CHRONOGRAPH: A Real-World Graph-Based Multivariate Time Series Dataset

Adrian Catalin Lutu[1,2]* **Ioana Pintilie**[1]* **Elena Burceanu**[1] **Andrei Manolache**[1,3,4]

[1]Bitdefender, Romania
[2]University of Bucharest, Romania
[3]University of Stuttgart, Germany
[4]International Max Planck Research School for Intelligent Systems, Germany
{alutu,ipintilie,eburceanu,amanolache}@bitdefender.com

Abstract

We present CHRONOGRAPH, a graph-structured *multivariate time series forecasting* dataset built from real-world production microservices. Each node is a service that emits a multivariate stream of system-level performance metrics, capturing CPU, memory, and network usage patterns, while directed edges encode dependencies between services. The primary task is forecasting future values of these signals at the service level. In addition, CHRONOGRAPH provides expert-annotated incident windows as anomaly labels, enabling evaluation of anomaly detection methods and assessment of forecast robustness during operational disruptions. Compared to existing benchmarks from industrial control systems or traffic and air-quality domains, CHRONOGRAPH uniquely combines (i) multivariate time series, (ii) an explicit, machine-readable dependency graph, and (iii) anomaly labels aligned with real incidents. We report baseline results spanning forecasting models, pretrained time-series foundation models, and standard anomaly detectors. CHRONOGRAPH offers a realistic benchmark for studying structure-aware forecasting and incident-aware evaluation in microservice systems. Our dataset and code are publicly available at `https://github.com/bit-ml/ChronoGraph`.

1 Introduction and Related Work

Forecasting the short- and medium-term evolution of service metrics is central to reliable operations in large-scale automated systems. Advances in connectivity, monitoring, and data collection have enabled such systems across domains ranging from manufacturing and transportation to IT infrastructure [1], where forecasts drive alerting, autoscaling, and capacity planning. Microservice architectures add unique challenges: hundreds of loosely coupled services form a dependency graph where disruptions (e.g., regressions, resource contention, upstream failures) can propagate across calls, making accurate forecasts dependent on both local temporal dynamics and cross-service influences. Existing graph benchmarks, however, do not reflect this setting: traffic [2] and air-quality [3, 4] datasets widely used for forecasting are univariate and lack incident annotations, while industrial control datasets such as SWaT [5] and WADI [6] include anomaly labels and are multivariate but provide only process diagrams rather than a true adjacency matrix. This leaves open the need for benchmarks that combine multivariate time series, explicit dependency structure, and labeled incidents in a single dataset.

This landscape has steered both forecasting and anomaly detection toward models that are largely topology-agnostic. Forecasters usually treat each signal independently or aggregate features without structural context. Similarly, most anomaly detection approaches operate on individual streams, with

*Equal contribution.

NeurIPS 2025 Workshop on Recent Advances in Time Series Foundation Models (BERT^{2}S).

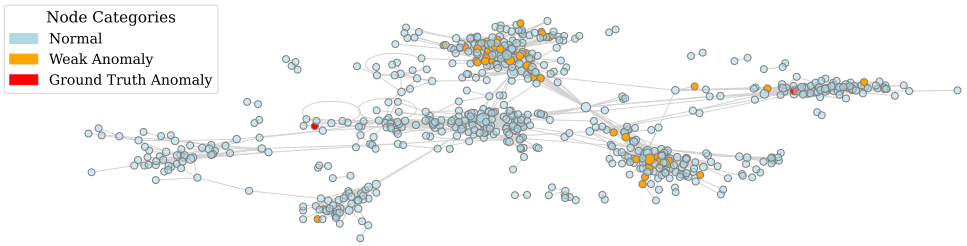


Figure 1: System architecture at a given time. Nodes represent microservices, and edges indicate inter-service communication. Ground-truth disruptions (red) and high-confidence predictions from our ensemble (orange) are overlaid. The spatial clustering of anomalies in densely connected regions suggests that abnormal behavior tends to propagate along the underlying graph topology.

anomalies often inferred indirectly from forecast residuals or reconstruction errors [7–15]. Foundation models for time series (e.g., Chronos, TabPFN-TS [16, 17]) extend forecasting capacity across domains, but they are typically applied per-series and thus cannot capture propagation effects across nodes. Existing graph-aware methods bypass actual dependencies by assuming global structures or learning dense, latent graphs end-to-end. Techniques include using attention over complete graphs [18], top-k similarity sampling [19], or latent graph sampling [20]. These induced, data-driven structures are often dense and may not align with the true dependency topology.

