemanom[64] |
| RADM [40] SR-CNN [112] TANG | oGAN[8] 4E[117] VELC[158] MA[18] EWMA[65] SARIMA[52] |
| NorM [14] Data Mining MoteESN [30] Nument MTAD-GAT [161] | taHTM[3] HealthESN[32] ANODE [96] Kalman Filter [52] |
| BoehmerGranh [13] MAXMOD Local Derifered Image-embe | edding-CAE [44] MGDD [126] Statistics |
| bochmeroraph [15] VALMOD[82] PS1[128] | PCI[157] |
| TARZAN[71] MERLIN [97] STAMP [156] MCOD [73] Isolation Fores | CBLOF[59] ARMA [18] st [83] EIF [58] pEWMA [25] MedianMethod [10] |
| NormA-SJ[15] ILOF [108] DAD [154] LOCI/aLOCI [103] Subsequence | e IF [83] Subsequence LOF [22] Holt-Winter's [1] |
| NormA-smpl [15] SurpriseEncoding [26] IF-LOF [36] Outlier 1 | COPOD [80] ARIMA [65] DSPOT [122] RePAD [76] Corector GeckoFSM [118] |
| SCRIMP++[163] Ensemble GI[43] Hybrid Isolation Forest[91] COF[130] BLOF[59] | DBStream [55] LOF [22] DILOF [95] AMD Segmentation [153] Holt's [65] |

"Anomaly Detection in Time Series: A Comprehensive Evaluation"

© 2024 by Zabbix. All rights reserved

Input types



ZABBIX SUMMIT 2024



Training types

Supervised

- Manual training
- Semi-supervised
 - Training on a clean data
- Unsupervised
 - No training



Image credit: Herluf Bidstrup

© 2024 by Zabbix. All rights reserved



Quantum Algo

Published as a conference paper at ICLR 2019

OUTLIER EXPOSURE

University of California, Berkeley

hendrycks@berkeley.edu

INTRODUCTION

Dan Hendrycks

DEEP ANOMALY DETECTION WITH

auxiliary dataset that improve performance

carefully proceeding with a more conservative fallback policy.

2023 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEAR Journal of Artificial Intelligence Research 46 (2013) 235-262

Ming-Chao Gu Qin,¹ Xiao-¹State Key Beijing Universit ²Comprehensive Research Center

LOW-COUNT TIME § Philipp Renz^{*,1}

Mantas Mazeika

ABSTRACT

It is important to detect anomalous inputs when deploying machine learning

systems. The use of larger and more complex inputs in deep learning magnifies

the difficulty of distinguishing between anomalous and in-distribution examples.

At the same time, diverse image and text data are available in enormous quantities.

We propose leveraging these data to improve deep anomaly detection by training

anomaly detectors against an auxiliary dataset of outliers, an approach we call Outlier Exposure (OE). This enables anomaly detectors to generalize and detect

unseen anomalies. In extensive experiments on natural language processing and

small- and large-scale vision tasks, we find that Outlier Exposure significantly

improves detection performance. We also observe that cutting-edge generative

models trained on CIFAR-10 may assign higher likelihoods to SVHN images

than to CIFAR-10 images; we use OE to mitigate this issue. We also analyze the

flexibility and robustness of Outlier Exposure, and identify characteristics of the

Machine Learning systems in deployment often encounter data that is unlike the model's tra

data. This can occur in discovering novel astronomical phenomena, finding unknown diseas

detecting sensor failure. In these situations, models that can detect anomalies (Liu et al., 2

Emmott et al., 2013) are capable of correctly flagging unusual examples for human intervention

Behind many machine learning systems are deep learning models (Krizhevsky et al., 2012) v

can provide high performance in a variety of applications, so long as the data seen at test tin

similar to the training data. However, when there is a distribution mismatch, deep neural net

classifiers tend to give high confidence predictions on anomalous test examples (Nguyen e

2015). This can invalidate the use of prediction probabilities as calibrated confidence estin

Several previous works seek to address these problems by giving deep neural network class

a means of assigning anomaly scores to inputs. These scores can then be used for detecting

of-distribution (OOD) examples (Hendrycks & Gimpel, 2017; Lee et al., 2018; Liu et al., 2

These approaches have been demonstrated to work surprisingly well for complex input spaces,

as images, text, and speech. Moreover, they do not require modeling the full data distribution

instead can use heuristics for detecting unmodeled phenomena. Several of these methods of

(Guo et al., 2017), and makes detecting anomalous examples doubly important.

