HARVARD | BUSINESS | SCHOOL

Algorithm-Augmented Work Performance and Domain Experience: The Countervailing Forces of Ability and Aversion

Ryan T. Allen
Prithwiraj Choudhury

Working Paper 21-073

Algorithm-Augmented Work Performance and Domain Experience: The Countervailing Forces of Ability and Aversion

Ryan T. Allen
Harvard Business School

Prithwiraj Choudhury
Harvard Business School

Working Paper 21-073

Funding for this research was provided in part by Harvard Business School.

Algorithm-Augmented Work Performance and Domain Experience: The Countervailing Forces of Ability and Aversion

Ryan T. Allen
Harvard Business School

Prithwiraj Choudhury
Harvard Business School

Abstract. How does a knowledge worker's level of domain experience affect their algorithm-augmented work performance? We propose and test theoretical predictions that domain experience has countervailing effects on algorithm-augmented performance: on one hand, domain experience enhances a worker's *ability* to accurately assess the quality of an algorithmic tool's advice; on the other hand, highly experienced workers exhibit more *aversion* to algorithmic advice, relative to their own judgment. We exploit a within-subjects experiment in which corporate IT support workers were assigned to resolve similar problems both manually (using their own judgment) and using advice generated by an algorithmic tool. Relative to solving problems using their own judgment, we confirm an inverted U-shape between IT domain experience and performance for problems where the algorithmic tool generated advice. While low experience workers' propensity to reject accurate algorithmic advice appears to be driven by lack of *ability* to accurately assess the algorithm's advice, highly experienced workers appear to reject algorithmic advice due to algorithmic *aversion*.

Keywords: automation, domain experience, algorithmic aversion, experts, algorithms, machine learning, decision-making, future of work

Acknowledgements: The authors are grateful to Hyunjin Kim, Rory McDonald, Marco Iansiti, and members of the Technology and Operations Management Doctoral Research Seminar at Harvard Business School for helpful reviews and comments on this paper. All errors remain our own.

Introduction

Workers in a variety of organizations increasingly use algorithmic tools to augment judgment and decision-making. Though algorithm-augmented work is not new, recent advances in artificial intelligence (AI) and machine learning (ML) technologies have increased the scope of tasks that can be algorithmically automated or augmented. For example, there has been a notable increase in the number of managers, clinicians, and judges that have adopted ML-trained algorithmic decision tools to help them hire personnel, diagnose diseases, and assign bail (Miller 2015, Cowgill 2018b, 2018a, Kleinberg *et al.* 2018, Arthur and Hossein 2019).

Much existing research compares the performance of algorithms and human experts for various decision and judgment tasks, often demonstrating the superior accuracy of even simple algorithms (Dawes 1979, Grove *et al.* 2000, Kleinberg *et al.* 2018, Miller 2018). Although this has established an important baseline for comparing human and algorithmic judgment, there are limits to the fruitfulness of the human vs. machine "horse race" line of questioning. In real organizations, even the most advanced algorithms often do not fully automate tasks—they augment the judgment of workers in organizations (Raisch and Krakowski 2020). Because algorithms and humans have different strengths and weaknesses, a broad research agenda is emerging around the question of how organizations can design hybrid work processes that combine the strengths of both humans and algorithmic tools (Shrestha, Ben-Menahem, and von Krogh 2019, Choudhury, Starr, and Agarwal 2020).

A first-order question in this emerging research agenda has been whether humans will accept and use algorithmic advice. A series of lab experiments document the notion of "algorithmic aversion," the tendency of people, *especially experts*, to rely more on their own judgment than the advice generated by an algorithm (Dietvorst, Simmons, and Massey 2015, Logg, Minson, and Moore 2019). Yet, this phenomenon has not been investigated in the context of real workers using a real algorithmic tool in an organizational setting. Furthermore, recent empirical evidence demonstrates that domain experience complements algorithm-augmented work because experts can catch and correct algorithmic bias/mistakes (Choudhury et al. 2020). Given the result that expertise is also related to aversion, it is unclear how the overall algorithm-augmented performance of workers varies over a range of domain experience. Thus in this paper, we ask: How does a knowledge worker's domain experience affect their performance on algorithm-augmented tasks (relative to performing the task using their own judgment)?

There are two countervailing forces that potentially shape how domain experience affects a worker's algorithm-augmented performance (relative to performing the task on their own). On the one hand, we expect domain experience to increase a worker's algorithm-augmented performance due to an increased *ability* to judge the accuracy of an algorithm's advice. Because algorithms excel at repeatable, measurable, and narrow tasks, humans can complement them by understanding relevant context, learning from a small number of cases, and adapting in unfamiliar situations (Autor 2015, Agrawal, Gans, and Goldfarb 2018, Brynjolfsson, Mitchell, and Rock 2018). These complementary human abilities are developed through experience within a specific domain (Simon 1991), and workers with greater domain experience are more likely to better assess the accuracy of algorithmic advice. Yet on the other hand, there is substantial evidence that experts (i.e., workers with high domain experience) might exhibit greater aversion in accepting advice (Liu 2017)—including algorithmic advice—relative to their own judgment (Logg et al. 2019).

This paper theorizes and tests predictions that are based on the interaction of these countervailing forces. We theorize that because *ability* developed through learning by doing increases at a decreasing rate (Becker 1962, Foster and Rosenzweig 1995, Mithas and Krishnan 2008) and *algorithmic aversion* is more prevalent among experts, overall algorithm-augmented performance will first rise with domain experience, then fall. We exploit a within-subjects experiment in which a sample of corporate IT support workers at a large Indian technology organization (hereafter TECHCO) were asked to resolve help tickets.[1] Tickets were assigned to be resolved: (1) manually (i.e., *without* using the algorithmic tool) using their previous ticket resolution system; and (2) by using a new ML-trained algorithmic tool that lists the most likely solutions to each ticket based on its text input. This enables us to compare algorithm-augmented performance of workers (i.e., where the algorithm generates advice for the worker), relative to self-performance (i.e., where the worker uses their own judgment to resolve tickets). We find that workers with the least and the most IT experience were less likely to resolve tickets using the new algorithmic tool than their peers in the intermediate range of IT experience—confirming an inverted U-shape in performance over the range of domain experience. The inverted-U pattern held only for the

[1] A ticket is a document that is generated when an employee at TECHCO flags a problem with the IT support team. In other words, each ticket documents a work order and is "resolved" when the underlying IT problem (e.g., a server is not working, an employee cannot log in remotely, etc.) is resolved.

tickets resolved using the algorithmic tool—a necessary condition for demonstrating that the decrease in performance of experts was specific to algorithm-augmented performance.

To illuminate the mechanisms, we analyze the individual log files of each participant in the experiment. We find that the inverted U-shaped relationship is driven by a propensity of both the low experience and the high experience participants to reject the algorithm's correct advice. In fact, these participants more frequently did not even attempt (i.e., select *any* algorithm-recommended solution) to resolve tickets using the algorithm. However, the mechanisms driving this pattern appear to be different for high experience vs. low experience participants. We show that low experience participants failed to resolve more difficult problems (indicative of a lack of *ability*), while high experience workers were equally likely to reject difficult or straightforward problems (consistent with *aversion*). Further bolstering the *ability* argument, we find that conditional upon attempting a solution, greater experience was linearly and positively associated with better performance using the algorithm. Interviews with the participants also confirm that low experience workers were unfamiliar with problems whose algorithmically generated solutions they rejected. By contrast, high experience workers were familiar with the problems but believed that accepting algorithmic advice might have unintended consequences affecting the overall IT production environment; they also believed that the tickets would best be addressed manually using their own judgment. These results all suggest that the countervailing forces of *ability* and *algorithmic aversion* across the range of domain experience may be a double-edged sword when implementing algorithm-augmented work processes.

Empirically, we contribute to the existing literature by providing evidence from one of the first experiments using actual workers in their daily work, with a wide range of relevant task experience. While researchers have conducted prior studies of algorithmic aversion in laboratory and online settings (e.g., Dietvorst, Simmons, and Massey 2015, Logg, Minson, and Moore 2019, Yeomans *et al.* 2019), to the best of our knowledge, this study represents the first experiment conducted within an organization to study how a broad range of domain experience affects algorithm-augmented task performance. Rather than just classifying workers as binary "experienced" or "inexperienced," we can explore the forces at play across a gradient of experience—which allows us to uncover the inverted-U relationship perhaps hidden by binary classifications. Furthermore, the rich

organizational dataset allows us to illustrate the countervailing mechanisms of ability and algorithmic aversion that drive the inverted U-shape.

We also make an important theoretical contribution relative to how domain experience affects performance of algorithm-augmented judgment. First, we build on the emerging literature on the complementarity of humans with algorithms (Choudhury, Starr, and Agarwal 2020, Raisch and Krakowski 2020) and argue that domain experience is an important complement for humans working with algorithms. Yet highly experienced humans may suffer from an "expert paradox" effect (Liu 2017), making them more averse to actually use algorithmic input (Dietvorst, Simmons, and Massey 2015, 2016). Given these two countervailing mechanisms, we theorize that an *intermediate level of domain experience* actually increases the likelihood of taking (correct) advice and leads to optimal performance.

Finally, we contribute to the organizations and technology strategy literature by demonstrating that although experience is necessary for successfully leveraging new technologies (Levitt and March 1988, Cohen and Levinthal 1990, Christensen and Knudsen 2013, Greenwood *et al.* 2019), this relationship may actually reverse for very high levels of experience. We also summarize scope conditions of our study that suggest a rich agenda for future research.

Theory Development

We theorize a framework in which domain experience influences algorithm-augmented performance via two countervailing mechanisms: (1) domain experience increases a worker's *ability* to accurately assess the advice of an algorithmic tool, but (2) also increases a worker's *aversion* to accepting advice generated by the algorithmic tool, leading them to reject correct algorithmic advice.

Algorithmic Decision Tools in Modern Organizations

Before delving into the theoretical framework, it is important to establish the definition of algorithmic decision tools. When we use the word "algorithm" in this paper, we are specifically referring to a tool that takes a set of

information as input, then systematically parses that information to make an assessment or decision recommendation as output.[2]

Although algorithm-augmented work is not new, recent developments in ML technologies have produced algorithms that can accurately automate tasks in ways that were not previously possible (Autor 2015). One example of a tool that was not previously possible is the natural language processing algorithmic tool studied in the research context of this paper. ML methods have created a new class of algorithmic tools by statistically inducing patterns from a training set of correctly labeled examples in data, allowing them to replicate correct classifications without explicit coding (Choudhury, Allen, and Endres 2020). These classification algorithms are being used to build decision-augmenting tools in many fields. For example, ML-built algorithmic tools are being deployed to help doctors read medical images (Arthur and Hossein 2019), to help managers make hiring decisions (Miller 2015, Cowgill 2018a), to help police officers decide where and how to police (Wang *et al.* 2013), and to help judges make bail decisions (Cowgill 2018b, Kleinberg *et al.* 2018). The changing nature of tasks to which algorithms are applied, and the growing scope of jobs affected by these tools, warrants new thinking about structuring decision-making in organizations, as they can be augmented by algorithms (Simon 1991, Csaszar and Eggers 2013, Shrestha, Ben-Menahem, and von Krogh 2019, Raisch and Krakowski 2020).

Domain Experience and *Ability* to Assess Algorithmic Advice

Large literatures in cognitive psychology (Chase and Simon 1973, Simon 1991, Ericsson, Krampe, and Tesch-Römer 1993) and specialized human capital in strategy and economics (Becker 1962, Castanias and Helfat 1991, Harris and Helfat 1997) have shown productivity benefits from domain experience. We adopt the definition of domain experience as domain-specific knowledge obtained by focused practice within that domain (Ericsson, Krampe and Tesch-Römer., 1993). Domain-specific knowledge, organized in a form similar to an indexed encyclopedia, leads to superior and faster reasoning (Simon 1991). This allows experts to recognize relevant signals when facing complexity, perceive meaning more rapidly, and react to stimuli appropriately (Chase and Simon 1973, Simon 1991, Salas, Rosen, and DiazGranados 2010). Recent empirical work also confirms that

[2] We are *not* referring to the algorithms that build the algorithmic tools. For example, ML algorithms can build classification models from large sets of training data, and those classification models can be used as algorithmic tools. We are referring to the latter, which is why we refer to "ML-built" or "ML-developed" algorithmic tools.

experienced workers are more likely to effectively adopt new practices. For example, physicians who had more procedural experience were more likely to quickly adopt new best practices for using stents to treat patients with stable coronary artery disease (Greenwood *et al.* 2019). This work theorizes that the expert physicians implemented the new practices more quickly because of decreased cognitive load and the ability to devote greater attention to the new information.

While domain experience increases task performance in general, domain experience is *especially* salient for assessing the quality of algorithmic advice. Domain experience allows a fuller grasp of context, which "makes focused perception possible, understandable and productive" (Partha and David 1994, pp. 493). For example, more experienced radiologists are able to detect subtle cues in X-ray images that less experienced radiologists cannot perceive (Lesgold *et al.* 1988). This grasp of context allows workers augmented by algorithms to efficiently and accurately perceive the quality of the algorithm's advice—and to know how to integrate the algorithm appropriately into the task as a whole. More broadly, this argument is central to the idea of vintage-specific human capital literature and the seminal paper by Chari and Hopenhayn (1991). In their model, learning by doing is a key complement to productively adopting new technology vintages. A recent experiment confirmed that domain experience was critical to algorithm-augmented work (Choudhury, Starr, and Agarwal 2020). Experimental subjects, who were novices in the task being performed, used an algorithmic tool to examine patents and identify relevant prior art. Without access to expert patent examiners with domain knowledge, participants in the experiment were unable to frame their searches to correctly interpret and leverage the advice generated by the ML-based algorithmic tool.

