"If security is required": Engineering and Security Practices for Machine Learning-based IoT Devices

Nikhil Krishna Gopalakrishna
Purdue University
gopalakn@purdue.edu

Dharun Anandayuvaraj
Purdue University
dananday@purdue.edu

Annan Detti
Purdue University
adetti@purdue.edu

Forrest Lee Bland
Purdue University
fbland@purdue.edu

Sazzadur Rahaman
University of Arizona
sazz@cs.arizona.edu

James C. Davis
Purdue University
davisjam@purdue.edu

ABSTRACT

The latest generation of IoT systems incorporate machine learning (ML) technologies on edge devices. This introduces new engineering challenges to bring ML onto resource-constrained hardware, and complications for ensuring system security and privacy. Existing research prescribes iterative processes for machine learning enabled IoT products to ease development and increase product success. However, these processes mostly focus on existing practices used in other generic software development areas and are not specialized for the purpose of machine learning or IoT devices.

This research seeks to characterize engineering processes and security practices for ML-enabled IoT systems through the lens of the engineering lifecycle. We collected data from practitioners through a survey (N=25) and interviews (N=4). We found that security processes and engineering methods vary by company. Respondents emphasized the engineering cost of security analysis and threat modeling, and trade-offs with business needs. Engineers reduce their security investment if it is not an explicit requirement. The threats of IP theft and reverse engineering were a consistent concern among practitioners when deploying ML for IoT devices. Based on our findings, we recommend further research into understanding engineering cost, compliance, and security trade-offs.

CCS CONCEPTS

• **Computer systems organization** → **Embedded and cyber-physical systems**; • **General and reference** → *Empirical studies*; • **Computing methodologies** → *Machine learning*; • **Security and privacy**;

KEYWORDS

Internet of Things, Machine Learning, Security and Privacy, Cyber-Physical Systems, Embedded Systems, Software Engineering

SERP4IoT 2022, May 2022, Pittsburgh, PA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Nikhil Krishna Gopalakrishna, Dharun Anandayuvaraj, Annan Detti, Forrest Lee Bland, Sazzadur Rahaman, and James C. Davis. 2022. "If security is required": Engineering and Security Practices for Machine Learning-based IoT Devices. In *Proceedings of The 4th International Workshop on Software Engineering Research & Practices for the Internet of Things (SERP4IoT 2022).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The Internet of Things (IoT) paradigm integrates cyber and physical components, connecting devices at the network edge ("Things") to one another and to more powerful resources over the network ("Internet") [15]. There are ~35 billion IoT devices worldwide, projected to double by 2025 [30, 57, 58]. IoT systems can leverage machine learning (ML) [38, 39] to make low-latency intelligent decisions [8, 67]. The resulting intelligent IoT systems could transform many sectors of the economy [42], however, the associated risks are also substantial. To minimize the risks, engineers should adopt ML methods on resource-constrained IoT devices in a secure, privacy-preserving way [16].

Despite the increasing importance of intelligent IoT systems to consumers, industry, and governments, we know relatively little about manufacturers' engineering practices [28, 46, 53]. Concerns about engineering practices are raised by high profile failures, including cyberattacks on waterworks systems leading to poisoned water supply [55], aggressive data collection practices [4, 48] and exploits leading to IoT botnets [1]. Researchers have investigated IoT software defects [46] and security flaws [12, 18, 20–23, 25, 34, 35, 47, 61] from the software perspective using program analysis and failure analysis. Also, researchers have proposed generic models of the secure software development life cycle (SDLC) for the development of ML models and the development of ML-enabled edge devices [28, 53]. However, the challenges of real-world adoption and current industry practices are largely unexplored.

Our goal is therefore to investigate the process of engineering ML-enabled IoT devices in industry. Our general research questions are: *What practices does the industry follow to develop and manage ML-based IoT devices? How is security treated in industry development life cycles?* We investigate these questions in a survey (N=25) and interviews (N=4) with industry practitioners.

Among other findings, our survey respondents and interview subjects emphasized tradeoffs between engineering cost and quality. Market forces reduce the quality and security of IoT products. As one interview subject (P2) said, *"it is a question of if it [better security]*

will be accepted by the market". Larger companies benefit from economies of scale, with in-house ML and security specialists to support IoT products. We also learned that businesses may give up some marketable functionality in order to reduce their risk, *e.g.,* not storing user data on IoT devices. Across several industry sectors, another common worry is the reverse engineering of proprietary ML models.

2 BACKGROUND

This research is motivated by an industry trend towards computing systems with intelligent components at the network edge, and the associated security and privacy implications. Definitions of an "IoT device" vary [56]; we consider devices with sensors and/or actuators, a network connection, and limited resources in memory, power, and computation [33, 66]. Resource-constrained IoT systems combine sensing and communication capabilities with low cost [50, 70].

Engineering process for IoT: Engineering processes for IoT systems are complex because IoT systems are inherently distributed and resource-constrained, and have physical components alongside virtual ones [68]. Figure 1 depicts a generic engineering lifecycle for ML-based IoT systems, which we used to design our study. This lifecycle combines several existing works [2, 28, 53]. In this model, IoT engineering is a five-step iterative process:

Specification: The purpose of the product is defined, perhaps constraining the hardware and software components.

Design: Decisions are made about system architecture, frameworks are selected, and evaluation techniques are chosen.

Development: Design decisions are implemented using development frameworks. The ML model is optimized by tuning hyperparameters, reducing the computational complexity of the model (eg: deep learning-based models), and manipulating network blocks [26, 40]. The implementation targets a hardware profile but not specific devices, to promote portability.

Deployment: The developed solution is deployed to the target hardware. Deployment-time optimizations such as pruning help fit the model into the IoT device constraints [39]. Optimization strategies are standardized, but the parameters vary based on the available resources of the target hardware [53].

