CCA: An ML Pipeline for Cloud Anomaly Troubleshooting

Lili Georgieva, Ioana Giurgiu, Serge Monney, Haris Pozidis, Viviane Potocnik, Mitch Gusat†

IBM Zurich Research Laboratory
Saumerstrasse 4, CH-8803 Rueschlikon/Switzerland
{lig, igi, smo, hap, viv, mig}@zurich.ibm.com

Abstract

The *Cloud Causality Analyzer (CCA)* is an ML-based analytical pipeline to automate the tedious process of *Root Cause Analysis (RCA)* of Cloud IT events. The 3-stage pipeline is composed of 9 functional modules, including dimensionality reduction (feature engineering, selection and compression), embedded anomaly detection, and an ensemble of 3 custom explainability and causality models for Cloud Key Performance Indicators (KPI). Our challenge is: **How to apply a reduced (sub)set of judiciously selected KPIs to detect Cloud performance anomalies, *and* their respective root causal culprits, all *without* compromising accuracy?**

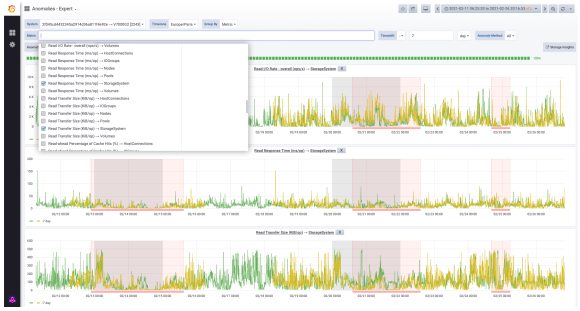

Figure 1: KPIs Dashboard GUI (anomalies in red).

Introduction

Anomaly Detection (AD)—the identification of novel or abnormal events—is ever more essential. Our challenge is to design explainable AD methods, capable to drill via causal inference from symptoms to root causes, and thus automate the deeply involved *Root Cause Analysis* of Cloud events.

We propose the *Cloud Causality Analyzer (CCA)*—a fully unsupervised pipeline that works on a compressed and channelized stream of multivariate timeseries – *Key Performance Indicators (KPIs)* from IBM's Cloud storage environments (Fig. 1). Besides KPI feature engineering and compression, CCA detects outliers, infers explanations, learns causal graphs and enables RCA-based anomaly troubleshooting. For scalability we introduce an upstream *dimensionality reduction* Front-end module that selects a compressed set of *representative* features, which are further ingested in a *channelized Temporal Convolutional Network (TCN)* AD model, a.k.a. Mid-end. Finally, CCA's Back-end uses SHAP explainability (Lundberg and Lee 2017) and two complementary causality models to identify the culprits for the detected anomalies – all ensembled in a hierarchical manner (Fig. 2).

Application: CCA is beta-tested by Cloud subject matter experts (SMEs) to troubleshoot anomalies on 250–$1K$+ KPIs across 10K+ large-scale storage systems. Following an alert for deteriorating performance, e.g., longer response times, CCA's Front-end presents to the SME two sets ("channels") of the most *Representative* KPIs: grouped as "*normal*" or "*abnormal*", based on their deviation from the typically correlated metrics during the analyzed window (one week). The 'channelized' Rep-KPIs are then ingested in CCA's Mid-end anomaly detection module to identify when the range anomaly occurred and which KPIs show the top anomalous symptoms. Thereafter, the Back-end investigates the root causal chains that explain which metrics caused the anomaly and 'infected' other KPIs. This in turn enables the support engineer to effectively troubleshoot the anomalous event based on the culprit KPIs.

Cloud Causality Analyzer

Front-end: Multi-channel DimRedux The *Cloud Causality Analyzer* performs multi-channel dimensionality reduction for multivariate KPI-based timeseries via *k-shape clustering* (Paparrizos and Gravano 2016), feature selection, channel population and inter-channel weighting.

(1) k-shape Clustering of KPIs creates k well-separated homogeneous clusters through a robust iterative refinement algorithm, scaling linearly with the N weekly features (Paparrizos and Gravano 2016). We validate the clusters' *cohesion* and *separation* by *silhouette* and *gap* scores, and interpret them using *t-SNE* (van der Maaten and Hinton 2008) and *UMAP* (McInnes, Healy, and Melville 2018) plots.