Yet the relationship between domain experience and the ability to interpret the quality of algorithmic advice is likely not linear. Some of the earliest work on the theory of human capital posited that knowledge and skills increase at a decreasing rate (Becker 1962). More specifically, the literature on learning by doing has documented diminishing returns to experience (e.g., Foster and Rosenzweig 1995) and in the more recent literature, Mithas and Krishnan (2008) document diminishing returns to *IT experience*, an insight relevant in the context of our study.

Taken together, empirical and theoretical work from disparate literatures suggest that domain experience should complement a worker's ability to interpret and accurately assess the quality of advice of an algorithmic

tool, and it should execute the human portion of a task. But there are diminishing returns to the contribution of domain experience to this ability, such that the marginal increase in ability is greater for workers with lower levels of experience. Thus, the literature leads us to expect that workers with more domain experience have more ability to accurately assess algorithmic advice—but this relationship has diminishing returns.

Domain Experience and *Aversion* to Algorithmic Advice

Although workers with more domain experience may be better equipped to assess the quality of algorithmic advice, they may also be more averse to accepting that advice. The term "algorithmic aversion" was coined by Dietvorst et al. (2015), but the literature on human distrust of algorithms dates back to as early as Meehl (1954), and it has been confirmed across many contexts (Grove and Meehl 1996, Grove *et al.* 2000, Sanders and Manrodt 2003, Fildes and Goodwin 2007, Vrieze and Grove 2009, Dietvorst, Simmons, and Massey, 2016, Christin, 2017).

Algorithm aversion seems to be especially salient for experts. One early study on the topic found that experts[3] tended to use helpful decision rules less than those with less experience and, consequently, they exhibited worse judgment (Arkes et al. 1986). More recently, Logg et al. (2019) confirmed these results in Study 4 of their paper (i.e., "Decision maker expertise"). They report that whereas laypeople placed more weight on algorithmic than human advice, experts[4] heavily discounted all advice sources; they preferred their own judgment over advice from an algorithm or from another human advisor. As a result, their forecasting performance suffered.

Discussing these results, Logg, Minson, and Moore (2019) attribute algorithmic aversion to mechanisms that explain experts' greater tendency to reject advice from both humans and algorithms—egocentrism (Soll and Mannes 2011) and individuals' overconfidence in their own judgment (Gino and Moore 2007, Logg, Haran, and Moore 2018). In their third experiment (i.e., "Role of the Self"), Logg et al. (2019) compare algorithmic advice to both advice from other human participants and to the self-judgment of participants, reporting that individuals were more confident in their own estimates than fellow participants. This reiterates prior empirical findings that individuals pay disproportionate credence to their own judgment. Prior literature also suggests that this mistrust

[3] Expertise was measured by a questionnaire about baseball (the relevant topic in the experiment)
[4] Experts in this experiment were defined as "professionals whose work in the field of national security for the U.S. government made them experts in geopolitical forecasting".

of algorithmic advice by experts may result from biased assimilation of information by experts (Liu 2017)[5]. In the context of the U.S. National Institutes of Health, Li (2017) showed that expert evaluators[6] were—ironically—both better informed and more biased about the quality of projects in their own areas. Experts' egocentric discounting of others' opinions has been attributed to differential information, namely the notion that experts have privileged access to their internal reasons for holding their own opinions, but not to the advisors' internal reasons (Yaniv and Kleinberger 2000). Teplitskiy et al. (2019) confirm a related idea by showing that experts[7] are less likely to accept advice, arguing that experts, unlike novices, are likely to have very fine-grained maps of intellectual space and may discount out-group information. And although those with more experience tend to make more accurate evaluations, this can be offset by the tendency to overestimate the confidence interval of their predictions (McKenzie, Liersch, and Yaniv 2008).

Several studies also offer explanations why experts' general advice aversion can be especially salient when advice is generated by algorithms. For example, Yeomans et al. (2019) find that although algorithmic systems outperform humans in making recommendations, people often choose not to rely on these recommender systems. This aversion partly stems from the fact that people believe the human recommendation process is easier to understand. People are generally averse to accepting recommendations from systems they cannot understand or cannot control (Herlocker *et al.* 2004). This has been observed by the resistance of clinical experts to diagnostic algorithmic decision rules, despite the superior performance of the decision rules (Grove and Meehl 1996). Experts with vast arrays of domain knowledge (often mistakenly) feel that they have access to important information unaccounted for by the algorithm, resulting in mistrust of the algorithmic output.

In summary, prior research has found that aversion against algorithmic advice is more prevalent among people with more domain experience. This is partly explained by experts' aversion to advice in general, but is especially salient for advice generated by algorithms.[8] To put it succinctly, workers with greater domain experience are more averse to using an algorithmic tool's advice.

---

[5] Expertise in laypeople was measured by multiple-choice questionnaires about domain-specific topics
[6] Expertise of evaluators was measured by the proximity of their previous academic papers to the papers being evaluated
[7] Expertise is measured using scientists' citations
[8] Though not salient in our context, it is worth noting that in other contexts there may be additional reasons for resistance against algorithms (Kellogg, Valentine, and Christin 2020). For example, new technologies and practices can disrupt existing routines and shift power dynamics and professional boundaries within organizations (Barley 1986, Edmondson, Bohmer, and Pisano 2001). Therefore, highly experienced workers may have more to lose to new technology that potentially

Domain Experience and Algorithm-augmented Performance (Relative to Self-performance)

Our study compares the performance of the individual when employing one's own judgment ("self-performance") to the performance of the individual when employing an "algorithm-augmented" judgment. We theorize that (relative to self-performance) algorithm-augmented performance increases as a function of increased *ability* to accurately assess algorithmic advice and decreases as a function of increased *algorithmic aversion*. To recap, our prior theorizing states that the ability to accurately assess algorithmic advice increases as a function of domain experience (at a decreasing rate), yet algorithmic aversion is more prevalent among high experience workers. Given this, we theorize that overall algorithm-augmented performance (relative to self-performance) will first rise with domain experience, then fall. The tipping point in this inverted U-shape is where the negative effect (i.e., domain experience increases aversion) overwhelms the diminishing positive effect (i.e., domain experience increases ability). For a visualization of the theorized relationships, see Figure 1. Bringing it all together, we hypothesize:

> *Hypothesis: Relative to self-performance, algorithm-augmented performance has an inverted U-shaped relationship with domain experience—marginally increasing for low experience workers and marginally decreasing for high experience workers.*

==========
INSERT FIGURE 1 ABOUT HERE
==========

Research Context and Experimental Design

Our study investigates the implementation of a new algorithmic tool within the context of corporate information technology (IT) work. IT is an ideal setting for our work because it cannot yet be fully automated by algorithms; humans must interface with algorithms in a "hybrid" work process (Shrestha, Ben-Menahem, and von Krogh 2019).

threatens their livelihood, status, or identity (Kellogg 2014). With more to lose, experienced workers may discount the value of an algorithm's contribution. People tend to exhibit "prior attitude bias" in which they readily accept information consistent with their attitudes and actively oppose inconsistent information (Kunda 1990). Newly encountered information triggers a complex network of emotions, beliefs, values, and thoughts. Information that does not neatly fit into the network is scrutinized and more often rejected (Ditto and Lopez 1992). Often this process occurs without people realizing it, and they falsely believe themselves to be completely objective and bias free (Pyszczynski and Greenberg 1987).

We partnered with a large Indian technology company, TECHCO, that was running an in-house experiment to test the performance of its IT staff using a new ML-built algorithmic tool that was designed to automate a significant portion of workers' daily tasks. Using a within-subjects design, IT professionals with varying levels of IT experience were instructed to solve a set of help tickets using the new algorithmic tool, and we compared that to their performance on a set of help tickets they solved manually without the tool. Because we could not randomly assign domain experience to workers, we test how domain experience moderated performance (in terms of the number of tickets resolved) using the algorithm (treatment) vs. resolving the tickets manually using the old system (control) . In the following sections, we provide more detailed explanations of IT support work and the algorithm used by the workers. Then we describe the within-subjects experimental design and our empirical approach.

IT Work in TECHCO

TECHCO, a technology company with more than 100,000 employees, houses a large internal IT department of roughly 500 support staff members who oversee the maintenance of networked computer systems within the organization. As in many large organizations, users (non-IT employees) alert TECHCO IT of technical issues by submitting "help tickets." A user fills out a form to provide details about the issue, then submits the ticket for IT to resolve. IT staff spend the day working through a queue of tickets that request help on issues like resetting passwords, granting administrative access to network users, and fixing security problems.

At TECHCO, IT staff are divided into three ascending levels approximately based on IT-related work experience. These levels overlap in years of experience: Level 1 (0-6 years of experience in our sample), Level 2 (3-10 years of experience in our sample), and Level 3 (7-15 years of experience in our sample). Although the ticket assignment system cannot tell how difficult each particular ticket will be, tickets assigned tags that tend to be more difficult or require higher permissions are sent to higher-level staff. In addition to experience levels, IT staff are assigned to an Operating System (OS) track: "Wintel" (a combination of the words "Windows" and "Intel") or "Linux," or a hybrid of both.[9] Although many of the same principles apply to resolving problems in

[9] TECHCO uses systems based on both Windows/Intel and Linux

both operating systems, there can be subtle distinctions such as directory paths or server protocols that make it easier for an IT specialist specialized in one system or the other.

Like many other IT departments, TECHCO IT staff have access to a large internal database of more than 7,000 "runbooks"—sets of instructions to solve specific recurring problems. The runbooks guide IT staff through complicated problems they may infrequently encounter. With the correct runbook as a guide, it is relatively simple to follow the step-by-step instructions to resolve about 90% of the tickets within TECHCO. However, it is not always obvious from the description in the ticket which runbooks among the thousands to use. According to the IT staff, it's not uncommon to spend 30 minutes to find the right runbook. Once the correct runbook is identified, workers follow the steps laid out in the runbook to resolve the ticket. Even for relatively simple issues, manual ticket resolution involves multiple windows and clicks—logging into a server, identifying a user in a directory, and enabling the user (for an illustrative example of resolving a ticket manually, see Appendix Figure A1).

AutomateIT: TECHCO's Algorithm for Augmenting IT Work

To reduce the resource cost of manual IT ticket resolutions, TECHCO assigned a team of machine learning (ML) engineers to build an algorithmic IT process automation tool. The project received considerable internal support and funding because the goal was not only to build the tool for internal IT use, but to market and sell the tool to other companies as a product. The result of the project was the new "AutomateIT"[10] tool. The tool automates both of the core steps of the IT ticket resolution process: runbook search and runbook execution. First, the tool uses ML-trained algorithms to match submitted help tickets to a list of textually similar runbook solutions. After a human selects a runbook from the list, the tool is equipped with software that automatically executes the runbook to resolve the ticket.

Due to intellectual property protection concerns, the organization did not want us to share in detail how the algorithm was trained. However, we are permitted to give a superficial overview. The AutomateIT system used natural language processing (NLP) techniques to standardize, tokenize, and topic extract text from more than 1 million help tickets and the corresponding 7,000 runbook solutions. The tickets had each been labeled

[10] AutomateIT is a pseudonym.

with the correct runbook, and a model was trained using term frequency-inverse document frequency (TF-IDF) token scores and directional n-grams to relate the text data in the tickets to each runbook label. The trained model takes as input the text data of a help ticket, and it outputs a similarity score for each of the 7,000 possible runbooks. All runbooks above a certain similarity score threshold (e.g., 60%) are displayed to the human user, who then selects one of the runbooks to execute. Before executing the runbook, the user inspects the default parameter values of the automated runbook to see if the parameter inputs will correctly resolve the ticket. If not, they can adjust the parameter values (e.g., change a server address) to correctly resolve the ticket. Because it is difficult for the algorithm to fill in the correct parameter values for each runbook based on the limited and unstructured information in the ticket, humans play a valuable role in checking and modifying the parameters of the runbooks before they execute. For an illustrative example of AutomateIT ticket resolution, see Appendix Figure A2.

Description of Within-Subjects Experimental Design and Execution

The experiment was executed by a team at TECHCO tasked with evaluating the effectiveness of the new AutomateIT tool. Because it was not possible to randomly assign domain experience, we estimate domain experience as a moderating effect on performance given self-judgment (control) vs. algorithm-augmentation (treatment). To estimate a moderating effect using a small number of participants, we opted to implement a within-subjects design. In a within-subjects design, each participant is subjected to both the treatment and the control conditions, which yields a causal estimate if the treatment and control exposures can be considered independent (Charness, Gneezy, and Kuhn 2012). The within-subjects design, which enables the researcher to use participant fixed effects, has several strengths relative to between-subjects designs. For instance, internal validity does not depend on random assignment because each person serves as their own control. By reducing error, this offers a boost in statistical power when testing a moderating effect using a small number of subjects (Judd, Kenny, and McClelland 2001). Although the authors of this paper had input on the experimental design, TECHCO ultimately finalized and executed it. Given practical limitations at the company, the intervention was not a perfectly designed experiment. Throughout the paper, we highlight issues with the research design and how we address each issue.

