Causal Machine Learning Approaches for Modelling Data Center Heat Recovery: A Physical Testbed Study.

DAVID ZAPATA GONZALEZ, Paderborn University, Germany
MARCEL MEYER, Paderborn University, Germany
OLIVER MÜLLER, Paderborn University, Germany

Data centers (DCs) form the backbone of our growing digital economy, but their rising energy demands pose challenges to our environment. At the same time, reusing waste heat from DCs also represents an opportunity, for example, for more sustainable heating of residential buildings. Modeling and optimizing these coupled and dynamic systems of heat generation and reuse is complex. On the one hand, physical simulations can be used to model these systems, but they are time-consuming to develop and run. Machine learning (ML), on the other hand, allows efficient data-driven modeling, but conventional correlation-based approaches struggle with the prediction of interventions and out-of-distribution generalization. Recent advances in causal ML, which combine principles from causal inference with flexible ML methods, are a promising approach for more robust predictions. Due to their focus on modeling interventions and cause-and-effect relationships, it is difficult to evaluate causal ML approaches rigorously. To address this challenge, we built a testbed of a miniature DC with an integrated waste heat network, equipped with sensors and actuators. This testbed allows conducting controlled experiments and automatic collection of realistic data, which can then be used to benchmark conventional and causal ML methods. Our experimental results highlight the strengths and weaknesses of each modeling approach, providing valuable insights on how to appropriately apply different types of machine learning to optimize data center operations and enhance their sustainability.

CCS Concepts: • **Information systems** → **Data centers**; • **Theory of computation** → *Inductive inference*; • **Hardware** → Power and energy.

Additional Key Words and Phrases: Data Center Operations, Heat Recovery, Causal Machine Learning

1 INTRODUCTION

Data centers (DCs) are the heart of the digital infrastructure, but also consume large amounts of energy [17]. By 2030, the energy demands of DCs are anticipated to grow significantly, with estimation reaching over 700 TWh, equivalent to 2% of global electricity use, which is a two-to-threefold increase compared to 2016 levels [15]. Most of the electric energy used in DCs is transformed into heat, which is currently mostly emitted into the atmosphere. It could, however, potentially be reused, for example, for district heating [34, 35]. Hence, the recovery of waste heat from DCs is a promising and timely initiative for minimizing their carbon footprint.

Modeling the thermal energy behavior of data centers (DCs) is critical for improving their efficiency and sustainability. While data-driven models have emerged as a fast and accurate alternative to traditional simulation-based approaches [36], most are built on statistical correlations and struggle to generalize to unobserved scenarios [20, 22, 27]. This limits their utility in evaluating interventions, such as changes in IT load or environmental conditions [7]. Addressing this gap requires approaches that better capture the causal structure of DC operations.

Fig. 1. Data center with heat recovery testbed

To overcome the limitations of traditional ML models, recent work has explored causal ML approaches to model energy systems [6, 12]. These methods combine causal inference with flexible ML tools to enable reasoning about interventions and counterfactuals. Since it is difficult or impossible to validate such models using purely observational data, and testing on real operational systems might be expensive or not possible, researchers argue that it is vital to use controlled physical testbeds to generate data with known dynamics [8, 31].

Inspired by these testbeds, we built a water-cooled data center testbed connected to a district heating network (Figure 1). This testbed, equipped with sensors and controllable actuators, enables controlled experiments. We collected detailed time-series data from these interventions to train and test correlation-based and causal ML models to predict the data center's waste heat potential, represented by the water temperature in the district heating network. By comparing their performance, we highlight the strengths and weaknesses of each approach.

We found that the predictive performance of the models varied substantially between overall performance and interventional scenarios. While conventional ML models performed in general well for predicting the water temperature, causal ML approaches excelled at predicting the effect of interventions. The results highlight the utility of physical testbeds for evaluating models of sustainable computing systems. In addition, our results demonstrate the potential of causal ML for robust and transparent data-driven modeling of such systems.

