UTRECHT UNIVERSITY

Department of Information and Computing Science

Applied Data Science master thesis

Causal CloudScape (CCS): A Novel Approach to Causal Discovery of High-Dimensional Low-Level Cloud Performance Metrics

First examiner:
Nishant Saurabh
(n.saurabh@uu.nl)

Candidate:
Tobias Zenner
(t.zenner@students.uu.nl)

Second examiner:
Siamak Farshidi
(s.farshidi@uu.nl)

Daily Supervisor:
Diogo Landau
(d.landau@uu.nl)

July 14, 2023

Abstract—**Traditional data center analysis is based on high-level, coarse-grained metrics. With modern workloads emerging such as Machine Learning, conventional analysis does not prove valid anymore. There is a need to create a better understanding of modern data center operations, through novel data-driven methods. Causal discovery is an active field of research, which aims at deriving causal relationships from data. Little research efforts have been done in developing such methods, due to the sparsity of holistic low-level metric cloud data sets, and the lack of scalable and effective causal discovery algorithms. In this paper, we present Causal CloudScape (CCS), a novel causal discovery framework, leveraging a recent and largely unexplored low-level cloud metric data set with over 60 billion measurements captured in 15-second intervals. We conduct a series of experiments through expert-guided model validation and conclude with answering formulated research questions during unsupervised deployments. With CCS, we identified potential root causes of performance anomalies and created a map to visualize and inspect causal relationships in complex systems such as cloud datacenters. These findings are expected to have significant potential for optimizing performance and planning in large-scale data centers.**

I. Introduction

In recent years, the prominence of cloud computing in our society has grown exponentially, bringing with it increased demand for energy and resource consumption. It is estimated that cloud data centers now account for around 1 percent of the world's energy consumption (Pesce, 2023), a figure that underscores the critical importance of optimizing performance in these large-scale facilities. Concurrently, with the introduction of trends such as Machine Learning (ML), modern cloud data centers are exhibiting large degrees of performance variability, making the task of performance optimization more complex and necessitating more sophisticated approaches to performance understanding. Key to performance understanding is the availability of large low-level metric data sets, next to effective and scalable statistical methods to uncover and map out causal relationships between cloud metrics.

However, there is a lack in both publicly available low-level cloud metric data sets and sophisticated causal statistical methods. Comprehensive data sets of low-level data center metrics are scarce because commercial providers are often reluctant to release such data sets due to concerns about commercial secrecy, privacy legislation, and the lack of sufficient incentives to compensate for the additional effort required. Common public datasets such as CloudLab [1] used by Maricq et al. (2018), are often collected over short periods with coarse time-granularity and do not include low-level machine metrics. Next to the lack of holistic low-level metric cloud data sets, little research efforts have been done in developing data-driven methods to understand complex relationships in cloud systems, which is largely due to high computational requirements and impractical runtimes of such algorithms. Figure 1 serves as an example here, in which Versluis et al. (2023), found that Cloud ML nodes, which rise in importance with the progression of ML applications, display a higher power consumption variability compared to Generic nodes. While this novel and rare data set is a first step towards bridging the gap of lacking relevant datasets, only correlation methods such as Pearson (Benesty et al., 2009), Kendalltau (Samara & Randles, 1988) and Spearman (Croux& Dehon, 2010) were applied. These methods help in uncovering linear and non-linear correlations between a metric pair, but correlation does not imply causation (Aslam, 2015). Thus, we cannot answer questions such as what are the causes of ML nodes exhibiting higher power consumption variability compared to Generic nodes. More broadly, such models fail at explaining root causes in large-scale systems with many complex relationships.

In our work, we aim to leverage a rare and rich low-level cloud metric data set with more than 60 billion measurements captured in 15-second time intervals, in order to develop the novel and data-driven causal discovery method, Causal CloudScape, or short CCS. Causal discovery is a process defined as identifying and deriving cause-and-effect relationships from data. The causal discovery approach presents a promising avenue for better understanding large and complex systems such as cloud data centers. By utilizing causal discovery, we can not only identify root causes such as power consumption variability across heterogeneous nodes, but also detect anomalies, inform AI models, and influence policy and planning, with the end goal of optimizing performance in large-scale cloud data centers. A major challenge of this approach are computational complexities, which are commonly exponential with the increase in number of features, in our case cloud metrics.

In this paper, we make the following contributions:

1) We explore ways to leverage holistic causal discovery to improve data center operations and propose the novel three-step method Causal CloudScape which runs in close to linear time-complexity.

2) We validate and evaluate Causal CloudScape in expert-guided and unsupervised settings.

3) We answer 5 research questions in the process of discovering causal relationships in a low-level cloud metric dataset and conclude with the hypothesis that high power consumption variability in ML nodes is due to GPU-related metrics, which are absent in Generic nodes.

We address the following research question (RQ1): **How can we develop a scalable method to discover new (causal) relationships in high-dimensional cloud metric datasets?** Further, we answer the following sub-questions:

- Are the time-dependencies between any metric pairs? (RQ1.1)
- Does the method produce stable results? (RQ1.2)
- What is the median runtime of CCS? (RQ1.3)
- Why are Generic nodes more stable in power consumption than ML nodes? (RQ1.4)
- Can we discover novel causal relationships? (RQ1.5)

To give the aforementioned contributions more context, we discuss related work and give a brief background on causal discovery in Section II. In Section III, we describe the dataset we have used, Section IV describes in detail our Causal CloudScape method. In Section V we report our experiments and tests in which we both validate our approach in an

[1]CloudLab is a dataset with 6.8M high-level irregular performance measurements such as execution times, collected on the CloudLab testbed. It was published in 2020. For more information on how and why it was collected, see https://zenodo.org/record/3686952/

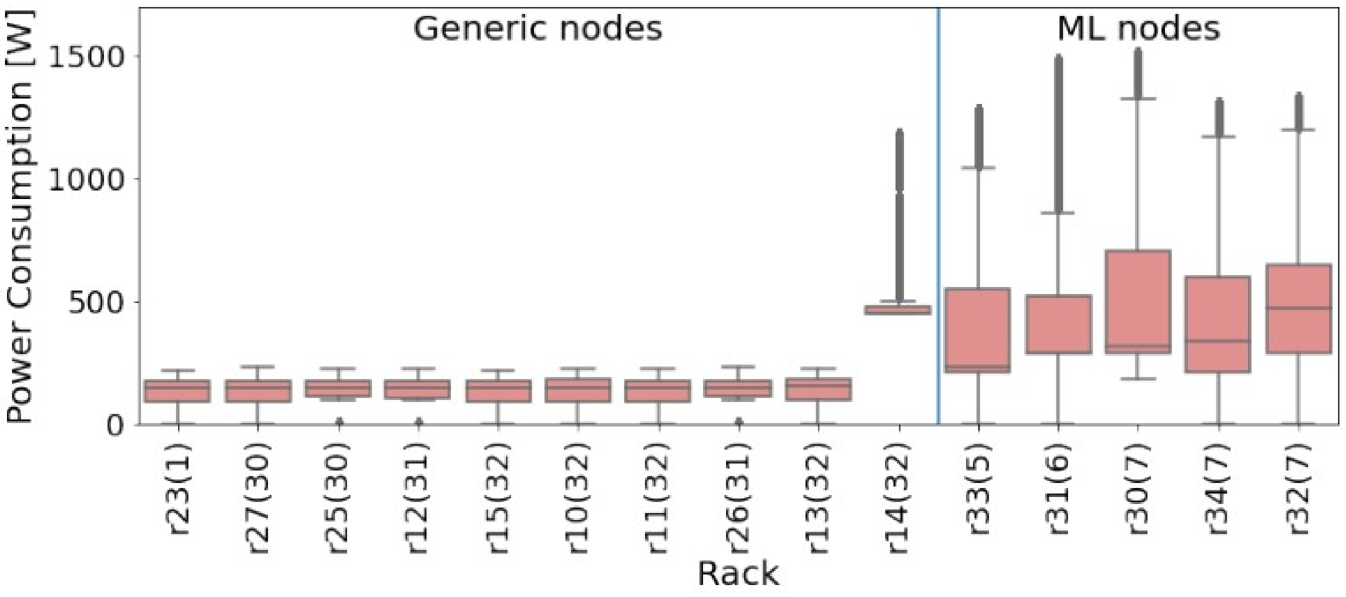

Fig. 1. Verlsuis et al. (2023) observed that ML/GPU nodes have higher power consumption variability than generic nodes.

expert-guided setting and discover novel causal relationships between low-level cloud performance metrics. We conclude with Section VI, in which we draw conclusions and discuss the limitations of our method next to providing directions for further work.

