IoT Anomaly Detection Via Device Interaction Graph: Technical Report

1 Potential Applications

While this work mainly discusses the smart home application and its security issues, we envision that our formalism of DIG and device anomalies can also apply to other IoT applications. For example, in industry IoT [1], a smart warehouse system may have the following business logic. A sensor first detects a low inventory level for an item, and the platform asks a programmed robot to put the item on an autonomous truck. Then the truck delivers the item to the warehouse. The above logic specifies an interaction chain Sensor $\rightarrow$ Robot $\rightarrow$ Truck, and our DIG is well-suited for describing it. With the DIG, one can further formalize various industrial IoT anomalies according to our anomaly definitions (e.g., robot misbehavior caused by command injection [1] and unsolicited truck movement triggered by robot [2]). As another example, in a water system, multiple sensors are deployed in different places to monitor the water quality [3]. Depending on their locations, sensors interact with each other, and their readings are naturally affected by the upstream sensor readings. A DIG that profiles the sensor network can help to detect water pollution (reflected in abnormal sensor readings) by checking the upstream sensor readings. Moreover, it supports tracking the flow of the polluted water, which is reflected in a sequence of abnormal readings.

2 Complexity Analysis

We here analyze the computational complexity of the Interaction Miner and Event Monitor modules. Interaction Miner involves the skeleton construction process and the CPT estimation process, and the former plays a dominant role in determining the computational complexity. Specifically, since our TemporalPC algorithm follows PC's conditional independence test framework, its computational complexity is of the same order as PC. Let n be the number of nodes in DIG, and k be the maximum degree of any node. The maximum number of conditional independence tests required by TemporalPC is $O(n^k)$ [4]. For Event Monitor, the computational complexity for validating a runtime event is $O(1)$, as it only involves a table lookup for calculating an anomaly score and a comparison operation with a constant threshold. Considering the realistic sources of device interactions, the value of k is usually limited, and the time complexity of our approach can be polynomial. Moreover, there have been attempts [5–7] which aim to improve the scalability of PC. Their solutions can be easily integrated into our TemporalPC algorithm. As a result, besides the smart home, our system can also be applied in other large-scale smart applications where the number of devices is significant, e.g., industrial IoT.

3 Evaluation

In this section, we post our evaluation result on the CASAS testbed.

3.1 Experimental Setup

CASAS collected 42,388 device events during a 30-day living experience, and the device information is shown in Table 1. One can check that most devices are presence sensors, and there is one contact sensor installed on the front door. Similar with the ContextAct testbed, there is no automation rules installed. As a result, we first generate three automation rules: (1) If M002 is activated, unlock D002; (2) If M004 is activated, unlock D002, and (3) If M011 is deactivated, unlock D002. By doing so, 15,911 events which simulate the automation executions are generated. We further preprocess the 58,299 events and partition the generated time series. Eventually, 80% of them are used as the training data (for DIG construction), while 20% are used for testing data (anomaly detection).

For the ground truth construction, since most devices are presence sensors, we mainly check the user movement traces during the 30 days. Then we identify the neighboring sensors, and construct the ground-truth interactions correspondingly. As a result, we identify 54 ground-truth interactions, which record user movement traces.

3.2 Interaction Mining Evaluation

With the training data, CausalIoT initiate the TemporalPC algorithm with $\tau = 2$ and $\alpha = 0.001$. The result shows that it identifies 53 ground-truth interactions with 100% precision and 98.2% recall. In particular, for all the installed three automation rules, CasualIoT successfully identifies them. Moreover, CausalIoT successfully rejects 10 spurious interactions, and all of them stem from the intermediate factor. For example, it rejects the M011 → M003, since M004 is an intermediate node in the path, i.e., M011 → M004 → M003. For the missing interaction D002 → M006, it is due to the low frequency. That is, users seldom move along this path, and as a result, CausalIoT cannot identify the association relationships between the two device states.

3.3 Anomaly Detection Evaluation

With the inferred interactions, we initiate the contextual anomaly detection evaluation. Recall that in our paper, we simulate 4 malicious cases for contex-

Table 1: Overview of device information

Abbr.	Attributes	# devices (CASAS)	# devices (ContextAct)	Value type	Description
M	Presence Sensor	7	5	Discrete	Movement detection
D	Contact Sensor	1	2	Discrete	Door/window state

tual anomaly. Unfortunately, since the CASAS testbed contains limited types of devices, we can only simulate case 2 (Burglar Intrusion) for evaluation purpose. Specifically, we inject 4,000 spoofed events of the presence and contact sensors, and use the precision and the recall as the metric. The evaluation results show that CausalIoT achieves a high detection accuracy with 94.2% precision and 98.8% recall. We also compare with the three baselines. In particular, the precision for Markov/OCSVM/HAWatcher is 35.2%/24.5%/19.7%, and the recall is 54.3%/77.4%/32.9%. One can check that compared with the three baselines, our CausalIoT achieves the best performance.

