CausalScaler: A Causality-Driven Autoscaling Framework for the Cloud

Zhemeng Yu[1,2], Yang Luo[1], Yucen Gao[1], Yinbo Sun[3], Xiaofeng Gao[1] *, Lintao Ma[3], and Guihai Chen[1]

[1] Shanghai Key Laboratory of Scalable Computing and Systems, College of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
{fish_meng,floating-dream,guo_ke}@sjtu.edu.cn,
{gao-xf,gchen}@cs.sjtu.edu.cn
[2] SJTU Paris Elite Institute of Technology, Shanghai Jiao Tong University, China
[3] Ant Group, Hangzhou, China
{yinbo.syb,lintao.mlt}@antgroup.com

Abstract. The unprecedented growth of cloud computing has highlighted critical challenges in resource autoscaling of heterogeneous cloud systems (HCS). Traditional approaches are limited by scope-restricted prediction, oversimplified metric interdependency, and rigid optimization mechanisms. We introduce CausalScaler, an innovative autoscaling framework addressing these challenges. We first design an integrated forecasting module that leverages holistic metric space information to capture diverse metric characteristics overlooked by traditional workload-only predictions. We then develop a dual-stream architecture combining explicit and implicit causal learning to identify fundamental causal relationships. Furthermore, we implement a frequency-aware decision-making module that adaptively optimizes customized objectives while maintaining system stability. Through extensive experiments on four real-world datasets and a month-long A/B test, CausalScaler achieves average improvements of 5.6% in forecasting accuracy and 7.8% in resource management efficiency over state-of-the-art baselines.

Keywords: Cloud System · Autoscaling · Resource Management · Time Series Forecasting.

1 Introduction

The escalating requirement for on-demand and scalable computing resources has driven the unprecedented growth of cloud computing. In recent years, the upsurge of machine learning and deep learning technologies has further pronounced

* X. Gao is the corresponding author. This work was supported by the National Key R&D Program of China [2024YFF0617700]; the National Natural Science Foundation of China [U23A20309, 62372296, 62272302], Shanghai Municipal Science and Technology Major Project [2021SHZDZX0102], and the Ant Group Research Fund [CCF-AFSGRF20230408].

this evolution, catalyzing the emergence of **Heterogeneous Cloud Systems (HCS)** that integrate heterogeneous computing resources [7].

In these large-scale cloud environments, resource management presents critical challenges, with **autoscaling** being a primary concern. Autoscaling mechanisms automatically adjust resources based on service demands to optimize utilization [17] while ensuring Service Level Agreement (SLA) requirements [14].

Delivering a wide spectrum of resource services, HCS exhibits several distinctive characteristics: (1) *Metric Diversity* - The system generates abundant metrics from heterogeneous resources, with each metric exhibiting distinct temporal dynamics such as periodicity and trending patterns. (2) *Hierarchical Causality* - These metrics form a three-layered structure: user-level metrics, resource-level metrics, and service-level metrics, where asymmetrical causal relationships propagate across different layers. (3) *Customized Objectives* - Facing user-centric platforms, besides resource efficiency, the scaling decisions must balance other aspects, including service quality and system stability. These characteristics of HCS pose challenges to conventional autoscaling approaches.

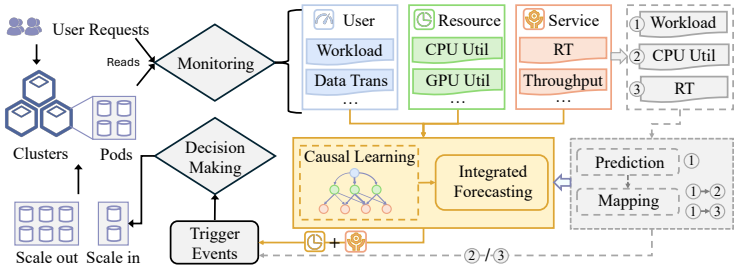

Fig. 1. Workflow comparison between traditional proactive scalers (gray) and CausalScaler. The decoupled workload-CPU prediction-mapping process is superseded by an integrated causality-aware forecasting module that leverages the complete system state. All resource and service metrics serve as scaling trigger candidates.