II. Background / Related Work

In this section, we present previous efforts on causal discovery within cloud computing on related cloud metric datasets and provide a brief outline of common causal discovery methods and their applicability in cloud computing.

A. Previous Efforts

Causal discovery in cloud computing presents significant challenges, evidenced by limited research efforts. The ”Causality Discovery as a Service” proposed by Wang et al. (2021) offers scalability on AWS and Azure, but focuses more on operational optimization than advancing core causal discovery methodologies. In another approach by Wang et al. (2021), Granger causality models are applied to detect dependencies between cloud application microservices. However, its efficacy is contingent on data properties, such as linearity and time-invariance, which are often not present in real-world cloud systems. Therefore, there is a pressing need for novel, robust, and adaptable causal discovery methods tailored for cloud computing, considering the limitations of existing research and the complexity of cloud systems.

B. Causal Discovery in Cloud Computing

Causal discovery can generally be divided into constraint-based or score-based methods. We will briefly explain both separately and discuss their applicability for causal discovery in cloud computing.

Constraint-based. Constraint-based causal discovery methods identify causal relationships by exploiting the conditional independencies in data, using algorithms such as PC (Duy Le et al., 2014), PCMCI+ (Runge, 2020), and FCI (Glymour et al., 2019), which iteratively removes edges in a fully connected graph based on statistical independence tests. Despite their theoretical appeal and established efficacy within limited dimensional settings, mentioned methods face considerable challenges when applied to high-dimensional data, such as the low-level cloud performance metrics that we focus on in this work. Each of these methods is characterized by an underlying complexity that increases exponentially with the number of variables. For instance, while efficient on low dimensions, within a subspace of just 20 variables, each of these algorithms exhibited runtimes on the order of days (tested on AWS EC2 t2.large instance with 8vCPU and 8GiB Memory), hence unscalable when expanded to even higher dimensions. Another limitation is decreased statistical power in high-dimensional settings, which can lead to inaccurate causal graphs due to a heightened risk of Type I and II errors.

Score-based. Score-based causal discovery methods estimate the likelihood of a causal structure by comparing the goodness-of-fit scores of different models, with the Bayesian Information Criterion often used to determine the best model. While promising approaches have been emerged recently, such as DAG with NOTEARs (Zheng et al., 2018), DAGMA (Bello et al., 2023), and CUTS+ (Zheng et al., 2023), their applicability on cloud computing datasets has, to the best of our knowledge, remained untested.

To summarize, constraint-based methods are due to the high dimensionality problem infeasible, and the application of score-based methods has remained largely untested. While previous work of very recent score-based algorithms has been tested across benchmarks such as climate models, it is uncertain that these will perform well on other problems (Glymour et al., 2018). In this paper, we take the approach of thoroughly testing various causal discovery methods in cloud specific datasets.

III. Dataset

In this section, we present the system model of LISA, the datacenter from which the data originates, and outline its operations.

A. Data Source and Infrastructure

The research data used in our study originates from the LISA system model [2], a compute cluster that forms part of the SURFsara [3] infrastructure. Managed by SURF, a collaborative Information and Communications Technology (ICT) organization for Dutch education and research, SURFsara operates as a national tier-1 data and computing facility offering a range of services. This extensive facility houses a heterogeneous infrastructure spread across 20 racks, comprising 349 nodes. These nodes vary in terms of processor types, clock speed, memory, sockets, cache, cores, accelerators, and interconnect. Racks are classified as generic (including only nodes with CPUs) or machine learning (ML) nodes with both CPUs and GPUs. The ML nodes are specially privileged and reserved primarily for ML workload. Additionally, the data center contains specific nodes for entry, administration, and compilation. These nodes are used for tasks such as library compilation and data processing and are not included in our analysis.

B. The LISA Dataset

The LISA dataset (SURF, 2021) stands out with its finest temporal and spatial granularity, compared to other open-sourced data center metric datasets. The data is captured at an interval of 15 seconds over a time period of nearly 8 months. A total of 327 diverse metrics were collected, providing up to 1.26 million samples per metric per node and culminating in 66 billion individual, high- and low-level metric measurements. Prometheus, complemented by libraries from Intel and NVIDIA was used to capture CPU metrics and GPU-related metrics respectively. These metrics include server-level (e.g., power consumption), hardware-sensor (e.g., fan speeds, temperature), and OS-level metrics (e.g., system load). Further, we grouped those metrics into six subgroups, as described in Table 1 [4]

[2]LISA is a compute cluster from SURF and primarily used for research at Dutch Universities. For more information, see: https://www.surf.nl/en/lisa-compute-cluster-extra-processing-power-for-research

[3]SURFsara is a Dutch High Performance Computing and e-Science support center that hosts large national infrastructure services and provides compute clusters such as LISA

[4]We used ChatGPT to automate the grouping of 327 metrics. For this, we extracted the subgroup of each metric, e.g. *netstat* from metric *node_netstat_lcmp_InMsgs* and used the following prompt: "For the list of following cloud datacenter low-level server-metrics, hardware-sensor and OS-level metrics, group these into 6 distinct categories. Assign labels to each of the 327 metrics. Following the list delimited by triple quotes: "'metric list + subgroup"' ". We experimented with the number of categories and sampled each to validate and concluded 6 the ideal number.

Despite concerted efforts and numerous outreachs to relevant parties, the job data was unfortunately not included in this research due to constraints and compliance with General Data Protection Regulation (GDPR) requirements, emphasizing the criticality of data privacy and protection in research contexts. However, incorporating this job data with over 800 users and more than 1 million submitted jobs presents a potential for enriching future research endeavors.

C. Data Cleanup

We followed the cleanup scripts from the dataset creators, resulting in a dataset covering the operation of 15 racks containing 315 nodes with nearly 64 billion measurements, spanning over 7 months. Additionally, we took steps to further clean and prepare the data. Firstly, to ensure data validity, we omitted any metric containing only one distinct value. The rationale behind this is that static data is unfit for the ranking step required to compute the Spearman and Kendall correlations (Croux, 2010), which we utilize in our analysis. Next, we plotted in Figure 3 the first and last timestamp of each metric, in order to easily identify any significant gaps in the dataset, that could potentially skew the research results. Versluis et al. (2023) reported, that for some metrics, the dataset contains gaps where the monitoring system was down, and for other metrics, data collection stopped halfway into May 2020. Further, as we found that Generic nodes exhibit more stable power consumption than ML/GPU nodes (Figure 1), we selected both Generic and ML nodes for our further analysis (Generic nodes: r10n20, r10n25; ML nodes: r30n4, r30n1). Selecting a second node by node type respectively helped ensure robustness and repeatability of the analysis methodology. When selecting these nodes, we also considered maximizing the number of relevant metrics, e.g. performance related metrics *node_load1* or *surfsara_power_consumption*. Additionally, we observed an anomalous event on 2020-01-03, when power usage dropped to zero and caused system-wide changes across a wide range of metrics. While this event could be a potential area for future research, it was excluded from the analysis in this study to maintain data consistency. Lastly, The remaining missing values (NaN) were eliminated, ensuring that only complete and valid data were included in the analysis.

IV. Causal CloudScape Method (CCS)

Given the small amount of research effort in creating scalable yet precise methods for causal discovery in low-level performance metrics of cloud data centers, we developed the novel method Causal CloudScape or short CCS. CCS is a scalable three-step causal discovery framework, designed to efficiently derive Directed Acyclic Graphs (DAGs) from high-dimensional cloud metric datasets. The major design challenge was runtime, thus we needed to create a rigorous pipeline to deal with high-dimensional time-series, that possibly contain various unknown characteristics, such as time-dependencies or trends.