Finally, we initiate the collective anomaly detection evaluation. Similar with the simulation of contextual anomalies, we only simulate case 1 (Burglar Wandering) for collective anomaly generation. Specifically, we first randomly select 1,000 positions for injection of spoofed contact/presence sensor events, which simulate the case of burglar intrusion. Then we propagate them based on the ground truth, and eventually add 1,000 movement traces to the testing data. The length of these traces is bounded by the parameter $k_{\max}$, and it ranges from 2 to 4. Evaluation results show that when $k_{\max}$ ranges from 2 to 4, the average length of injected anomalies is 2/2.476/2.979. CausalIoT can successfully detect 98.6%, 98.2%, 94.4% anomaly chains. For all the detected chains, CausalIoT can also fully reconstruct them.

4 Related Work

4.1 Causality

The understanding of the causal relationship has evolved from the deterministic functional equations [8–10] to the stochastic functional equations [11, 12] in the last century. Specifically, [12] proposed a *structural causal model* which uses structural equations to describe the causal relationship among variables. Researchers also base on the bayesian network [13] and propose a graphical representation of the model, i.e., the causal bayesian network (or causal graph for simplicity). Compared with the bayesian network, the edges in the causal graph describe more complex relationships among variables: They encode both the conditional dependence relationship and causal relationship in terms of intervention [14]. Due to its transparent and interpretable representation of causal relationships, the structural causal model has been widely used in economics [15, 16], epidemiology [17, 18], and social sciences [19, 20]. Our work makes a first step towards utilizing the causal model in the IoT ecosystem to interpret the prevalent device interactions.

Prior works propose various methods to infer the causal model [4, 21–26]. Specifically, constraint-based methods aim to efficiently search for a causal graph [4, 21, 22], and they can encode various independence test methods to handle different types of data/dependence formats. Equation-based methods leverage the noise term in the structural equations and test asymmetry between the causal and effect [23, 24]. However, they cannot identify the causal relationship without any assumption on the function [27]. Generally speaking, any methods described above can be extended to handle the temporal setting. Considering their benefits and popularity, we extend one of the most famous constraint-based methods, the PC algorithm, and use it for causal discovery for device interaction mining.

4.2 Detection of Smart Home Security Threats

Many prior trials focus on detecting threats in specific IoT applications and functionalities. In particular, [28–36] extensively study the security issues of the automation rules. However, their methods largely depend on the accessibility of the source codes, and cannot handle the case where the automation rules come from multiple platforms and each platform uses different programming languages. Instead, CausalIoT uses a data-driven approach to adequately address the challenge of automation inference. Moreover, our work can detect and track unsolicited automation executions at runtime, which these works cannot address.

[37–39] leverage the idea of "contextual integrity" and design a set of security policies for device access control. Specifically, they highlight that one can use external conditions (e.g., user location or the state of other devices) to justify a device operation. Unfortunately, they require significant human efforts (e.g., user survey or runtime application prompt) to infer the user preference for device operations. Our work focuses on studying the inter-device relationship for profiling normal device behavior. We use the causal primitive to describe the device interaction, and we design a TemporalPC algorithm for automatic DIG construction. It can also handle more sophisticated security threats which are not considered by prior work (e.g., unsolicited interaction execution).

Finally, many anomaly detection systems utilize data mining techniques to profile the normal system behavior [40–44]. The main difference between these detectors and our work is that we exploit the nature of the widespread device interactions and use them to profile legitimate smart home usage. As a result, our work achieves better detection accuracy. Moreover, the knowledge of device interactions can provide helpful information for explaining the detection result, which was unsupported by prior work.