To catalyze graph-aware forecasting in multivariate time series, we release CHRONOGRAPH, a benchmark where nodes carry multivariate service metrics and edges capture call dependencies. Unlike existing datasets, it combines explicit topology with labeled incidents, enabling the evaluation of both forecasting and anomaly detection methods. This design exposes the limitations of topology-agnostic baselines and foundation models, and motivates structure-aware approaches that integrate temporal dynamics with graph information. Finally, we hope that resources like CHRONOGRAPH will help connect the multivariate time series and temporal graph learning communities [21, 22] by providing a shared benchmark. Our contributions are as follows:

1. **A real-world multivariate time series dataset.** We release CHRONOGRAPH, built from six months of production microservice telemetry covering $\sim$ 700 services, each represented by 5-dimensional time series with $\sim$ 8,000 time steps per service along with 8-dimensional time series capturing the communication between interacting services.
2. **Explicit graph topology and incident propagation.** The dataset includes a service dependency graph showing empirical anomaly propagation. This helps models capture topological correlations and supports the evaluation of topology-aware forecasting.
3. **Baseline evaluation.** We benchmark forecasting methods, time-series foundation models, and standard anomaly detectors. We highlight key limitations of current approaches: their difficulty with long-term forecasting and their inability to leverage the system's graph structure, and we discuss potential paths for improvement.

2 Data Collection and Labeling

Data. CHRONOGRAPH comprises six months of operational telemetry collected from a production microservice platform operated by a large enterprise. For each of the 708 services, we monitor five system-level metrics: CPU usage, memory usage, memory working set, and network traffic rates (incoming and outgoing). Measurements are recorded per container and aggregated to the service level (mean across containers) at a fixed 30-minute interval. After removing series with prolonged discontinuities (e.g., due to maintenance or decommissioning) and aligning timestamps, each service is represented as a multivariate time series with 8005 time steps and five variables. In addition, the observed inter-service communication defines a directed graph, where nodes are associated with multivariate time series and edges represent service dependencies along 8-dimensions: number of requests, return codes, and latency. This explicit topology enables structure-aware forecasting and allows analysis of how disturbances propagate through the system.

Table 1: Forecasting performance across models, averaged (mean±std) over all services. We report results on the full test window (3202 steps) and on the first 500 steps. All models achieve substantially lower errors on the shorter horizon, highlighting the limited long-term stability of current approaches.

Model	window=3202			window=500		
	MAE (↓)	MSE (↓)	MASE (↓)	MAE (↓)	MSE (↓)	MASE (↓)
Prophet	$\mathbf{0.125}_{\pm 0.067}$	$\mathbf{0.044}_{\pm 0.054}$	$7.182_{\pm 11.210}$	$0.069_{\pm 0.044}$	$0.013_{\pm 0.022}$	$3.143_{\pm 3.663}$
Chronos	$0.150_{\pm 0.173}$	$0.343_{\pm 2.426}$	$7.902_{\pm 12.707}$	$\mathbf{0.044}_{\pm 0.030}$	$\mathbf{0.007}_{\pm 0.015}$	$\mathbf{1.938}_{\pm 1.731}$
TabPFN-TS	$\mathbf{0.125}_{\pm 0.125}$	$0.089_{\pm 1.172}$	$\mathbf{6.205}_{\pm 9.315}$	$0.109_{\pm 0.061}$	$0.026_{\pm 0.031}$	$5.082_{\pm 11.013}$

Table 2: Performance of individual models and an ensemble, evaluated on human-labeled service disruptions (partial coverage). The ensemble (*) combines Prophet, Isolation Forest, and Autoencoder.

Method	$F1_K$ (↑)	ROC_K (↑)	FP rate (↓)	FN rate (↓)	$F1$ (↑)
Prophet	**20.57**	**62.97**	2.02	97.98	2.39
Isolation Forest	17.49	56.39	46.9	**50.48**	**7.08**
OC-SVM	14.46	54.31	22.13	77.08	5.50
Autoencoder	13.86	59.79	0.38	99.58	0.72
TabPFN-TS	12.37	54.08	0.55	99.79	0.31
Chronos	12.41	49.78	2.49	97.84	2.49
Ensemble*	16.92	60.95	**0.20**	99.58	0.73

Service Disruption Labels. The dataset also includes a curated set of anomaly labels derived from internal incident reports. Human-written incident entries were parsed to extract affected services and timestamps, which were then mapped to fixed-length windows centered on the reported time. This procedure yields 17 labeled anomaly segments associated with specific services.