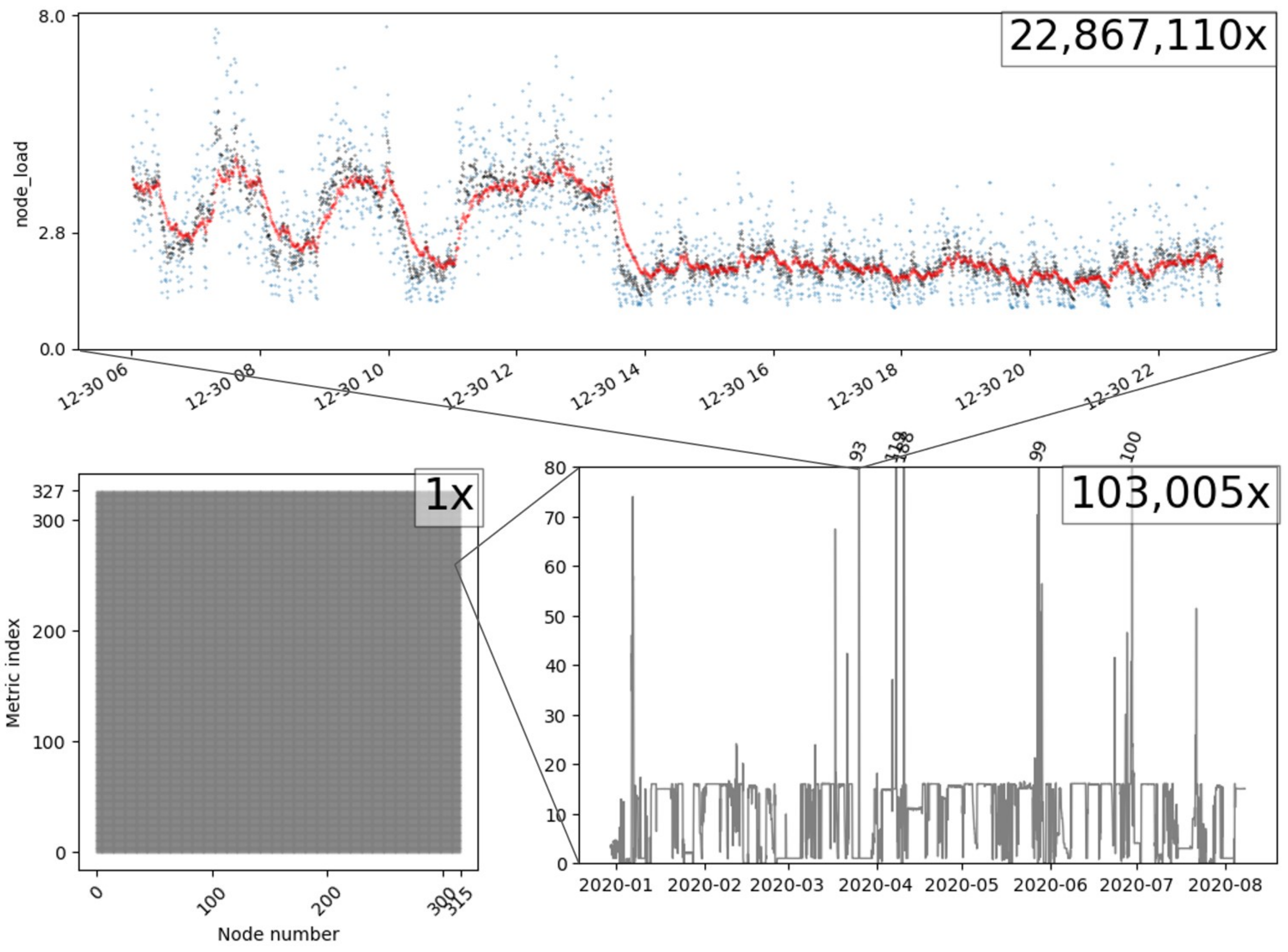

Fig. 2. LISA Dataset Overview, zoomed in. Each pixel in the first zoom (1x) represents one metric in one node. 103,005x Zoom shows metric node_load in r10n25 over 8 months. 22,867,110x Zoom shows node_load, averaged over 1min (back), 5min (blue), 15min (red), on a single day.

TABLE I
OVERVIEW METRICS GROUPED USING GPT4.

Metric Group	**Metric Count**	**e.g.**
Network	114	node_netstat_Icmp_InMsgs
System Performance	76	node_memory_Mapped
Disk and File System	44	node_disk_reads_completed
Process and Resource Management	42	node_context_switches
Misc	38	surfsara_ambient_temp
Time-related	13	node_load1

TABLE II
GENERIC OUTLINE OF THE LISA DATASET.

Dataset	**Item**	**Value**
Public Data	Start Date	2019-12-29
	End Date	2020-08-07
	Sampling frequency [s]	15
	Max. samples per metric per node	1,258,646
	Number of metrics	327
	Number of measurements	66,541,895,243
Clean data	Number of valid racks	15
	Number of valid nodes	315
	Number of valid measurements	63,978,689,791

A. *System Architecture*

Figure 4 shows the architecture of CCS, which consist of three components. Algorithm 1 describes CCS in pseudocode, and Table 3 explains hyperparameters. Following an overview and detailed explanation by steps.

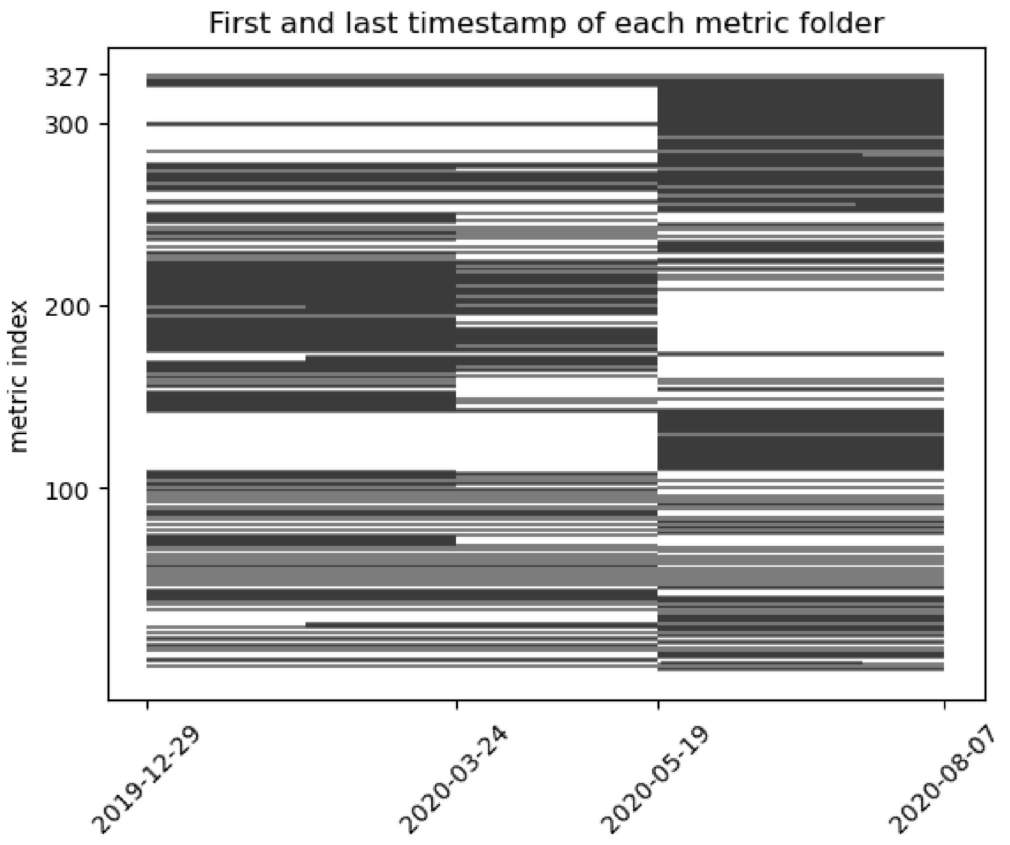

Fig. 3. Plotting the first and last timestamp of each metric helped guide sample selection. Most metrics relevant to system performance were recorded between 2019-12-29 and 2020-03-24.