The participants in the experiment were 154 TECHCO IT support staff members who volunteered to participate in the experiment. Volunteers were told the purpose of the experiment—to test a new algorithmic tool that will assist them in their work—and were strongly encouraged to participate in the experiment by their managers. In field interviews, participants unanimously expressed that they did not feel that their jobs were threatened by the tool, but rather viewed the tool as a welcome automation of routine tasks so they could focus on other aspects of their job. Given that these tasks represented only a fraction of the participants' overall job and the consensus that the tool would not affect their future employment status, we do not consider job substitution to be a salient factor in our context.

Each volunteer was given one hour of training on how to use the new AutomateIT tool prior to the experiment. In the training session, they were trained on how the tool worked, and the trainers explained that the purpose of the tool was to assist them in their day-to-day activity of resolving tickets. After the training, each participant was assigned to one of five experiment sessions, which took place over the course of a week. Among those who volunteered and received training, all but one participated (he/she called in sick). In each session, about 30 participants were given four hours in a proctored conference room to resolve the assigned tickets on their normal work laptop. The four-hour time limit was determined based on the normal production time it would take to resolve 8 tickets to avoid significant time pressure. Each participant was assigned tickets to resolve using both the manual ticket resolution system and the new AutomateIT tool. IT staff at TECHCO had used the manual system on a daily basis; but prior to the training session, none had used the AutomateIT tool or even knew of its existence. The tickets assigned within each session were different, to ensure that no specific answers would leak from one session to another. Proctors ensured that there was no communication between subjects and that no one looked at others' laptop screens.

In accordance with a within-subjects experimental design, each participant received four (control) "manual" self-performance tickets to be resolved using the old system and four (treatment) tickets to be resolved using the new augmenting AutomateIT system. Therefore, each participant was assigned only four *unique* tickets among the eight total assigned tickets.[11] This feature of the within-subjects experiment allows us to directly

[11] Several employees mistakenly received tickets that were not repeated in both manual and AutomateIT resolution modes (overall 126 tickets were not repeated). For example, they were given three tickets to solve in both manual and AutomateIT

compare AutomateIT tickets to manual tickets while completely controlling for differences between individual participants and differences between tickets. Everything about the recurring tickets was the same except for small changes in the problem description, which varied the parameters necessary for ticket resolution (e.g., user ID and IP address).

Using a within-subjects experimental design requires that treatment and control conditions are independent—yet in our experimental design, it is possible that a recurring ticket (e.g., resolving the same manual ticket that was already resolved using AutomateIT) could be easier to resolve due to learning effects. We took several steps to verify independence of treatment and control. First, we randomly determined the sequence of the assignment to resolve either the manual tickets or the automatic tickets first. Second, we control for recurring tickets in our models and verify that there is not a positive and significant learning effect. Third, we include a supplementary analysis in the appendix using a model with minimal functional form assumptions to test the independence of treatment and control on ticket resolution (Appendix Table A1). Fourth, in field interviews there was a consensus among participants that there were no major learning effects from receiving the same ticket twice. According to the participants and proctors, the algorithm's recommendation and solution emerged as if from a "black box." That is, because the AutomateIT tool did not display the steps that it followed to resolve a ticket, it did not reveal how to resolve a ticket manually when it executed a solution. And although resolving a ticket manually could give the participant a better understanding of the problem, they still would not be sure that the algorithmic tool's runbook description matches the steps they took to manually resolve the ticket. In other words, the "know-how" (i.e., the steps taken for ticket resolution) was often completely different for manual and AutomateIT ticket resolution.

The TECHCO proctors who ran the experiment created the tickets that were assigned to participants. These proctors were very familiar with the routine work of the IT support staff and crafted tickets based on past typical tickets (i.e., they slightly modified problem descriptions from previous tickets). In order to simulate normal working conditions (and to ensure the results were relevant for the actual daily work of the IT staff), a different set of tickets was created for each employee level. A pool of 40 Level 1 tickets were based on previous

resolution systems (six tickets), and the remaining two tickets were unique. We also ran a subsample analysis excluding these cases, which did not meaningfully affect the results (see Appendix Table A3, column 1).

typical Level 1 tickets, 40 Level 2 tickets were based on previous typical Level 2 tickets, and 40 Level 3 tickets were based on typical Level 3 tickets. Among each pool of potential Level 1, Level 2, and Level 3 tickets, 28 were "Wintel" OS problems, and 12 were "Linux" OS problems (40 total per level). For the experiment, tickets from each of the three pools of tickets were randomly assigned to each participant based on their employee experience level (Level 1, 2, or 3).

Because the company wanted to simulate daily working conditions by assigning each employee tickets that were within their "employee level," there is a potential issue when estimating the moderating effect of domain experience. Differences in performance across employees with varying levels of domain experience could be due to the fact that different problems were assigned to employee levels 1, 2, and 3—which correlates with years of domain experience. We use a variety of approaches throughout the paper to ensure that the different ticket assignments the three employee levels are not driving the results. Here we highlight three specific ways we address this issue. First, it is important to keep in mind that we are using a within-subjects design and that our models include participant and ticket-level fixed effects. Therefore, the relevant comparison is not performance across participants with varying levels of experience, but the *difference in algorithm-augmented and manual ticket resolution performance within each participant* across participants with varying levels of experience. Because each ticket is assigned in both automatic and manual resolution mode, if differences in the tickets assigned to each employee level are driving the results, it must be because the tickets somehow systematically affect participants' treatment of identical sets of manual vs. automatic tickets *differently*. Therefore, we do not need to assume that different tickets are not affecting performance differently across the range of experience—we only need the weaker assumption that the same set of tickets are not affecting the difference between manual and automatic tickets differently for the sets of tickets, within the different employee levels. Second, we confirmed that the experimental tickets used for each employee level (1, 2, and 3) were not significantly different in the rate of correct predictions by the algorithmic tool (all were ~90% correct). This suggests that, as designed by the proctors, the tickets were similar enough that the algorithm could provide the same accuracy to all participants regardless of their employee level and, therefore, would not be a meaningful confound (see Appendix Figure A3). Finally, we ran a complete subsample analysis using *only* Level 2 employees, thereby completely eliminating any potential confounding issues between employee levels, which we include in the robustness checks. This

analysis (presented in the results section) confirms that there was a statistically significant inverted U-shape across the range of experience that was not driven by the ticket assignment to the different employee levels.

AutomateIT Ticket Resolution

Each participant was assigned four tickets to resolve using the AutomateIT tool. This ticket resolution environment was exactly the same as the actual AutomateIT tool, except that in the experiment, the proctors checked to make sure that each ticket contained the correct runbook as one of the likely runbook solutions (this was not communicated to the participants).

In the AutomateIT ticket resolution mode, we can observe two outcomes of participant actions. Before opening the ticket, participants can view the list of tickets they are assigned, which includes a preview of the description of the problem. After they select a specific ticket, they can observe more details about the ticket, including the list of likely runbook solutions. At this point, they select a runbook—or if they do not think any of the options are correct, they may "release" the ticket, resulting in an unresolved ticket. The first outcome we observe is whether the runbook selected by the participant was the correct runbook—and whether that runbook matches the algorithm's top predicted solution (which may be correct or incorrect). However, it is possible to select the correct runbook but still fail to resolve the ticket. If the parameters of the runbook are incorrect and not corrected by the human participant, the ticket would remain unresolved despite the correct runbook selection. Therefore, the second outcome we observe is whether the participant actually resolved the ticket.

Manual Ticket Resolution

Each participant was assigned four tickets to resolve "manually" using the old ticket resolution system. In the experiment, participants used the same system they had already been using for their day-to-day work, and experimenters sent them the experiment tickets through the system. In this resolution mode, we observe only one relevant outcome since there is no algorithmic advice step. We simply observe whether each ticket is successfully resolved. The ticket is considered resolved if the user reports that he/she resolved the problem, which is the standard practice for the user's regular daily work.

Variable Descriptions

In this section, we describe the measures used throughout our analysis. Table 1 presents summary statistics for each of the variables we use, which includes a balance test between AutomateIT (treatment) and manual (control) tickets.

Dependent Variables

Our primary measure of performance is $Ticket\ Resolved_i$, a binary variable that is set to 1 for resolved tickets (otherwise 0).

Independent Variables

Our treatment compares tickets resolved using a participant's own judgment in the manual system (control) versus tickets resolved using AutomateIT, where the worker is provided with algorithmic advice (treatment). This comparison is captured by the binary variable $Is\ AutomateIT\ Ticket_t$ (1 for AutomateIT tool, 0 for the manual system). Our moderator variable of interest, domain experience, is measured by the participant's years of IT experience ($Years\ of\ IT\ Experience_i$).[12] Because the participants in our sample were so homogeneous and had such similar work experience, years of IT experience is our best proxy for domain experience because it is a measure of their exposure to solving domain-relevant problems (Ericsson, Krampe, Tesch-Römer 1993). Mithas and Krishnan (2008) argue that IT competencies are acquired through learning by doing, and thus that "technical competencies of IT professionals are reflected in their on-the-job IT experience" (Mithas and Mayuram S Krishnan 2008, pp. 417). They demonstrate that firms value the "IT experience" of IT workers, and it has additionally shown that firms value "IT experience" of IT workers (measured using number of years worked in an IT role) more than they value "non-IT experience" of such workers (Mithas and Krishnan 2008). They also point out that while firm-specific IT experience (i.e., familiarity with the IT systems of the firm) is valuable, general IT experience (i.e., IT experience in other companies) is also valuable given the standardization of hardware, software (e.g., use of enterprise resource planning systems and application service providers), and methodologies (e.g., capability maturity models, ISO) across firms. With this in mind, we argue that years of IT experience is an appropriate measure of IT domain experience.

Controls

[12] The construction of this measure is similar to how Greenwood et al. (2019) measure expertise. The authors measured the number of quarters the physician practiced medicine since graduation from medical school.

To separate domain expertise from firm-specific human capital, we also control for each participant's years of experience at the company ($Years\ at\ Company_i$). To measure whether the ticket matches the employee's OS track experience, the binary variable $Ticket\ Matches\ OS\ Track_{it}$ is set to 1 if the ticket is for a problem that matches the technology track of the employee (e.g., the employee is on the "Wintel" track and the ticket is a Windows/Intel related problem). We also mark whether each employee's OS track is Wintel, Linux, or a hybrid Linux/Wintel track. Lastly, we control for the $Ticket\ Order_{it}$ in which each ticket was opened (from first through eighth) and for whether each ticket was a $Recurring\ Ticket_{it}$ (1 if the participant had already seen the same ticket using the other resolution system; otherwise 0).[13]

==========
INSERT TABLE 1 ABOUT HERE
==========

In addition to the balance test for AutomateIT vs. manual tickets in the summary statistics in Table 1, we also check for balance across levels of IT experience in the Appendix, Table A2. Although this is not a traditional balance test (because we do not randomly assign IT experience but instead employ a "within-subjects" experimental design), Table A2 reveals a final issue with the experimental design that should be addressed: The ticket order of AutomateIT tickets are not evenly distributed across participants with varying levels of domain experience. Therefore, it is important that we control for $Ticket\ Order_{it}$ and $Recurring\ Ticket_{it}$ throughout the paper, as well as interact them with $Is\ AutomateIT\ Ticket_t$. In all models, we also interact both $Recurring\ Ticket_{it}$ and $Ticket\ Order_{it}$ with $Years\ of\ IT\ Experience_i + Years\ of\ IT\ Experience_i^2$ to ensure that the ordering of tickets unevenly distributed across the range of participant domain experience did not explain our results. This is related to the issue that in the experimental setup, some employees did not receive four recurring tickets (some received three tickets that recurred and two that did not). Therefore, aside from adding controls, we also ran our models without the participants who did not receive recurring tickets (Appendix Table A3, column 1), and the results remained unchanged. Because ticket ordering was more balanced across

[13] Because we did not have the exact time stamps for tickets that were manually released, we had to impute the order for some tickets. For these tickets, we randomized their order within the range of tickets with the same resolution mode. For example, if a participant had three AutomateIT tickets assigned at the beginning of their session and one ticket that was released and, therefore, had no time stamp, we would randomly assign that ticket order to be 1, 2, 3, or 4. This procedure eliminates systematic biases for ordering of released tickets and ensures that the *Recurring Ticket* variable is accurate.

domain experience for Level 2 employees, we also ran a subsample analysis using only Level 2 employees, which yielded results consistent with our main findings (see Appendix Table A3, column 2).