Challenge 1: Integrated Learning. As shown in Fig. 1, traditional proactive autoscaling solutions typically adopt a two-step fragmented approach: first forecasting the future workload, then determining resource requirements based on pre-established mapping relationships between workload and metrics. This decoupled process severely lacks flexibility facing *metric diversity*, as it only predicts workload and fails to leverage the rich historical information embedded in the numerous metrics, preventing a holistic understanding of the system state.
Challenge 2: Causal Discovery. The above mentioned mapping approach assumes workload directly affects all other system metrics. However, this cursory treatment fails to capture intricate interactions under *hierarchical causality*. For instance, inferring throughput solely based on workload-throughput relationship is overly simplistic, as factors like CPU utilization will also impact throughput. Furthermore, conventional correlation analysis often misinterprets coincidental

relationships or common causes as genuine dependencies. Therefore, causal analysis is needed to reveal key impact pathways within the hierarchical structure.

Challenge 3: Adaptive Scaling. Existing autoscaling solutions usually adopt a rigid optimization framework that trivializes complex system objectives by optimizing a single metric while treating others as simple threshold constraints. In light of *customized objectives*, this narrow approach lacks flexibility when confronting different scenarios where system priorities shift between different objectives. Additionally, current methods overlook that frequent scaling operations can introduce system instability and cause extra costs.

To this end, we propose **CausalScaler**, a causality-driven autosaling framework tailored for multi-metric heterogeneous cloud service systems.

The main contributions of this paper can be summarized as follows.

- We propose an integrated forecasting module that transcends two-stage prediction-and-mapping approaches. This module leverages holistic data to extract metric-centric insights and incorporates inter-metric relationship learning, enabling it to adapt flexibly to any multi-metric system.
- We integrate both explicit and implicit causal learning into the forecasting module to identify fundamental causal interdependencies that profoundly determine the prediction, thereby reinforcing forecast robustness.
- We introduce a scaling-frequency-aware decision-making module that adaptively supports diverse scenarios.
- We conducted extensive experiments on forecasting and decision-making to validate the effectiveness of CausalScaler.

2 Methodology

2.1 Causal Predictor

In **Challenge 1**, we point out the importance of leveraging metric-centric patterns. Thus, we employ the idea of [4] to encode each metric's time series independently, which facilitates inter-intra-metric relationship learning. Specifically, we extract the pattern of i-th metric temperal sequence $X^i \in \mathbb{R}^T$ as:

$$E^i = \text{Embedding}(X^i) : \mathbb{R}^T \mapsto \mathbb{R}^d, \tag{1}$$

where d is the dimension of the embedding for each metric, which gives the system state embedding of N metrics as $\mathbf{E}^N = \{E^1, E^2, \ldots, E^N\} \in \mathbb{R}^{N \times d}$.

5.2.1 Causal Aggregator for Explicit Causal Learning To leverage the causal relationships in **Challenge 2**, we first propose causal aggregator, which serves to statistically quantify causal relationships and reorganize the metrics.

(1) Causal Strength Quantification. To quantify inter-metric causal relationships, we define a Causal Strength Index (CSI) that measures the improvement in predicting one metric (X^j) given another metric's historical context (X^i):

$$CSI^{(l,k)}_{X^i \to X^j} = \sum_t p(X^j_{t+1}, X^j_{[t-k:t]}, X^i_{[t-l:t]}) \log_2 \frac{p(X^j_{t+1}|X^j_{[t-k:t]}, X^i_{[t-l:t]})}{p(X^j_{t+1}|X^j_{[t-k:t]})},$$

where l and k are respectively two lookback window sizes for X^i and X^j. By such formulation, CSI can simultaneously capture non-linear dependencies, directional information flow, and temporal evolution between time series.