University of Chicago

mantas@ttic.edu

Kurt Cutajar[†]

Thomas Dietterich

Oregon State University

tgd@oregonstate.edu

Toward Supervised An

Detection in the Internet of Things (IoT) Envi

Eleonora Achiluzzi, Menglu Li, Md Fahd Al Georgy, and Rasha

Exploring the Use of Data-Driven Approaches fo

Toronto Metropolitan University

Published a

{eachiluzzi, menglu.li, mgec

Abstract-The Internet of Things (IoT) is a system that connects physical computing devices, sensors, software, and other technologies. Data can be collected transferred, and exchanged with other devices over the network without requiring human interactions. One challenge the development of IoT faces is the existence of anomaly data in the network. Therefore, research on anomaly detection in the IoT environment has become popular and necessary in recent years. This survey provides an overview to understand the current

TAnoGAN: Time Series Anomaly Detection with Generative Adversarial Networks

BRE

Md Abul Bashar School of Computer Science Centre for Data Science Queensland University of Technology Brisbane, Oueensland 4000, Australia Email: m1.bashar@qut.edu.au

aboratory Technische Universität Berl

Rerlin Germany

Darmetad

ttering Cancer Center

gen, Dep. of Computer Science

7077 Göttingen, Germany

ogy Center

Richi Navak School of Computer Science Centre for Data Science Queensland University of Technology Brisbane, Oueensland 4000, Australia Email: r.nayak@qut.edu.au

Abstract—Anomaly detection in time series data is a significant problem faced in many application areas such as manufacturing, medical imaging and cyber-security. Recently, Generative Adversarial Networks (GAN) have gained attention for generation and anomaly detection in image domain. In this paper, we propose a novel GAN-based unsupervised method called TAnoGan for detecting anomalies in time series when a small number of data potential of GAN in time series domain.

points are available. We evaluate TAnoGan with 46 real-world time series datasets that cover a variety of domains. Extensive experimental results show that TAnoGan performs better than traditional and neural network models.

I. INTRODUCTION The ubiquitous use of networked sensors and actuators in places like smart buildings, factories, power plants and data centres as well as the emergence of the Internet of Things (IoT) have resulted in generating substantial amounts of time series data. These data can be used to continuously monitor the

unmodeled phenomena by using representations from only in-distribution data In this paper, we investigate a complementary method where we train models to detect unmore working conditions of these environments to detect anomalies. data by learning cues for whether an input is unmodeled. While it is difficult to model the full

time series data [10]. Recently a GAN framework coupled with the mapping of data to latent space has been explored for anomaly detection [3], [2]. While GAN has been extensively investigated in image domain for generation and anomaly detection, only a few works (e.g. [10], [2]) have explored the

In this paper, we propose a novel method, Time series Anomaly detection with GAN (TAnoGan)1, for unsupervised anomaly detection in time series data when a small number of data points are available. Detecting anomalies in time series using GAN requires modelling the normal behaviour of time series data using the adversarial training process and then detecting anomalies using an anomaly score that indicates how much the data points have deviated from the normal behaviour [3], [12], [2]. For learning the anomaly score, we first map the real time series data space to a latent space and then reconstruct the data from latent space. The anomaly score is

Proc. of the International Workshop on User Understanding from Big

Zhiyuan Yao

Cisco Meraki

Paris, France

Develop End-to-End Anomaly Detec

Emanuele Mengoli École Polytechnique Paris, France em.mengoli@gmail.com vzhivuan@cisco.com