Audit: Here the software components have been deployed to the hardware components, and engineers determine whether the system specification is met. Concerns may be raised about performance goals, fault tolerance [31, 59], or security vulnerabilities. Engineers consider traditional threat models as well as those specific to the use of ML. For example, researchers have proposed attacks involving corrupted training data [69] or reverse engineering a model [49].

Security in IoT: Security is a cross-cutting concern for engineered systems [51]. Security is increasingly incorporated throughout the engineering life cycle (Figure 1) [41]. However, IoT developers find security challenging and complicated [46]. Engineering teams feel responsible for security, but often lack a formal security process [9, 45, 63]. Functionality and deadlines are often prioritized over security [14, 24, 43], and adding security to resource-constrained devices penalizes power consumption, latency, and throughput [11, 60].

Although this engineering process model for ML-based IoT development is a promising start, the research community still lacks insight into industry practices. This knowledge gap hinders our understanding of industry-wide problems and challenges towards building and maintaining secure ecosystems. This study is a step towards filling that gap.

3 RESEARCH QUESTIONS

To understand the processes and challenges of engineering secure ML-enabled IoT systems, we posed five research questions across two themes. The first theme explores ML engineering in a resource-constrained context, with implications for IoT system trustworthiness (*e.g.,* affecting security and privacy). The second theme examines cybersecurity practices for these systems.

Theme 1: Applying machine learning on IoT devices

RQ1: What are the common practices for bringing ML to resource-constrained edge devices? (Process model steps 3a-3d)

RQ2: What are the challenges and consequences developers face due to resource limitations in developing ML software for edge devices? (Steps 3a-3d)

Theme 2: Engineering secure IoT systems

RQ3: How do engineers incorporate security into the IoT engineering process? (Steps 1-5)

RQ4: How do engineers reason about trust in ML-based IoT systems? (Step 4)

RQ5: What other factors affect security practices in IoT engineering? (Steps 1-5)

4 METHODOLOGY

Given our research questions, we chose an exploratory methodology [54] — a mixed quantitative and qualitative approach to explore a phenomenon and develop new research questions. We elicited coarse data with a survey, and detailed insights using interviews.

4.1 Survey

Instrument design: We designed a ∼10-minute, 32-question survey instrument aligned with our research questions. We drew on existing literature for seven demographic questions [10, 13], and developed the other questions using best practices in survey design [29]. The initial set of questions were based on our own industry experience working with ML on IoT devices, and then refined through discussion with practitioners. To test validity and length, we administered the survey to two practitioners and further refined it based on their feedback.

Survey distribution: Given the specialized nature of the engineering security practices under consideration, we distributed the survey widely: on the public platforms Reddit, Hacker News, and TowardsAI; through our personal networks via Facebook and LinkedIn; and on our departmental mailing list. We also asked survey respondents to share the link with their colleagues (snowball sampling [36]). The survey was published in the last week of March 2021 and closed after 5 weeks. We incentivized survey participation with a 1-in-50 chance of winning a $50 gift card.

Analysis method: We analyzed the data using reports generated using the Qualtrics platform. We examined the data from each question, aggregated across all participants. In order to have a uniform scale of results, we have represented all the data in the

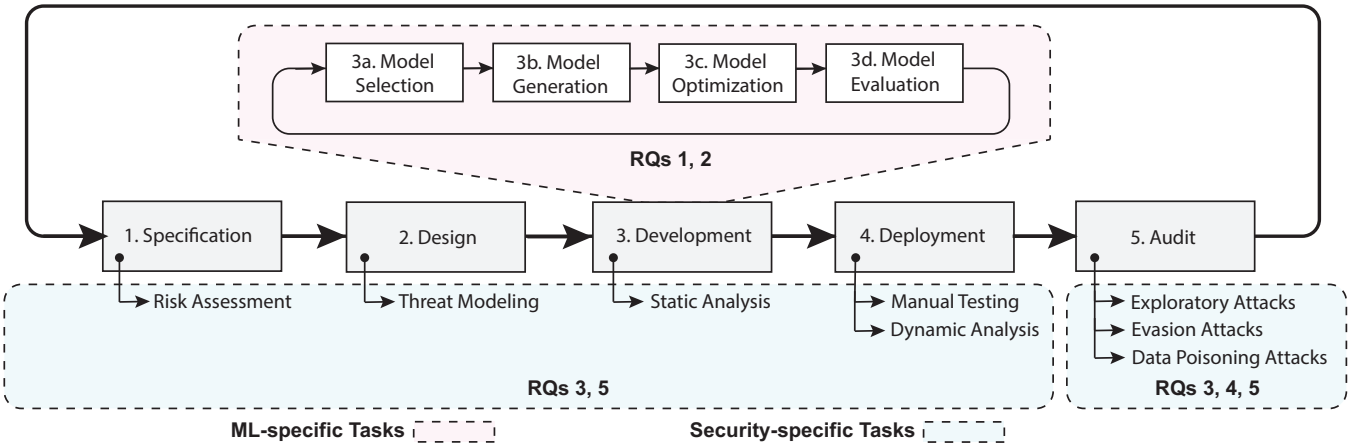

Figure 1: An engineering lifecycle for machine learning-based IoT devices. It combines several models including the SDLC [2, 28, 53].

survey in terms of the percentage of total responses in the diagrams for the purpose of visualization.

4.2 Interviews

Protocol design: We designed our interview protocol as an extension of the survey questions. We observed survey responses and developed questions around areas where the survey respondents disagreed or gave unexpected answers. The interview followed a semi-structured interview, with 8 planned questions to permit a 30-40 minute conversation with each subject [27]. To test validity and length, we piloted the interview protocol with one practitioner.

Participant recruitment: We recruited interviewees from the survey respondent pool. Survey respondents had experience in ML and and IoT engineering, making them good candidates for a longer interview. Survey respondents could indicate if they were interested in a follow up interview, incentivized with a $25 gift card. We contacted all interested respondents, and interviewed any who replied and completed the interview consent form.

Participant privacy: Audio recordings of interviews were transcribed by a third-party service. We anonymized participant PII (*e.g.,* names of people and companies) before analysis.