2 BACKGROUND

2.1 Modeling of Data Centers Operations

Approaches for modeling environmental and operational conditions in data centers (DCs) can be grouped into three categories: simplified

Authors' Contact Information: David Zapata Gonzalez, Paderborn University, Paderborn, Germany, david.zapata@uni-paderborn.de; Marcel Meyer, Paderborn University, Paderborn, Germany, marcel.meyer@uni-paderborn.de; Oliver Müller, Paderborn University, Paderborn, Germany, oliver.mueller@uni-paderborn.de.

physics-based models, computational fluid dynamics/heat transfer (CFD/HT) simulations, and data-driven methods [2]. Simplified models offer computational efficiency but rely on strong assumptions that limit applicability in real scenarios [10]. CFD/HT simulations solve complex physical equations and provide fine-grained, accurate predictions, but are computationally expensive and time-consuming to program [2], motivating the search for more efficient alternatives [7]. Data-driven approaches like machine learning (ML) provide a promising middle ground. Supervised ML involves developing a statistical model to predict an output based on one or more input variables [11]. The model is trained on historical data and then used to predict outcomes for new, unseen cases stemming from the same data distribution. At this, the primary objective is accurate prediction, so the model is optimized to minimize the error between its predictions and the actual values. It is to note that ML leverages statistical association between variables, without necessarily considering the true cause-and-effect relationships of the data-generating process [4, 18].

ML techniques have been extensively used for predictive modeling of DC operations, covering aspects such as electricity consumption, server room temperatures, IT workloads, or energy costs. Jin et al. [13] reviewed power consumption prediction models in DCs, emphasizing the necessity of accurate models for effective energy and thermal management. They found that polynomial and linear regression models are the most accurate for predicting server power consumption, and argued that future models should incorporate the interdependence between temperature and electricity consumption in DCs. In another example, Saxena et al. [26] applied regression methods to predict power management in Azure Virtual Machines and found that hybrid ensemble models and modified neural networks are better for highly diversified workloads. In addition, Lin et al. [16] evaluated several ML models and found that tree-based models, including XGBoost and LightGBM, delivered superior accuracy for temperature prediction in air-cooled DC simulations, while Tabrizchi et al. [30] found that modified convolutional and long-short-term neural networks can make accurate temperature predictions in DCs. Other studies have used trained ML models for downstream optimization tasks. For instance, Yang et al. [32] used different ML models (e.g. NNs, LightGBM, Random Forest, and Recurrent NNs) to predict the Power Use Effectiveness of DCs and later used them to determine the optimal set point of the condenser water of the chiller that cools down the DC. They estimated that their approach can save 1500 MWh of energy per year in a real DC.

These studies yield promising results in terms of predictive performance under observational settings. However, they all rely on ML approaches that do not account for causality or the effects of interventions within the physical system. As a result, such models may struggle to accurately predict intervention outcomes or scenarios not covered in the training data.

2.2 Causal Machine Learning

Causal Machine Learning (causal ML) can overcome the shortcomings of traditional correlational ML by emphasizing the identification of actual causal mechanisms that drive the data-generating process [18]. A cornerstone of causal ML is the use of Graphical Causal Models (GCMs), which depict causal relationships within a system through Directed Acyclic Graphs (DAGs) [18]. In these graphs, nodes signify variables, while directed edges represent causal effects between them.

Causal ML differs from conventional ML by selecting features with a direct causal effect on the target, reducing confounding bias [33]. Subsequently, non-parametric ML models can accurately model the causal relationship, accounting for non-linearities and interactions [18, 19].

Graphical Causal Models (GCMs) can be derived either from expert domain knowledge or inferred with causal discovery methods. In this paper, we focus on the latter, leveraging constraint-based causal discovery techniques that identify causal structure based on conditional independence between the variables. In our experiments, we use a refined version of the widely recognized Peter-Clark (PC) algorithm [29], tailored for time series data (i.e., data with lags) and referred to as PC1. This adaptation concentrates on iterative independence tests for the most relevant lagged variables [3, 25]. Also, we use the Peter-Clark Momentary Conditional Independence (PCMCI) method, which enhances the process by better addressing autocorrelation in time series data [25].

It is important to note that causal discovery methods for time series rely on critical assumptions such as stationarity, meaning that statistical and causal properties remain constant over time, or causal sufficiency, which assumes no hidden confounders. These assumptions are difficult to satisfy in practice, often causing false positive or negative causal links [1, 9]. Nonetheless, causal discovery methods can select variables with the best indication of causal effects, leading to more robust models [33].

3 METHODOLOGY

We use causal discovery methods from the Python library Tigramite [23] (time-series graph-based measures of information transfer), which is designed for time-series data and identifies causal relationships based on conditional independence tests. Because such methods assume stationarity in the input time series, we first assess each series using the Augmented Dickey-Fuller test from the Statsmodels library [28], adopting a p-value cutoff of 0.01. When non-stationarity is detected, we transform the series via differencing (subtracting each value from its immediate predecessor). The causal discovery methods we used are the PC1 and PCMCI algorithms, explained in Section 2.2. They deliver a causal graph of the system (an example of a causal graph is shown in Section 4.2). A combination of a variable and a lag is a *feature*, where a lag refers to the value of the variable at a previous time step. For the training of **Causal** ML models, we use only the variable and the corresponding time lag that has a connection with the target variable in the causal graph.