Empirical Approach

We estimate various specifications of the following equation using OLS:

$$Y_{it} = \beta_1 Is\ AutomateIT\ Ticket_t + \beta_2 Years\ of\ IT\ Experience_i + \beta_3 Years\ of\ IT\ Experience_i^2 + \beta_4 Is\ AutomateIT\ Ticket_t * Years\ of\ IT\ Experience_i * + \beta_5 Is\ AutomateIT\ Ticket_t * Years\ of\ IT\ Experience_i^2 + \gamma \boldsymbol{X} + \delta Is\ AutomateIT\ Ticket_t * \boldsymbol{X} + \alpha + \epsilon_{it},$$

where *i* indexes individual-level attributes, *t* indexes ticket-level attributes, and Y_{it} represents if the ticket was resolved. The effect of interest is captured by the terms β_4 and β_5, the quadratic fit of the influence of years of experience on solving AutomateIT tickets (relative to manual tickets). $\boldsymbol{X}$ represents a vector of controls: years at the company, whether the ticket matches the user's OS track, the user's OS track, the ticket order, and whether it is a recurring ticket. To adjust for imperfect random assignment in the order of manual and automatic tickets, the controls also include interactions between $Ticket\ Order_{it}$ and $Recurring\ Ticket_{it}$ with $Years\ of\ IT\ Experience_i + Years\ of\ IT\ Experience_i^2$. All controls are included twice in the model—once alone and once interacted with $Is\ AutomateIT\ Ticket_t$. Interacting $Is\ AutomateIT\ Ticket_t$ with each variable allows the equation to estimate different effects for tickets that were resolved manually vs. using the AutomateIT system. It is also important that we interact the controls with $Is\ AutomateIT\ Ticket_t$ to allow the controls to have different effects on the two groups. Lastly, α represents a vector of fixed effects, for the employee level (1, 2, or 3) and experimental session (of five possible sessions), participant-level fixed effects, and/or ticket-level fixed effects.

Results

Figure 2 compares the raw percentage of tickets resolved in the manual and AutomateIT resolution modes across the range of IT experience. Although the overall percentage of resolved tickets was statistically indistinguishable for tickets resolved using manual and AutomateIT (see Table 1), there was considerable heterogeneity across the range of experience. For tickets resolved using AutomateIT, there was a clear inverted U-shaped relationship between the years of IT experience and the percentage of tickets resolved. The same

relationship did not exist for tickets resolved manually. However, because the assignment of ticket order was not completely random (recall Table A2), it is possible that adding controls could change the results.

==========
INSERT FIGURE 2 ABOUT HERE
==========

Columns (1)-(3) in Table 2 display regression results that confirm the relationships displayed in Figure 2, after adding relevant controls and fixed effects. The main model in column (1) confirms that participants with more years of IT experience were more likely to accurately resolve AutomateIT tickets. It also confirms our hypothesis, showing that the effect changes signs as participants with high levels of experience are significantly less likely to resolve tickets. This is captured by the statistically significant quadratic relationship for AutomateIT tickets ($0.080 * Years\ of\ IT\ Experience_i - 0.005\ Years\ of\ IT\ Experience_i^2$). Notably, this relationship is not present for manual tickets. Column (2) confirms that the relationship is robust to problem fixed effects, thereby ruling out alternative explanations due to unobserved problem heterogeneity (though not quite statistically significant at $\alpha = 0.05$ in our small sample). Column (3) adds participant fixed effects, so the coefficient estimate tells us that there is a significant quadratic relationship between $Ticket\ Resolved_i$ and $Years\ of\ IT\ Experience_i$ for AutomateIT tickets *relative to* manual tickets *within* each participant. In other words, even accounting for differences between individuals, there is an inverted-U relationship between AutomateIT tickets and experience relative to manual tickets. Figure 3 displays the predictions of ticket resolution conditional on experience as predicted by the regression model in column (3), which provides a visual confirmation of the hypothesized inverted U-shape of algorithm-augmented performance (relative to self-performance) over domain experience.

==========
INSERT TABLE 2 ABOUT HERE
==========
==========
INSERT FIGURE 3 ABOUT HERE
==========

Robustness Check: Two Lines Test

Forcing a quadratic fit to data can be misleading and exhibit a high rate of false-positive hypotheses. To ensure this is not an issue in our model, we apply Simonsohn's (2018) two-lines test. This test estimates two regression lines—one for low and one for high values of *x*—without imposing a quadratic functional form assumption.

Figure 4 displays the results of our test, which included all the variables and controls as the model in Table 2, column (1). The test confirms that there is a sign change in probability of resolving an AutomateIT ticket for low vs. high levels of experience in our participants.

==========
INSERT FIGURE 4 ABOUT HERE
==========

Robustness Check: Random Forest Partial Dependence Plots

As an extra validation of the inverted U-shaped relationship tested in our model, Figure 5 displays partial dependence plots for a random forest model trained on the data. Because machine learning algorithms like the random forest can flexibly fit a model while balancing bias and variance, it can serve as a check that the models we tested in Table 2 are a good fit of the data, and not simply being forced on the data by the researcher (for methodology and rationale, see Choudhury, Allen, and Endres 2020). For example, it is possible that there were hidden nonlinear relationships or interactions between variables that we ignorantly did not include in our model, which may have changed the way we understand the relationship between $Ticket\ Resolved_i$ and $Years\ of\ IT\ Experience_i$. This method must be used with caution and may not be valid if variables are causal descendants of other variable in the model or of they are omitted variables (Zhao and Hastie 2018). However, we do not expect such issues in our experimental data. The random forest algorithm independently found the same inverted-U relationship between $Ticket\ Resolved_i$ and $Years\ of\ IT\ Experience_i$, which serves as further validation that we fit a reasonable model to the data.

==========
INSERT FIGURE 5 ABOUT HERE
==========

Robustness Check: Level 2 Subsample Analysis

We briefly return to the potential identification issue in assigning a different set of questions to each employee level. Our first step in checking this issue was to confirm with the proctors that all questions were similar levels of difficulty and then to confirm with the data that the rate of correct AutomateIT predictions was the same across levels of experience. As a final check, we ran a complete subsample analysis using *only* Level 2 employees. Although this narrows the data to a much smaller sample size, analyses using only these employees eliminates any

potential confounding issues between employee levels. We chose to examine Level 2 employees because they contain the largest sample of employees in the range of experience where the change in direction of the inverted U-shape takes place. Results from the regression model are included in the Appendix (Table A3, column 2). For visualization of the predicted effects of the regression model, see Figure 6. The analysis confirms that there was a statistically significant inverted U-shape across the range of experience that was not driven by the ticket assignment to the different employee levels.

==========
INSERT FIGURE 6 ABOUT HERE
==========

Mechanisms

Now that we have confirmed the hypothesized inverted U-shaped relationship for performance over the range of domain experience, we explore whether our proposed mechanisms of ability and aversion drive this relationship. To explore these mechanisms, we separately consider the phases of the ticket resolution process: algorithmic runbook recommendation (i.e., the algorithm's predicted most likely runbook solution), runbook selection, and parameter input (see Figure 7).

==========
INSERT FIGURE 7 ABOUT HERE
==========

Figure 8 shows that most (79%) of the errors of commission were due to "releasing" (i.e., not selecting any runbook) tickets, both for the low and the high domain experience participants. We propose that there are two potential reasons for releasing a ticket given a correct algorithmic recommendation: (1) due to lack of ability in understanding the problem and solution, so that it is difficult to evaluate whether the algorithmic advice is correct or useful; or (2) due to an unconscious or conscious rejection of the algorithm's recommendation due to an aversion against the algorithm's advice. According to our theory, we expect that the former is driving errors of commission for the low experience workers and the latter for the high experience workers.

==========
INSERT FIGURE 8 ABOUT HERE
==========

Next, we focus on rates of error and success among runbooks that were "attempted" (i.e., any runbook was selected). Figure 9 plots three lines that we dub "errors of commission" (i.e., a worker Type 1 error; rejecting runbook selection given correct algorithmic advice), "errors of omission" (i.e., a worker Type 2 error; incorrect runbook selection given incorrect algorithmic advice), and "correcting false positives" (i.e., correctly selecting a runbook despite incorrect algorithmic advice) for attempted tickets. The figure confirms a U-shaped relationship with low and high experience workers more likely to reject correct algorithmic advice ("error of commission"). These also coincide with higher rates of correcting false positives, indicative that both high and low experience workers were less compliant with algorithmic advice. Next, we explore different explanations for high and low experience participants' noncompliance with algorithmic advice.

==========
INSERT FIGURE 9 ABOUT HERE
==========

High and Low Experience Participants' Differences in Time Spent on Released vs. Attempted Tickets

If high experience participants are more likely to release tickets independent of their ability to solve the ticket, we would expect them to release the tickets relatively quickly, without much deliberation. Conversely, we would expect low experience participants to spend more time puzzling out the problem before they release it. Although we do not observe how much time a participant spends on a ticket before releasing it, we can observe the average amount of time it took others of the same employee level to resolve the ticket. Figure 10 displays the difference in the average amount of time participants spent on tickets that were released vs. attempted. The figure confirms that participants with fewer years of experience were much more likely to release tickets that took longer to solve, whereas participants with many years of experience relatively indiscriminately release tickets. This pattern is consistent with the theorized mechanisms driving the release of tickets by low vs. high experience workers.

==========
INSERT FIGURE 10 ABOUT HERE
==========

Domain Experience and Ticket Resolutions for Attempted Tickets

As another empirical test of the different mechanisms driving high vs. low experience participants, we examine how domain experience moderates the likelihood of ticket resolution for attempted tickets. According to our theorizing, we would expect that if we could somehow remove algorithm aversion for high experience participants, more domain experience would lead to relatively better performance with the AutomateIT tool. As a proxy for completely removing aversion, we consider the subset of tickets that were actually attempted—which we argue represents a sample of tickets that participants at least attempted to resolve. Our regression models show that more experience is positively and linearly related to a greater predicted probability of resolving the ticket. [14] The predicted marginal effects of the regression model are displayed in Figure 11.

==========
INSERT FIGURE 11 ABOUT HERE
==========

Qualitative Exploration of Mechanisms

To further explore the validity of our proposed mechanisms, we conducted field interviews with participants across a range of expertise. There was wide agreement among interviewees that low experience employees released tickets because they simply did not have enough experience to know what to do with the ticket. One participant with two years of experience said, "It might not be possible for us to [resolve the ticket] at the time because we are unaware of it." Usually they might get help from senior colleagues, but in the experiment, they just decided to release the ticket. Another worker with three years of experience emphasized the role of experience in the ability to leverage the tool: "It's just a honed experience. So the more you get experience into a particular technology, the more you are able to work on that." They then went on to say that their lack of experience explained why they released tickets in the experiment. It also explained why they were the most likely to fail to input the correct parameters even if they selected the correct runbook.

Highly experienced employees agreed that experience played a key role in the ability to use the tool to resolve tickets. Yet they were unsurprised that the high experience employees were relatively likely to release tickets, even when AutomateIT had recommended the correct runbook. It was not that they did not recognize

[14] The linear coefficient for the interaction *Is AutomateIT Ticket * Years of IT Experience* is just below the threshold of significance ($\alpha = 0.05$) when participant level fixed effects are included (t-stat = 1.85). This is largely due to a decreased sample size

the runbook or did not know what to do with it. Rather, it was difficult for them to be sure whether the AutomateIT tool's recommendation in the form of a simple label (e.g., "Start AWS Cluster") was actually the correct course of action. One participant with 13 years of experience said that when using the algorithm, "Certain information is not always available to make a decision whether a runbook is okay or not. So in these kinds of situations, if information is not available for us, we would just be executing the runbook blindly…We cannot blindly run the executor script or runbook." Another participant with 12 years of experience agreed: "We cannot go blindly before seeing anything and just execute the runbooks."

These high domain experience employees also perceived themselves as possessing a much deeper understanding of the intricacies and interconnectedness of the back-end systems than lower-level employees, and they said this is why they would more often release AutomateIT tickets. One participant explained that "When a ticket comes, a detailed ticket description will confuse an [inexperienced employee] who is just trained to go in, match the situation, click, and execute. But being a senior person, we have to go in and investigate...If we perform the runbook and production goes down, so what will be the impact? An inexperienced employee will never think like that." Another emphasized the doubt that the algorithmic tool would resolve the ticket without messing up interconnected software and systems: "And when we say experience, it comes from knowing the environment. If there is one thing, they are linked to other things…an experienced person has a very vast understanding and very vast picture of…where he can link multiple things. So those are the things where he gets a little doubt [that the algorithm will work]."

Our interpretation of these interviews confirmed that there were very different mechanisms driving the lower algorithm-augmented performance of low vs. high experience workers. The low experience workers simply did not know whether the runbook was an appropriate solution for the ticket. The high experience workers, however, possessed deep knowledge of the intricately interconnected IT systems and doubted that the algorithm would be as careful as a human manually assessing and resolving the ticket. They rejected the algorithm's advice not necessarily because it was wrong, but because they believed a human could do a better job. These qualitative explanations fit with our quantitative observations. For example, given the correct selection of a runbook, high experience workers rarely failed to input the correct parameters (unlike low experience workers).

Alternative Explanations

We briefly consider two alternative explanations that could explain the pattern we observe. First, it is possible that the low experience workers had false overconfidence, which is sometimes found in novices who have just begun to learn a new skill (Sanchez and Dunning 2018). We do not think this is likely considering the pattern of both high and low experience participants releasing tickets (as displayed in Figure 8). If the story were overconfidence, we would expect relatively more attempted tickets and fewer correctly resolved tickets. Second, it is possible that participants with different levels of experience were learning more quickly to use the algorithm or to trust its advice (similar to the mechanism for aversion proposed by Dietvorst et al. 2015). However, a wide battery of regressions and visualizations of learning over time showed no significant differences in learning across the levels of domain experience in our context (e.g., see Appendix Table A1; other results available upon request). The lack of any learning effects may be because of the short time period (four hours) in which the experiment was conducted.