4.3 Collected data

Survey: We received a total of **25** survey responses, of which **12** were fully completed. Given the few full responses, we also analyzed the available data from partial responses.

Interview: We interviewed 4 experts, with a range of positions and professional experience. The interviews comprised 140 minutes of audio recordings.

5 RESULTS AND ANALYSIS

We present results corresponding to our RQs. To simplify the presentation, we synthesize survey and interview data for each question.

5.1 Demographics

Survey respondents (Figure 2) hold bachelor's degrees in computer science, software engineering, computer engineering, or electrical

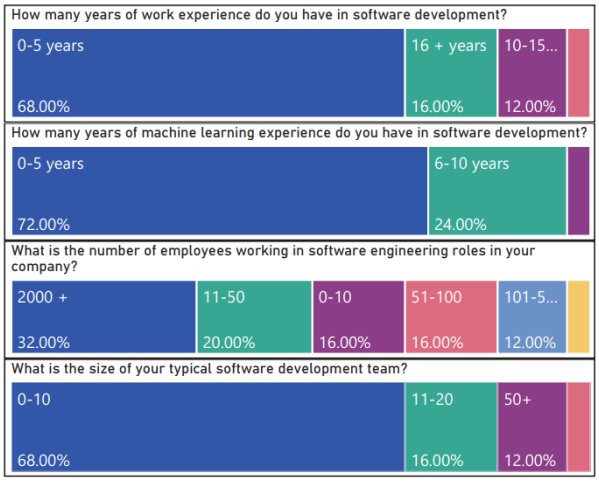

Figure 2: Demographics of survey respondents.

engineering; work primarily in the sectors of consumer electronics (27%), IT & telecommunications (22%), automotive (20%), and healthcare & biomedical (15%); and learned about ML techniques from university coursework (41%), self-taught (37%), and from corporate training (20%). They work at a range of company sizes, from under 50 employees (36%) to over 2,000 (32%). They have a range of experience applying ML in software engineering, ~30% more than 5 years and ~70% fewer. At their companies, they reported an almost equal distribution of ML deployment experience: from initial exploration/prototyping stages to "multiple projects" to extensive multi-platform experience (Figure 3).

Interview subjects (Table 1) had a range of job roles, and experience in sectors including manufacturing, consumer electronics, defense, and medical devices.

5.2 Theme 1: Machine Learning for IoT Devices

RQ1: Common ML practices for IoT. *ML modeling:* ML algorithms are one ingredient of next-generation IoT systems. We asked survey respondents and interview subjects where their models come from. Survey respondents rely on academic research, re-using

Table 1: Interview Subjects

Identifier	Role (Company type)	Experience
P1	Principal Architect (HW vendor)	20 years
P2	Senior developer (HW vendor)	20 years
P3	Chief Architect (Start-up)	30 years
P4	ML Engineer (ML services)	3 years

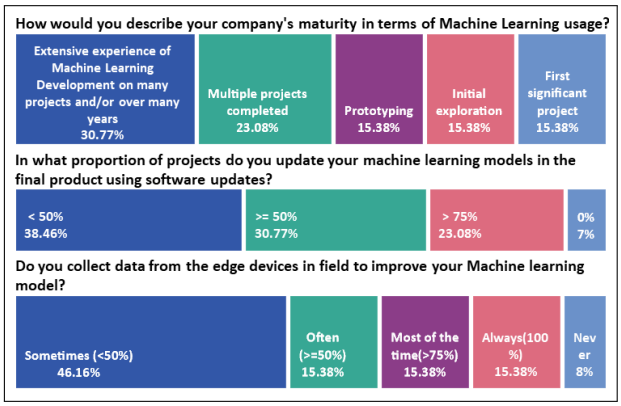

Figure 3: Survey data on ML maturity, software updates, and data collection.

models entirely or tailoring them to their company's needs (Table 2). Notably, none of the survey respondents indicated that they follow product line development (*i.e.,* , reuse models from one product to the next) for their ML models. P3 characterized the sources used by his start-up:

> *"In the ML world, [if] you don't read a paper every single day, you are in trouble...IEEE papers and...we also look at results that come out of Google, Facebook, Amazon and Microsoft."*

Companies with deeper expertise also develop models internally; P1 said, *"[My company's] research team does a lot of research around machine learning, and we...use the frameworks developed by them."*

Table 2: Source of ML models in practice.

For the machine learning models, do you:	Proportion
Adopt existing models from academic research with significant changes	46%
Adopt existing models from academic research without significant changes	38%
Develop new models from scratch	15%
Adopt or iterate over existing models within the company used in other applications	0%

ML development: TensorFlow/TF-Lite and PyTorch were the most popular modeling frameworks; Python and C/C++ were the most popular languages. To train and validate models, survey respondents follow standard practices: splitting training and testing data, applying K-fold/cross validation, etc.

Engineering processes: Our data show that the industry movement towards incremental development and agile methodologies [41] includes IoT systems development. Among survey respondents, 48% report using *"Agile"* as their software development process, the most popular response. Our interview subjects concurred. As interviewee P3 said:

> *"We tend to follow the agile flow...2 years ago we [were] mostly waterfall, the old-fashioned way...now...95% of...[our] programs [are agile]."*

This adoption includes the Continuous Integration/Continuous Deployment (CI/CD) approach. Half of the survey respondents said their teams incorporate ML models into the rest of their IoT systems during CI, 25% said *"Before deployment"*, and only 16% said their integration occurred at software release time. Interviewee P3 said:

> *"At every stage of our Agile flow...[we have a] CI/CD-based validation flow...as part of the weekly sprints trying to meet accuracy, latency and throughput."*

After IoT device deployment, many survey respondents report that they improve the ML models in their products by collecting new data and sending software updates (Figure 3).