5 Discussion

While the evaluation result shows that CausalIoT outperforms the existing work, we consider it the first step towards interaction-aware anomaly detection at smart homes. As a result, we list some limitations that we plan to address.
Unmeasured confounder. In Section VI, we highlight that some unmeasured factors (e.g., the sunrise) come from the environment. These factors result in spurious interactions, which eventually reduce the detection accuracy and interpretation capability. The best solution is collecting environmental information about smart homes (e.g., the weather). Since CausalIoT can be easily extended, we can introduce additional nodes that represent the environmental factors, and study their relationship with the device. By doing so, the generated device interaction graph will encode more fruitful information, and the inferred interactions among devices will be more accurate.
Stationary assumption. While TemporalPC achieves good detection accuracy, it is initiated in an offline way. The inferred device interaction graph only describes the interactions during the collection period, and the stationary assumption is needed for applying the graph to handle the runtime device event. As a result, it is susceptible to the scenario where the interaction graph changes. For example, there may be behavioral deviations, or users may add/remove a device. While CausalIoT can use a new training dataset collected within a short period and efficiently re-train the interaction graph, other alternatives are worth exploiting, e.g., runtime inference of the interaction graph.
Multi-user scenarios. Finally, our work focuses on the interaction graph construction in a single-user setting. However, when extending to the multi-user scenario, the graph construction can be complicated as different users may have different preferences about smart home usage. Even worse, some preferences may be conflicted (e.g., user A deploys an automation rule “Turn on the light when the TV is on”, while user B deploys a conflicted rule “Turn off the light when the TV is on”). Some auxiliary information (e.g., temporal information) may help to distinguish the device usage for different users and assist in the interaction mining for each user.

References

[1] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero, “An experimental security analysis of an industrial robot controller,” in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 268–286.

[2] H. P. Breivold and K. Sandström, “Internet of things for industrial automation–challenges and technical solutions,” in *2015 IEEE International Conference on Data Science and Data Intensive Systems*. IEEE, 2015, pp. 532–539.

[3] F. Giannoni, M. Mancini, and F. Marinelli, “Anomaly detection models for iot time series data,” *arXiv preprint arXiv:1812.00890*, 2018.

[4] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman, *Causation, prediction, and search.* MIT press, 2000.

[5] P. Bühlmann, M. Kalisch, and M. H. Maathuis, “Variable selection in high-dimensional linear models: partially faithful distributions and the pc-simple algorithm,” *Biometrika*, vol. 97, no. 2, pp. 261–278, 2010.

[6] J. Ramsey, J. Zhang, and P. L. Spirtes, “Adjacency-faithfulness and conservative causal inference,” *arXiv preprint arXiv:1206.6843*, 2012.

[7] M. Kalisch and P. Bühlman, “Estimating high-dimensional directed acyclic graphs with the pc-algorithm.” *Journal of Machine Learning Research*, vol. 8, no. 3, 2007.

[8] S. Wright, “Systems of mating. i. the biometric relations between parent and offspring,” *Genetics*, vol. 6, no. 2, p. 111, 1921.

[9] T. Haavelmo, “The statistical implications of a system of simultaneous equations,” *Econometrica, Journal of the Econometric Society*, pp. 1–12, 1943.

[10] O. D. Duncan, *Introduction to structural equation models.* Elsevier, 2014.

[11] J. Pearl, “Causal diagrams for empirical research,” *Biometrika*, vol. 82, no. 4, pp. 669–688, 1995.

[12] ——, “Causal inference in statistics: An overview,” *Statistics surveys*, vol. 3, pp. 96–146, 2009.

[13] R. E. Neapolitan *et al.*, *Learning bayesian networks.* Pearson Prentice Hall Upper Saddle River, NJ, 2004, vol. 38.

[14] J. Pearl, “Theoretical impediments to machine learning with seven sparks from the causal revolution,” *arXiv preprint arXiv:1801.04016*, 2018.

[15] A. Abadie, A. Diamond, and J. Hainmueller, “Synthetic control methods for comparative case studies: Estimating the effect of california’s tobacco control program,” *Journal of the American statistical Association*, vol. 105, no. 490, pp. 493–505, 2010.

[16] J. D. Angrist and J.-S. Pischke, “The credibility revolution in empirical economics: How better research design is taking the con out of econometrics,” *Journal of economic perspectives*, vol. 24, no. 2, pp. 3–30, 2010.

[17] M. A. Hernán and J. M. Robins, “Using big data to emulate a target trial when a randomized trial is not available,” *American journal of epidemiology*, vol. 183, no. 8, pp. 758–764, 2016.

[18] T. J. VanderWeele and P. Ding, “Sensitivity analysis in observational research: introducing the e-value,” *Annals of internal medicine*, vol. 167, no. 4, pp. 268–274, 2017.

[19] G. Imbens, “Instrumental variables: an econometrician’s perspective,” National Bureau of Economic Research, Tech. Rep., 2014.

[20] C. F. Manski, “Identification problems in the social sciences,” *Sociological Methodology*, pp. 1–56, 1993.

[21] P. Spirtes and C. Glymour, “An algorithm for fast recovery of sparse causal graphs,” *Social science computer review*, vol. 9, no. 1, pp. 62–72, 1991.