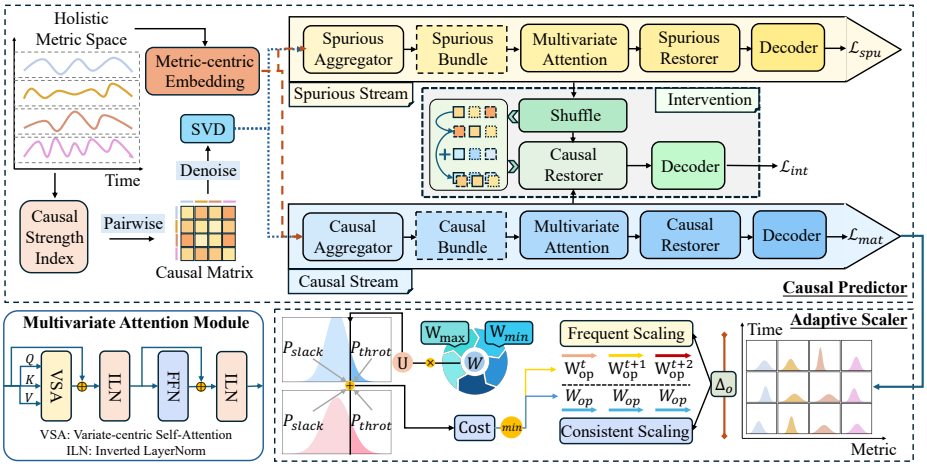

Fig. 2. Pipeline of CausalScaler: the Causal Predictor (top), the Adaptive Scaler (bottom), with the attention module structure detailed in the bottom left.

(2) Causal Matrix Denoising. Computing CSI between metric pairs yields a causal matrix $\mathbf{G} \in \mathbb{R}^{N \times N}$. We then apply Singular Value Decomposition (SVD) to remove noise from confounding factors and estimation uncertainties:

$$\mathbf{G}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{2}$$

where $\mathbf{\Sigma} \in \mathbf{R}^{N \times N}$ is a diagonal matrix containing singular values $\sigma_i = \mathbf{\Sigma}_{ii}$, quantifying the importance of each compositional direction in the causal structure. Hence, SVD decomposes $\mathbf{G}$ into a new linear system, where we can identify causal patterns and spurious correlations by the magnitude of singular values.

(3) Causal/Spurious Bundle Construction. Therefore, by retaining the top-N_c singular values and their corresponding vectors, we construct a noise-reduced

causal aggregator $\mathbf{G}_c \in \mathbb{R}^{N_c \times N}$ which preserves significant causal patterns:

$$\mathbf{G}_c = \mathrm{diag}(\sigma_1, \ldots, \sigma_{N_c})\mathbf{V}^T_{1:N_c}, \tag{3}$$

where N_c denotes the number of retained causal bundles. Complementarily, we define the spurious aggregator as $\mathbf{G_s} = \mathbf{I} - \mathbf{G_c}$ to aggregate non-causal correlations. Then, we apply these aggregators on system state embedding to obtain the aggregated causal and spurious bundles:

$$\mathbf{E_c} = \mathbf{G}_c\mathbf{E}^N, \mathbf{E_s} = \mathbf{G_s}\mathbf{E}^N \tag{4}$$

5.2.2 Dual-stream Intervention Module for Implicit Causal Learning According to backdoor adjustment theory [6], the spurious bundles could act as confounders, which form backdoor paths between causal bundles and prediction. We then leverage spurious bundles as intervention signals to distill causal features with universal predictive power.

(1) Causal-Spurious Dual Stream. The causal stream and spurious stream share an identical architecture, so we illustrate on the causal stream. With causal bundles as inputs, the causal stream first proceeds the forecasting with Multivariate Attention module, which employs variate-centric self-attention to learn the inter-bundle relationships. After that, we construct a learnable Causal Restorer $\mathbf{R}_c \in \mathbb{R}^{N_c \times N}$ to map the causal bundles back to original metrics:

$$\tilde{\mathbf{E}}_c = \mathbf{R}_c^T\mathbf{E}_c, \tag{5}$$

where $\tilde{\mathbf{E}}_c \in \mathbb{R}^{N \times d}$. Finally, given restored causal/spurious representations $\tilde{\mathbf{E}}_c$ and $\tilde{\mathbf{E}}_s$, we use a Decoder Φ to generate causal and spurious predictions:

$$\mathbf{F}_c = \Phi(\tilde{\mathbf{E}}_c), \quad \mathbf{F}_s = \Phi(\tilde{\mathbf{E}}_s), \tag{6}$$

where $\mathbf{F}_c, \mathbf{F}_s \in \mathbb{R}^{N \times S \times D}$. Here we obtain parameterized Gaussian distributions as the predictions for future S timestamps, so $D = 2$.