RQ2: ML challenges and consequences for IoT. *ML on resource-constrained devices:* IoT engineers work within hardware constraints. Over 90% of survey respondents said they meet constraints by changing the software, not the hardware. To meet their resource constraints, our survey respondents said they use neural network pruning techniques including regularization, second-order methods, and variational dropout. As they do so, survey respondents said they struggle with decreased model performance (38%), memory constraints (23%), and insufficient expertise (23%) (Figure 4). Interviewee P3 went into more detail:

> *"From a technical perspective, one of the biggest problems that we face is the inability of standard tools to be able to squash a model into something that fits with a push of a button."*

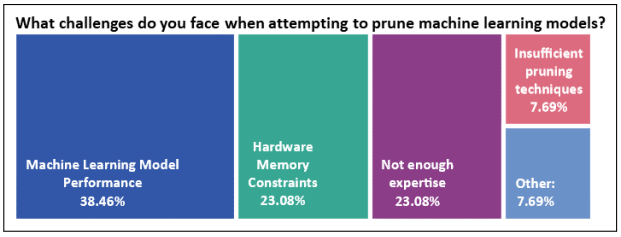

Figure 4: Survey data on ML resource constraints.

Our interview participants went into detail about their strategy for estimating ML model performance: back-of-the-envelope calculations. As P2 said:

> *"I prefer Excel sheet because bringing emulator to a state that you can perform simulation takes time. And also building machine learning algorithms takes time. So it's better [to make a] crude estimate...using Excel sheet...and then simply prepare ML algorithms that simply relies on this crude estimate."*

Working with customers: P4 noted the challenges of ensuring robustness as a customer requirement:

> *"[Clients] give us the validation data set, but not the test data set...Then they used to run the inference at their end on the same device and validate if it works well on the test data set. Even slight changes...distortions...used to give bad accuracy...So if your model should be robust to such kind of things, then you need to have such kind of data in your training data set."*

Edge-Cloud collaboration: Survey respondents described different architectures for data processing. Two-thirds follow a hybrid strategy, with lighter-weight processing on IoT devices and heavy-weight processing using Fog or Cloud systems. Edge-only processing was the second most common, and Cloud-only processing was rare. When placing computation on Edge devices, engineers reported working around the resource limitations of IoT devices.

5.3 Theme 2: Secure IoT engineering

Most survey respondents have experienced CVEs in their current product. One-third have dealt with 1–3 CVEs, and one-third with 4 or more. Understanding how they incorporate security and reason about trust in their engineering processes may help reduce CVEs.

RQ3: Incorporating security into IoT engineering. *Security Analysis:* We asked survey respondents to describe the processes their teams follow for security analysis. Code review (42%) and white-box analyses (21%) were the primary ways in which security checks are realized (Figure 5). Survey respondents and interview participants also described conducting security reviews and creating mitigation plans. Interviewee P3 discussed integrating security into the ML development process:

> *"It's not as if every member in the team is...[a security expert, but] they are [generally] aware of the pitfalls and needs. But...[we ensure that] a few experts are always there in the reviews."*

For interviewees working in smaller companies, security analysis was part of every developer's job. In larger organizations, interviewees said that security analysis was done by dedicated security teams. However, developers are still involved and have some familiarity with security analysis methods.

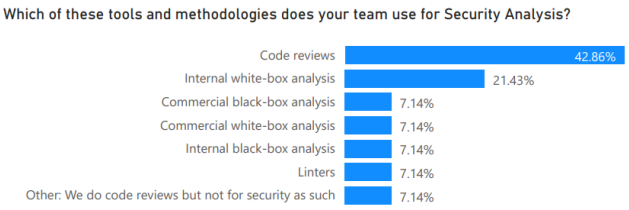

Figure 5: Methods for security analysis. *This question was accidentally single-response, so we suppose the respondents interpreted this as* primary *method.*

Threats and threat models: Our subjects said that security threat analysis was a common part of the development process, but with varying priority depending on the company size and available resources. Our interviewees indicated that the major threat they considered was the loss of intellectual property — reverse engineering of their ML models. Interviewee P1 said the biggest security challenge they face is in-memory re-engineering: *"We try to mimic scenarios that can breach security...We are careful about snoop-out transactions."* At P1's organization the same threat is considered:

> *"One common area where you can snoop things out in hardware is the Memory Management Unit...So if your MMU gets compromised, then you...have physical addresses and you can do whatever you want with it. So, secure hardware design become critical."*

By nature, IoT devices interact with humans and the physical world. This makes privacy a concern for both developers and end-users. Interviewee P2 said privacy was the most difficult aspect of security analysis, and described his company's approach like this:

> *"The privacy, it is the hard problem...it will be really visible to the market...We are trying to not store any private data that could be...used by hacker in any way....we are simply not trying to tackle such cases. And from my previous work...It was always an issue because it is a really hard problem. And it is really easy to...lose your name, lose your brand."*

RQ4: Trust in IoT systems. Our interviewees identified trust in researchers, vendors, data, and tool chains.

As highlighted in Table 2, survey respondents indicated trust in researchers through the common adoption of research prototypes. Interviewee P3 described how his company's ML model training process is dependent on security features provided by cloud computing platforms:

> *"So in the fully cloud-based solutions we are largely dependent on...the goodness of the cloud. It's almost impossible to see what Azure, AWS, etc., are doing under the hood. So there's a large level of dependence on their security procedures."*

P3 also noted his trust in development tool chains:

> *"We...are not doing a whole lot of analysis on weakness of...tools like TensorFlow. If TensorFlow...has a security hole, there is not much we do about it. ... [W[e have wrappers that ensure there is some levels of encryption, unhackability before it...goes on to the eventual edge IoT device. But if you were to question the IDEs and tools chains having security bugs, there is nothing we can do about it."*

P1 pointed out his assumptions of trustworthy data:

> *"We have to ensure that the [training] data...is from a trusted source, otherwise it becomes a nightmare."*