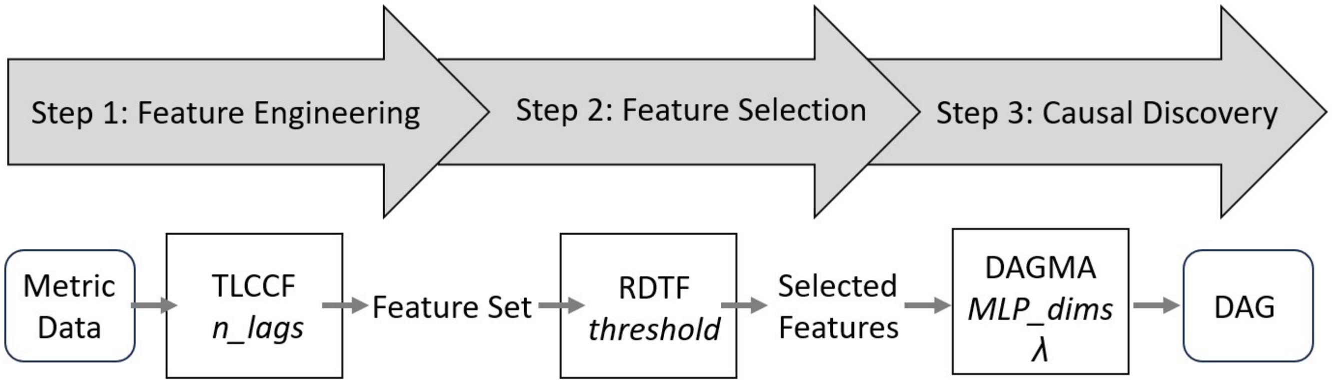

Fig. 4. System Architecture CCS. Metric Data as input and DAG as output.

1) Feature Engineering: A time-lagged cross-correlation function (TLCCF) is used to search linear and non-linear feature space of temporal differences between cloud metrics.
2) Feature Selection: A Recursive Decision Tree Function (RDTF) is used to select enhanced metric features (i.e. differenced lags) according to the hyperparameter threshold.
3) Causal Discovery: Directed Acyclic Graph Discovery via M-matrices (DAGMA) is used to create DAGs.

1. Step - Feature Engineering: The first step in our approach involves using a time-lagged cross-correlation function (TLCCF) to explore both linear and non-linear feature spaces of temporal differences between cloud metrics. We used three correlation methods - Pearson (Benesty et al., 2009), Spearman (Myers, 2006), and Kendall's tau (Samara & Randles, 1988) - to capture the potential linear and non-linear correlations among the metrics. The TLCCF is parameterized by $nlags$, a hyperparameter that represents the number of time steps in each direction that a metric pair is shifted. This allows us to

TABLE III
CAUSAL CLOUDSCAPE HYPERPARAMETERS.

CCS Step	Hyperparameter	Definition
1. Feature Engineering	n_lags	Number of time-steps.
	p	Significance value of correlation coefficient. Default to 0.05
2. Feature Selection	threshold	Numerical value x indicating head(x) of feature importance per function.
3. Causal Discovery	MLP_dims	Dimension of Multi-layer perceptron.
	λ	Learning rate of gradient descent function.

Algorithm 1 CausalCloudScape (metric data)

```
STEP 1: TLCCF(n_lags)
for lag from $-n\_lags$ to $+n\_lags$ do:
    calculate pearson, kendalltau, spearmanr for all metric pairs
    return max(r) if $p \geq 0.05$
STEP 2: RDTF(threshold)
for metric in metric data do:
    run LGBM(metric)
    return feature_importances filtered by threshold
STEP 3: DAGMA(MLP_dims,$\lambda$)
run DAGMA
return DAG as $g_i$
```

capture the dynamic dependencies between the metrics that could change over different points in time. The correlations calculated are then subjected to significance testing. Only those with p-values less than or equal to the set threshold of 0.05 are considered significant, thus ensuring the statistical robustness of our analysis and filtering out any spurious correlations. The function takes as input the cleaned metric dataset, and returns after processing the significant maximum correlation coefficient by metric pair.

2. Step - Feature Selection: Following TLCCF output, we employ a Recursive Decision Tree Function (RDTF), specifically utilizing the LightGBM framework (Ke, 2017), to dissect the intricate relationships among the various metrics. Decision trees are advantageous for this purpose because they create binary splits based on feature values, which offer an intuitive way to visually navigate and interpret complex data relationships (Figure 5). By recursively implementing this method, we are able to develop a hierarchy of decision nodes. Each node in this hierarchy provides additional insight into the dependencies and interactions among different metrics. This methodology aids in pinpointing the most influential metrics in a given context and also brings to light any unexpected metric interactions. To enhance the process, we deployed LightGBM with a specific hyperparameter, $threshold$. This hyperparameter selection allows us to filter out and retain only the most significant features for each metric, thereby reducing the input feature space for the next step.

3. Step - Causal Discovery: To create DAGs between generated features, we deploy the novel causal discovery method Directed Acyclic Graph via M-Matrices (DAGMA). DAGMA, a novel causal discovery method (Bello et al., 2023), introduces an acyclicity characterization based on the log-determinant function, designed to identify directed acyclic graphs (DAGs). Leveraging the nilpotency property of DAGs and the concept of M-matrices, the proposed log-det function outperforms existing characterizations by better detecting large cycles, offering more manageable gradients, and running much faster in practice. DAGMA uses the log-det function as a regularizer, presenting an optimization scheme that results in solutions guaranteed to be DAGs. The approach demonstrates considerable speed-ups and smaller structural Hamming distances compared to state-of-the-art methods. An alternation of the implementation of Bello et al. (2023) was followed. Additionally, the DAGMA function was enhanced by hyperparameters $MLPdims$ and $lambda$. The former determines the shape of the Multilayer perceptron, we set its standard value to [d, 10, 1], where d is the dimension of the input matrix, aka the number of final features of the previously cleaned and processed cloud metric features. The latter is the learning rate with standard values set to 0.02.

B. CCS Evaluation Metric

To evaluate the performance of our CCS, we introduce the metric $Stability$. This quantifies the consistency between two directed graphs, $g1$, and $g2$, derived from the same data but through different iterations. Essentially, the metric measures the fraction of identical edges in both graphs. A higher value indicates that the CCS method has managed to generate more stable and consistent graph structures across different iterations.

The metric is defined as follows:

$$Stability(g1, g2) = \frac{\sum_{i,j}[A_{g1_{i,j}} == A_{g2_{i,j}}]}{\sum_{i,j}[A_{g1_{i,j}}]} \tag{1}$$

where $Ag1$ and $Ag2$ represent the adjacency matrices of the graphs $g1$ and $g2$ respectively and (i, j) are the pairs of vertices.

Beyond the "Stability" metric designed for CCS evaluation, we examined other popular metrics such as Structural Hamming Distance (SHD), Structural Intervention Distance (SID), and the number of links. However, each of these presented limitations. SHD and SID (Cheng et al., 2022) assume knowledge of the ground truth graph, which is not available in our case of exploratory causal discovery in cloud performance metrics. The "number of links" metric (Marcot, 2012) measures the complexity of a model but does not reflect the consistency of structures generated in different iterations. While these metrics presented limitations for our specific cloud performance metrics problem, they could potentially be leveraged in further research addressing other problems or datasets.

C. Additional Methods Explored

In our quest for a viable causal discovery method suitable for high-dimensional data, we developed and evaluated additional strategies that combined existing methods or leveraged novel techniques. However, these did not produce stability scores as high as those attained with our CCS approach.

1. Louvain-community clustering + PCMCI+: We hypothesized that partitioning the metrics into clusters using the Louvain method (Blondel et al., 2008) could create manageable subspaces for PCMCI+. While this approach alleviated some of the computational burden, it could not fully overcome the high-dimensionality challenge. Louvain community clusters were often found to be up to the size of 50 metrics. More importantly, the stability scores from this strategy were less than satisfactory, especially when reducing cluster sizes, indicating a lower level of confidence in the derived causal relationships.

2. DAG with NO TEARS: Another explored strategy was the implementation of the DAG with NO TEARs algorithm (Zheng, 2018). This approach, unlike the constraint-based methods, attempts to learn the structure of a DAG directly, avoiding the expensive combinatorial search of the graph space. We used the open-source repository CausalNex 0.12.1 by QuantumBlack (2023). However, the input data was required to be discretized. While valid results were yielded, applying discretization techniques was time-intensive and only yielded results when systematically discretized with expert-informed discretization techniques such as equal-width binning based on information gain, or Machine-Learning based discretization techniques. However, we contend that the reliance on such expert-guided discretization significantly diminishes the method's generalizability. Both time-consuming and inefficient, this approach lacks scalability, a critical deficiency in swiftly changing domains like cloud performance monitoring.