(2) Intervention Module. Inspired by [12], we implicitly strengthen causal learning via interventions. Specifically, we combine the causal representations with randomly shuffled spurious bundles:

$$\mathbf{E}_{\text{interv}} = \mathbf{E}_c \otimes \mathcal{S}(\mathbf{E}_s), \tag{7}$$

where $\mathbf{E}_{\text{interv}}$ denotes intervened patterns, $\mathcal{S}$ represents the random shuffle operation and $\otimes$ denotes the combination operation of addition or concatenation.

(3) Loss Construction. Using the intervened patterns $\mathbf{E}_{\text{interv}}$, we generate intervened predictions $\mathbf{F}_{\text{interv}} = \Phi(\mathbf{R}_c^T\mathbf{E}_{\text{interv}})$. To identify robust causal patterns that maintain stable predictive power under interventions, we encourage the $\mathbf{F}_{\text{interv}}$ to align with the $\mathbf{F}_c$ by minimizing their Kullback-Leibler (KL) Divergence:

$$\mathcal{L}_{\text{int}} = \frac{1}{NS}\sum_{n=1}^{N}\sum_{s=1}^{S} D_{\text{KL}}(\mathcal{N}(\mu_c^{n,s}, \sigma_c^{n,s^2}) \| \mathcal{N}(\mu_{\text{interv}}^{n,s}, \sigma_{\text{interv}}^{n,s^2})), \tag{8}$$

where $\mu^{n,s}$ and $\sigma^{n,s}$ denote the distribution parameters for the n-th metric at time s. Similarly, following [12], we constrain spurious predictions to follow a standard normal distribution to remove spurious influence:

$$\mathcal{L}_{\text{spu}} = \frac{1}{NS}\sum_{n=1}^{N}\sum_{s=1}^{S} D_{\text{KL}}(\mathcal{N}(\mu_{\text{s}}^{n,s}, \sigma_{\text{s}}^{n,s}) \| \mathcal{N}(0,1)), \tag{9}$$

In addition, since we treat the causal prediction as final outputs, we leverage Negative Log-Likelihood (NLL) Loss to measure prediction accuracy:

$$\mathcal{L}_{\text{mat}} = -\frac{1}{NS}\sum_{n=1}^{N}\sum_{s=1}^{S} \log p(y_s^n | \mu_c^{n,s}, \sigma_c^{n,s}), \tag{10}$$

where y_s^n is the ground truth value. Finally, we can define the total loss for our forecasting strategy with balanced hyperparameters λ_1, λ_2:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{mat}} + \lambda_1 \mathcal{L}_{\text{spu}} + \lambda_2 \mathcal{L}_{\text{int}}, \tag{11}$$

2.2 Adaptive Scaler

Given the predicted distributions for the metric system $\mathbf{F}_c \in \mathbb{R}^{N\times S\times D}$, our framework first allows users to flexibly select their monitoring objectives $\mathbf{O} \in \mathbb{R}^{M\times S\times D} \subset \mathbf{F}_c$ and specify their relative importance. Based on these selected metrics, we adaptively control the scaling frequency through distributional discrepancy analysis. When detecting significant viriability, i.e., the distribution discrepancy $\Delta_{\mathbf{O}} = \frac{1}{S-1}\sum_{t=0}^{S-1} D_{\text{KL}}(\mathbf{O}_{:,t,:} | \mathbf{O}_{:,t+1,:})$ is larger than the threshold Δ, we adopt **frequent scaling** where scaling decisions are made iteratively based on newly predicted distributions at each timestamp [17]. Conversely, when the discrepancy is minimal, we employ **consistent scaling** by making a single scaling decision that applies to all future S timestamps, which effectively reduces the overhead of container cold starts [8] and promotes system stability.