RQ5: Other factors that affect IoT engineering. *Process requirements and regulations:* During our survey we asked participants about restrictions on their engineering processes and products (Table 3). About half comply only with general quality processes (*e.g.*, P3: *"We are an ISO 9001:2015 company. We rigorously follow the ISO standards."*). Other survey respondents comply with governmental safety and security regulations (26%), and with privacy regulations like GDPR and HIPAA (22%). In P1's organization, they prefer to work with metadata instead of data because of HIPAA requirements:

> *"Once you start working with meta-data, then you don't really need...any private information...so, it becomes much easier."*

P1 expanded on the difficulties of regulatory compliance:

> *"For example, anytime I'm working with the medical data, that becomes a very, very tricky situation...[you must] set up proper working environment and...ensure that the data is not leaving your trusted network...not just personal data, but also [its] trends"*

Table 3: Survey data on process requirements and regulations

What regulations do you have to comply with during your software development process?	Proportion
General engineering processes like ISO XXX	44%
Governmental privacy regulations like GDPR / HIPAA / FERPA	26%
Governmental safety / security regulations like the IoT Cybersecurity Improvement Act (requires following NIST guidelines).	22%
Security engineering processes like OWASP SAMM.	7%

Engineering Cost: Several of our subjects pointed out a balance between security and engineering cost. A survey respondent wrote that the most challenging aspect is:

> *"Addressing vulnerabilities properly...within the project budget...[and] supporting cryptographic functionality for encryption, storage, data transmission, and key/certificate management."*

Interviewee P2 observed that user visibility may justify engineering costs:

> *"First of all you must decide if security is required. If you push security to a level that is hard to maintain, and it is adding significant value to the Bill Of Materials cost, then it is a question if it will be accepted by the market. I believe, the argument about security is if it will be visible to the user."*

Interviewee P1 observed that his company invests security resources non-uniformly, with less effort in analyzing software that they release open-source:

> *"When we do an open-source release, we don't worry much about it...any shortcomings we get notified very quickly by the open-source community and we can fix it. Of course, it is not a good thing to release something insecure to open-source which is not adequately tested or verified...We mainly ensure that previous occurrences of security breaches are tested and we make it part of the design process."*

6 DISCUSSION

6.1 Comparison to prior findings

Our findings overlapped with prior knowledge in many aspects. In terms of development tools, our participants followed industry-wide practices such as using ML frameworks like TensorFlow and PyTorch and development toolchains based on the Visual Studio/-Code IDEs. Our participants follow iterative development processes. The use of hybrid Edge-Cloud architectures is widespread. Power, memory constraints, and computational constraints are known to be major challenges within IoT systems. Our participants are aware of security issues such as data poisoning.

The main difference between the research literature and our findings is the discussion of engineering cost. Our participants — perhaps especially those in consumer electronics — reduce security for cheaper production costs. Similarly, there are many interesting methods of emulation, load-balancing, and system validation proposed in the research literature, but most respondents' organizations do not use these methods. Unlike researchers' goals of unbreakable systems, our subjects balance how much security is possible (relative to its engineering cost) and required (relative to market demand). The research literature generally does not consider the engineering cost of proposed techniques. Lastly, the many sources of unverified trust — open-source code, academic research, and development toolchains — was greater than what we understood in the literature.

6.2 Advice for practitioners

Our study revealed a significant gap between how the academic community and industry perceive IoT security. This suggests potential value in cybersecurity workforce development [7]. Outside academia, government guidelines (*e.g.,* from US-NIST [5] and EU-ENISA [3]) describe secure development lifecycles. NIST [6] recommends a thorough study on the customers, users, expected use cases, security risks, and goals during planning, execution, and post-deployment. Our subjects did not describe such a process.

Given the success of automated code analysis methods such as static analysis, black-box and grey-box fuzzing in identifying system vulnerabilities in IT software, we were surprised by practitioners' continued emphasis on code review and white-box analysis in their IoT systems. We recommend practitioners integrate such methods into their product development process [44].

6.3 Future work for researchers

Based on the challenges faced by the practitioners we studied, we suggest three directions for future research.

First, the IoT domain is characterized by tight profit margins and low-cost parts. Many of our research subjects were therefore concerned about the engineering cost of securing IoT devices. It would be helpful for researchers to offer engineering cost-aware security processes suited to the constraints of IoT systems engineering, and practical measurements of this cost. Past research works primarily focus on trade-offs between security and resource costs, such as operation delay and energy [19, 65]. Our work identifies the importance of considering engineering costs, not just the runtime implications. Our work also complements ongoing research to help consumers understand how security affects the cost of commodity IoT devices [32].

Second, practitioners leverage open science and open-source software for their ML modeling and their development toolchains. This accelerates development, but introduces substantial risk. For ML, we recommend that ML researchers carefully document their

research prototypes and the limitations of their work, and that they can achieve broader impact by participating in community efforts to develop exemplary ML models (*e.g.,* TorchVision [52] and the TensorFlow Model Garden [62]). Additional studies of how best to reproduce and transfer ML knowledge will be helpful [13, 17, 37]. More broadly, given the reliance of our participants on open-source tools, trustworthy software supply chains will improve the safety and security of IoT systems [64].

Third, the difficulties experienced by practitioners in following the compliance restrictions and regulations identified in Table 3 poses a potential research area. For example, researchers could study the impact of security compliance on security outcomes of IoT applications, and the tradeoff with engineering cost.

7 THREATS TO VALIDITY

Construct validity: Our survey instrument and interview protocol were intended as direct measures of the constructs of interest (*i.e.,* engineering practices), and we used pilot studies as a check.

Internal validity: Our study reports on practices without inferences about cause and effect, so internal validity is not a concern.

External validity: The primary limitation of our study is in its external validity, *i.e.,* generalizability. Our goal was to describe current practices in IoT engineering, focused on machine learning and cybersecurity. As is common with studies of this kind, we used a human-subjects method with a self-report design, which assumes the respondents were truthful. Beyond the trustworthiness of our data, we emphasize that we had relatively few survey responses (N=25). We cannot claim saturation; our results are likely not representative of the entire state of practice. As mitigating factors, our survey reached participants from several industry sectors, and our interview subjects included experts with a long tenure in industry and experience at several companies.

