Secure Antifragile Manufacturing

Earl T. Barr Jenschwich Charoenchai Steve Hailes

Manufacturing is increasingly lean and flexible. To achieve even greater flexibility, we first describe the core of a research agenda that applies Chaos Engineering and Machine Learning (ML) to solve scheduling problems in manufacturing. This creates a new attack surface; we outline its security implications and introduce mitigations. We close with an opportunity: the use of Chaos engineering will not only improve scheduling performance, but will improve cybersecurity in manufacturing.

1 Antifragile Manufacturing

Flexible manufacturing systems (FMS) must make many logistical decisions, like routing and ordering inputs in real time, to keep a factory operating efficiently. The current state of practice is to cast these decisions as an optimisation problem, using techniques from operations research (OR), and build a bespoke model for each factory. Formulating, testing, and deploying these models are costly. Of the logistical challenges FMS poses, production scheduling is one of the most important [6]. Just in Time (JIT) delivery reduces inventories and buffer cost, but complicates scheduling and exacerbates the problem of running out of parts. OR scheduling models tend not to effectively handle disturbances in complex environments due to their computational cost [5]. Thus, OR modelers must assume the production process to be stationary, and resort to buffers to handle dynamicity [14], delivering less than ideal JIT performance.

Remanufacturing is a form of FMS that exacerbates the scheduling problem. To improve sustainability and reduce environmental impact, remanufacturing recycles inputs. Thus, remanufacturing requires scheduling models that incorporate disassembly and assembly stages while maintaining low inventory [13]. Worse, the arrival times and quantity of recycled inputs are more variable than traditional inputs [15]. Here, too, buffers ameliorate this problem but, once again, conflict with the JIT principle.

An *antifragile* system [24] is one that not only handles, but thrives on and exploits, rare events, like accidents or shortages, by recovering faster than its competitors. Inspired by antifragility, chaos engineering — a new type of reliability engineering — builds antifragile systems by simulating rare events, even injecting them into production systems [19]. Many IT firms have successfully integrated chaos engineering into their development lifecycle. Netflix, in particular, randomly terminates virtual-machine instances, in production, to force their engineers to build resiliency into their service [3]. Netflix is leveraging the fact that their traffic is bursty, like all web traffic, so their service has spare capacity to tolerate injecting errors in production. Factories are rarely ever built with spare capacity, so we cannot introduce faults into factories: the cost is prohibitive. We use high fidelity simulation instead to apply chaos engineering to manufacturing. To achieve the required fidelity, we will work with domain experts to build a probability distribution over FMS events. Our reliance on simulation does not introduce a new dependency or uncertainty into manufacturing: manufacturers have long used simulation to validate and test their processes [12], notably the OR scheduling models we seek to replace.

The frequency distribution of FMS events is skewed and fat-tailed. The tail is fat enough that, even though specific failures are rare, rare failures themselves are surprisingly common [26, 29]. To sample rare events, we will do two things. First, we will work with domain experts to approximate the frequency of tail events and model their effects, understand which common events to remove from consideration, and re-normalise the distribution. The resulting tail-only distribution will only contain anticipated rare events. Second, we will augment the distribution to model unknown events. We will use smoothing to reallocate probability from our renormalised tail to UNK, an event that represents all unknown events. UNK's effects will be a random subset of all known effects, *i.e.* those already known to the model. Example effects include production unit or conveyor belt failures. For tractability, we assume that these effects are independent by combining co-occurring real world events into a single event, if necessary. We cannot check, because we cannot know, whether a random subset of events we generate is possible, but reaching our goal — system-wide resilience — is aided by training a controller to handle even impossible event sets. We will return to our domain experts to elicit UNK's probability and its distribution over effects.

An antifragile FMS must be able to assess and react to FMS events, including the rare events we inject, so we will turn to Reinforcement learning (RL), the ML approach to the fundamental problem of optimising the outcome of a sequence of decisions [30]. Unlike model-based methods, model-free RL learns from its environment. A key promise of RL is its ability to react to dynamic phenomena and disturbances [25, 18, 10]. We will train our scheduler as an RL agent against our FMS simulator, which will alternate between its high fidelity and a rare event injection modes. The rare event mode is what makes our approach chaos engineering: it will hothouse our scheduler in an environment in which rare events are not so rare. After sampling a rare event, our simulator will return to its high fidelity mode so our scheduler can recover. Our RL scheduler will not make unilateral decisions: we will train it to ask factory operators to choose or confirm actions. Combining RL with chaos engineering will build an antifragile FMS that reliably reduces buffering costs. Unlike traditional schedulers, whose model must periodically be manually updated [4], our RL scheduler will learn to tolerate and recover from rare production events by changing the production schedule on the fly, while still matching existing scheduler performance on common events.