3 Empirical Evaluation

Objectives. We pursue two aims: (i) *Benchmark forecasting models* on CHRONOGRAPH, comparing forecasting methods and time-series foundation models and (ii) *Evaluate anomaly detection methods* on the multivariate time series using the disruption labels as ground truth. We describe the evaluation metrics, models, and experimental protocol in appendix A.1.

3.1 Forecasting

To assess the temporal stability of forecasting performance, we evaluate each model on both the full test sequence (3202 time steps) and a shorter horizon comprising the first 500 steps. As shown in Table 1, which reports average errors across all services, all models exhibit significantly lower errors in the short-term window. This consistent performance drop over longer horizons suggests that current methods lack the capacity to model long-range temporal dynamics effectively. Among them, Chronos shows the strongest decline: while it achieves the best performance on the smaller 500-point subset, it performs worst on the full sequence. In contrast, TabPFN-TS remains the most stable across both settings, and on the full sequence its performance is comparable to Prophet.

3.2 Anomaly Detection

The results in Table 2 show that performance remains modest across the evaluated methods. Prophet achieves the strongest individual results, though it still suffers from a very high false negative rate. Isolation Forest and OC-SVM detect more anomalies but at the cost of high false positive rates, while autoencoder-based detection and foundation models (TabPFN-TS, Chronos) perform poorly overall. The ensemble of Prophet, Isolation Forest, and Autoencoder offers somewhat more balanced performance, but remains far from reliably identifying disruption periods. Taken together, these results indicate that methods commonly used on time-series, when applied in a topology-agnostic fashion, struggle to account for the complex dynamics of microservices. Incorporating the dependency graph into forecasting or detection may offer a promising path toward more effective approaches.

It is important to stress that our ground-truth labels are sparse and limited to reported service disruptions. As a result, the dataset likely contains anomalous behaviors that were not escalated into

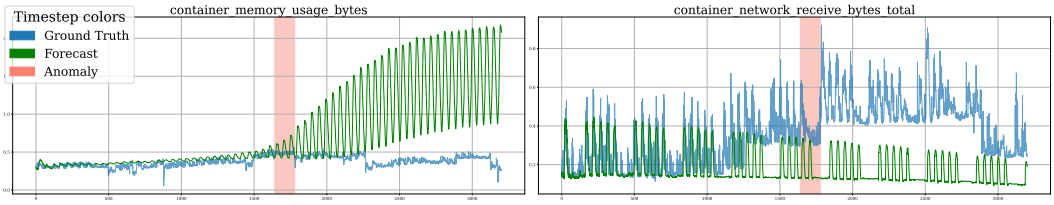


Figure 2: Chronos forecasts on two metrics of the same microservice during a disruption. **Left:** predictions drift when the anomaly occurs (red band). **Right:** the model captures only a periodic baseline, missing the bursty variability of network traffic.

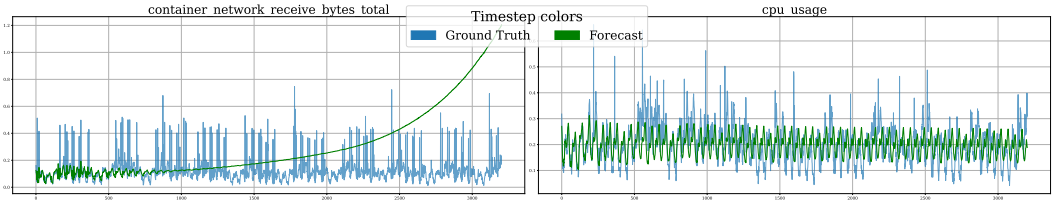


Figure 3: Chronos forecasts on two metrics of a microservice. **Left:** network traffic is accurate at first but then drifts upward even without an anomaly, illustrating the limited stability of long-horizon predictions. **Right:** CPU usage is forecast accurately across the full test time window.

incidents, either because they were transient or because the affected services self-recovered. While detecting these currently counts as false positives, they are operationally valuable, potentially helping operators identify issues before they become service-impacting.