Discussion and Conclusion

We began with the question of how a worker's domain experience moderates their performance when using algorithmic tools to complete a task (relative to using their own judgment in completing the task). This is a first-order question for understanding how decision-making and other work processes may change with the increasing prevalence of algorithmic tools within organizations such that workers might be increasingly asked to complement their own judgment with algorithmic advice. We theorized a framework in which domain experience moderates algorithm-augmented performance via two countervailing mechanisms. We argue that domain experience enhances the ability to assess algorithmic advice but also increases aversion to accepting such advice—leading to an overall inverted U-shape in algorithm-augmented performance over the range of domain experience.

We tested this hypothesis using data from a within-subjects experiment that assigned IT support workers to resolve "help tickets," both manually and using a new algorithmic tool. The algorithm was developed using ML methods, which matched the text in each help ticket to the most probable "runbook" solutions—a typical example of the new wave of algorithms finding applications in organizations. In summary, we find that both the low experience workers and high experience workers perform relatively poorly when working with the

algorithmic tool—a relationship that does not exist for tickets resolved manually. The inverted U-shaped relationship is driven by the tendency of both the low experience and the high experience workers to reject correct algorithmic advice (i.e., "errors of commission"). However, mechanisms driving the low and high experience workers are different. We show evidence consistent with the assertion that the overall inverted U-shape of the algorithm-augmented performance curve across domain experience is driven by both ability to assess accuracy of algorithmic advice and algorithmic aversion increasing with domain experience.

We showed that the decrease in performance for high experience workers exists *only* for tickets resolved using AutomateIT (not manually), indicating that there is something special about the AutomateIT tickets that particularly affects the high experience workers. We propose this is due to experts' algorithm aversion. We also document that low experience workers were more likely to release more difficult (time consuming) problems, while high experience workers released tickets relatively indiscriminately. Finally, we documented that for the tickets that were attempted, the relationship between performance and domain experience was positive and linear. These indications from the data are consistent with our theorized predictions. Taken together, the results confirm that domain experience is indeed necessary for the ability to accurately judge and use algorithmic assessments (Choudhury, Starr, and Agarwal 2020), but high levels of experience seem to be related to a greater aversion to accepting algorithmic advice (Logg, Minson, and Moore 2019).

The generalizability of our empirical study has important limitations and scope conditions, suggesting a rich agenda for future research. First, this study provides just one specific task within the context of a single domain. Future studies could explore how varying task or domain-specific attributes. These may include varying the stakes of decision-making (low in our context), amount of uncertainty (low in our context), the transparency of the algorithm (low in our context), or different worker attributes (e.g., artists vs. IT workers). Relatedly, how receptive an expert is to algorithmic advice may also change greatly based on how *different* the advice is from the expert's priors—though we believe we give a safe estimate of this since in our context the algorithm was relatively accurate and in line with workers' priors. Second, our study was bounded in time—a four-hour experimental session. Future work should explore whether the observed effects persist after several days, weeks, or months. Third, while a strength of this study is that it employs an objective measure of "accuracy" in judgment (i.e., whether the ticket was resolved or not), we acknowledge that other organizational decisions may

not lend themselves to an objective measure of accuracy—especially for relatively uncertain tasks. Finally, in our context, we do not observe whether high experience workers' algorithm aversion is because they do not trust the algorithm's advice or because they do not trust the black-box algorithm to execute it correctly. This is a potentially important distinction that can be left to future research to determine which factor is more salient, and whether increasing the transparency of algorithms would reduce expert aversion.

Another related area ripe for future research is how human-algorithm complementarity changes based on the origins of the algorithm. For example, does a human's view of the algorithm change if it was designed by domain experts vs. by data scientists or by teams of both? It is possible that our measure of domain experience is correlated with the age of the worker, though we are less concerned with this given that all workers were in the age group of 23 to 35 years (the company did not give us exact worker ages).

Limitations notwithstanding, this paper makes several meaningful contributions toward understanding complementarities between humans and algorithms in organizations. Although substantial previous literature compares the performance of humans vs. algorithms, this paper contributes to the nascent conversation on complementarities between them (Raj and Seamans 2019, Shrestha, Ben-Menahem, and von Krogh 2019, Choudhury, Starr, and Agarwal 2020, Raisch and Krakowski 2020). Our study indicates that there are two important countervailing forces to be aware of when humans use algorithms in their work: *ability* and *algorithmic aversion*. Although workers with domain experience have a greater ability to leverage an algorithm in their work, highly experienced workers may not accept algorithmic advice due to greater algorithm aversion. This insight helps reconcile literatures that emphasize the role of either ability or algorithmic aversion separately. More generally, this contribution sits at the intersection of the technology strategy and the technology in work and organizations literatures. The optimal intermediate knowledge base recognizes two seemingly opposing perspectives: that experience is necessary for productive adaptation to new technologies and practices (Cohen and Levinthal 1990, Greenwood *et al.* 2019, Choudhury, Starr, and Agarwal 2020) and that experience can make individuals and organizations averse to technological change or to taking advice (Barley 1986, Levitt and March 1988, Kellogg 2014, Logg, Minson, and Moore 2019, Teplitskiy *et al.* 2019). To reiterate, an important theoretical contribution (that we validate empirically) relates to the fact that an *intermediate level of domain experience* increases the likelihood of taking (correct) advice and leads to optimal performance.

We also contribute to the literature on algorithmic aversion/appreciation. We confirm previous findings (based on lab experiments) that in organizational settings, experts may reject advice more often than nonexperts (Arkes, Dawes, and Christensen 1986, Logg, Minson, and Moore 2019). But rather than just classifying workers as binary "experienced" or "inexperienced," we can explore the forces at play across a gradient of experience—which allows us to uncover the inverted U-shaped relationship perhaps hidden by binary classifications. In doing so, we make a nuanced contribution to the literature on algorithmic aversion/appreciation. In this literature (e.g., Logg et al. 2019) algorithmic aversion/appreciation has been often measured using whether or the lab participant "ignores" the advice given.[15] We demonstrate that not only experts but also novices can "ignore" algorithmic advice; but this ignoring of algorithmic advice likely relates to different mechanisms—that is, the ability to judge accuracy of advice (for relative novices) and algorithmic aversion (for experts). In other words, "ignoring" algorithmic advice (as for the novices in our sample) could be related to the inability to judge algorithm accuracy, rather than the classic view of algorithmic aversion. Future research can further investigate mechanisms that cause novices/experts to accept or ignore algorithmic advice.

In conclusion, this study employs a within-subjects experimental design to provide (to the best of our knowledge) the first set of empirical results on how domain experience affects algorithm-augmented performance, relative to self-performance. Our theoretical framework highlights two countervailing mechanisms—the ability to assess the accuracy of algorithmic advice and algorithmic aversion—that might differently affect worker performance along a continuum of domain experience. While the algorithmic aversion literature and the emerging literature on complementarities between human capital and algorithmic tools have independently discussed the effect of domain experience on algorithm-augmented performance, we contribute to this emerging conversation by synthesizing these two perspectives and then by theorizing and demonstrating that an intermediate level of domain experience might be optimal for algorithm-augmented decision-making. Our findings have implications for how organizations should think about where (i.e., for what level of workers) to augment decision-making with algorithms within the organization, as well as the kinds of organizations that may

[15] As an example, Logg et al. (2019, p. 92) use a variable entitled "weight on advice" that takes the value of 0% when a participant ignores advice.

be most likely to benefit from algorithm-augmented decision-making based on the range of worker domain experience.

REFERENCES

Agrawal, A., Gans, J. and Goldfarb, A. (2018) *Prediction Machines: The simple economics of artificial intelligence*. Harvard Business Press.

Arkes, H. R., Dawes, R. M. and Christensen, C. (1986) 'Factors influencing the use of a decision rule in a probabilistic task', *Organizational Behavior and Human Decision Processes*, 37(1), pp. 93–110. doi: 10.1016/0749-5978(86)90046-4.

Arthur, F. and Hossein, K. R. (2019) 'Deep learning in medical image analysis: A third eye for doctors', *Journal of stomatology, oral and maxillofacial surgery*.

Autor, D. H. (2015) 'Why Are There Still So Many Jobs? The History and Future of Workplace Automation', *Journal of Economic Perspectives*, 29(3), pp. 3–30. doi: 10.1257/jep.29.3.3.

Barley, S. R. (1986) 'Technology as an occasion for structuring: Evidence from observations of CT scanners and the social order of radiology departments', *Administrative Science Quarterly*, 9780521862(Hs 05004), pp. 83–100. doi: 10.1017/CBO9780511618925.005.

Becker, G. S. (1962) 'Investment in human capital: A theoretical analysis', *Journal of Political Economy*.

Brynjolfsson, E., Mitchell, T. and Rock, D. (2018) 'What Can Machines Learn and What Does It Mean for Occupations and the Economy?', *AEA Papers and Proceedings*, 108, pp. 43–47. doi: 10.1257/pandp.20181019.

Castanias, R. P. and Helfat, C. E. (1991) 'Managerial resources and rents', *Journal of Management*.

Chari, V. V. and Hopenhayn, H. (1991) 'Vintage Human Capital, Growth , and the Diffusion of New Technology', *Journal of Political Economy*, 99(6), pp. 1142–1165.

Charness, G., Gneezy, U. and Kuhn, M. A. (2012) 'Experimental methods: Between-subject and within-subject design', *Journal of Economic Behavior and Organization*. Elsevier B.V., 81(1), pp. 1–8. doi: 10.1016/j.jebo.2011.08.009.

Chase, W. G. and Simon, H. A. (1973) 'Perception in chess', *Cognitive Psychology*, 4(1), pp. 55–81. doi: 10.1016/0010-0285(73)90004-2.

Choudhury, P., Allen, R. T. and Endres, M. G. (2020) 'Machine Learning for Pattern Discovery in Management Research', *Strategic Management Journal*.

Choudhury, P., Starr, E. and Agarwal, R. (2020) 'Machine Learning and Human Capital Complementarities: Experimental Evidence on Bias Mitigation', *Strategic Management Journal*.

Christensen, M. and Knudsen, T. (2013) 'How Decisions Can Be Organized - and Why It Matters', *Journal of Organization Design*, 2(3), p. 41. doi: 10.7146/jod.8566.

Christin, A. (2017) 'Algorithms in practice: Comparing web journalism and criminal justice', *Big Data & Society*, 4(2), p. 205395171771885. doi: 10.1177/2053951717718855.

Cohen, W. M. and Levinthal, D. A. (1990) 'Absorptive Capacity : A New Perspective on Learning and Innovation', *Administrative Science Quarterly*, 35(1), pp. 128–152. doi: 10.1177/0149206310369939.

Cowgill, B. (2018a) 'Bias and Productivity in Humans and Algorithms: Theory and Evidence from Résumé Screening', *Columbia Business School*, 29, pp. 1–58. Available at: http://conference.iza.org/conference_files/MacroEcon_2017/cowgill_b8981.pdf.

Cowgill, B. (2018b) 'The Impact of Algorithms on Judicial Discretion : Evidence from Regression Discontinuities'.

Csaszar, F. A. and Eggers, J. P. (2013) 'Organizational decision making: An information aggregation view', *Management Science*, 59(10), pp. 2257–2277. doi: 10.1287/mnsc.1120.1698.

Dawes, R. M. (1979) 'The robust beauty of improper linear models in decision making', *American psychologist*.

Dietvorst, B. J., Simmons, J. P. and Massey, C. (2015) 'Algorithm Aversion: People Erroneously Avoid Algorithms After Seeing Them Err', *Journal of Experimental Psychology: General*, 143(6), pp. 1–13. doi: 10.1037/xge0000033.supp.

Dietvorst, B., Simmons, J. P. and Massey, C. (2016) 'Overcoming Algorithm Aversion: People Will Use Imperfect Algorithms If They Can (Even Slightly) Modify Them', *Ssrn*, 64, pp. 1155–1170. doi: 10.2139/ssrn.2616787.

Ditto, P. H. and Lopez, D. F. (1992) 'Motivated skepticism: Use of differential decision criteria for preferred and nonpreferred conclusions', *Journal of Personality and Social Psychology*, 63, pp. 568–584.