For robust resource allocation, we propose a dual-boundary triggering mechanism that simultaneously considers both under-provisioning and over-provisioning risks. Given a Allocation Unit (AU) configuration $\mathbf{U}$, we transverse the range of allocated numbers $W \in [W_{\min}, W_{\max}]$ to evaluate two critical probabilities: the slack probability $\mathbf{P}_{\text{slack}} = P(\mathbf{O} < W \times \mathbf{U})$ indicating potential resource waste, and the throttling probability $\mathbf{P}_{\text{throt}} = P(\mathbf{O} > W \times \mathbf{U})$ indicating potential resource shortage. The optimal allocation is determined by minimizing:

$$\text{Cost} = \mathbf{W}_{\text{slack}} \times \mathbf{P}_{\text{slack}} + \mathbf{W}_{\text{throt}} \times \mathbf{P}_{\text{throt}} \tag{12}$$

where user-defined weights $\mathbf{W}_{\text{slack}}, \mathbf{W}_{\text{throt}} \in \mathbb{R}^{M\times S}$ balance the trade-off between resource efficiency and service quality while enabling fine-grained control over different metrics' importance.

3 Experiments

3.1 Experimental Setup

Datasets To evaluate CausalScaler, we use four real-world datasets: (1) Ali dataset from Alibaba Cluster Trace[4] with resource utilization of 50 machines over 8 days, (2) Fisher dataset[5] tracking 10 containers over 30 days, (3) GPU dataset from 100 payment service clusters covering 3,441 GPUs over a week, and (4) CPU dataset with 120,000 CPU cores across 100 clusters over a week.

Table 1. Forecasting Results with different lookback windows. The best results are in bold and the second best results are underlined. Models: DpAR (DeepAR), MGT (MG-TSD), TSMx (TSMixer), DLin (DLinear), PTST (PatchTST), PyFM (Pyraformer), iTFM (iTransformer), CrFM (Crossformer). Note: .0067 denotes 0.0067.

T	Data	Metric	DpAR	CSDI	MGT	TSMx	DLin	PTST	PyFM	iTFM	CrFM	Ours
72	GPU	MSE	.0225	.0374	.0331	.2013	.0350	.0202	.0227	.0179	.8261	**.0169**
		CRPS	.0706	.1214	.1454	.3686	.1176	.0719	.0781	.0708	.6652	**.0633**
	CPU	MSE	.3593	.2674	.1258	.2997	.1365	.0327	.0676	.0337	.3639	**.0316**
		CRPS	.2142	.5457	.3317	.2347	.1920	.0708	.0891	.0711	.3067	**.0652**
	Ali	MSE	.0044	.0172	.0318	.0410	.0112	.0044	.0042	.0047	.0891	**.0037**
		CRPS	.0399	.0730	.0535	.1550	.0821	.0328	.0327	.0340	.2359	**.0298**
	Fisher	MSE	.0695	.0766	.0650	.0581	.0500	.0467	.0461	.0470	.0587	**.0455**
		CRPS	.1481	.1634	.1840	.1484	.1448	.1143	.1141	.1160	.1545	**.0993**
	Avg	MSE	.1139	.0997	.0639	.1500	.0582	.0260	.0352	.0258	.3344	**.0244**
		CRPS	.1182	.2259	.1786	.2267	.1341	.0725	.0785	.0730	.3406	**.0644**
144	GPU	MSE	.0291	.0409	.0357	.0880	.0419	.0267	.0241	.0248	1.2853	**.0235**
		CRPS	.0807	.1302	.2628	.2160	.1372	.0893	.0859	.0892	.9662	**.0785**
	CPU	MSE	.4260	.2680	.1718	.2465	.1748	.0604	.1084	.0683	.4071	**.0582**
		CRPS	.2170	.5273	.4722	.2449	.2066	.0921	.1146	.0962	.3866	**.0908**
	Ali	MSE	.0054	.0116	.0323	.0246	.0145	.0049	.0056	.0055	.0962	**.0045**
		CRPS	.0492	.0588	.0729	.1235	.0855	**.0351**	.0401	.0390	.2496	.0356
	Fisher	MSE	.0604	.0968	.0663	.0863	.0574	.0472	.0469	.0489	.0616	**.0452**
		CRPS	.1390	.2282	.1991	.1691	.1606	.1159	.1172	.1200	.1596	**.1023**
	Avg	MSE	.1302	.1043	.0765	.1114	.0722	.0348	.0462	.0369	.4626	**.0328**
		CRPS	.1215	.2361	.2517	.1884	.1475	.0831	.0895	.0861	.4405	**.0768**

Parameter Settings For all datasets, the sampling interval is 10 minutes using max-pooling and min-max normalization is applied with a 7:1:2 train-validation-test split. Experiments are conducted on an NVIDIA P100 GPU with batch size 128, hidden dimension 512, learning rate 0.001 and training epochs 30.