3. CUTS+: CUTS+ (Cheng et al., 2023) is a novel algorithm that improves causal discovery in high-dimensional, irregular time-series data by employing a Coarse-to-Fine-Discovery approach to break down complex time-series into manageable groups and a Message-Passing Graph Neural Network for enhanced predictions. Given its specificity for irregular time series, which our dataset lacks, and the minimal temporally relevant features within our dataset, its application was deemed unsuitable. Yet, its potential utility in future research, particularly in datasets with irregular time series or substantial temporal dependencies, remains noteworthy.

D. Software Reproducibility

In order to ensure reproducibility, we load the raw data from the SURF Machine Metric data set, which can be accessed as open-access data at https://doi.org/10.5281/zenodo.4459519. Furthermore, all of our analysis and plotting code is openly available on our GitHub repository at https://github.com/EC-labs/ADS-tobias.

V. Experimential Results

As part of developing CCS, we conducted a series of iterative tests on an expert-guided sub-selection of the dataset related to disk-swapping events, where causal relationships were known. Further, we deployed the method in an unsupervised holistic metric selection over an extended period of time and answered RQ1.1-RQ1.5.

A. Setup

Table 4 provides an overview of the experimental setups. The expert-guided tests were run locally due to the small sample size. The unsupervised causal discovery tests were computationally more expensive, thus were deployed using AWS EC2 services.

B. Expert-Guided Disk Swapping

We focus on our expert-guided validation of disk-swapping events. Disk swaps occur when free memory is low, which leads to page swaps and context switches. We selected three distinct disk swap events and created sub-datasets of node r10n25 of sampling sizes $n1 = 248$, $n2 = 70$, and $n3 = 70$, including both first a metric set we assume to have strong causal relationships and second a metric set we assume to have an absence of causal interplay, to account for type I and type II errors. First, the metric set with strong causal relationships.

- *node_memory_MemFree*: The amount of free memory available.
- *node_memory_pswpin*: data transferred from disk to memory per second in kilobytes.
- *node_context_switches*: number of the system's context switches.
- *node_load1*: average system load over the last minute.

Second, the metric set with weak causal relationships.

- *node_forks*: number of forks.
- *node_netstat_TCPHPacks*: number of Transmission Control Protocol packets sent or received. Following the CCS framework.

Step 1 - Feature Engineering: Next to time-lagged features, we generated additional differenced and smoothed features, such as *node_load1_diff* or *node_load1_smoothed*. We chose differencing as a feature to mitigate non-stationary properties of the metric time series, meaning to stabilize the mean of a time series by removing changes in the level, eliminating trend and seasonality, and therefore helping to make the series stationary. The latter smoothing feature was generated subsequently, as we observed frequent high fluctuations in differenced data, thus smoothing helped to mitigate noise.

Step 2 - Feature Selection: We applied RDTF on the resulting set of features including original, normalized metrics and generated features of the previous step. With *threshold = 3*, we yielded the most important features of differenced and smoothed metrics with a smoothing factor of 2, meaning averaging the data over 30 seconds, for the metrics *node_context_switches*, *node_memory_pswpin*, *node_forks*, and *node_netstat_TCPHPacks*. All these were aggregated data, thus non-stationary. The remaining set of metrics were used in their original form, as stationary.

Step 3 - Causal Discovery: Using the entire sample size of the three disk-swapping events, including datapoints before

TABLE IV
EXPERIMENTAL SETUP.

Experiment	Input Data Shape	Remote [Y/N]	AWS EC2 Instance type
B. Expert-Guided	(248, 6)	N	-
	(70, 6)	N	-
	(70, 6)	N	-
C. Unsupervised	(10000, 103)	Y	c5.xlarge (4vCPU, 8GiB Memory)
	(10000, 117)	Y	c5.2xlarge (8vCPU, 16GiB Memory)
	(10000, 117)	Y	t2.large (2vCPU, 8GiB Memory)

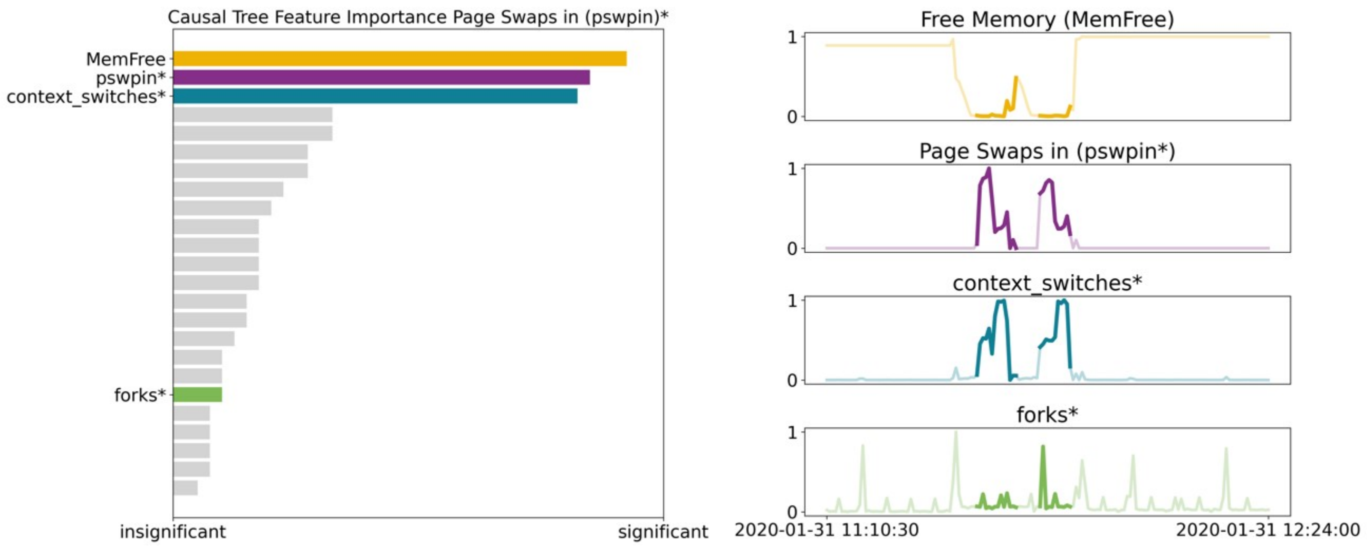

Fig. 5. Left figure visualizes the CCS feature selection through Step 2: RDTF plot shows feature importances for metric *node_memory_pswpin*. Right figure shows a normalized subselection of metrics. Metrics marked with * are differenced and smoothed.

and after, we used DAGMA to generate DAGs. We noticed that for small sample sizes, undirected DAGs produced better results than directed DAGs. Thus we transformed the output into an undirected adjacency matrix. Figure 6 reports the results of all three tests.

The strongest relationship is intuitively between *node_context_switches* and *node_memory_pswpin*, as data is transferred from memory to disk, triggering context switches, and establishing a consistent causal relationship. Similarly, the consistent relationships among all three tests of node pairs (*node_memory_pswpin*, *node_memory_pswpin*), and (*node_context_switches*, *node_memory_MemFree*) were anticipated. As disk swapping (pswpin) increases, it usually means that the system is freeing up memory (MemFree), so there is a direct statistical relation. Context switches often occur when the system is low on free memory, leading to a consistent relationship with metric *node_memory_MemFree*. Finally, the relationship between (*node_memory_MemFree*, *node_load1*) is also consistent between all tests. The rationale here is that the average system load can increase when memory is low (MemFree). It's the system's reaction to cope with the high demand for memory resources, hence the relationship.

C. Unsupervised Causal Discovery

In a second series of tests, we deployed the CCS method in an unsupervised holistic metric selection over an extended period of time and responded to RQ1.1-RQ1.5.

RQ1.1: Cross-correlation coefficients rarely increase between two lagged metrics.

RQ1.2: CCS yielded 95.5% Stability across 5 tests in ML node r30n4 and 96.1% Stability across 3 tests in generic node r10n25.