To ensure a comprehensive comparison with other traditional methods, we include a baseline model that uses **All** features as well as several conventional feature selection algorithms from the scikit-learn library [21]. These include Recursive Feature Elimination **RFE**, which iteratively removes the least important features using LinearRegression; Principal Component Analysis **PCA**, which reduces redundancy by transforming and selecting features that capture at least 85% of the explained variance; tree-based selection **Tree**,

which identifies important features via RandomForestRegressor and retains those with importance scores above zero; and Lasso regression **Lasso** with an alpha of 0.1, favoring variables with non-zero coefficients after L1 regularization.

The features (both causal and traditional) are then used in the modeling phase. We use for modeling several regression models, including Linear Regression **LR**, ElasticNet **Enet**, Multilayer Perceptron Regressor **MLP** from the scikit-learn package [21], XGBRegressor **XGB** from the Xgboost library [5], and LGBMRegressor **LGBM** from the LightGBM library [14]. For all models except Linear Regression, we run random hyperparameter search and three-fold cross-validation. The final evaluation is done on the test set, and we compare the models with the Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE).

4 EXPERIMENTS

4.1 Description of the testbed

The testbed's P&ID is shown in Figure 2. It is divided into two sub-systems: the DC side and the residential building side, each operating an independent water circuit. On the DC side, a water pipeline begins at the heat exchanger (HEX) on the surface of the first Raspberry Pi (Pi), flows to another HEX on a second Pi, and later goes into a water-to-water HEX. At this point, the water temperature is monitored using a water temperature transmitter 1 (TT1). The HEX transfers the waste heat from this water loop to the second one. After passing through the HEX, the water temperature is measured again using TT2. The water then flows through a water-to-air HEX, which is cooled by a fan, before returning to a water tank. From there, it is pumped back to the Raspberry Pi, completing the circuit. The water in this circuit is heated by the CPUs of the Raspberry Pis' (heat source) and can be cooled down by the water-to-water HEX or by the water-air HEX with the fan (heat sink).

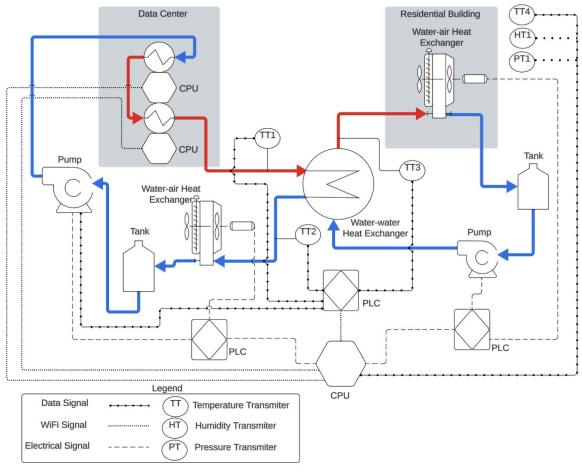

Fig. 2. Piping and Instrumentation Diagram (P&ID) of the data-center testbed, where water-cooled Raspberry Pi's generate heat, which is transferred via a heat exchanger for heating in a separate water loop.

In our experimental setup, we want to investigate the waste heat potential and, therefore, turn off the water-to-air HEX in the data center circuit in most settings, transferring as much heat as possible to the residential building circuit through the water-to-water HEX.

On the residential building side, the water temperature is measured at TT3 before entering a water-to-air HEX, which is also cooled by a fan. The water-to-air HEX emulates a heating system in a household that draws different amounts of heat from the water. The water then flows into the tank, passes through a pump, and is directed back into the HEX. The water in this circuit is heated up by the water-to-water HEX and cooled down with the water-to-air HEX and the fan (heat sink).

Also, an air temperature transmitter (TT4) is installed outside the building to monitor the external weather conditions. Similarly, we installed a pressure sensor (PT1) and a humidity sensor (HT1), not because we expect a strong influence to the experimental setup, but to acknowledge that datasets often include extraneous variables that may be unnecessary and could introduce confounding bias in data-driven modeling.