2 Securing Antifragile Manufacturing

Integrating ML into manufacturing will introduce new attack surfaces — both the standard security vulnerabilities of ML and those new to its application in manufacturing — and the stakes are high: factories are extremely expensive, full of fragile equipment, and often contain toxic substances or destructive potential. By gaining access to devices in the production process, adversaries can perturb, delete or delay signals to manipulate ML behaviour. An adversary might create, amplify, alter, or obfuscate faults. Their intention could be to damage a plant, people or products, or to slow production. Concretely, the adversary could, say, steal up to date information about a company's production decisions [21] and sell it to competitors. As another example, the adversary could launch a ransomware attack on a factory, raising the stakes over current ransomware attacks. Indeed, the stakes are so high that nation states are adversaries here. In 2014, adversaries caused massive damage by controlling a blast furnace of a German steel factory [27]. In 2010, Stuxnet attacked Bushehr, a nuclear power plant, and set back Iran's nuclear programme for years [17]; attacks on Iranian nuclear power plants are ongoing [9].

To the usual panoply of software and network vulnerabilities, Cyber-physical systems (CPS), such as a traditional production system, add three new attack surfaces: sensors, actuators, and controllers, as well as the networks that interconnect them [2, 17, 27]. The attack surface of the production control system is what separates CPS from purely digital systems. Integrating ML into FMS adds still more attack surfaces. Training data is a conventional ML attack surface [8]. Because training ML requires a lot of data, poisoned inputs are hard to detect. During operation, ML systems are susceptible to adversarial examples [11]. Some ML models do not explain or justify their decisions; this is the explainability problem [20]. Its pernicious consequence for cybersecurity is that it can make it harder for administrators to realise that an ML-augmented system is under attack or has been exploited. Coupling ML to FMS means that ML vulnerabilities permit the adversary to take physical actions.

To launch an attack through these surfaces severally or in combination, an adversary needs to first gain access. The attacker may either obtain direct physical access, use an insider to launch an insider attack [23], attack the supply chain of the FMS, or commit a cyber attack if the FMS is connected to the internet. After gaining access, adversaries can physically interfere with actuators. They can tamper with accessible sensors [2] or replace them with a malicious one using the supply chain [28]. Insiders can carry malware in a device like a USB stick or a mobile phone and then plant it into a controller, similarly to the attack in the Stuxnet case. With an ML-enhanced FMS (ML-FMS), adversaries can introduce physical adversarial examples [22] in the view of sensors to fool ML. They can attach an adversarial sticker to a product to deceive a detection camera. Adversaries can also poison ML training data using insiders. To attack from the outside, they can disguise themselves as third party service providers, like a training framework provider, or attack the providers directly to spoil the training framework [21].

In light of these new attack surfaces, a natural question is "Is the risk of augmenting FMS with ML worth the benefit?". We believe the answer is yes, for three reasons. First, enhancing an FMS with ML has huge potential economic gains. A production process is a dynamic, rapidly evolving environment, where the unexpected often happens in practice [26, 29]. ML models are built to generalise and handle unseen events. Thus, integrating ML into a production system promises a system that can effectively react to unanticipated disturbances. An ML-FMS may, therefore, allow a company to undercut its competitors by reducing the cost of buffers by optimising production process and by achieving higher reliability. Second, ML-FMS is more resistant than general CPS to attacks that exploit some of the vulnerabilities above. For example, adversaries can easily construct adversarial inputs, such as incorrect traffic signs, for CPS like autonomous vehicles, whose sensors interact with an open world. A factory's sensors, in contrast, take data from spaces that are physically secured. Because, in general, ML-FMS are trained in controlled settings, these surfaces are hard to reach. Third, the risks that ML brings can be effectively mitigated, as we detail next.

Akhtar *et al.* and Chakraborty *et al.* detail three effective defence mechanisms [1, 7]: 1) training with perturbed data to build a robust model; 2) modifying the architecture to build a robust model; and 3) using other models to detect irregularities in the vulnerable model. The first defence is self-explanatory; An example for the second mechanism is injecting noise to the neural network layer to shift the classifier boundary to defend against adversarial inputs [16]. ML-FMS falls in the last category. The very simulator we use to train our ML-FMS allows us to detect sensor irregularities: we can run it, out of band, in parallel to the production system, and compare its results to those of the production sensors. Chaos engineering allows us to go further: the antifragility of antifragile ML-FMS naturally extends to attacks. We propose to do just that — build a secure antifragile ML-FMS that can actively resist attacks. Instead of generating rare events that are pertinent to general operation, we will use Chaos engineering to inject cyber attacks and security-relevant rare events into the simulation. In this way, we will train a secure antifragile ML-FMS that identifies and mitigates cyber security attacks.