8 CONCLUSION

In this research attempted to broaden the existing understanding of IoT engineering practices related to machine learning and cybersecurity. Through our survey and interviews, we found that the main challenge engineers face when creating an IoT product is balancing among engineering cost, performance, trust, and security. We found that organizations place unverified trust in open-source and academic resources; going so far as to incorporate academic prototypes of ML techniques into their IoT products. Cybersecurity investment varies based on resources, engineering cost and organizational priorities; one organization even explicitly relies on the open-source community to find vulnerabilities in their software. Practitioners have not yet adopted academic research in engineering practices and government recommendations that might address some of their problems. In the future, we recommend that software engineering and cybersecurity researchers incorporate engineering cost considerations into their work, as this was a concern raised by many of our research subjects.

DATA AVAILABILITY

An artifact is available with the data collection instruments, survey data, and interview transcripts. See https://tinyurl.com/ysy7ntmx.

RESEARCH ETHICS

The study was approved by our institution's IRB.

REFERENCES

[1] 2016. Hackers Used New Weapons to Disrupt Major Websites Across U.S. https://www.nytimes.com/2016/10/22/business/internet-problems-attack.html. Accessed June 08, 2021.

[2] 2016. *A Primer on Continuous Delivery.* https://feeney.mba/a-primer-on-continuous-delivery.html

[3] 2017. Baseline Security Recommendations for IoT. https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot. Accessed June 09, 2021.

[4] 2017. Your Roomba May Be Mapping Your Home, Collecting Data That Could Be Shared. https://www.nytimes.com/2017/07/25/technology/roomba-irobot-data-privacy.html. Accessed June 08, 2021.

[5] 2019. IoT Device Cybersecurity Capability Core Baseline. https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259A.pdf. Accessed June 09, 2021.

[6] 2020. Foundational Cybersecurity Activitiesfor IoT Device Manufacturers. https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259.pdf. Accessed June 09, 2021.

[7] 2020. IoT Non-Technical Supporting Capability Core Baseline. https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259b-draft.pdf. Accessed June 09, 2021.

[8] 2020. System brings deep learning to "internet of things" devices. https://news.mit.edu/2020/iot-deep-learning-1113.

[9] Yasemin Acar, Christian Stransky, Dominik Wermke, Charles Weir, Michelle L. Mazurek, and Sascha Fahl. 2017. Developers Need Support, Too: A Survey of Security Advice for Software Developers. *Proceedings - IEEE Cybersecurity Development Conference, SecDev* (2017). https://doi.org/10.1109/SecDev.2017.17

[10] Deniz Akdur, Vahid Garousi, and Onur Demirörs. 2018. A survey on modeling and model-driven engineering practices in the embedded software industry. *Journal of Systems Architecture* 91 (2018), 62–82. https://doi.org/10.1016/j.sysarc.2018.09.007

[11] Sultan Alharby, Nick Harris, Alex Weddell, and Jeff Reeve. 2018. The Security Trade-offs in Resource Constrained Nodes for IoT Application. *International Journal of Electrical, Electronic and Communication Sciences: 11.0.* 12, 1 (2018), 56–63. https://www.researchgate.net/publication/322747058%0Ahttp://www.waset.org/downloads/15/papers/18ae010177.pdf

[12] Omar Alrawi, Chaz Lever, Manos Antonakakis, and Fabian Monrose. 2019. SoK: Security Evaluation of Home-Based IoT Deployments. In *2019 IEEE Symposium on Security and Privacy, SP.*

[13] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software Engineering for Machine Learning: A Case Study. In *International Conference on Software Engineering: Software Engineering in Practice.* https://doi.org/10.1109/ICSE-SEIP.2019.00042

[14] Hala Assal and Sonia Chiasson. 2019. "Think secure from the beginning": A survey with software developers. *Conference on Human Factors in Computing Systems - Proceedings* (2019). https://doi.org/10.1145/3290605.3300519

[15] Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The Internet of Things: A survey. *Computer Networks* (2010). https://doi.org/10.1016/j.comnet.2010.05.010

[16] Saurabh Bagchi, Tarek F. Abdelzaher, Ramesh Govindan, Prashant Shenoy, Akanksha Atrey, Pradipta Ghosh, and Ran Xu. 2020. New Frontiers in IoT: Networking, Systems, Reliability, and Security Challenges. *IEEE Internet of Things Journal* 7, 12 (2020), 11330–11346. https://doi.org/10.1109/JIOT.2020.3007690

[17] Vishnu Banna, Akhil Chinnakotla, and et al. 2021. An Experience Report on Machine Learning Reproducibility: Guidance for Practitioners and TensorFlow Model Garden Contributors. *arXiv* (2021).

[18] Iulia Bastys, Musard Balliu, and Andrei Sabelfeld. 2018. If This Then What?: Controlling Flows in IoT Apps. In *Conference on Computer and Communications Security, CCS.*

[19] Chiara Bodei, Stefano Chessa, and Letterio Galletta. 2019. Measuring Security in IoT Communications. *Theoretical Computer Science* 764 (April 2019), 100–124. https://doi.org/10.1016/j.tcs.2018.12.002

[20] Will Brackenbury, Abhimanyu Deora, Jillian Ritchey, Jason Vallee, Weijia He, Guan Wang, Michael L. Littman, and Blase Ur. 2019. How Users Interpret Bugs in Trigger-Action Programming. In *Conference on Human Factors in Computing Systems CHI.*

[21] Z. Berkay Celik, Earlence Fernandes, Eric Pauley, Gang Tan, and Patrick D. McDaniel. 2019. Program Analysis of Commodity IoT Applications for Security and Privacy: Challenges and Opportunities. *ACM Comput. Surv.* 52, 4 (2019).

[22] Z. Berkay Celik, Patrick D. McDaniel, Gang Tan, Leonardo Babun, and A. Selcuk Uluagac. 2019. Verifying Internet of Things Safety and Security in Physical Spaces. *IEEE Secur. Priv.* 17, 5 (2019).