[4] https://github.com/alibaba/clusterdata/tree/v2018
[5] https://github.com/chrisliu1995/Fisher-model/tree/master

3.2 Comparison of Forecasting Models

Baselines and Evaluation Metrics We evaluate CausalScaler's forecasting performance against 9 state-of-the-art baselines, including probabilistic forecasting models (DeepAR [11], CSDI [13], MG-TSD [2]) and multivariate forecasting methods (TSMixer [1], DLinear [15], PatchTST [5], Crossformer [16], Pyraformer [3], and iTransformer [4]). Forecasting performance is evaluated using Mean Squared Error (MSE) for deterministic accuracy and Continuous Ranked Probability Score (CRPS) for probabilistic prediction quality.

Forecasting Performances We evaluate our model with different lookback windows of 72 and 144 timestamps. As shown in Table 1, CausalScaler consistently achieves superior performance across all datasets and historical length settings, with average improvements of 5.6% in MSE and 9.4% in CRPS compared to the best baseline. Notably, diffusion-based models (CSDI and MG-TSD) perform poorly on CPU datasets due to limited capability in capturing periodicity and rapid fluctuations. While attention-based models show good scalability generally, Crossformer exhibits degraded performance due to naive modeling of cross-variate dependencies. MLP-based approaches (TSMixer and DLinear) show limited effectiveness in handling diverse metric patterns in HCS. These results demonstrate that CausalScaler's forecasting module, with its metric characteristic modeling and causal learning mechanism, achieves robust performance.

3.3 Ablation Study

As shown in Fig. 3, we conducted ablation studies with a history length of 72. Removing the metric embedding and multivariate attention mechanism ("w/o M") leads to performance degradation across all datasets, particularly on CPU data where metrics show strong periodicity. Similarly, replacing the causal aggregator with random matrices ("w/o C") decreases performance, especially on the GPU dataset, which is a typical HCS dataset with numerous complex metrics. These results demonstrate that both components are essential: the metric-centric design captures individual metric characteristics, while the causal architecture models hierarchical interdependencies among metrics.

3.4 Comparison of Autoscaling Frameworks

Baselines and Evaluation Metrics To evaluate scaling decisions, we compare CausalScaler with four representative approaches: a Rule-based method using predefined rules, Autopilot [10], FIRM [9], and FSA [14]. We evaluate our framework through an A/B test on a payment application's cloud platform over one month. The test focuses on GPU-based services with CPU components, representing typical heterogeneous cloud systems. The evaluation metrics include (1) Resource Hours that measures GPU and CPU consumption time, (2) Resource Utilization that reflects resource efficiency, and (3) Success Rate that indicates the ability to meet service demands without resource shortages.

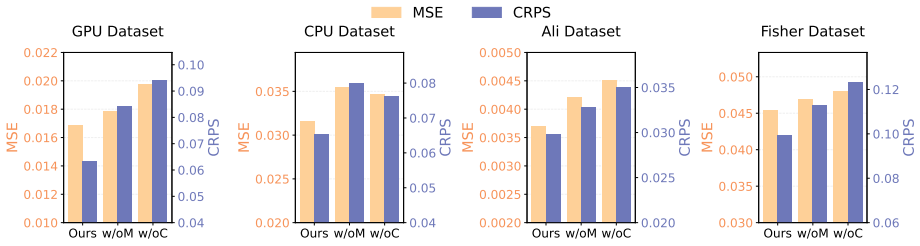

Fig. 3. Ablation Study. We compare two substitutions with the original CausalScaler on four datasets with historical length of 72, both MSE and CRPS are evaluated.

Table 2. Autoscaling Results. The best results are in bold.