RQ1.3: CCS has a median runtime of 19.04 hours running 2 tests simultaneously on c5.2xlarge with 8vCPU and 16GiB RAM.

RQ1.4: Power consumption in ML nodes have GPU-related metrics as causal factor, which is absent in Generic nodes.

RQ1.5: There is a causal relationship from *surf-sara_power_usage* to *0:GeForce GTX 1080 Ti.3*.

To discover new causal relationships and test the model in an unsupervised way, we deployed CCS on two datasets from Generic node r10n25 and ML node r30n4. Again, following the CCS framework. Causal relations are written using the following notations: *Causal Factor*$\rightarrow$*Effect*.

Step 1 – TLCCF: To obtain a holistic view of temporal patterns across metric pairs, we plotted the results of applying TLCCF in Figure 7. We found that despite a high number of significant correlation pairs according to p-values between shifted time series, the difference was never higher than 0.03 and mostly smaller than 0.005, compared to the baseline correlation coefficient at lag 0. We observe and conclude that there are no major temporal influential factors in the LISA data set (**RQ1.1**). This observation makes sense because

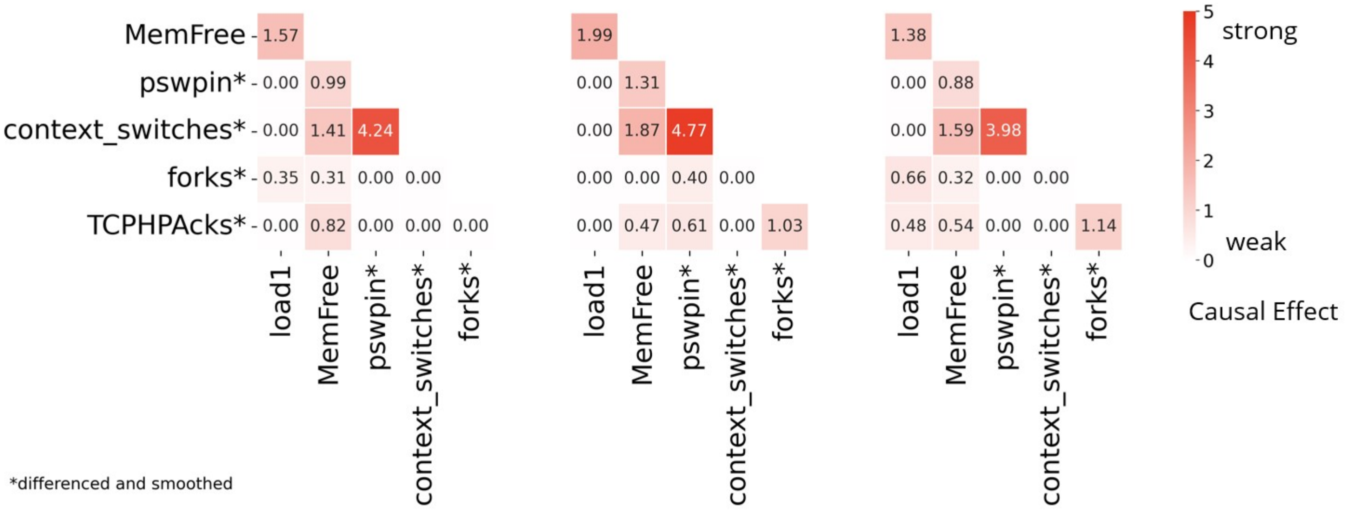

Fig. 6. Undirected DAG as adjacency metric across three distinct disk swapping events. From left to right, the sample input size is $n1 = 248$, $n2 = 70$, and $n3 = 70$ data points. Due to the small sample size of the input data, an undirected adjacency matrix was chosen to display the results.

the time granularity of recorded metrics is 15 seconds, and computational processes often happen sub-sequentially in time spans of milliseconds.

Step 2 – RDTF: We yielded 103 relevant features for Generic node r10n25 and 117 features for ML node r30n4.

Step 3 – DAGMA: We conducted 5 tests on the ML node and 3 tests on the Generic node. For efficiency and robustness, we used random sampling with a relatively large sample size of $n = 10,000$, which we found a good sample size considering the runtime and quality of results in terms of stability. On ML node tests, we recorded 95.5% stability across 5 tests, and 96.1% across 3 tests in Generic node tests (**RQ1.2**). We tested several AWS EC2 instance types for runtime and cost optimization and recorded the following runtimes as in Table 5. The fastest runtimes were recorded on c5.2xlarge (8vCPU, 16GiB Memory), with median of 19:04:54 hours (**RQ1.3**), running 2 tests parallel to optimize costs. The tests had a steady CPU utilization of 397% and 396% respectively.

Next, we plotted the resulting DAGs (Figure 9-13). We found many obviously logical causal relationships, such as *node_load1* $\rightarrow$ *node_load5*, *node_load1* $\rightarrow$ *node_load15* and *node_load5* $\rightarrow$ *node_load15*. *node_load5* and*node_load15* are derived metrics from *node_load1*, as these present the average system load of the past 5 and 15 minutes, respectively. Also, we found other logical performance related causal relationships, such as *node_disk_read_time_ms_diff* $\rightarrow$ *node_disk_reads_completed_diff* and *node_disk_write_time_ms_diff* $\rightarrow$ *node_disk_writes _completed_diff*. These metrics represent the differenced disk read/write times in milliseconds and the number of completed reads/writes, which is logical, the faster the read/write times, the higher the completed reads/writes. The differenced metrics were chosen by the RDTF function (CCS step 2), as these are aggregated and non-stationary processes. Besides obvious relationships, we also found novel, consistent relationships. First, we observed across ML-tests how GPU-related metrics such as *1:GeForce GTX 1080 Ti.3* are causing surfsara_power_usage. As ML nodes uniquely make usage of GPU for optimized computation, these relationships were only present in the ML tests, and absent in Generic tests (**RQ1.4**). This interesting observation leads to the hypothesis that high power consumption variability in ML nodes (Versluis et al., 2023) is caused by GPU devices. To test this hypothesis further, it would be interesting to compare these findings with other nodes and datasets. Interestingly, the causal relationship surfsara_power_usage $\rightarrow$ 0:GeForce GTX 1080 Ti.3 was observed across all tests. While this direction of the causal relationship seemed as a false positive initially, we found that on several occasions, the cooling system is a limiting factor for the performance of ML nodes, only handling up to 5.5 kW per rack, as observed by Versluis et al. (2023) after inquiring the datacenter operators. Thus, these directed relationships could be indicating thermal throttling (**RQ1.5**).

VI. Discussion and Conclusions

We outlined the importance of developing novel methods to create a better understanding of complex systems such as cloud data centers, through causal discovery. We provided an overview of the scarce, current development in this field and proposed Causal CloudScape (CCS), a novel method to discover causal relationships in high-dimensional cloud metric datasets. We applied CCS on the high-dimensional and novel low-level cloud metric dataset from the LISA system. We validated CCS through deployment and evaluation in an expert-guided setting with known causal structures and applied the framework in two unsupervised holistic settings, both on Generic and ML nodes. Through answering five different research questions on the efficacy of CCS, we showed the framework's potential in analyzing root causes, identifying performance anomalies, and understanding causal relationships in low-level metrics of cloud data centers. By parameterizing CCS, we provide a scalable and efficient framework,