[23] Z. Berkay Celik, Gang Tan, and Patrick D. McDaniel. 2019. IoTGuard: Dynamic Enforcement of Security and Safety Policy in Commodity IoT. In *26th Annual Network and Distributed System Security Symposium, NDSS.*

[24] Mengsu Chen, Felix Fischer, Na Meng, Xiaoyin Wang, and Jens Grossklags. 2019. How reliable is the crowdsourced knowledge of security implementation?. In *International Conference on Software Engineering (ICSE).*

[25] Long Cheng, Christin Wilson, Song Liao, Jeffrey Young, Daniel Dong, and Hongxin Hu. 2020. Dangerous Skills Got Certified: Measuring the Trustworthiness of Skill Certification in Voice Personal Assistant Platforms. In *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security.* ACM.

[26] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* (2017), 1800–1807. https://doi.org/10.1109/CVPR.2017.195 arXiv:1610.02357

[27] Rafael Maiani de Mello and Guilherme Horta Travassos. 2016. Surveys in Software Engineering: Identifying Representative Samples. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '16).* Association for Computing Machinery, Article 55, 6 pages. https://doi.org/10.1145/2961111.2962632

[28] Joa Pedro Dias and Hugo Sereno Ferreira. 2018. State of the software development life-cycle for the internet-of-things. *arXiv* (2018). arXiv:1811.04159

[29] Paul Dodemaide, Prof. Lynette Joubert, Dr Nicole Hill, and Dr Mark Merolli. 2020. Online Survey Design and Social Media. In *Proceedings of the Australasian Computer Science Week Multiconference (ACSW '20).* Association for Computing Machinery, Article 36, 8 pages. https://doi.org/10.1145/3373017.3373054

[30] B. Dorsemaine, J. Gaulier, J. Wary, N. Kheir, and P. Urien. 2015. Internet of Things: A Definition Taxonomy. In *2016 9th International Conference on Next Generation Mobile Applications, Services and Technologies.* 72–77. https://doi.org/10.1109/NGMAST.2015.71

[31] El Mahdi El Mhamdi and Rachid Guerraoui. 2017. When Neurons Fail. *Proceedings - 2017 IEEE 31st International Parallel and Distributed Processing Symposium, IPDPS 2017* (2017), 1028–1037. https://doi.org/10.1109/IPDPS.2017.66 arXiv:1706.08884

[32] Pardis Emami-Naeini. 2020. *Informing Privacy and Security Decision Making in an IoT World.* Ph. D. Dissertation. Carnegie Mellon University.

[33] Michael Fagan, Katerina N Megas, Karen Scarfone, and Matthew Smith. 2020. Foundational Cybersecurity Activities for IoT Device Manufacturers. *NIST Interagency/Internal Report 8259* (2020). https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259.pdf

[34] Earlence Fernandes, Jaeyeon Jung, and Atul Prakash. 2016. Security Analysis of Emerging Smart Home Applications. In *IEEE Symposium on Security and Privacy, SP.*

[35] Earlence Fernandes, Amir Rahmati, Jaeyeon Jung, and Atul Prakash. 2018. Decentralized Action Integrity for Trigger-Action IoT Platforms. In *25th Annual Network and Distributed System Security Symposium, NDSS.*

[36] Fereshteh Ghaljaie, Mahin Naderifar, and Hamideh Goli. 2017. Snowball Sampling: A Purposeful Method of Sampling in Qualitative Research. *Strides in Development of Medical Education* 14, 3 (2017). https://doi.org/10.5812/sdme.67670 arXiv:http://sdme.kmu.ac.ir/article_90598_3632edfb2e97c38d73c0bdea8753195c.pdf

[37] Kim Hazelwood, Sarah Bird, David Brooks, Soumith Chintala, Utku Diril, Dmytro Dzhulgakov, Mohamed Fawzy, Bill Jia, Yangqing Jia, Aditya Kalro, James Law, Kevin Lee, Jason Lu, Pieter Noordhuis, Misha Smelyanskiy, Liang Xiong, and Xiaodong Wang. 2018. Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective. In *HPCA.* https://doi.org/10.1109/HPCA.2018.00059

[38] Robert M. Hierons. 1999. Machine Learning, by Tom M. Mitchell, McGraw-Hill, 1997 (Book Review). *Softw. Test. Verification Reliab.* 9, 3 (1999), 191–193.

[39] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *NIPS Deep Learning and Representation Learning Workshop* (2015), 1–9. arXiv:1503.02531 http://arxiv.org/abs/1503.02531

[40] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* (2017). arXiv:1704.04861

[41] Jez Humble and Gene Kim. 2018. *Accelerate: The science of lean software and devops: Building and scaling high performing technology organizations.* IT Revolution.

[42] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. 2014. Industry 4.0. *Business & information systems engineering* (2014).

[43] Dirk Van Der Linden, Pauline Anthonysamy, Bashar Nuseibeh, Thein Than Tun, Marian Petre, Mark Levine, John Towse, and Awais Rashid. 2020. Schrodinger's security: Opening the box on app developers' security rationale. *Proceedings - International Conference on Software Engineering* (2020). https://doi.org/10.1145/3377811.3380394

[44] MH Lloyd and PJ Reeve. 2009. IEC 61508 and IEC 61511 assessments-some lessons learned. (2009).

[45] Tamara Lopez, Helen Sharp, Thein Tun, Arosha Bandara, Mark Levine, and Bashar Nuseibeh. 2019. Hopefully we are mostly secure': Views on secure code in professional practice. *Proceedings - 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2019* (2019). https://doi.org/10.1109/CHASE.2019.00023

[46] Amir Makhshari and Ali Mesbah. 2021. IoT Bugs and Development Challenges. *ICSE 2021* (2021), 460–472. https://doi.org/10.1109/icse43902.2021.00051

[47] Xianghang Mi, Feng Qian, Ying Zhang, and XiaoFeng Wang. 2017. An empirical characterization of IFTTT: ecosystem, usage, and performance. In *Proceedings of the 2017 Internet Measurement Conference, IMC.*

[48] Hooman Mohajeri Moghaddam, Gunes Acar, Ben Burgess, Arunesh Mathur, Danny Yuxing Huang, Nick Feamster, Edward W. Felten, Prateek Mittal, and Arvind Narayanan. 2019. Watching You Watch: The Tracking Ecosystem of

Over-the-Top TV Streaming Devices. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS*. ACM.