Edmondson, A. C., Bohmer, R. M. and Pisano, G. P. (2001) 'Disrupted Routines: Team Learning and New Technology Implementation in Hospitals', *Administrative Science Quarterly*, 46(4), p. 685. doi:

10.2307/3094828.
Ericsson, K. A., Krampe, R. T. and Tesch-Römer., C. (1993) 'The role of deliberate practice in the acquisition of expert performance', *Psychological review*.
Fildes, R. and Goodwin, P. (2007) 'Against your better judgment? How organizations can improve their use of management judgment in forecasting', *Interfaces*. INFORMS, 37(6), pp. 570–576.
Foster, A. D. and Rosenzweig, M. R. (1995) 'Learning by doing and learning from others: Human capital and technical change in agriculture', *Journal of political Economy*. The University of Chicago Press, 103(6), pp. 1176–1209.
Gino, F. and Moore, D. A. (2007) 'Effects of task difficulty on use of advice', *Journal of Behavioral Decision Making*. Wiley Online Library, 20(1), pp. 21–35.
Greenwood, B. N. *et al.* (2019) 'The role of individual and organizational expertise in the adoption of new practices', *Organization Science*, 30(1), pp. 191–213. doi: 10.1287/orsc.2018.1246.
Grove, W. M. *et al.* (2000) 'Clinical versus mechanical prediction: a meta-analysis', *Psychological assessment*.
Grove, W. M. and Meehl, P. E. (1996) 'Comparative efficiency of informal (subjective, impressionistic) and formal (mechanical, algorithmic) prediction procedures: The clinical-statistical controversy.', *Psychology, Public Policy, and Law*, 2(2), pp. 293–323. doi: 10.1037//1076-8971.2.2.293.
Harris, D. and Helfat, C. E. (1997) 'Specificity of CEO human capital and compensation', *Strategic Management Journal*.
Herlocker, J. L. *et al.* (2004) 'Evaluating Collaborative Filtering Recommender Systems', *ACM Transactions on Information Systems*, 22(1), pp. 5–53. doi: 10.1007/978-3-540-72079-9_9.
Judd, C. M., Kenny, D. A. and McClelland, G. H. (2001) 'Estimating and testing mediation and moderation in within-subject designs', *Psychological Methods*, 6(2), pp. 115–134. doi: 10.1037/1082-989X.6.2.115.
Kellogg, K. C. (2014) 'Brokerage Professions and Implementing Reform in an Age of Experts', *American Sociological Review*, 79(5), pp. 912–941. doi: 10.1177/0003122414544734.
Kellogg, K. C., Valentine, M. A. and Christin, A. (2020) 'Algorithms at work: The new contested terrain of control', *Academy of Management Annals*, 14(1), pp. 366–410. doi: 10.5465/annals.2018.0174.
Kleinberg, J. *et al.* (2018) 'Human decisions and machine predictions', *The Quarterly Journal of Economics*, (January), pp. 237–293. doi: 10.1093/qje/qjx032.Advance.
Kunda, Z. (1990) 'The case for motivated reasoning', *Psychological Bulletin*, 108, pp. 480–498.
Lesgold, A. *et al.* (1988) 'Expertise in a complex skill: Diagnosing x-ray pictures', *American Psychological Association*.
Levitt, B. and March, J. G. (1988) 'Organizational Learning', *Annual Review of Sociology*. Annual Reviews 4139 El Camino Way, P.O. Box 10139, Palo Alto, CA 94303-0139, USA , 14(1), pp. 319–338. doi: 10.1146/annurev.so.14.080188.001535.
Li, D. (2017) 'Expertise versus Bias in Evaluation: Evidence from the NIH', *American Economic Journal: Applied Economics*, 9(2), pp. 60–92.
Liu, B. S. (2017) 'Knowledge, attitudes, and biased evaluation of science: Testing the expertise paradox', (September). Available at: https://www.researchgate.net/profile/Brittany_Liu/publication/320065323_Knowledge_attitudes_and_biased_evaluation_of_science_Testing_the_expertise_paradox/links/59cbbb8faca272bb050c5978/Knowledge-attitudes-and-biased-evaluation-of-science-Testing-the-expe.
Logg, J. M., Haran, U. and Moore, D. A. (2018) 'Is overconfidence a motivated bias? Experimental evidence.', *Journal of Experimental Psychology: General*. American Psychological Association, 147(10), p. 1445.
Logg, J., Minson, J. and Moore, D. (2019) 'Algorithm appreciation: People prefer algorithmic to human judgment', *Organizational Behavior and Human Decision Processes*. doi: 10.2139/ssrn.2941774.
McKenzie, C. R. M., Liersch, M. J. and Yaniv, I. (2008) 'Overconfidence in interval estimates: What does expertise buy you?', *Organizational Behavior and Human Decision Processes*, 107(2), pp. 179–191. doi: 10.1016/j.obhdp.2008.02.007.
Meehl, P. E. (1954) 'Clinical versus statistical prediction: A theoretical analysis and a review of the evidence.' University of Minnesota Press.
Miller, A. P. (2018) 'Want less-biased decisions? Use Algorithms', *Harvard Business Review*.
Miller, C. C. (2015) 'Can an Algorithm Hire Better Than a Human?', *The New York Times*.
Mithas, S. and Krishnan, M. S. (2008) 'Human capital and institutional effects in the compensation of information technology professionals in the United States', *Management Science*, 54(3), pp. 415–428. doi:

10.1287/mnsc.1070.0778.
Mithas, S. and Krishnan, Mayuram S (2008) 'Human capital and institutional effects in the compensation of information technology professionals in the United States', *Management Science*. INFORMS, 54(3), pp. 415–428.
Partha, D. and David, P. A. (1994) 'Toward a new economics of science', *Research policy*. Elsevier, 23(5), pp. 487–521.
Pustejovsky, J. E. and Tipton, E. (2018) 'Small-sample methods for cluster-robust variance estimation and hypothesis testing in fixed effects models', *Journal of Business & Economic Statistics*. Taylor & Francis, 36(4), pp. 672–683.
Pyszczynski, T. and Greenberg, J. (1987) 'Toward and integration of cognitive and motivational perspectives on social inference: A biased hypothesis-testing model', in Berkowitz, L. (ed.) *Advances in experimental social psychology*. New York: Academic Press, pp. 297–340.
Raisch, S. and Krakowski, S. (2020) 'Artificial Intelligence and Management: The Automation-Augmentation Paradox', *Academy of Management Review*, pp. 1–48. doi: 10.5465/2018.0072.
Raj, M. and Seamans, R. (2019) 'Primer on artificial intelligence and robotics', *Journal of Organization Design*.
Salas, E., Rosen, M. A. and DiazGranados, D. (2010) 'Expertise-based intuition and decision making in organizations', *Journal of Management*, 36(4), pp. 941–973. doi: 10.1177/0149206309350084.
Sanchez, C. and Dunning, D. (2018) 'Research: Learning a Little About Something Makes Us Overconfident', *Harvard Buisness Review*.
Sanders, N. R. and Manrodt, K. B. (2003) 'The efficacy of using judgmental versus quantitative forecasting methods in practice', *Omega*. Elsevier, 31(6), pp. 511–522.
Shrestha, Y. R., Ben-Menahem, S. M. and von Krogh, G. (2019) 'Organizational Decision-Making Structures in the Age of Artificial Intelligence', *California Management Review*, (July). doi: 10.1177/0008125619862257.
Simon, H. A. (1991) 'Bounded Rationality and Organizational Learning', *Organization Science*, 2(1), pp. 125–134.
Simonsohn, U. (2018) 'Two-Lines: A Valid Alternative to the Invalid Testing of U-Shaped Relationships With Quadratic Regressions', *Advances in Methods and Practices in Psychological Science*, pp. 1–32. doi: 10.2139/ssrn.3256708.
Teplitskiy, M. *et al.* (2019) 'Do Experts Listen to Other Experts? Field Experimental Evidence from Scientific Peer Review'. Available at: https://www.hbs.edu/faculty/Pages/item.aspx?num=56067.
Vrieze, S. I. and Grove, W. M. (2009) 'Survey on the use of clinical and mechanical prediction methods in clinical psychology.', *Professional Psychology: Research and Practice*. American Psychological Association, 40(5), p. 525.
Wang, T. *et al.* (2013) 'Learning to detect patterns of crime', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8190 LNAI(PART 3), pp. 515–530. doi: 10.1007/978-3-642-40994-3_33.
Yaniv, I. and Kleinberger, E. (2000) 'Advice Taking in Decision Making: Egocentric Discounting and Reputation Formation', *Organizational Behavior and Human Decision Processes*, 83(2), pp. 260–281. doi: 10.1006/obhd.2000.2909.
Yeomans, M. *et al.* (2019) 'Making sense of recommendations', *Journal of Behavioral Decision Making*, (January). doi: 10.1002/bdm.2118.
Zhao, Q. and Hastie, T. (2018) 'Causal Interpretations of Black Box Models'.

Table 1. Summary Statistics

Variables	AutomateIT	Manual	tStat	pVal
Dependent Variables				
Ticket Resolved	0.703	0.68	0.866	0.387
	(0.457)	(0.467)		
Independent Variable				
Years of IT Experience	6.026	6.026	0	1
	(3.9)	(3.9)		
Control Variables				
Years at Company	2.248	2.248	0	1
	(2.154)	(2.154)		
OS Track: Linux	0.353	0.353	0	1
	(0.478)	(0.478)		
OS Track: Wintel	0.484	0.484	0	1
	(0.5)	(0.5)		
OS Track: Hybrid Linux/Wintel	0.163	0.163	0	1
	(0.37)	(0.37)		
Ticket Matches OS Track	0.6	0.603	-0.117	0.907
	(0.49)	(0.49)		
Ticket Order	4.338	4.662	-2.474	0.013
	(2.171)	(2.399)		
Recurring Ticket	0.428	0.467	-1.379	0.168
	(0.495)	(0.499)		
Level 1 Employee	0.392	0.392	0	1
	(0.489)	(0.489)		
Level 2 Employee	0.281	0.281	0	1
	(0.45)	(0.45)		
Level 3 Employee	0.327	0.327	0	1
	(0.469)	(0.469)		
AutomateIT-specific Variables				
Correct Runbook Recommendation by Algorithm	0.902			
	(0.298)			
Correct Runbook Selection by Participant	0.796			
	(0.403)			
Error of Omission (Acceptance of Incorrect Algorithmic Recommendation)	0.062			
	(0.242)			
Error of Commission (Rejection of Correct Algorithmic Recommendation)	0.142			
	(0.349)			
Correcting False Positive (Rejection of Incorrect Algorithmic Recommendation)	0.036			
	(0.186)			

Notes. Means displayed with standard deviations in parentheses. P-values are displayed for a standard t-test of the difference in means between values of variables for the sample of AutomateIT vs. manual tickets.

Table 2. OLS Regressions: Ticket Resolution and Handling Time

	Dependent Variable: Ticket Resolved		
	(1)	**(2)**	**(3)**
Is AutomateIT Ticket * Years of IT Experience	**0.080 (0.029)****	**0.067 (0.027)***	**0.085 (0.029)****
Is AutomateIT Ticket * Years of IT Experience Squared	**-0.005 (0.002)***	**-0.004 (0.002)+**	**-0.006 (0.002)***
Is AutomateIT Ticket	-0.163 (0.104)	-0.121 (0.088)	-0.043 (0.111)
Is AutomateIT Ticket * Controls	Yes	Yes	Yes
Years of IT Experience	0.004 (0.043)	-0.040 (0.043)	
Years of IT Experience Squared	0.002 (0.003)	0.004 (0.003)	
Controls	Yes	Yes	Yes
Employee Experience Level Fixed Effects	Yes		Yes
Experiment Session Fixed Effects	Yes		Yes
Problem Fixed Effects		Yes	
Participant Fixed Effects			Yes
Adj. R^2	0.086	0.141	0.221
Num. Obs.	1,224	1,224	1,224

Notes. This table displays OLS regression results, using *Ticket Resolved* (whether a ticket was resolved or not) as the dependent variable. Each column includes fixed effects for employee level, experimental session, participant, and/or the problem to be solved. Each column also includes controls for all the control variables listed in Table 1, plus *Ticket Order* and *Recurring Ticket* interacted with the *Years of IT Experience* + *Years of IT Experience*.[2] Asterisks indicate statistical significance at p-value cutoffs: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$, $^{+}p<0.1$. All regressions use standard errors clustered at the participant level, using "CR2" bias-reduced standard errors from the "estimatr" package in R (Pustejovsky and Tipton, 2018).