quantifying responsibility Abstract-Anomaly detection plays a crucial role in ensuring network robustness. However, implementing intelligent alerting systems becomes a challenge when considering scenarios in straightforward exercise, t significantly when faced which anomalies can be caused by both malicious and non-TIN malicious events, leading to the difficulty of determining anomaly causal factors. The identif great interest to users and atterns. The lack of labeled data in the computer networking BY RE domain further exacerbates this issue, impeding the development of robust models capable of handling real-world scenarios. supervised methods require truth data to establish caus To address this challenge, in this paper, we propose an end-to-end anomaly detection model development pipeline. This In this paper, we prese allows continuously evalu framework makes it possible to consume user feedback and Ming Jin¹ enable continuous user-centric model performance evaluation and optimization. We demonstrate the efficacy of the framework anomaly detection model. tion of our framework is a Pin-Yu Cl by way of introducing and bench-marking a new forecasting model – named Lachesis – on a real-world networking problem. abnormally high amount c networking switches devel ¹Monash | Experiments have demonstrated the robustness and effectiveness The development of an of the two proposed versions of Lachesis compared with other ⁶The Hon models proposed in the literature. Our findings underscore the ability to continuously qua potential for improving the performance of data-driven products root causes presents a nun {ming.jin, yuantang.li}@monash.edu, pin-yu.chen@ibm.com yuxliang@outlook.com, s.pan@griffith.edu.au, gingsongedu@c {weiming.wsy,lintao.mlt,chuzhixuan.czx,james.z,peter.sxm}@

ABSTRACT

Time series forecasting holds significant importance in many real-wor systems and has been extensively studied. Unlike natural language pro and computer vision (CV), where a single large model can tackle mu models for time series forecasting are often specialized, necessitating signs for different tasks and applications. While pre-trained foundat have made impressive strides in NLP and CV, their development in domains has been constrained by data sparsity. Recent studies have re large language models (LLMs) possess robust pattern recognition and abilities over complex sequences of tokens. However, the challenge effectively aligning the modalities of time series data and natural 1 leverage these capabilities. In this work, we present TIME-LLM, a ming framework to repurpose LLMs for general time series forecasti backbone language models kept intact. We begin by reprogrammin aline ended with and an action of high feedback disc de high feedback at the feedback of the feedback at the f



ZABBIX

SUMMIT

2024

compared tegers 1 ing linear tography for the r tion of qu tum Mac a promis made gre ing [16], l [20], and Anoma data that been exte fraud det care [26]. been pro semi-supe unsuperv as they d proposed (called L(unsuperv rithm is t containin ever, simi

Quanti

computat

is quite ti * liwenmir † gaof@bu

Types

Statistical methods

Z-score, Moving Average, ARIMA, ...

Machine learning and data analysis methods

• STL, FFT-based anomaly detection, ...

Deep learning methods

LSTM autoencoders, RNN, ...



ZABBIX

SUMMIT

2024

Deep learning methods

Autoencoders

- Recurrent Neural Networks (RNNs)
- Generative Adversarial Networks (GANs)
- Convolutional Neural Networks (CNNs)

LLMs (even!)



"most anomaly detection algorithms, especially ones based on deep learning, have ten or more parameters"

Wu, R., et al. "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress" (2020)

"we are not aware of a single paper that presents forceful reproducible evidence that deep learning outperforms much simpler methods"

Wu, R., et al. "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress" (2020)







"Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress"

© 2024 by Zabbix. All rights reserved





"We found that deep learning approaches are not (yet) competitive despite their higher processing effort on training data"

Schmidl, S., et al. "Anomaly Detection in Time Series: A Comprehensive Evaluation." (2022)

"Our experiments showed that the classical machine learning methods ... outperform the deep learning methods."

Rewicki, F., et al. "Is it worth it? Comparing six deep and classical methods for unsupervised anomaly detection in time series" (2023)





"The experiments showed that the statistical approaches perform best on univariate time-series ... They also require less computation time compared to the other two classes.

Although deep learning approaches have gained huge attention by the ... community in the last years, our results have revealed that they are not ... able to achieve the accuracy values of the statistical methods on the univariate time-series benchmarks"

Braei, M., et al. "Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art" (2020)





"... those [deep learning] methods, though potentially useful ..., do not bring much additional value for the task of TAD [anomaly detection] and their complexity is definitely not justified.