References

[1] Akhtar, N., Mian, A.: Threat of adversarial attacks on deep learning in computer vision: A survey. Ieee Access **6**, 14410–14430 (2018)

[2] Ashibani, Y., Mahmoud, Q.H.: Cyber physical systems security: Analysis, challenges and solutions. Computers & Security **68**, 81–97 (2017)

[3] Basiri, A., Behnam, N., De Rooij, R., Hochstein, L., Kosewski, L., Reynolds, J., Rosenthal, C.: Chaos engineering. IEEE Software **33**(3), 35–41 (2016)

[4] Bradley, S.P., Hax, A.C., Magnanti, T.L.: Applied mathematical programming. Addison-Wesley (1977)

[5] Buxey, G.: Production scheduling: Practice and theory. European Journal of Operational Research **39**(1), 17–31 (1989)

[6] Buzacott, J.A., Yao, D.D.: Flexible manufacturing systems: a review of analytical models. Management science **32**(7), 890–905 (1986)

[7] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., Mukhopadhyay, D.: Adversarial attacks and defences: A survey. arXiv preprint arXiv:1810.00069 (2018)

[8] Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)

[9] Corera, G.: Iran nuclear attack: Mystery surrounds nuclear sabotage at natanz. https://www.bbc.com/news/world-middle-east-56722181, accessed: 2021-05-17

[10] Gabel, T., Riedmiller, M.: Adaptive reactive job-shop scheduling with reinforcement learning agents. International Journal of Information Technology and Intelligent Computing **24**(4), 14–18 (2008)

[11] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)

[12] Gosavi, A., et al.: Simulation-based optimization. Springer (2015)

[13] Guide Jr, V.D.R.: Production planning and control for remanufacturing: industry practice and research needs. Journal of operations Management **18**(4), 467–483 (2000)

[14] Maccarthy, B.L., Liu, J.: Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. The International Journal of Production Research **31**(1), 59–79 (1993)

[15] Morgan, S.D., Gagnon, R.J.: A systematic literature review of remanufacturing scheduling. International Journal of Production Research **51**(16), 4853–4879 (2013)

[16] Nguyen, L., Wang, S., Sinha, A.: A learning and masking approach to secure learning. In: International Conference on Decision and Game Theory for Security. pp. 453–464. Springer (2018)

[17] Peng, Y., Lu, T., Liu, J., Gao, Y., Guo, X., Xie, F.: Cyber-physical system risk assessment. In: 2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. pp. 442–447. IEEE (2013)

[18] Qu, S., Chu, T., Wang, J., Leckie, J., Jian, W.: A centralized reinforcement learning approach for proactive scheduling in manufacturing. In: 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA). pp. 1–8. IEEE (2015)

[19] Rosenthal, C., Jones, N.: Chaos engineering. O'Reilly Media, Incorporated (2020)

[20] Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**(5), 206–215 (2019)

[21] Song, C., Ristenpart, T., Shmatikov, V.: Machine learning models that remember too much. In: Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security. pp. 587–601 (2017)

[22] Song, D., Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramer, F., Prakash, A., Kohno, T.: Physical adversarial examples for object detectors. In: 12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18) (2018)

[23] Stolfo, S.J., Bellovin, S.M., Hershkop, S., Keromytis, A.D., Sinclair, S., Smith, S.W.: Insider attack and cyber security: beyond the hacker, vol. 39. Springer Science & Business Media (2008)

[24] Taleb, N.N.: Antifragile: Things that gain from disorder, vol. 3. Random House Incorporated (2012)

[25] Tesauro, G., Jong, N.K., Das, R., Bennani, M.N.: A hybrid reinforcement learning approach to autonomic resource allocation. In: 2006 IEEE International Conference on Autonomic Computing. pp. 65–73. IEEE (2006)

[26] Tsarouhas, P.H., Varzakas, T.H., Arvanitoyannis, I.S.: Reliability and maintainability analysis of strudel production line with experimental data–a case study. Journal of food engineering **91**(2), 250–259 (2009)

[27] Tuptuk, N., Hailes, S.: Security of smart manufacturing systems. Journal of manufacturing systems **47**, 93–106 (2018)

[28] Urciuoli, L., Männistö, T., Hintsa, J., Khan, T.: Supply chain cyber security–potential threats. Information & Security: An International Journal **29**(1) (2013)

[29] Vineyard, M., Amoako-Gyampah, K., Meredith, J.R.: Failure rate distributions for flexible manufacturing systems: An empirical study. European journal of operational research **116**(1), 139–155 (1999)

[30] Wiering, M., Van Otterlo, M.: Reinforcement learning. Adaptation, learning, and optimization **12**(3) (2012)