4.2 Data Description and Causal Discovery

Table 1 provides an overview of the variables in the system, including their mean values from the first training dataset. Our testbed continuously produces data, which we aggregate into 5-second intervals. Note that a feature is a combination of a variable and a corresponding time lag (e.g., water temperature in the house lagged by 3 timesteps); we use 4 as maximum lag, so the total number of available features is 52 (13 variables multiplied by 4 lags). Temperatures are reported in degrees Celsius, pressure in hectopascals, and relative humidity in percent.

Table 1. Description of the variables in the experiments

Variable	Description	Mean
cpu_temp_1	Pi 1 CPU (C°)	33.9
cpu_temp_2	Pi 2 CPU (C°)	31.5
env_humidity	Room Humidity (HT1)	31.9
env_pressure	Room Pressure (PT1)	1006
env_temp	Room Temperature (C°) (TT4)	23.7
dc_fan	DC fan state (%)	0.0
dc_pump	DC pump state (%)	73.0
house_fan	House fan state (%)	72.3
house_pump	House pump state (%)	100.0
stress_ctrl	Stress control CPU (%)	24.5
water_temp_in_HEX	Water in HEX (C°) (TT1)	21.9
water_temp_out_HEX	Water out HEX (C°) (TT2)	21.6
water_temp_house	House water (C°) (TT3)	21.7

We generated five datasets from experiments conducted on different dates, each performed under distinct environmental temperature conditions. In these experiments, actuator configurations, used as system interventions, were randomly changed. The amount of heat input into the system corresponds to the flow rate (DC water loop speed) and the temperature difference before and after the CPUs. High CPU loads result in higher energy inputs into the water circuit. The heat transferred to the house circuit via a heat exchanger is extracted with varying intensity, depending on the fan speed, using an air-water heat exchanger.

In our experiments, we aimed to replicate a range of real-world scenarios, including full waste heat utilization, the necessity to additionally cool the DC, fluctuating heat demand, changing environmental conditions, and different levels of data center utilization. Experiment 1 tested diverse CPU loads, house heating patterns, and DC water loop speeds. Experiment 2 maintained full house heating and water loop speed, varying CPU loads, and introduced multiple environmental temperature drops. Experiment 3 replicated Experiment 2 but with more stable environmental temperatures. Experiment 4 introduced variability across CPU loads, water loop speeds, house heating and occasionally activating the data center's HEX fan. Experiment 5 is similar to Experiment 1 but features a rising environmental temperature.

The datasets generated for these five experiments vary in length, ranging from 5,347 to 8,256 observations. Each observation comprises sensor measurements and current intervention variables[1]. We split the gathered datasets into 70% training and 30% testing sets, ensuring that the training data precedes the test data to prevent temporal leakage. Across most experiments, the setup, coupled with varying environmental temperatures, introduces partly out-of-distribution patterns in the test set.

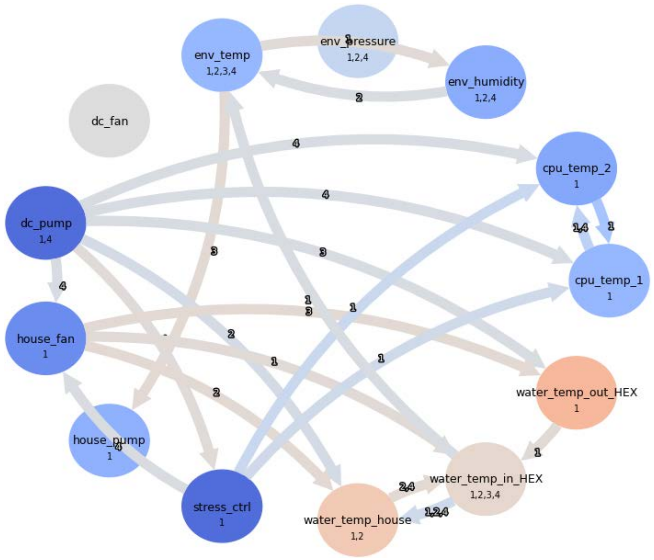

Fig. 3. Stationary DAG representing the causal links between the variables in a testbed´s training dataset

For the causal models, we perform causal discovery on the training set of every experiment. For example, Figure 3 shows an example of the resulting stationary causal graph [24] of the first experiments' train set, using PC1 and an alpha value of 0.05. Nodes represent variables, links indicate causal influences, link colors indicate correlation strength (blue: positive, red: negative), and node colors and numbers indicate autocorrelation. The dc_fan was not activated in this experiment, and therefore has no color or links in the causal graph. The key variable to predict is water_temp_house (water temperature measured on TT3). The graph correctly shows that increasing water_temp_in_HEX and dc_pump speed (which increases water flow and heat exchange) raise water_temp_House, while higher house fan speed decreases it.