What is even more worrisome, is that they managed to create up to now an illusion of progress"

M. Saquib Sarfraz, et al. "Position: Quo Vadis, Unsupervised Time Series Anomaly Detection?" (2024)





Zabbix requirements

- Performance and efficiency
- Easy to use (the less knobs and switches the better)
- Easy to understand and interpret results
- Subsequence-based anomalies
- Multiple seasonalities



ZABBIX

SUMMIT

2024

Distance-based methods

| Learn. Algorithm | TL OOM ERR | AUC-ROC all datasets |
|------------------------|--------------|----------------------|
| • Sub-LOF [22] | 2% 0% 0% | |
| ▼ GrammarViz [120] | 3 % 0 % 0 % | |
| DWT-MLEAD [134] | 0% 0% 0% | |
| • VALMOD [82] | 1 % 9 % 11 % | |
| SAND [17] | 5 % 1 % 22 % | |
| Left STAMPi [156] | 2% 0% 1% | |

"Anomaly Detection in Time Series: A Comprehensive Evaluation"



ZABBIX

SUMMIT

2024

- Method, not an algorithm
 - STAMP, STOMP, SCRIMP, SCRIMP++, SCAMP, VALMOD, MERLIN, ...
- Distance-based
 - Calculates nearest neighbours
- Subsequences of constant length
 - Motifs similar subsequences
 - Discords anomalies
- Fast on long subsequences
- Predictable time



ZABBIX summit 2024

- Applicable for subsequence and point-based anomalies
- Easily parallelizable
- ► No false positives
- No tuning parameters
- Used in many domains
 - IT infrastructure, IoT, space and sattelites, medicine, seismology, industrial equipment, water distribution, ...





Calculation



024 by Zabbix. All rights reserved







Time Series

ZABBIX SUMMIT 2024

- 50

40

- 30

- 20

10

Calculation (naïve approach)

- Get distances between all subsequences
- Get minimum for gevery row
- Compose profile