(2) Rep–KPI Selection. We characterize each cluster with a fixed number of *representative KPIs* (*R-KPIs*)—the features that best describe the KPI patterns in the respective cluster (Thalheim et al. 2017). However, for AD, we distinguish two sets of R-KPIs: (i) *central*, nearest the centroid, to represent "*normality*"; (ii) *peripheral*, the farthest from the centroid, to capture the less frequent "*abnormalities*".

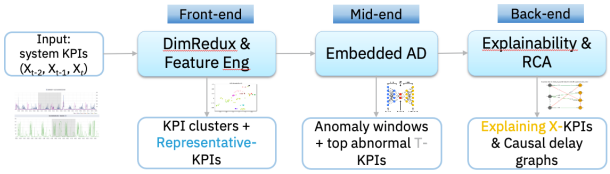

Figure 2: *Cloud Causality Analyzer*: Simplified pipeline.

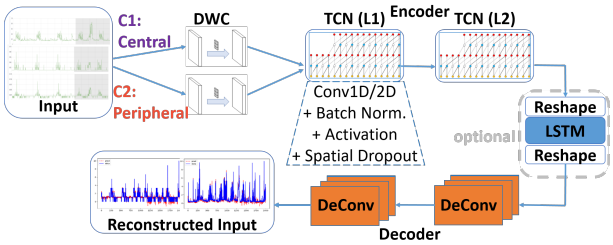

Figure 3: CCA Mid-end: Embedded-AD implemented as a TCN–based Autoencoder with 2-channel Attention.

(3) Population of *Central* and *Peripheral* Channels. The two KPI *channels* are intra-sorted (i) by their individual shape-based distance scores relative to the centroids, and then (ii) frequency-based aggregated across one month.

Mid-end: Embedded Compressed AD CCA's FE feeds the *compressed* and *ranked* Representative *central* and *peripheral* KPIs into a custom *embedded AD (e-AD)*. We assume that the majority of the data is 'normal', i.e., will be well-learned and reconstructed by a lossy Autoencoder (AE) model, whereas the novelties (not necessarily abnormal) will be detected as having high reconstruction errors (residuals).

(1) TCN-based AE with 2-channels, up/down-weighted, or inter-sorted by *'Attention'-lite* heuristics, comprising two separable *Depth-wise Convolutional* (*DWC*) layers. We initialize the DWC weights in the range $[0, 1]$, which after training we use as attention scores. The *temporal convolutions* provide three key benefits: (i) a flexible Receptive Field that is sensitive to feature ordering *and* size, (ii) causal modelling, preventing any future-past leakage, and (iii) dilated convolutions and residual modules for a deeper history (vs. our prior LSTM-based models).

(2) Residual Error –based AD post-processor is used to identify the *anomaly windows* relative to k-standard deviations. We output only sustained anomalies of multiple time steps; thus we discard the punctual outliers and short-lived bursts ($\leq$ 15-min). The *Top-k contributors* (*T-KPIs*) are extracted based on their autoencoder reconstruction error.

The *embedded-Anomaly Detector* is validated against ground-truth estimates from human SME labels and a reference AD pipeline. Overall, we achieve ca. 67% $(50 - 88\%)$ accuracy with 15x less KPIs vs. the reference.

Back-end: AD Explainability & RCA CCA entails a *hierarchical ensemble* for Cloud event-troubleshooting that discovers: (i) SHAP-based punctual explanations via *SHAP-XAD*, and (ii) Granger- (Granger 1969) and TCDF-based (Nauta, Bucur, and Seifert 2019) causal graphs, as follows.

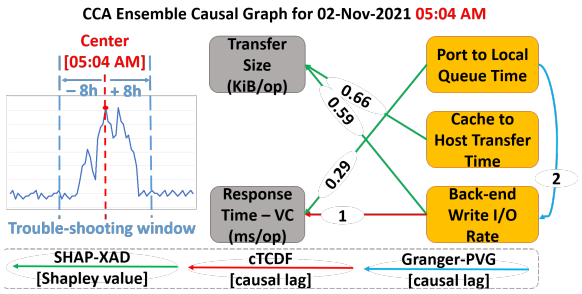

Figure 4: CCA Back-end: Ensemble-based RCA troubleshooting of a Cloud Storage performance anomaly within a +/-8hr window (left), and a causality graph (right).