[49] Blaine Nelson, Benjamin I.P. Rubinstein, Ling Huang, Anthony D. Joseph, Steven J. Lee, Satish Rao, and J. D. Tygar. 2012. Query strategies for evading convex-inducing classifiers. *Journal of Machine Learning Research* 13 (2012), 1293–1332. arXiv:1007.0484

[50] Taehyeun Park, Nof Abuzainab, and Walid Saad. 2016. Learning How to Communicate in the Internet of Things: Finite Resources and Heterogeneity. *IEEE Access* 4 (2016). https://doi.org/10.1109/ACCESS.2016.2615643

[51] Roger S Pressman. 2005. *Software engineering: a practitioner's approach*. Palgrave macmillan.

[52] Pytorch. 2017. TORCHVISION - https://pytorch.org/vision/stable/index.html. https://pytorch.org/vision/stable/index.html?highlight=torchvision#module-torchvision

[53] Bin Qian, Jie Su, Zhenyu Wen, Devki Nandan Jha, Yinhao Li, Yu Guan, Deepak Puthal, Philip James, Renyu Yang, Albert Y. Zomaya, Omer Rana, Lizhe Wang, MacIej Koutny, and Rajiv Ranjan. 2020. Orchestrating the Development Lifecycle of Machine Learning-based IoT Applications: A Taxonomy and Survey. *Comput. Surveys* 53 (2020). Issue 4. https://doi.org/10.1145/3398020

[54] Paul Ralph, Sebastian Baltes, Domenico Bianculli, et al. 2020. ACM SIGSOFT Empirical Standards. *CoRR* abs/2010.03525 (2020). arXiv:2010.03525

[55] Frances Robles and Nicole Perlroth. 2021. 'Dangerous Stuff': Hackers Tried to Poison Water Supply of Florida Town. *The New York Times* (Feb. 2021).

[56] Ruben M. Sandoval, Sebastian Canovas-Carrasco, Antonio Javier Garcia-Sanchez, and Joan Garcia-Haro. 2019. A reinforcement learning-based framework for the exploitation of multiple rats in the iot. *IEEE Access* 7 (2019). https://doi.org/10.1109/ACCESS.2019.2938084

[57] F. Schwandt. 2016. Internet of things (IoT) connected devices installed base worldwide from 2015 to 2025 in billions - Statista. (2016). https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide

[58] Eugene Siow, Thanassis Tiropanis, and Wendy Hall. 2018. Analytics for the Internet of Things: A Survey. *ACM Comput. Surv.* 51, 4, Article 74 (July 2018), 36 pages. https://doi.org/10.1145/3204947

[59] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. 2017. Certified defenses for data poisoning attacks. *Advances in Neural Information Processing Systems* 2017-December, i (2017), 3518–3530. arXiv:1706.03691

[60] Manuel Suarez-Albela, Tiago M. Fernandez-Carames, Paula Fraga-Lamas, and Luis Castedo. 2018. A practical performance comparison of ECC and RSA for resource-constrained IoT devices. *2018 Global Internet of Things Summit, GIoTS 2018* (2018). https://doi.org/10.1109/GIOTS.2018.8534575

[61] Milijana Surbatovich, Jassim Aljuraidan, Lujo Bauer, Anupam Das, and Limin Jia. 2017. 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, WWW*.

[62] TensorFLow. 2021. Models datasets - https://github.com/tensorflow/models. https://github.com/tensorflow/models

[63] Tyler W. Thomas, Madiha Tabassum, Bill Chu, and Heather Lipford. 2018. Security during application development: An application security expert perspective. *Conference on Human Factors in Computing Systems - Proceedings* (2018). https://doi.org/10.1145/3173574.3173836

[64] Santiago Torres-Arias. 2020. *In-toto: Practical Software Supply Chain Security*. Ph. D. Dissertation. New York University Tandon School of Engineering.

[65] Wiebke Toussaint and Aaron Yi Ding. 2020. Machine Learning Systems in the IoT: Trustworthiness Trade-offs for Edge Intelligence. In *2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI)*. 177–184. https://doi.org/10.1109/CogMI50398.2020.00030

[66] Jeffrey Voas. 2016. Demystifying the Internet of Things. *Computer* 49, 6 (2016), 80–83. https://doi.org/10.1109/MC.2016.162

[67] Fangxin Wang, Miao Zhang, Xiangxiang Wang, Xiaoqiang Ma, and Jiangchuan Liu. 2020. Deep Learning for Edge Computing Applications: A State-of-the-Art Survey. *IEEE Access* 8 (2020). https://doi.org/10.1109/ACCESS.2020.2982411

[68] Elecia White. 2011. *Making Embedded Systems: Design Patterns for Great Software*. O'Reilly Media.

[69] Han Xiao, Huang Xiao, and Claudia Eckert. 2012. Adversarial label flips attack on support vector machines. *Frontiers in Artificial Intelligence and Applications* 242 (2012), 870–875. https://doi.org/10.3233/978-1-61499-098-7-870

[70] Shuochao Yao, Yiran Zhao, Huajie Shao, Sheng Zhong Liu, Dongxin Liu, Lu Su, and Tarek Abdelzaher. 2018. FastDeepIoT: Towards understanding and optimizing neural network execution time on mobile and embedded devices - SenSys 2018 - Proceedings of the 16th Conference on Embedded Networked Sensor Systems. *SenSys2018*. https://doi.org/10.1145/3274783.3274840