Metrics	NoScale	Rule-Based	Autopilot	FIRM	FSA	CausalScaler
CPU_Hour	1.02e+7	5.56e+6	5.35e+6	3.72e+6	3.10e+6	**2.89e+6**
GPU_Hour	1.13e+6	6.50e+5	6.17e+5	4.09e+5	3.54e+5	**3.21e+5**
CPU_Uti	16.27%	29.87%	31.03%	44.62%	53.55%	**56.78%**
GPU_Uti	18.66%	32.46%	34.20%	51.59%	59.60%	**64.24%**
SuccR	100%	99.92%	99.88%	99.23%	95.87%	**99.90%**

Autoscaling Performances As shown in Table. 2, CausalScaler outperforms all baselines with up to 9.3% reduction in time consumption and 7.8% increase in resource utilization, while only experiencing a minimal 0.1% decrease in success rate compared to the most conservative non-scaling approach. From an economic perspective, we achieved significant savings by reducing CPU and GPU time by 210,000 and 33,000 hours respectively, resulting in cost savings exceeding $100,000. These results demonstrate CausalScaler's ability to effectively balance resource efficiency and service stability in HCS environments.

4 Conclusion

In this paper, we propose CausalScaler, a novel autoscaling framework for heterogeneous cloud systems that addresses critical limitations in traditional approaches. Our framework innovates through an integrated forecasting module utilizing holistic metric space, a dual-stream architecture for robust causal learning, and a frequency-aware decision module for balanced resource management. Extensive experiments demonstrate CausalScaler's superior performance in both forecasting accuracy and resource efficiency.

References

1. Ekambaram, V., Jati, A., Nguyen, N., Sinthong, P., Kalagnanam, J.: Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 459–469 (2023)

2. Fan, X., Wu, Y., Xu, C., Huang, Y., Liu, W., Bian, J.: Mg-tsd: Multi-granularity time series diffusion models with guided learning process. In: The Twelfth International Conference on Learning Representations
3. Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A.X., Dustdar, S.: Pyraformer: Low-complexity pyramidal at-tention for long-range time series modeling and forecasting. The tenth international conference on learning representations (2022)
4. Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: itransformer: Inverted transformers are effective for time series forecasting. In: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024 (2024)
5. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. In: The Eleventh International Conference on Learning Representations
6. Pearl, J.: Interpretation and identification of causal mediation. Psychological methods **19**(4), 459 (2014)
7. Persico, V., Grimaldi, D., Pescape, A., Salvi, A., Santini, S.: A fuzzy approach based on heterogeneous metrics for scaling out public clouds. IEEE Transactions on Parallel and Distributed Systems **28**(8), 2117–2130 (2017)
8. Qian, H., Wen, Q., Sun, L., Gu, J., Niu, Q., Tang, Z.: Robustscaler: Qos-aware autoscaling for complex workloads. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE). pp. 2762–2775. IEEE (2022)
9. Qiu, H., Banerjee, S.S., Jha, S., Kalbarczyk, Z.T., Iyer, R.K.: {FIRM}: An intelligent fine-grained resource management framework for {SLO-Oriented} microservices. In: 14th USENIX symposium on operating systems design and implementation (OSDI 20). pp. 805–825 (2020)
10. Rzadca, K., Findeisen, P., Swiderski, J., Zych, P., Broniek, P., Kusmierek, J., Nowak, P., Strack, B., Witusowski, P., Hand, S., et al.: Autopilot: workload autoscaling at google. In: Proceedings of the Fifteenth European Conference on Computer Systems. pp. 1–16 (2020)
11. Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T.: Deepar: Probabilistic forecasting with autoregressive recurrent networks. International journal of forecasting **36**(3), 1181–1191 (2020)
12. Sui, Y., Wang, X., Wu, J., Lin, M., He, X., Chua, T.S.: Causal attention for interpretable and generalizable graph classification. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2022)
13. Tashiro, Y., Song, J., Song, Y., Ermon, S.: Csdi: Conditional score-based diffusion models for probabilistic time series imputation. Advances in Neural Information Processing Systems **34**, 24804–24816 (2021)
14. Wang, S., Sun, Y., Shi, X., Shiyi, Z., Ma, L.T., Zhang, J., Zheng, Y., Jian, L.: Full scaling automation for sustainable development of green data centers. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. IJCAI '23 (2023)
15. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 11121–11128 (2023)
16. Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: The eleventh international conference on learning representations (2023)
17. Zou, D., Lu, W., Zhu, Z., Lu, X., Zhou, J., Wang, X., Liu, K., Wang, K., Sun, R., Wang, H.: Optscaler: A collaborative framework for robust autoscaling in the cloud. Proceedings of the VLDB Endowment **17**(12), 4090–4103 (2024)