| | | | | | | © 20 | 24 b | y Zab | bix. | All rig | ghts | reser | ved | | | | | | | |
|------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|--------------|---------|----------|-------|-------|-------|-------|-------|-------|--------|------|
| | | o | | 2 | | 4 | | | 6 | | 8 | | 10 | | 12 | | 14 | | 16 | |
| | | 2.20 | 4.44 | 6.11 | 5.1 | 5 4.4 | 14 6. | .02 5 | .15 | 2.20 | 17.03 | 29.49 | 29.45 | 21.38 | 5.28 | 8.80 | 5.67 | 7 7.0 | 7 5.67 | 5.28 |
| | | | | | | | | | | 1 | 4atrix | Profile | | | | | | | | |
| | | 0 | | 2 | | 4 | | 6 | S | 8 ubseque | nce Ind | 10 ex | | 12 | | 14 | | 16 | | |
| | | 11.03 | 16.76 | 13.61 | 9.09 | 15.58 | 16.76 | 11.51 | 10.27 | 32.68 | 45.47 | 40.57 | 27.02 | 5.28 | 12.60 | 10.53 | 9.70 | 9.37 | 0.00 | |
| | 16 - | 11.07 | 13.03 | 10.81 | 13.40 | 13.32 | 11.20 | 13.34 | 12.13 | 29.25 | 40.48 | 40.02 | 33.30 | 13.42 | 11.28 | 5.67 | 9.30 | 0.00 | 9.37 | |
| | | 7.07 | 10.49 | 14.31 | 11.46 | 7.61 | 13.26 | 14.47 | 7.27 | 24.27 | 39.99 | 42.81 | 33.22 | 11.85 | 8.80 | 9.10 | 0.00 | 9.30 | 9.70 | |
| | 14 - | 14.03 | 16.13 | 16.27 | 16.78 | 15.04 | 15.79 | 18.01 | 14.70 | 30.78 | 44.29 | 45.62 | 36.97 | 12.45 | 6.31 | 0.00 | 9.10 | 5.67 | 10.53 | - |
| file | | 15.64 | 18.12 | 20.75 | 18.80 | 15.51 | 19.69 | 21.46 | 15.80 | 30.49 | 46.82 | 50.11 | 39.27 | 12.28 | 0.00 | 6.31 | 8.80 | 11.28 | 12.60 | |
| | 12 - | 14.87 | 20.77 | 18.84 | 12.93 | 18.68 | 21.58 | 16.24 | 13.70 | 35.56 | 49.97 | 45.19 | 28.75 | 0.00 | 12.28 | 12.45 | 11.85 | 13.42 | 5.28 | |
| | N 10 | 28.48 | 33.38 | 26.78 | 22.04 | 33.95 | 32.66 | 21.38 | 27.21 | 45.60 | 50.77 | 28.93 | 0.00 | 28.75 | 39.27 | 36.97 | 33.22 | 33.30 | 27.02 | |
| for | psequ | 35.26 | 29.59 | 33.61 | 39.10 | 32.79 | 29.49 | 36.99 | 36.84 | 22.67 | 0.00 | 34.61 | 50.77 | 49.97 | 46.82 | 44.29 | 39.99 | 40.48 | 45.47 | |
| | ence 8 | 22.13 | 17.03 | 26.78 | 27.86 | 17.15 | 21.39 | 28.81 | 23.26 | 0.00 | 22.67 | 43.60 | 45.60 | 35.56 | 30.49 | 30.78 | 24.27 | 29.25 | 32.68 | |
| 2 | Inde | 2.20 | 8.70 | 10.71 | 5.54 | 7.30 | 11.46 | 8.96 | 0.00 | 23.26 | 36.84 | 36.66 | 27.21 | 13.70 | 15.80 | 14.70 | 7.27 | 12.13 | 10.27 | F |
| | 6 - × | 9.00 | 13.17 | 6.11 | 5.15 | 14.57 | 11.70 | 0.00 | 8.96 | 28.81 | 36.99 | 29.45 | 21.38 | 16.24 | 21.46 | 18.01 | 14.47 | 13.34 | 11.51 | |
| | | 9.35 | 6.02 | 6.68 | 13.58 | 9.93 | 0.00 | 11.70 | 11.46 | 21.39 | 29.49 | 32.67 | 32.66 | 21.58 | 19.69 | 15.79 | 13.26 | 11.20 | 16.76 | |
| | 4 - | 5.94 | 4.44 | 13.37 | 12.63 | 0.00 | 9.93 | 14.57 | 7.30 | 17.15 | 32.79 | 39.36 | 33.95 | 18.68 | 15.51 | 15.04 | 7.61 | 13.32 | 15.58 | |
| | | 6.78 | 12.87 | 9.82 | 0.00 | 12.63 | 13.58 | 5.15 | 5.54 | 27.86 | 39.10 | 33.69 | 22.04 | 12.93 | 18.80 | 16.78 | 11.46 | 13.40 | 9.09 | |
| | 2 - | 9.43 | 10.57 | 0.00 | 9.82 | 13.37 | 6.68 | 6.11 | 10.71 | 26.78 | 33.61 | 29.49 | 26.78 | 18.84 | 20.75 | 16.27 | 14.31 | 10.81 | 13.61 | |
| | Ū | 6.68 | 0.00 | 10.57 | 12.87 | 4.44 | 6.02 | 13.17 | 8.70 | 17.03 | 29.59 | 35.84 | 33.38 | 20.77 | 18.12 | 16.13 | 10.49 | 13.03 | 16.76 | |
| | 0 - | 0.00 | 6.68 | 9.43 | 6.78 | 5.94 | 9.35 | 9.00 | 2.20 | 22.13 | 35.26 | 36.14 | 28.48 | 14.87 | 15.64 | 14.03 | 7.07 | 11.07 | 11.03 | - |

Raw data - load







© 2024 by Zabbix. All rights reserved

STL analysis – not good



ZABBIX summit 2024





© 2024 by Zabbix. All rights reserved

Matrix Profile analysis







Raw data – CPU utilization









Matrix Profile analysis: Z-normalization











Matrix Profile analysis: Z-normalization off



© 2024 by Zabbix. All rights reserved





Raw data – ECG



^{© 2024} by Zabbix. All rights reserved





Matrix Profile analysis











Zabbix requirements

- Performance and efficiency
- Easy to use
- Easy to understand and interpret results
- Subsequence-based anomalies
- Multiple seasonalities





Multiple seasonalities

Contenders

Prophet

- Released by Facebook in 2017
- Unlimited seasonalities
- Easy to use
- Designed for business data
- Primary goal is forecasting

ZABBIX summit 2024

Forecasting at Scale

Sean J. Taylor*† Facebook, Menlo Park, California, United States sjt@fb.com

and

Benjamin Letham[†] Facebook, Menlo Park, California, United States bletham@fb.com