3.3 Limitations of Current Approaches

Limited Forecast Horizon. Our results show that existing models struggle with long-term forecasting. Qualitative examples (Fig.2, Fig.3) illustrate that forecasts quickly diverge when the underlying series becomes volatile or when disruptions occur. Quantitatively, Table 1 shows a clear gap: while models achieve reasonable results on the first 500 test points, performance degrades substantially when evaluated over the full horizon. This indicates that current approaches are effective at short-term prediction but lack the ability to sustain accurate forecasts as dynamics grow more complex.

Lack of Propagation Capabilities. A second limitation is that none of the models account for the service dependency graph. In Figure 1, we plot the predictions of the system with the lowest FPR in Table 2, which is the ensemble. Although its overall performance is limited, it combines predictions from diverse anomaly detection methods. As a result, its sparse positive predictions may reflect genuine irregularities in the series that did not escalate into labeled service failures. Predicted anomalies often cluster along connected services, suggesting that disturbances propagate through the system. These findings highlight the potential of topology-aware methods: incorporating the graph structure may improve long-term forecasting and anomaly detection by modeling both disruption spread and normal service interactions. While we do not explore such methods here, we consider them a promising direction for future work.

4 Conclusions and Future Work

We introduced ChronoGraph, a real-world dataset collected from a large-scale enterprise microservice platform. Its provenance ensures that signals and disruptions reflect real operational challenges, providing a realistic basis for model evaluation. Unlike existing benchmarks, ChronoGraph combines a) multivariate time series, b) a dependency graph, and c) labeled disruption windows. We benchmarked standard forecasting and anomaly detection methods across hundreds of services. Forecasting models performed well on short horizons but degraded over longer windows, revealing limited capacity to model evolving dynamics. Anomaly detectors showed modest accuracy, with predictions often clustering along connected services-suggesting propagation effects and short-lived irregularities not labeled as disruptions. These findings indicate that combining multivariate dynamics with service dependencies could improve both forecasting and anomaly detection. We present ChronoGraph as a foundation for advancing graph-based approaches and research in these tasks.

Acknowledgements

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No: 101120237 (ELIAS).

References

[1] Tiago Zonta, Cristiano André da Costa, Rodrigo da Rosa Righi, Miromar José de Lima, Eduardo Silveira da Trindade, and Guann Pyng Li. Predictive maintenance in the industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, 150: 106889, 2020. ISSN 0360-8352. doi: https://doi.org/10.1016/j.cie.2020.106889. URL `https://www.sciencedirect.com/science/article/pii/S0360835220305787`.

[2] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv: Learning*, 2017. URL `https://api.semanticscholar.org/CorpusID:3508727`.

[3] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. U-air: when urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 1436–1444, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450321747. doi: 10.1145/2487575.2488188. URL `https://doi.org/10.1145/2487575.2488188`.

[4] Guannan Geng, Qingyang Xiao, Shigan Liu, Xiaodong Liu, Jing Cheng, Yixuan Zheng, Dan Tong, Bo Zheng, Yiran Peng, Xiaomeng Huang, Kebin He, and Qiang Zhang. Tracking air pollution in china: Near real-time pm2.5 retrievals from multiple data sources, 2021. URL `https://arxiv.org/abs/2103.06520`.

[5] Aditya P. Mathur and Nils Ole Tippenhauer. Swat: a water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks, CySWater@CPSWeek 2016, Vienna, Austria, April 11, 2016*, pages 31–36. IEEE Computer Society, 2016.

[6] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur. WADI: a water distribution testbed for research in the design of secure cyber physical systems. In Panagiotis Tsakalides and Baltasar Beferull-Lozano, editors, *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWATER@CPSWeek 2017, Pittsburgh, Pennsylvania, USA, April 21, 2017*, pages 25–28. ACM, 2017.

[7] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga. USAD: unsupervised anomaly detection on multivariate time series. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3395–3404. ACM, 2020.

[8] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Dae-ki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[9] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 1409–1416. AAAI Press, 2019.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3431–3440. IEEE Computer Society, 2015.

[11] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis, editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 2828–2837. ACM, 2019.