However, some links are incorrect: for example, CPU temperatures are not linked in the GCM to water_temp_in_HEX at the DC side, and the environment temperature incorrectly has a link to the house pump motor speed. As discussed in Section 2.2, causal discovery can produce false links due to violated assumptions. While expert knowledge could be used to refine the graph [6], we avoid manual adjustments, as our objective is to assess the performance of automatic causal discovery methods and the resulting models, comparing them to traditional approaches, even though the automatically identified features may include some incorrect causal relationships. For the subsequent modeling using **causal** methods, we selected only the lag variables that had a direct causal link to water_temp_house, as identified by the causal discovery process.

To evaluate all modeling approaches, we first apply both traditional and causal feature selection techniques, followed by training ML models with the resulting features to predict the house's water temperature one step in the future.

5 RESULTS

Our evaluation has two parts: first, we assess all models on the entire test set. Second, we evaluate the models two minutes post-intervention to asses their robustness.

Evaluation 1. The results of each experiment are presented in Table 2 with the feature selection method (Feat. Selec.), the model name, the number of features used (F. Nº), and the metrics. The models are sorted by lowest MAE, where we only showcase the top-performing model for each feature selection method. For a comprehensive overview of all models and feature selection methods (in total 54 models per experiment), please refer to the online appendix.

The Table 2 reveals that, across all experiments, models employing the causal approach for selecting the features consistently demonstrated superior performance, with an average error of 0.034 degrees Celsius and a percentage error of 14.8%. However, the performance difference between these models and the best model with traditional feature selection or all variables ranges from only 0.3% to 14.6%. The PCA feature selection exhibited overall poor performance.

Interestingly, LR was the top performer in nearly all experiments, with the exception of Experiment 4, where tree-based models outperformed the others (RF and LGBM). This difference can be explained by the underlying characteristics of the algorithms: linear regression is capable of extrapolating to values outside the range observed during training, while tree-based models are limited to interpolation within the span of their training data, as they lack leaves for unseen values. As our interventions are random throughout the train and test sets, we sometimes have an actuator´s parameter set that is not exactly present in the train dataset. In Experiment 4, interventions stayed within the training data range, enabling tree-based models to excel.

Evaluation 2. In this evaluation, we consider the models listed in Table 2 and calculate the MAE at each timestep over the two

[1]An online appendix with time-series visualizations, raw data, all experimental results, and the corresponding code for complete reproducibility is available in the repository: https://github.com/zapataunipaderborn/testbed_experiments/

Table 2. Evaluation results for the experiments.

Exp.	Feat. Selec.	Model	F. N°	MAE	MSE	MAPE
1	Causal	LR	20	0.0357	0.0021	0.1581
	Tree	LR	2	0.0375	0.0023	0.1660
	RFE	LR	26	0.0376	0.0023	0.1665
	Lasso	LR	18	0.0409	0.0028	0.1808
	All	LR	52	0.0413	0.0028	0.1823
	PCA	XGB	2	1.1451	1.5389	5.0212
2	Causal	LR	16	0.0329	0.0017	0.1417
	Lasso	LR	16	0.0381	0.0025	0.1639
	Tree	LR	6	0.0408	0.0032	0.1755
	RFE	LR	26	0.0455	0.0047	0.1963
	All	RF	52	0.0518	0.0043	0.2223
	PCA	RF	2	0.5349	0.3802	2.2828
3	Causal	LR	4	0.0319	0.0016	0.1411
	All	LR	52	0.0331	0.0019	0.1458
	Tree	LR	9	0.0358	0.0020	0.1580
	Lasso	LR	16	0.0359	0.0022	0.1585
	RFE	LR	26	0.0517	0.0047	0.2265
	PCA	ENet	2	1.6007	2.7418	7.0543
4	Causal	RF	19	0.0354	0.0020	0.1521
	Lasso	LGBM	19	0.0355	0.0020	0.1528
	Tree	LR	9	0.1289	3.5293	0.5507
	All	RF	52	0.1409	0.0873	0.6034
	RFE	LR	26	0.1621	2.8015	0.6931
	PCA	MLP	3	1.2977	1.7766	5.5705
5	Causal	LR	18	0.0342	0.0019	0.1481
	Tree	LR	9	0.0348	0.0020	0.1509
	All	LR	52	0.0443	0.0031	0.1911
	RFE	ENet	26	0.0506	0.0040	0.2184
	Lasso	LR	17	0.0743	0.0078	0.3192
	PCA	MLP	2	1.1619	1.6752	4.9813

minutes following an intervention, averaging the results across all experiments and interventions. This yields a table of MAE values for each timestep. To present these results in a concise and informative way, we display them in Figure 4. We don't show the results of the PCA feature selection due to its poor performance.