Abstract

Forecasting is a common data science task that helps organizations with capacity planning, goal setting, and anomaly detection. Despite its importance, there are serious challenges associated with producing reliable and high quality forecasts – especially when there are a variety of time series and analysts with expertise in time series modeling are relatively rare. To address these challenges, we describe a practical approach to forecasting "at scale" that combines configurable models with analyst-in-the-loop performance analysis. We propose a modular regression model with interpretable parameters that can be intuitively adjusted by analysts with domain knowledge about the time series. We describe performance analyses to compare and evaluate forecasting procedures, and automatically flag forecasts for manual review and adjustment. Tools that help analysts to use their expertise most effectively enable reliable, practical forecasting of business time series.

 $\mathit{Keywords:}$ Time Series, Statistical Practice, Nonlinear Regression



Multiple seasonalities

Contenders

► MSTL

- New, published in 2021
- Extension of STL
- Unlimited seasonalities, detects seasons automaticaly
- Tuning parameters
- Faster than Prophet

MSTL: A Seasonal-Trend Decomposition Algorithm for Time Series with Multiple Seasonal Patterns ZABBIX

SUMMIT

2024

Kasun Bandara^{a,*}, Rob J Hyndman^b, Christoph Bergmeir^c

^aSchool of Computing and Information Systems, Melbourne Centre for Data Science, University of Melbourne ^bDepartment of Econometrics and Business Statistics, Monash University ^cDepartment of Data Science and AI, Monash University

Abstract

The decomposition of time series into components is an important task that helps to understand time series and can enable better forecasting. Nowadays, with high sampling rates leading to high-frequency data (such as daily, hourly, or minutely data), many real-world datasets contain time series data that can exhibit multiple seasonal patterns. Although several methods have been proposed to decompose time series better under these circumstances, they are often computationally inefficient or inaccurate. In this study, we propose Multiple Seasonal-Trend decomposition using Loess (MSTL), an extension to the traditional Seasonal-Trend decomposition using Loess (STL) procedure, allowing the decomposition of time series with multiple seasonal patterns. In our evaluation on synthetic and a perturbed real-world time series dataset, compared to other decomposition benchmarks, MSTL demonstrates competitive results with lower computational cost. The implementation of MSTL is available in the R package *forecast*.

 $\mathit{Keywords:}~$ Time Series Decomposition, Multiple Seasonality, MSTL, TBATS, STR

*Corresponding Author Name: Kasun Bandara, Affiliation: School of Computing and Information Systems, Melbourne Centre for Data Science, University of Melbourne, Melbourne, Australia, Postal Address: School of Computing and Information Systems, The University of Melbourne, Victoria 3052, Australia, E-mail address: Kasun.Bandara@unimelb.edu.au

Coming to Zabbix

Trigger functions

trendmp(/host/key,eval,detect,subseq)

Number of discords

trendmseason(/host/key,eval,detect)

Anomaly rate



Further plans and research

Include anomaly detection in standard templates

- MERLIN, VALMOD
- Multivariate anomaly detection
- Untie from trends
- Mark anomalies on graphs

References

- Schmidl, S., et al. "Anomaly Detection in Time Series: A Comprehensive Evaluation.", 2022
- Rewicki, F., et al. "Is it worth it? Comparing six deep and classical methods for unsupervised anomaly detection in time series", 2024
- Braei, M., et al. "Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art", 2020
- Wu, R., et al. "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress", 2020
- M. Saquib Sarfraz, et al. "Position: Quo Vadis, Unsupervised Time Series Anomaly Detection?", 2024
- Keogh, E., et al. "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets", 2016
- ▶ Taylor, S.J., Letham, B. "Forecasting at Scale", 2017
- Bandara, K. "MSTL: A Seasonal-Trend Decomposition Algorithm for Time Series with Multiple Seasonal Patterns", 2021
- Hoang, D., et al. "The UCR Time Series Classification Archive", <u>https://www.cs.ucr.edu/~eamonn/time_series_data_2018/</u>
- Numenta, "Numenta Anomaly Benchmark", <u>https://github.com/numenta/NAB</u>



Thank you!