Figure 1. Theoretical Model Visualization

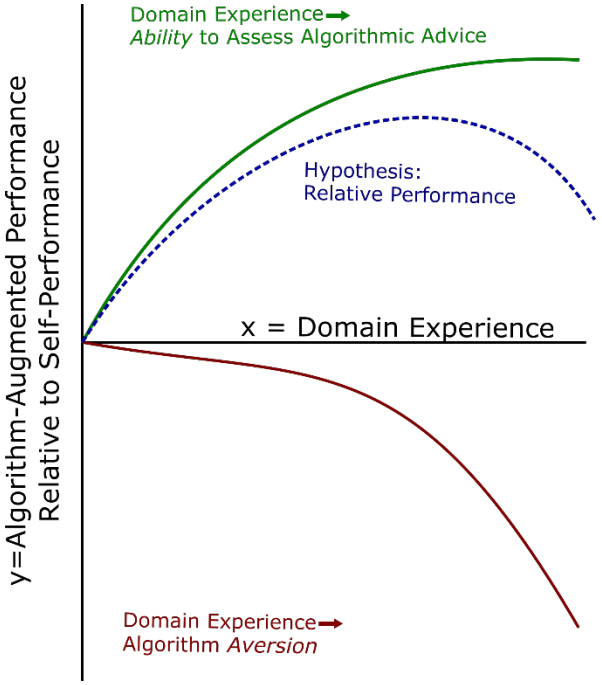
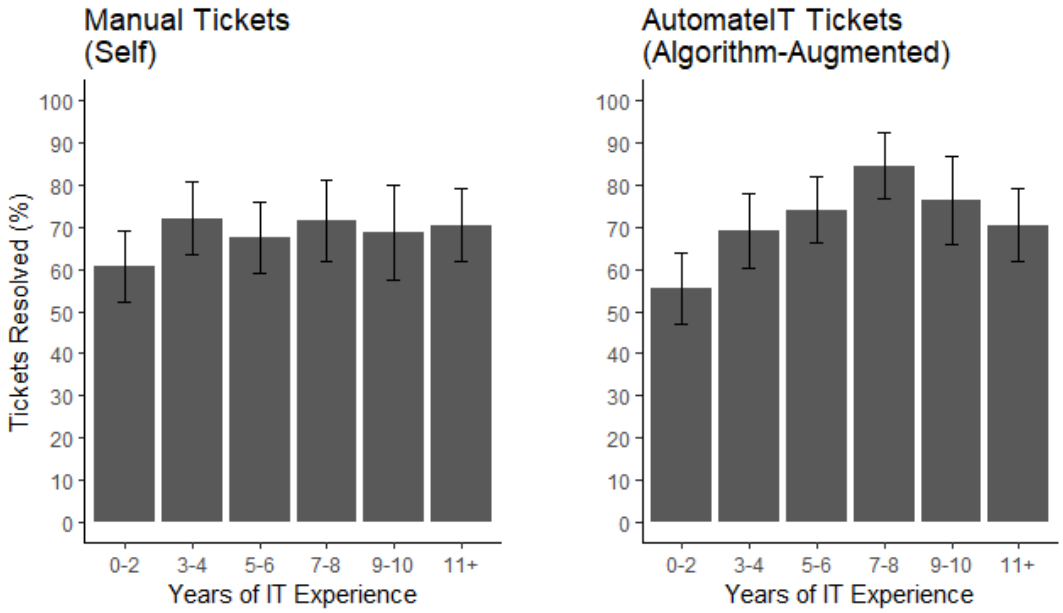
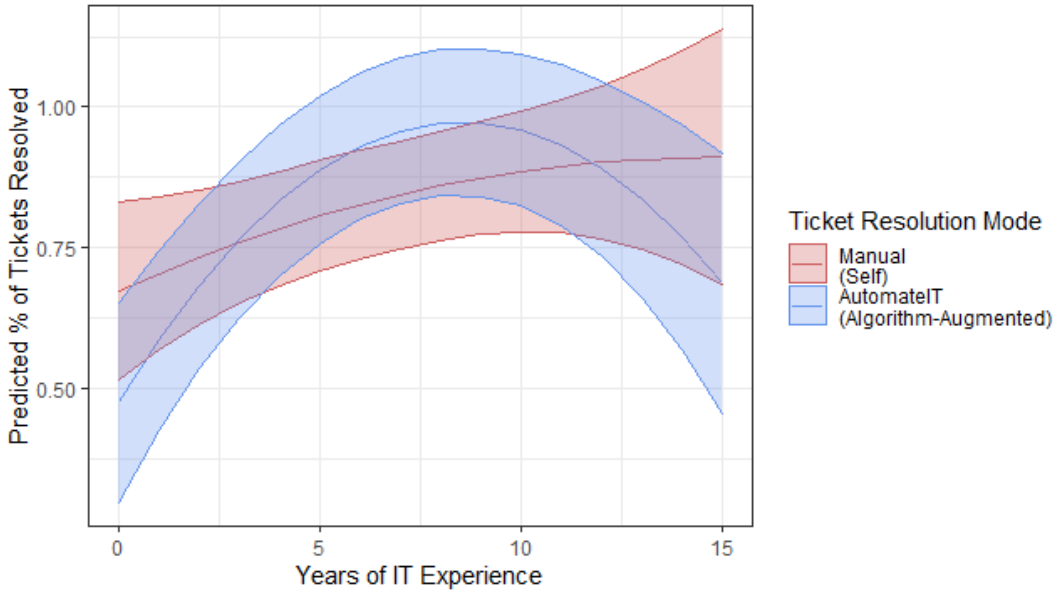


Figure 2. Raw Percentage of Tickets Resolved Across Range of Experience for Manual and AutomateIT Tickets



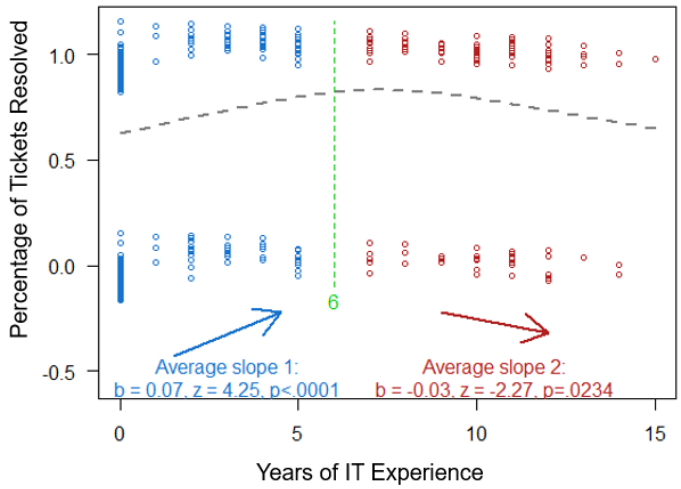
Notes. Gray bars represent the raw percentage of tickets that were resolved (95% confidence interval error bars) for participants with varying levels of IT experience. The left panel is for tickets resolved manually, and the right panel is for tickets resolved with algorithmic assistance from the AutomateIT tool.

Figure 3. Predicted Effects for Ticket Resolutions and Years of IT Experience



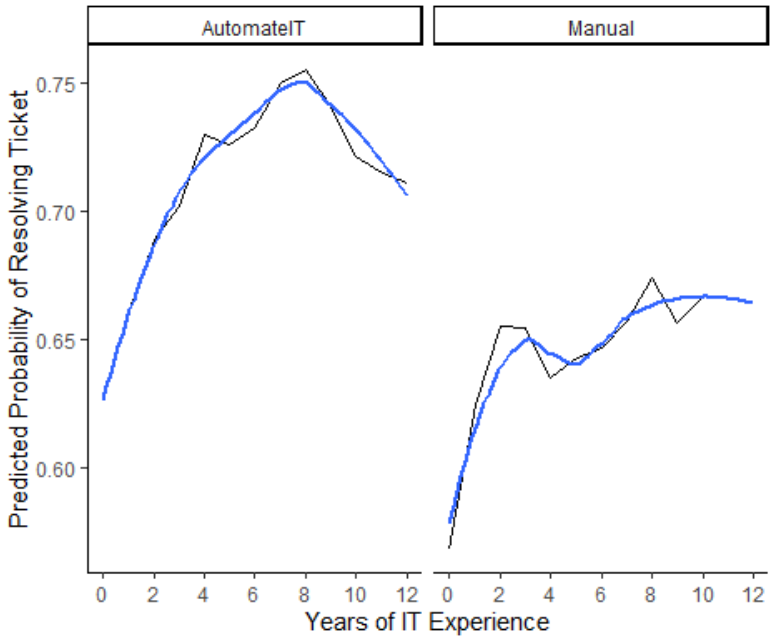
Notes. The figure displays predicted percentage of tickets resolved for AutomateIT and Manual tickets, conditional on Years of IT Experience. Predicted values were obtained using the model in Table 2, column (3). The figure was built using the "ggpredict" function from the "ggeffects" package in R.

Figure 4. Two Lines Test for AutomateIT Ticket Resolutions and Years of IT Experience



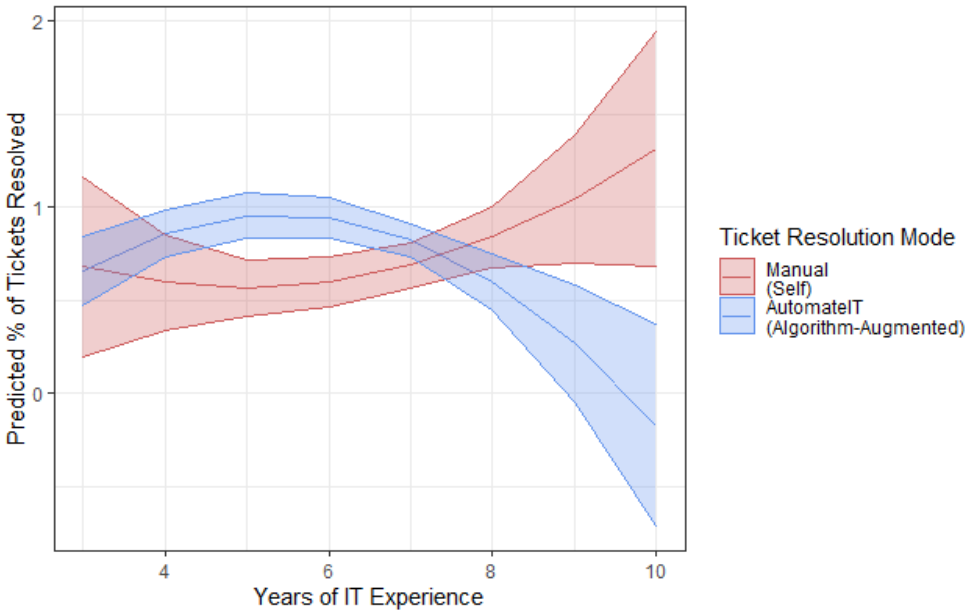
Notes. This figure displays results from Simonsohn's (2018) "two lines" test applied to the variables *Ticket Resolved* and *Years of IT Experience*. We ran the test including all the same variables and controls as in the model in Table 2, column (1). The test finds whether there is a significant relationship on each side of an algorithmically determined optimal breakpoint (represented by the dotted vertical green line).

Figure 5. Partial Dependence Plot for Random Forest Model of Experimental Data



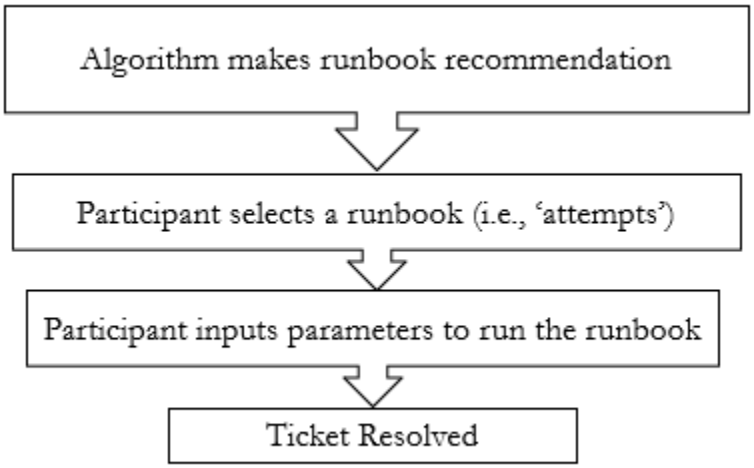
Notes. This figure displays partial dependence plots for a random forest model trained on the experimental data. The plots display the predicted probability of resolving a ticket conditional on *Years of IT Experience* for AutomateIT (left) and manual (right) tickets. The random forest model included all the variables and controls in Table 2, column (1). It was fitted using the "ranger" package in R, using repeated tenfold cross-validation (three repeats). We searched 40 random hyperparameter combinations. The optimally tuned model for the area under the curve (AUC) metric was: *mtry* = 5, *min. node. size* = 12, *splitrule* = extratrees. The cross-validation AUC of the model was 0.72, and the holdout test AUC was 0.67.

Figure 6. Level 2 Subsample Analysis: Predicted Effects for Ticket Resolutions and Years of IT Experience



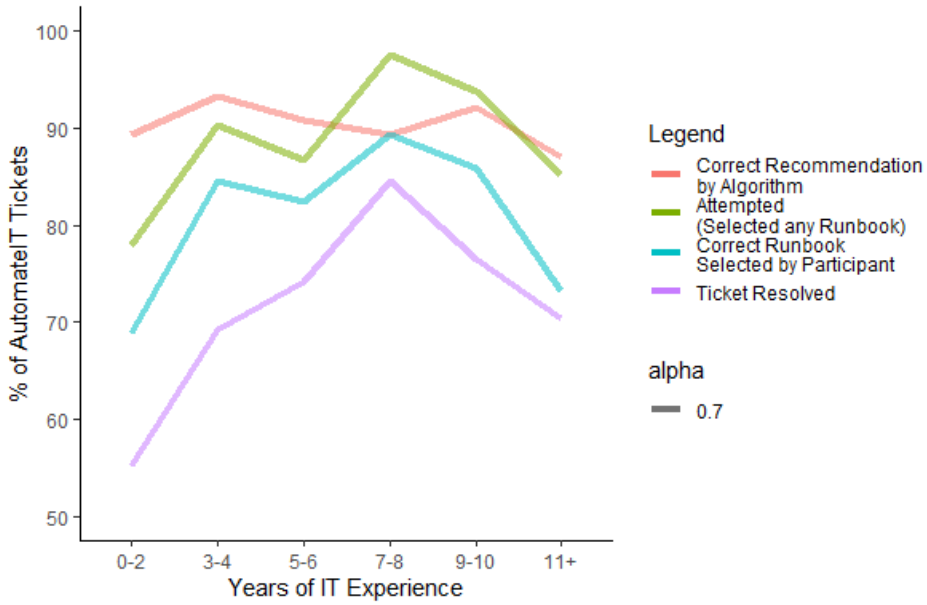
Notes. The figure displays predicted percentage of tickets resolved for AutomateIT and manual tickets, conditional on Years of IT Experience for the subsample of Level 2 employees only. The model used for these predictions was the same as the model used in Table 2, column 1. The regression model coefficients are displayed in Appendix Table A3, column 2. The figure was built using the "ggpredict" function from the "ggeffects" package in R.