[12] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In Feida Zhu, Beng Chin Ooi, and Chunyan Miao, editors, *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 3220–3230. ACM, 2021.

[13] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate time-series anomaly detection via graph attention network. In Claudia Plant, Haixun Wang, Alfredo Cuzzocrea, Carlo Zaniolo, and Xindong Wu, editors, *20th IEEE International Conference on Data Mining, ICDM 2020, Sorrento, Italy, November 17-20, 2020*, pages 841–850. IEEE, 2020.

[14] Ioana Pintilie, Andrei Manolache, and Florin Brad. Time series anomaly detection using diffusion-based models. *ICDM 2023 Workshop AI4TS*, 2023.

[15] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks. In Igor V. Tetko, Vera Kurková, Pavel Karpov, and Fabian J. Theis, editors, *Artificial Neural Networks and Machine Learning - ICANN 2019: Text and Time Series - 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17-19, 2019, Proceedings, Part IV*, volume 11730 of *Lecture Notes in Computer Science*, pages 703–716. Springer, 2019.

[16] Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR. Expert Certification.

[17] Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. The tabular foundation model tabpfn outperforms specialized time series forecasting models based on simple features. *arXiv preprint arXiv:2501.02945*, 2025. URL https://arxiv.org/abs/2501.02945v2.

[18] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate time-series anomaly detection via graph attention network. pages 841–850, 11 2020. doi: 10.1109/ICDM50108.2020.00093.

[19] Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 4027–4035. AAAI Press, 2021.

[20] Zekai Chen, Dingshuo Chen, Xiao Zhang, Zixuan Yuan, and Xiuzhen Cheng. Learning graph structures with transformer for multivariate time series anomaly detection in iot. *IEEE Internet of Things Journal*, pages 1–1, 2021. doi: 10.1109/JIOT.2021.3100509.

[21] Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal graph benchmark for machine learning on temporal graphs. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 2056–2073. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/066b98e63313162f6562b35962671288-Paper-Datasets_and_Benchmarks.pdf.

[22] Julia Gastinger, Shenyang Huang, Mikhail Galkin, Erfan Loghmani, Ali Parviz, Farimah Poursafaei, Jacob Danovitch, Emanuele Rossi, Ioannis Koutis, Heiner Stuckenschmidt, Reihaneh Rabbany, and Guillaume Rabusseau. TGB 2.0: A benchmark for learning on temporal knowledge graphs and heterogeneous graphs. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=EADRzNJFn1.

[23] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[24] Abdul Fatir Ansari, Michael Bohlke-Schneider, Henry Elder, Nikhil Gauraha, Saurabh Kapoor, Dustin Lange, David Salinas, Kari Torkkola, and Yuyang Wang. Chronos: Learning the language of time series. *arXiv preprint arXiv:2310.06625*, 2023.

[25] Sean J. Taylor and Benjamin Letham. Forecasting at scale. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 793–802, 2018.

[26] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[27] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11. ACM, 2014.

[28] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[29] Bernhard Schölkopf, John Platt, John Shawe-Taylor, Alexander J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

A Appendix

A.1 Further Details on the Experimental Setup

Metrics. We evaluate forecasting capabilities using the mean absolute error (MAE), mean sqaured error (MSE), and the mean asbsolute scaled error (MASE). Conventional metrics for anomaly detection, such as precision, recall, and $F1$-score, can overly penalize false negatives inside a longer anomaly segment. Conversely, when combined with point-adjustment (PA), they can severely overestimate anomaly detection performance: a segment is counted as correctly detected if *any* point within it is flagged. To overcome this, we adopt $F1_K - AUC$ and $ROC_K - AUC$ as our primary evaluation metrics. $F1_K - AUC$ integrates performance across different K ratios of correctly predicted points [23], providing a more balanced view of segment-level accuracy. $ROC_K - AUC$ further extends this idea by jointly evaluating across both anomaly score thresholds and K values, thereby enabling fairer model comparisons regardless of hyperparameter sensitivity [14].