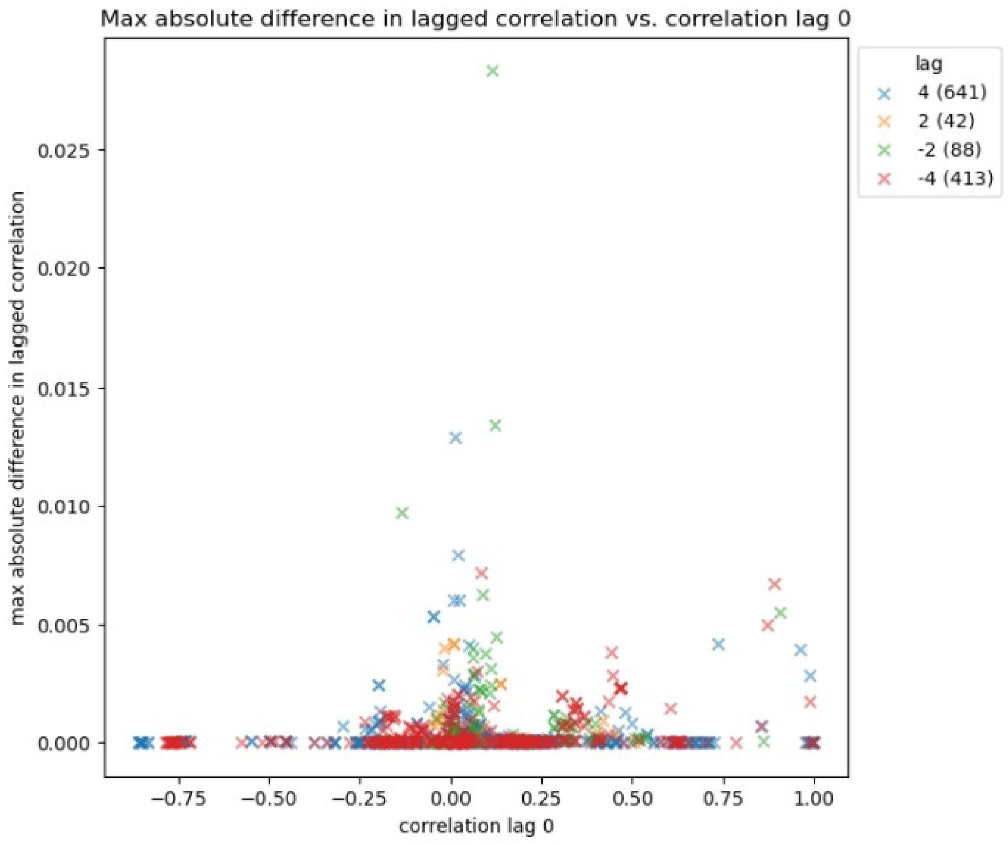

Fig. 7. Results of TLCFF averaged of nodes r10n25, r10n20, r30n4 and r30n1, validated across 50 day periods. Each datapoint shows the correlation coefficient at lag 0, plotted against the highest difference in lagged correlation between two metric pairs.

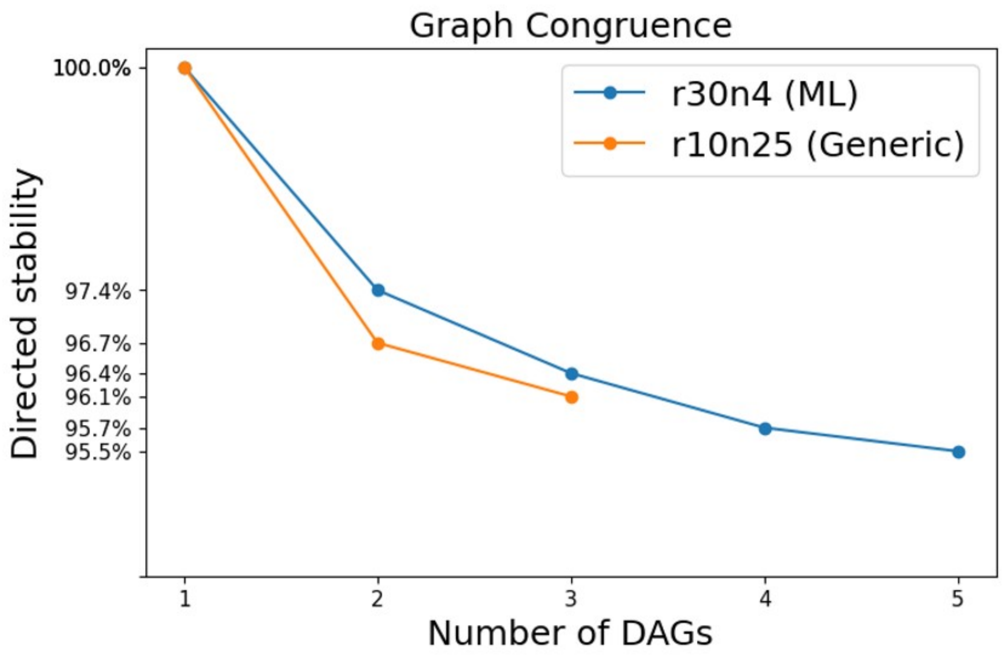

Fig. 8. Stability metric by subsequent testing. Each iteration within a given node contains distinct data points, sampled from a 50-day time period, with sample size $n = 10000$

TABLE V
CCS Execution times. Runtime logs in format [min, median, max] hh:mm:ss

Node test	Input Data Shape	AWS EC2 instance type	Runtime logs
r10n25 (3)	(10000, 103)	c5.xlarge (4vCPU, 8GiB Memory)	[52:57:00, 52:58:55, 52:59:36]
r30n4 (2)(2)	(10000, 117)	c5.2xlarge (8vCPU, 16GiB Memory)	[18:05:47, 19:04:54, 19:13:15]
r30n4 (1)	(10000, 117)	t2.large (2vCPU, 8GiB Memory)	[58:30:56]

which can potentially be generalized beyond cloud domain-specific use cases.

A. Limitations

While the Causal CloudScape (CCS) method offers a novel approach to the causal discovery of high-dimensional low-level cloud performance metrics, there are three limitations that should be considered. First and foremost are implicit assumptions and model-specific limitations of the causal discovery process (CCS step 3). As part of the causal discovery process, we have to make implicit assumptions about our data. We assume that the causal structure is *acyclic*, meaning that there are no loops or circles in causal structures. Also, we assume data *sufficiency*, meaning that we have included all causal factors at play in our model. In practice, this assumption is mostly never fully fulfilled, as reality is messy. However, it is important to make this assumption explicit when interpreting results., especially since job data is expected to reveal new causal influences. On model limitations, DAGMA is a very recent and not widely studied method for deriving DAGs. While the developers conducted extensive testing and benchmarking across diverse datasets, it is not as widely adopted yet as traditional methods, such as PC or GES. Thus keeping this in mind, we envision more extensive testing and development of novel models. The second limitation is that we used only limited feature selection techniques. While we were able to create a process (CCS steps 1 2) to effectively create and derive the most important input features, we also noticed how some metrics contain patterns that were not fully exploited yet, such as seasonality. Versluis et al. (2023) found a diurnal distribution of job arrivals, office / non-office hours. The third limitation is the one of DAG evaluation. Due to the nature of the problem, we cannot use metrics such as SHD or SID (Section IV, B.), as the ground truth of the causal structure is widely unknown. Thus, we have used the stability metric, which effectively shows the congruence amongst DAGs. As this metric only shows such stability, we envision that further efforts also include metrics such as a number of links and other metrics, depending on the nature of the research.

B. Future Work

We envision that further research applies CCS more thoroughly to the LISA dataset, which is still largely unexploited, such as by including job data in the analysis. Further, it would be interesting to compare CCS results with other holistic and rich datasets and also focus on holdout data such as performance anomalies or changepoints (Zhao et al., 2021).

Acknowledgment

We would like to thank Alexandru Uta for directing our attention on the LISA system and its mostly unexplored, yet novel and rich dataset.

References

[1] Aslam, A. (2015). Research ideas: Correlation does not imply causation. British Dental Journal, 219(2), 49. https://doi.org/10.1038/sj.bdj.2015.585

[2] Aleksander Maricq, Dmitry Duplyakin, Ivo Jimenez, Carlos Maltzahn, Ryan Stutsman, and Robert Ricci. 2018. Taming Performance Variability. In Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation (Carlsbad, CA, USA) (OSDI'18). USENIX Association, USA, 409–425.