[22] D. Colombo, M. H. Maathuis *et al.*, “Order-independent constraint-based causal structure learning.” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3741–3782, 2014.

[23] S. Shimizu, P. O. Hoyer, A. Hyvärinen, A. Kerminen, and M. Jordan, “A linear non-gaussian acyclic model for causal discovery.” *Journal of Machine Learning Research*, vol. 7, no. 10, 2006.

[24] P. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf, “Nonlinear causal discovery with additive noise models,” *Advances in neural information processing systems*, vol. 21, 2008.

[25] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, “Toward causal representation learning,” *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.

[26] H. Mao, H. Liu, J. X. Dou, and P. V. Benos, “Towards cross-modal causal structure and representation learning,” in *Machine Learning for Health*. PMLR, 2022, pp. 120–140.

[27] K. Zhang, Z. Wang, J. Zhang, and B. Schölkopf, “On estimation of functional causal models: general results and application to the post-nonlinear causal model,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 7, no. 2, pp. 1–22, 2015.

[28] Z. B. Celik, P. McDaniel, and G. Tan, “Soteria: Automated iot safety and security analysis,” in *2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18)*, 2018, pp. 147–158.

[29] Z. B. Celik, G. Tan, and P. D. McDaniel, “Iotguard: Dynamic enforcement of security and safety policy in commodity iot.” in *NDSS*, 2019.

[30] Q. Wang, P. Datta, W. Yang, S. Liu, A. Bates, and C. A. Gunter, “Charting the attack surface of trigger-action iot platforms,” in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 1439–1453.

[31] M. Surbatovich, J. Aljuraidan, L. Bauer, A. Das, and L. Jia, "Some recipes can do more than spoil your appetite: Analyzing the security and privacy risks of ifttt recipes," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 1501–1510.

[32] D. T. Nguyen, C. Song, Z. Qian, S. V. Krishnamurthy, E. J. Colbert, and P. McDaniel, "Iotsan: Fortifying the safety of iot systems," in *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*, 2018, pp. 191–203.

[33] H. Chi, Q. Zeng, X. Du, and J. Yu, "Cross-app interference threats in smart homes: Categorization, detection and handling," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).* IEEE, 2020, pp. 411–423.

[34] W. Ding and H. Hu, "On the safety of iot device physical interaction control," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 832–846.

[35] W. Ding, H. Hu, and L. Cheng, "Iotsafe: Enforcing safety and security policy with real iot physical interaction discovery," in *the 28th Network and Distributed System Security Symposium (NDSS 2021)*, 2021.

[36] M. O. Ozmen, X. Li, A. Chu, Z. B. Celik, B. Hoxha, and X. Zhang, "Discovering iot physical channel vulnerabilities," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022.

[37] W. He, M. Golla, R. Padhi, J. Ofek, M. Dürmuth, E. Fernandes, and B. Ur, "Rethinking access control and authentication for the home internet of things (iot)," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 255–272.

[38] S. Manandhar, K. Moran, K. Kafle, R. Tang, D. Poshyvanyk, and A. Nadkarni, "Towards a natural perspective of smart homes for practical security and safety analyses," in *2020 IEEE Symposium on Security and Privacy (SP).* IEEE, 2020, pp. 482–499.

[39] Y. J. Jia, Q. A. Chen, S. Wang, A. Rahmati, E. Fernandes, Z. M. Mao, A. Prakash, and S. Unviersity, "Contexlot: Towards providing contextual integrity to appified iot platforms." in *NDSS*, vol. 2, no. 2, 2017, pp. 2–2.

[40] J. Choi, H. Jeoung, J. Kim, Y. Ko, W. Jung, H. Kim, and J. Kim, "Detecting and identifying faulty iot devices in smart home with context extraction," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).* IEEE, 2018, pp. 610–621.

[41] P. A. Kodeswaran, R. Kokku, S. Sen, and M. Srivatsa, "Idea: A system for efficient failure management in smart iot environments," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, 2016, pp. 43–56.

[42] S. Munir and J. A. Stankovic, "Failuresense: Detecting sensor failure using electrical appliances in the home," in *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems.* IEEE, 2014, pp. 73–81.

[43] A. K. Sikder, H. Aksu, and A. S. Uluagac, "6thsense: A context-aware sensor-based attack detector for smart devices," in *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 2017, pp. 397–414.

[44] K. Kapitanova, E. Hoque, J. A. Stankovic, K. Whitehouse, and S. H. Son, "Being smart about failures: assessing repairs in smart homes," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 51–60.