(1) SHAP-based eXplainable AD (SHAP-XAD) is CCA's punctual explainability that applies *Kernel-SHAP* to derive the local feature importance scores for each *e-AD* prediction; this generates the eXplanatory *KPIs* per each anomalous *Top-KPI* (Fig.4).

(2) Metric-dependency Extractor based on the Front-end's Central and Peripheral Representative KPIs, as below.

(2a) Granger-PVG tests for Granger causality across targeted time periods, reflecting short delay lags ($\leq$ 10 time steps) in the causality chains. We build *Permutation-Validation of Granger (PVG)* to validate these causalities by discarding: (i) relationships that persist after random-shuffling of the "cause", and (ii) bidirectional causations.

(2b) cTCDF. In contrast to the pairwise *Granger*, *TCDF* (Nauta, Bucur, and Seifert 2019) is a NxN-parallel model, similar to *Neural Granger* (Tank et al. 2021), for discovering causal relationships (incl. instantaneous) with delays. These are validated against confounders via a permutation-based method (*PIVM*). In CCA's *compressed-TCDF (cTCDF)*, we leverage the mid-end TCN's *kernel size, number of hidden layers* and *dilation coefficients* to re-tune the Receptive Field. We ingest the compressed Rep-KPIs to obtain validated, causal graphs with lags (0-16 timesteps).

(3) RCA Ensemble troubleshooting example: (i) a 16-hour *troubleshooting window* is centered at each top e-AD anomalous point *and* is explained by *SHAP-XAD*, (ii) in this window we perform the causal exploration via both *Granger-PVG* & *cTCDF*, (iii) to build the ensemble graph, as shown in Fig. 4. The associated video illustrates CCA's operation and the steps taken by a user, typically a Cloud SME, to troubleshoot detected storage systems' failures.

Conclusion & Next Steps

CCA proposes a novel RCA automation solution for Cloud KPI features selection, incl. their reduction from 100s to 10s of multivariate timeseries, anomaly detection and discovery of the potential causal culprits. CCA brings distributed robustness in a 3–stage pipeline and is used in conjunction with a data-lake-based Dashboard to detect and troubleshoot Cloud events. We are working on fault-localization in the systems' topology, pruning the ensemble graphs, incorporating the *k-shape* clustering into the ensemble logic and analyzing causal chains' dynamics during real-life failures.

Acknowledgments

We are grateful to Roman Pletka, Slavisa Serafijanovic, Artur Dox, Jonas Pfefferle and Martin Petermann from IBM Research for their contributions and fruitful discussions.

References

Granger, C. W. J. 1969. Investigating Causal Relations by Econometric Models and Cross-Spectral Methods. *Econometrica*, 37(3): 424–438.

Lundberg, S.; and Lee, S. 2017. A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874.

McInnes, L.; Healy, J.; and Melville, J. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. http://arxiv.org/abs/1802.03426. Accessed: 2021-11-19.

Nauta, M.; Bucur, D.; and Seifert, C. 2019. Causal Discovery with Attention-Based Convolutional Neural Networks. *Machine Learning and Knowledge Extraction*, 1(1): 312–340.

Paparrizos, J.; and Gravano, L. 2016. K-Shape: Efficient and Accurate Clustering of Time Series. *SIGMOD Rec.*, 45(1): 69–76.

Tank, A.; Covert, I.; Foti, N.; Shojaie, A.; and Fox, E. B. 2021. Neural Granger Causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Thalheim, J.; Rodrigues, A.; Akkus, I. E.; Bhatotia, P.; Chen, R.; Viswanath, B.; Jiao, L.; and Fetzer, C. 2017. Sieve: Actionable Insights from Monitored Metrics in Microservices. *CoRR*, abs/1709.06686.

van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9: 2579–2605.