[3] Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson Correlation Coefficient. In Springer eBooks (pp. 1–4). https://doi.org/10.1007/978-3-642-00296-0_5

[4] Bello, K., Aragam, B., & Ravikumar, P. (2023). DAGMA: Learning DAGs via M-matrices and a Log-Determinant Acyclicity Characterization. Carnegie. https://arxiv.org/pdf/2209.08037.pdf

[5] Blondel, V. D., Guillaume, J., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10), P10008. https://doi.org/10.1088/1742-5468/2008/10/p10008

[6] Croux, C., Dehon, C. (2010). Influence Functions of the Spearman and Kendall Correlation Measures. Journal of the Italian Statistical Society. https://doi.org/10.2139/ssrn.1585216

[7] Cheng, L., Guo, R., Moraffah, R., Sheth, P., Candan, K., Liu, H. (2022). Evaluation Methods and Measures for Causal Learning Algorithms. JOURNAL OF IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, 00(0). https://arxiv.org/pdf/2202.02896.pdf

[8] Cheng, Y. (2023, May 10). CUTS+: High-dimensional Causal Discovery from Irregular Time-series. arXiv.org. https://arxiv.org/abs/2305.05890

[9] Duy Le, T., Hoang, T., & Li, J. (2014). fast PC algorithm for high dimensional causal discovery with multi-core PCs. JOURNAL OF LATEX CLASS FILES, 13(9). https://arxiv.org/pdf/1502.02454.pdf

[10] Glymour, C., Zhang, K.,& Spirtes, P. (2019). Review of Causal Discovery Methods Based on Graphical Models. Frontiers in Genetics, 10. https://doi.org/10.3389/fgene.2019.00524

[11] IBM, & Wang, Q. (2021). Detecting Causal Structure on Cloud Application Microservices Using Granger Causality Models. IBM Research. https://kesyren.github.io/download/paper/2021IEEECLOUD.pdf

[12] Ke, G. (2017). LightGBM: a highly efficient gradient boosting decision tree. https://proceedings.neurips.cc/paper/2017/hash

[13] Marcot, B. G. (2012). Metrics for evaluating performance and uncertainty of Bayesian network models. Ecological Modelling, 230, 50–62. https://doi.org/10.1016/j.ecolmodel.2012.01.013

[14] Myers, L., & Sirois, M. (2006). Spearman Correlation Coefficients, Differences between. Encyclopedia of Statistical Sciences. https://doi.org/10.1002/0471667196.ess5050.pub2

[15] Pesce, M. (2023, March 29). Cloud Computing's Coming Energy Crisis. IEEE Spectrum. https://spectrum.ieee.org/cloud-computings-coming-energy-crisis

[16] Runge, J. (2020, August 27). Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. PMLR. https://proceedings.mlr.press/v124/runge20a.html

[17] Samara, B., & Randles, R. H. (1988). A test for correlation based on Kendall's tau. Communications in Statistics, 17(9), 3191–3205. https://doi.org/10.1080/03610928808829798

[18] Valur, L. O. K. (2021). SURF Machine Metric Dataset [2019-12-29, 2020-08-07]. Zenodo. https://doi.org/10.5281/zenodo.4459519

[19] Versluis, L., Cetin, M., Greeven, C., Laursen, K. H., Podareanu, D., Codreanu, V., Uta, A., & Iosup, A. (2023). Less is not more: We need rich datasets to explore. Future Generation Computer Systems, 142, 117–130. https://doi.org/10.1016/j.future.2022.12.022

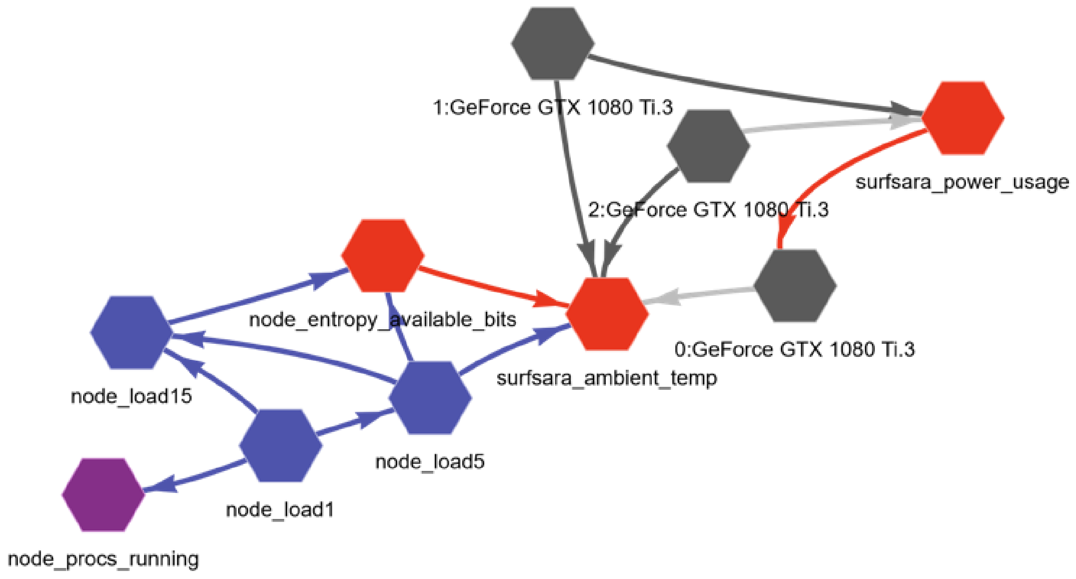

Fig. 9. Subgraph from CCS DAG r30n4. The consistent node from surfsara_power_usage to 0:GeForce GTX 1080 Ti.3 leads to the hypothesis of Thermal Throttling (RQ1.5). Ref to Fig 13.

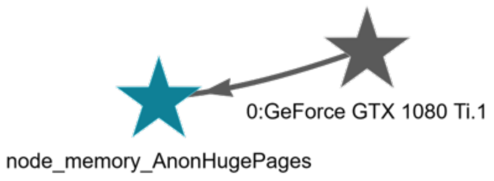

Fig. 10. Subgraph from CCS DAG r30n4. Ref to Fig 13.

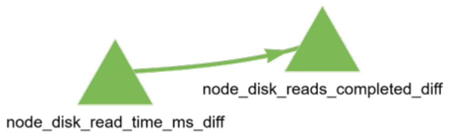

Fig. 11. Subgraph from CCS DAG r30n4. Ref to Fig 13.

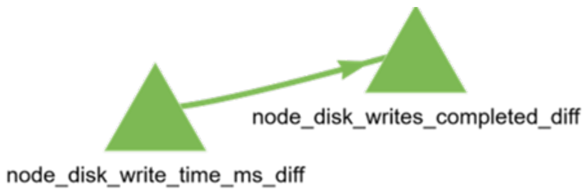

Fig. 12. Subgraph from CCS DAG r30n4. Ref to Fig 13.

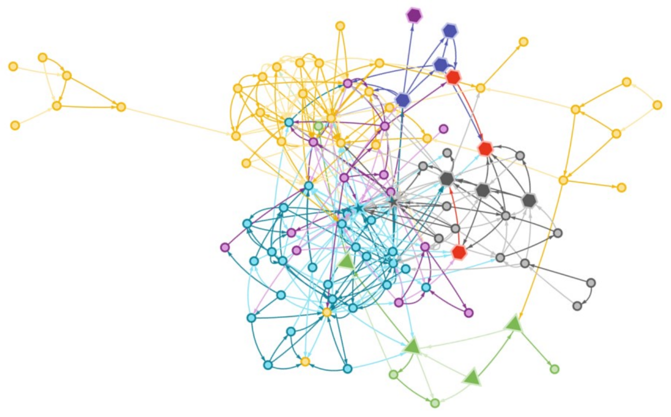

Fig. 13. CCS DAG r30n4. Lines in bold represent directed edges that were consistent across all tests. The nodes were highlighted according to manual metric grouping as described in Section II, C. Legend: Network (Yellow), System Performance (Light-Blue), Disk and File System (Green), Process and Resource Management (Purple), Misc (Red), Time-related (Dark-Blue). For a annotated version refer to https://github.com/EC- labs/ADS-tobias.

[20] Wang, X., Guo, P., & Wang, J. (2021). Large-Scale Causality Discovery Analytics as a Service. Umbc. https://par.nsf.gov/servlets/purl/10303949
[21] Welcome to CausalNex's API docs and tutorials! — causalnex 0.12.1 documentation. (n.d.). https://causalnex.readthedocs.io/en/latest/
[22] Zhao, Y., Duplyakin, D., Ricci, R., & Uta, A. (2021). Cloud Performance Variability Prediction. https://doi.org/10.1145/3447545.3451182
[23] Zheng, X. (2018, March 4). DAGs with NO TEARS: Continuous Optimization for Structure Learning. arXiv.org. https://arxiv.org/abs/1803.01422