Models. In order to capture the strengths and weaknesses of fundamentally different modeling paradigms, we choose a broad set of approaches spanning statistical models, representation learning, foundation models, and classical machine learning methods. We include two time series foundation models: Chronos-Bolt Base [24], designed for zero-shot and few-shot time series tasks, and TabPFN-TS [17], a transformer-based prior-data-fitted network adapted for time series. Within the class of statistical forecasting methods, we employ Prophet [25], a decomposable model for trend, seasonality, and holiday effects, since ARIMA [26] in our setting mostly defaulted to forecasting the mean under noisy conditions. For anomaly detection, in addition to using residuals from the forecasting models, we also consider Autoencoders [27], which detect anomalies through reconstruction error, along with two classical AD methods, Isolation Forest [28] and the One-Class SVM (OC-SVM) [29], as they remain strong baselines for high-dimensional outlier detection.

Training details. We apply a 60/40 train–test split to ensure sufficient history for model learning while retaining ample data for evaluation. For Prophet, we train a separate model for each dimension to capture individual temporal dynamics. For Chronos, we adopt a rolling prediction strategy: we forecast the first 64 time steps, append them to the context, and then use this extended context to predict the next 64 time steps, repeating the process until the forecast horizon is reached. In contrast, TabPFN-TS is trained jointly on the entire MTS, leveraging cross-series dependencies to model shared temporal patterns. The resulting forecasts from all models are then used for anomaly detection by computing z-scores of the residuals, which are subsequently evaluated using the anomaly detection protocol described in [14]

A.2 Additional Figures

Figure 4 highlights anomaly detection qualitative examples for both Prophet and the ensemble model. While Prophet successfully detects multiple points inside the ground truth anomaly segment, it has a higher FPR than the ensemble. Figure 5 further compares the distributions of anomaly counts per time step detected by the ensemble model versus the Prophet model. While most time steps contain few or no anomalies, a small subset exhibit higher anomaly counts, suggesting that anomalies may propagate along the graph structure. The y-axis is log-scaled, highlighting the heavy-tailed nature of anomaly occurrence, where rare but extreme anomaly events are present. Finally, Figures 6 and 7 present the distributions of anomalies per time step identified by the Prophet model and the ensemble approach, respectively.