In this scenario, all models perform worse than in the previous evaluation, as predicting outcomes immediately after interventions is inherently more challenging than making predictions across the entire test set. The results indicate that the model trained with the causal feature selection approach yields the best performance. Notably, the difference in performance between the causal models and the rest of the models is substantially larger in this case. Previously, the differences were small, ranging from 0.3% to 14.6%, whereas now they range from 20% to 50%, depending on the lag after the intervention.

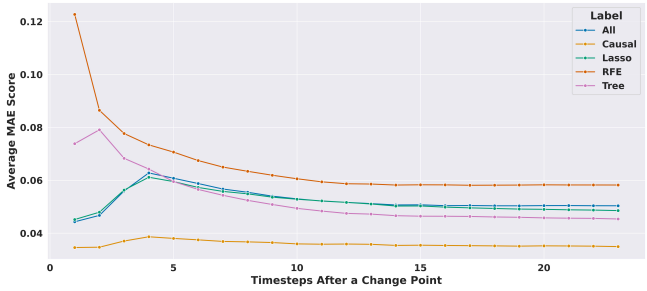

Fig. 4. MAE per time step post-intervention across feature selection methods using the best models from each experiment.

6 DISCUSSION, LIMITATIONS AND FUTURE WORK

By developing a DC testbed with heat recovery capabilities, we created a platform to test and evaluate both conventional ML and causal ML approaches for modeling a proxy for heat recovery potential. Our work builds on the insights of Gamella et al. [8] with a focus on applications in DC. We demonstrate that these environments are essential for advancing ML research, as they allow for robust model testing under real-world scenarios and controlled interventions. This enables a comprehensive evaluation of modeling approaches that can be used to optimize control systems to reduce energy demand and enhance waste energy utilization in data centers.

Currently, most ML applications in the context of DCs rely on observational data from real systems or from simulations [13, 16, 26, 30, 32], and the evaluation of causal approaches relies mainly on data from simulations [6, 12]. Our experiments provide empirical evidence in a controlled environment involving interventions in the real DC testbed. The results show that under observational settings, models trained with causally selected features performed slightly better than those using all available features or those selected through traditional correlation-based methods. Under intervention scenarios, where changes are actively introduced to the system, causal ML approaches significantly outperformed standard ML approaches. These results highlight the importance of causal modeling for understanding and predicting the behavior of sustainable computing systems, where interventions play a crucial role in both optimizing operations and accurately evaluating control strategies in data center environments.

It is important to consider that causal discovery methods may be susceptible to errors in feature selection when their underlying assumptions are violated [1, 9]. Such violations of assumptions are likely inevitable in real-world scenarios, such as in the DC testbed. Nonetheless, causal discovery methods can still effectively identify features resulting in models that perform robustly, especially for predictions following interventions. However, if the objective is to precisely analyze the causal effect of a particular variable on the target, it is essential to further refine the causal model and ensure the correct causal features are included, for example, by adjustments with domain knowledge.

Based on our results, we argue that testing these new causal approaches on real physical systems is essential. ML practitioners

should choose modeling approaches based on the specific task: if the resulting models should be used as the basis for optimization approaches, analyzing interventions, or for out-of-distribution predictions, causal ML approaches can make a significant contribution; for only in-distribution prediction tasks, however, traditional ML models continue to serve as reliable standard tools.

Even though the basic physical processes in real data centers will be similar, the causal graph generated from the testbed cannot be directly transferred because the temporal relationships between the variables in data centers will be different (e.g., how long it takes for heat to travel from the CPUs to the extraction point for waste heat utilization). Nevertheless, DC operators can follow a similar approach to that used in the study: first, identify the causal structure of the data center using relevant measured variables, and only then proceed with machine learning (ML) modeling. As the scale of the DC increases, along with the number of sensors, causal discovery methods can become computationally expensive. To address this, variables may be grouped to reduce complexity; for instance, by averaging CPU loads across all processors in a rack, or aggregating temperature readings from sensors located in proximity within the same water pipeline. Similarly, in large-scale data centers, it is more important to measure the correct variables that have a causal influence on the target variable than to simply increase the number of variables monitored.

Moreover, it is important to conduct controlled interventions on key system variables, such as the cooling system or the valves that regulate water flow to the servers. These interventions should cover the full safe operating range of the system, including extreme and median settings (e.g., fully open, fully closed, and halfway open valve positions). Such systematic interventions provide a strong empirical basis for causal discovery algorithms to accurately determine cause-and-effect relationships. Moreover, while as many interventions as possible should be conducted to improve causal identification, the feasibility depends on factors such as the strength of the causal effect, the level of sensor noise, and the cost of performing the interventions. These trade-offs should be evaluated on a case-by-case basis.

Furthermore, DCs can significantly benefit from more accurate and robust predictive models to support decision-making, scenario evaluation, and control strategies. Since control systems in DCs continuously implement interventions within the system, causal ML models could serve as a foundation for advanced control frameworks such as model predictive control (MPC) or reinforcement learning (RL). The final impact of improved predictions on economic and sustainability metrics, such as Power Usage Effectiveness (PUE) or the fraction of heat recovered, depends on how sensitive the specific application is to prediction errors. For these systems, even small inaccuracies can accumulate over time, potentially resulting in a substantial impact when models are used repeatedly.

Our study has some limitations that future research should address. The effectiveness of causal models depends heavily on the accuracy of the underlying causal graphs. In our experiments, causal discovery performed well, but it might be less reliable when based solely on observational data without interventions. Additional graph editing by domain experts could be necessary. Future work could establish the conditions under which causal discovery methods are trustworthy, for example, investigating the required data quantity, the number of interventions, and sampling frequency. In addition, an analysis can be performed to examine the relationship between the complexity of interventions in the different experiments, such as whether one or multiple variables were changed, and the results of the causal discovery, as well as the overall model performance. Moreover, comparing simplified physics-based models, CFD/HT simulations, and machine learning approaches, including causal and physics-informed ML methods, could offer valuable insights into their complementary strengths and modeling capabilities.

ACKNOWLEDGEMENTS

This study was supported by the Federal Ministry for Environment, Nature Conservation, and Nuclear Safety of Germany under Grant No. 67KI32008B (DC2HEAT - Data center HEat Recovery with AI-Technologies), and we gratefully acknowledge their support.

REFERENCES

[1] Charles K Assaad, Emilie Devijver, and Eric Gaussier. 2022. Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research* 73 (2022), 767–819.

[2] Jayati Athavale, Minami Yoda, and Yogendra Joshi. 2019. Comparison of data driven modeling approaches for temperature prediction in data centers. *International Journal of Heat and Mass Transfer* 135 (2019), 1039–1052.

[3] Tom Beucler, Frederick Iat-Hin Tam, Milton S Gomez, Jakob Runge, Andreas Gerhardus, et al. 2023. Selecting robust features for machine-learning applications using multidata causal discovery. *Environmental Data Science* 2 (2023), e27.

[4] Gianluca Bontempi and Maxime Flauder. 2015. From dependency to causality: a machine learning approach. *J. Mach. Learn. Res.* 16, 1 (2015), 2437–2457.

[5] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[6] Xia Chen, Jimmy Abualdenien, Manav Mahan Singh, André Borrmann, and Philipp Geyer. 2022. Introducing causal inference in the energy-efficient building design process. *Energy and Buildings* 277 (2022), 112583.

[7] Quentin Clark, Fatih Acun, Ioannis C Paschalidis, and Ayse Coskun. 2024. Learning a Data Center Model for Efficient Demand Response. *ACM SIGENERGY Energy Informatics Review* 4, 5 (2024), 98–105.

[8] Juan L Gamella, Jonas Peters, and Peter Bühlmann. 2025. Causal chambers as a real-world physical testbed for AI methodology. *Nature Machine Intelligence* (2025), 1–12.

[9] Uzma Hasan, Emam Hossain, and Md Osman Gani. 2023. A survey on causal discovery methods for iid and time series data. *arXiv preprint arXiv:2303.15027* (2023).

[10] Pei Huang, Benedetta Copertaro, Xingxing Zhang, Jingchun Shen, Isabelle Löfgren, Mats Rönnelid, Jan Fahlen, Dan Andersson, and Mikael Svanfeldt. 2020. A review of data centers as prosumers in district energy systems: Renewable energy integration and waste heat reuse for district heating. *Applied energy* 258 (2020), 114109.

[11] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. 2013. *An introduction to statistical learning*. Vol. 112. Springer.