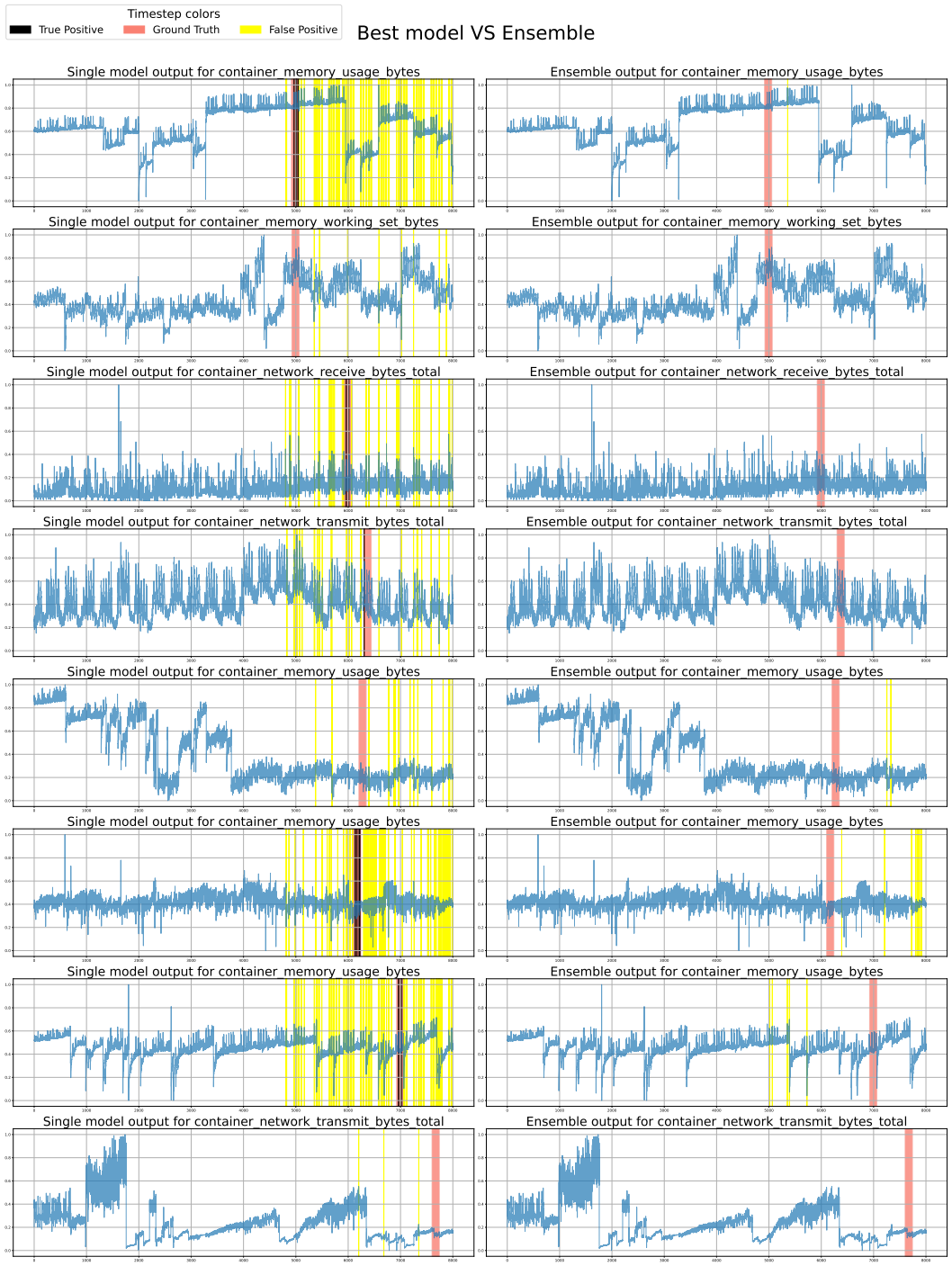
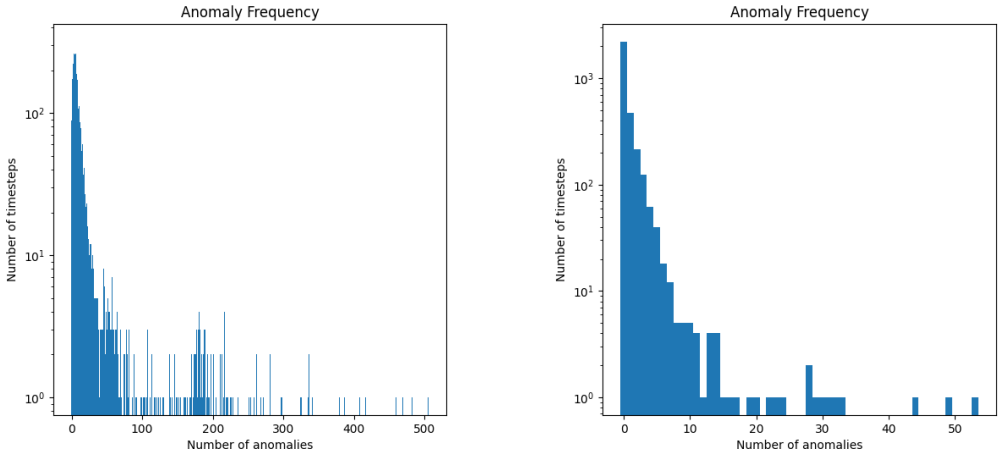


Figure 4: Comparison of Prophet model and ensemble anomaly detection outputs across multiple container metrics



(a) Anomaly Frequency Prophet (b) Anomaly Frequency Ensemble

Figure 5: Comparison of anomaly frequency distributions.

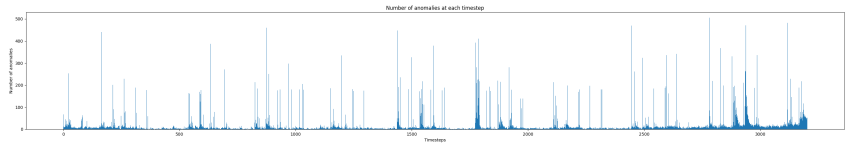


Figure 6: Anomalies at each time step found by Prophet

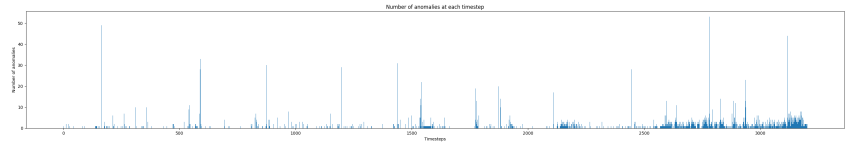


Figure 7: Anomalies at each time step found by the ensemble