[12] Fuyang Jiang and Hussain Kazmi. 2025. What-if: A causal machine learning approach to control-oriented modelling for building thermal dynamics. *Applied Energy* 377 (2025), 124550.

[13] Chaoqiang Jin, Xuelian Bai, Chao Yang, Wangxin Mao, and Xin Xu. 2020. A review of power consumption models of servers in data centers. 265 (2020), 114806. doi:10.1016/j.apenergy.2020.114806

[14] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).

[15] Martijn Koot and Fons Wijnhoven. 2021. Usage impact on data center electricity needs: A system dynamic forecasting model. *Applied Energy* 291 (2021), 116798.

[16] Jianpeng Lin, Weiwei Lin, Wenjun Lin, Jiangtao Wang, and Hongliang Jiang. 2022. Thermal prediction for Air-cooled data center using data Driven-based model. 217 (2022), 119207. doi:10.1016/j.applthermaleng.2022.119207

[17] Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. 2020. Recalibrating global data center energy-use estimates. *Science* 367, 6481 (2020), 984–986.

[18] Judea Pearl. 2009. *Causality*. Cambridge university press.

[19] Judea Pearl. 2014. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Elsevier.

[20] Judea Pearl. 2019. The seven tools of causal inference, with reflections on machine learning. *Commun. ACM* 62, 3 (2019), 54–60.

[21] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[22] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms.* The MIT Press.

[23] Jakob Runge. 2018. Causal network reconstruction from time series: From theoretical assumptions to practical estimation. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28, 7 (2018).

[24] Jakob Runge, Andreas Gerhardus, Gherardo Varando, Veronika Eyring, and Gustau Camps-Valls. 2023. Causal inference for time series. *Nature Reviews Earth & Environment* 4, 7 (2023), 487–505.

[25] Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. 2019. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science advances* 5, 11 (2019), eaau4996.

[26] Deepika Saxena, Jitendra Kumar, Ashutosh Kumar Singh, and Stefan Schmid. 2023. Performance Analysis of Machine Learning Centered Workload Prediction Models for Cloud. 34, 4 (2023), 1313–1330. doi:10.1109/TPDS.2023.3240567

[27] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proc. IEEE* 109, 5 (2021), 612–634.

[28] Skipper Seabold and Josef Perktold. 2010. Statsmodels: econometric and statistical modeling with python. *SciPy* 7, 1 (2010), 92–96.

[29] Peter Spirtes, Clark N Glymour, and Richard Scheines. 2000. *Causation, prediction, and search.* MIT press.

[30] Hamed Tabrizchi, Jafar Razmara, and Amir Mosavi. 2023. Thermal prediction for energy management of clouds using a hybrid model based on CNN and stacking multi-layer bi-directional LSTM. 9 (2023), 2253–2268. doi:10.1016/j.egyr.2023.01.032

[31] Philipp Wiesner, Ilja Behnke, Paul Kilian, Marvin Steinke, and Odej Kao. 2023. Vessim: A Testbed for Carbon-Aware Applications and Systems. doi:10.48550/ARXIV.2306.09774 Version Number: 3.

[32] Zhen Yang, Jinhong Du, Yiting Lin, Zhen Du, Li Xia, Qianchuan Zhao, and Xiaohong Guan. 2022. Increasing the energy efficiency of a data center based on machine learning. *Journal of Industrial Ecology* 26, 1 (2022), 323–335.

[33] Kui Yu, Xianjie Guo, Lin Liu, Jiuyong Li, Hao Wang, Zhaolong Ling, and Xindong Wu. 2020. Causality-based feature selection: Methods and evaluations. *ACM Computing Surveys (CSUR)* 53, 5 (2020), 1–36.

[34] Xiaolei Yuan, Yumin Liang, Xinyi Hu, Yizhe Xu, Yongbao Chen, and Risto Kosonen. 2023. Waste heat recoveries in data centers: A review. *Renewable and Sustainable Energy Reviews* 188 (2023), 113777.

[35] Caiqing Zhang, Hongxia Luo, and Zixuan Wang. 2022. An economic analysis of waste heat recovery and utilization in data centers considering environmental benefits. *Sustainable Production and Consumption* 31 (2022), 127–138.

[36] Qingxia Zhang, Zihao Meng, Xianwen Hong, Yuhao Zhan, Jia Liu, Jiabao Dong, Tian Bai, Junyu Niu, and M Jamal Deen. 2021. A survey on data center cooling systems: Technology, power consumption modeling and control strategy optimization. *Journal of Systems Architecture* 119 (2021), 102253.