Figure 7. Ticket Resolution Phases



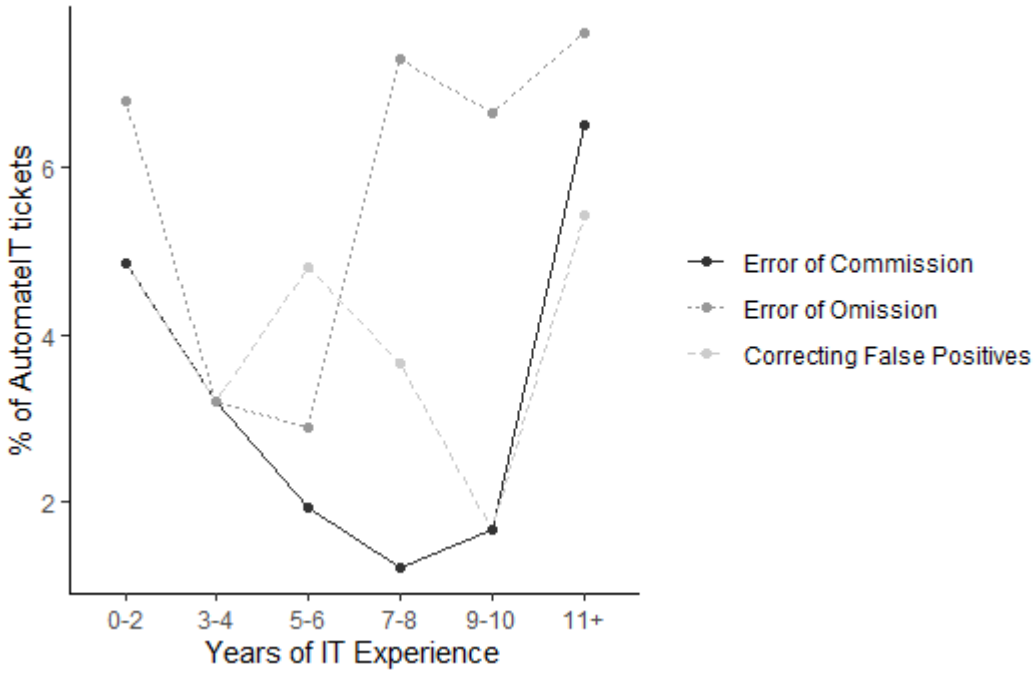
Notes. In order to successfully resolve a ticket, the participant must: (1) decide whether to select any runbook (attempt); and (2) input the correct parameters for the selected runbook in order to execute the runbook. If the participant correctly executes both steps, the ticket is resolved.

Figure 8. Rates of Correct Algorithmic Predictions, Runbook "Attempts," Correct Participant Runbook Selection, and Ticket Resolution



Notes. This figure displays the raw percentage of AutomateIT tickets for which the algorithm gave the correct runbook as the top recommendation (red line), tickets that were "attempted" (green line), the human participant selected the correct runbook (blue line), and the ticket was resolved (purple line).

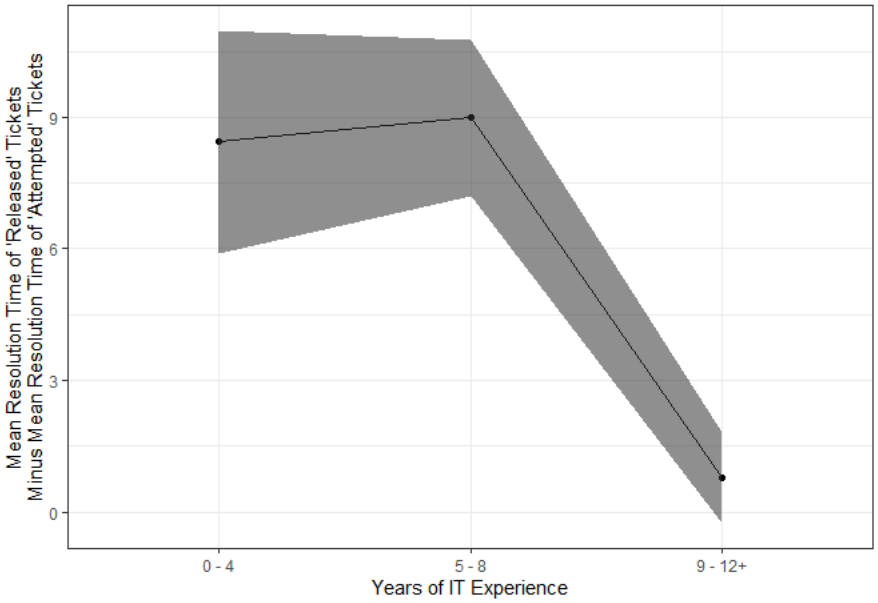
Figure 9. Rates of Error and Correction among AutomateIT Tickets



Notes. This figure displays the raw percentage of "attempted" (i.e., selected any runbook) AutomateIT ticket errors of commission (participant selects incorrect runbook given correct algorithmic advice), errors of omission (participant selects incorrect runbook given incorrect algorithmic advice), and correcting false positives

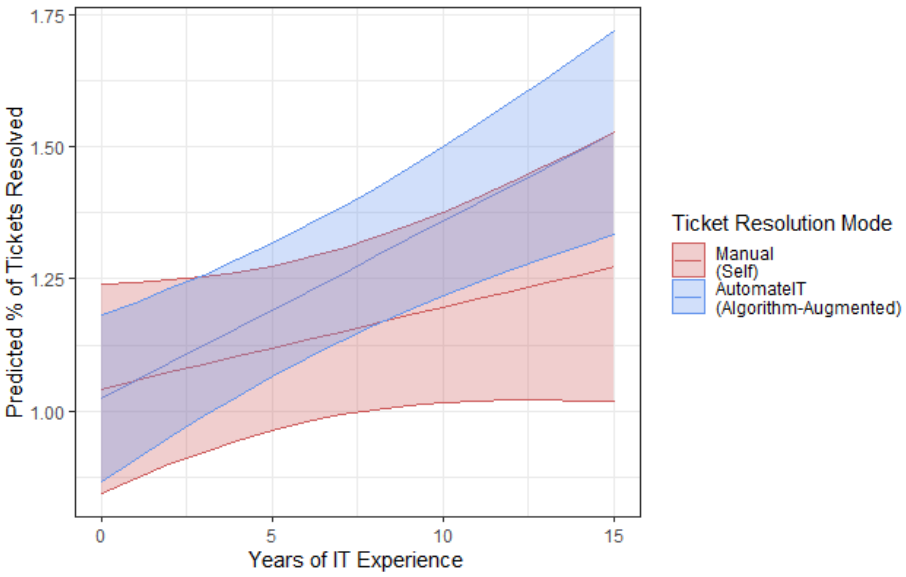
(participant selects correct runbook given incorrect algorithmic advice). It appears that high and low experience workers have the highest rates of errors of commission, errors of omission, and correcting false positives—indicative of noncompliance with algorithmic advice.

Figure 10. Difference in Time Spent on Released vs. Attempted Tickets for Different Levels of Domain Experience



Notes. The figure demonstrates that low experience participants released tickets that would have required relatively more time to solve (on average nine minutes longer for released tickets). This is contrasted with high experience participants, who released tickets that would take as long to solve as the tickets they attempted (statistically indistinguishable from 0). We were not able to observe how long participants spent tickets before releasing them, but instead measured how long a ticket *would* take to resolve based on other participants of the same employee level who resolved the same ticket. Thus the y-axis of this plot is the mean resolution time for tickets that were "released" minus the mean resolution time for tickets that were "attempted."

Figure 11. Predicted Rate of Ticket Resolution for "Attempted" Tickets



Notes. The figure displays predicted percentage of tickets resolved for AutomateIT and manual tickets, conditional on Years of IT Experience for the subsample of tickets that were "attempted" (i.e., any runbook was selected). The model used for these predictions was the same as the model used in Table 2, column 3. The coefficient estimates are included in Appendix Table A3, column 3. The figure was built using the "ggpredict" function from the "ggeffects" package in R.

APPENDIX

Table A1. Testing learning effects for recurring tickets: independence of treatment and control

Resolution Mode	Ticket Order	Average Marginal Effect	Standard Error	z-score	p-value
AutomateIT	2	0.13	0.08	1.71	0.09
AutomateIT	3	0.01	0.07	0.08	0.93
AutomateIT	4	0.05	0.08	0.67	0.50
AutomateIT	5	0.08	0.08	1.03	0.30
AutomateIT	6	0.16	0.08	2.11	0.03
AutomateIT	7	0.09	0.09	1.04	0.30
AutomateIT	8	0.01	0.09	0.09	0.93
Manual	2	-0.04	0.08	-0.46	0.64
Manual	3	0.07	0.09	0.79	0.43
Manual	4	-0.07	0.09	-0.85	0.40
Manual	5	-0.17	0.09	-1.90	0.06
Manual	6	-0.12	0.08	-1.39	0.17
Manual	7	-0.16	0.09	-1.82	0.07
Manual	8	-0.16	0.09	-1.75	0.08

Notes: This table demonstrates that there are no significant learning effects for recurring tickets (recall that each ticket recurred as the 5th-8th ticket after appearing as a 1st-4th ticket) for both AutomateIT and Manual resolution mode. The table displays the average marginal effects of ticket order dummy variables (tickets that occurred 2nd-8th out of 8 tickets), both for AutomateIT and Manual tickets. The marginal effects were predicted by the following OLS model, which used dummy variables to minimize functional form assumptions:

$$Y_{it} = \beta\ Is\ AutomateIT\ Ticket_t * (Years\ of\ IT\ Experience\ Dummies_i + Ticket\ Order\ Dummies_t + \boldsymbol{X_{it}}) + \gamma\ (Years\ of\ IT\ Experience\ Dummies_i * Ticket\ Order\ Dummies_t) + \alpha_t + \epsilon_{it}$$

where *i* indexes participant-level attributes, *t* indexes ticket-level attributes, and Y_{it} represents if the ticket was resolved. $Years\ of\ IT\ Experience\ Dummies_i$ represents a vector of dummy variables for every two years of IT experience (0-2 years, 2-4 years, 4-6 years, etc.). $Ticket\ Order\ Dummies_t$ represents a vector of dummy variables for the order in which the ticket was resolved, 1st – 8th. $\boldsymbol{X}$ represents a vector of controls: years at the company, whether the ticket matches the user's OS track, the user's OS track, the ticket order, and whether it is a recurring ticket. All variables are included twice in the model—once alone and once interacted with $Is\ AutomateIT\ Ticket_t$. Lastly, α represents a vector of ticket-level fixed effects.

Table A2. Balance Test Across Range of IT Experience

Years of IT Experience	0-2	3-4	5-6	7-8	9-10	11+	p-value
Is AutomateIT Ticket	0.500	0.500	0.500	0.500	0.500	0.500	1
	0.501	0.501	0.501	0.501	0.502	0.501	
Ticket Matches OS Track	0.606	0.567	0.637	0.637	0.594	0.565	0.486
	0.490	0.497	0.482	0.482	0.493	0.497	
Recurring Ticket is AutomateIT	0.242	0.404	0.775	0.488	0.516	0.194	<0.001
	0.430	0.493	0.419	0.503	0.504	0.398	
Ticket Order of AutomateIT	3.197	3.885	5.858	5.048	5.188	3.426	<0.001
	1.767	2.128	1.760	2.017	2.007	1.920	
Observations	264	208	240	168	128	216	

Notes: This table displays mean values for workers, broken out by workers with 0-2, 3-4, 5-6, 7-8, 9-10 and 11+ years of IT experience. Standard deviations are reported in parentheses. P-values are from a Pearson's Chi-squared test. *Recurring Ticket is AutomateIT* indicates the average number of AutomateIT tickets that were recurring tickets. *Ticket Order of AutomateIT* indicates the average order in which AutomateIT tickets were assigned. For example, if a worker was assigned AutomateIT tickets for their 5^{th}-8^{th} tickets, their average AutomateIT Ticket Order would be $(5+6+7+8)/4 = 6.5$. Thus a lower value represents that the AutomateIT tickets were assigned earlier in the experimental session relative to manually resolved tickets. The statistically significant imbalance across this variable represents a potential concern for identification. Throughout the paper, we address this concern using a variety of methods, including controlling for both *Recurring Ticket* and *Ticket Order* interacted with *Years of IT Experience* + *Years of IT Experience*^{2}.

Table A3. Subsample OLS Regressions

Dependent Variable: Ticket Resolved			
	(1) Subsample: had 4 recurring tickets	**(2) Subsample: Level 2 Participants**	**(3) Subsample: "Attempted" tickets**
Is AutomateIT Ticket * Years of IT Experience	**0.103 (0.037)***	**0.867 (0.410)***	**0.018 (0.0098)+**
Is AutomateIT Ticket * Years of IT Experience Squared	**-0.008 (0.003)***	**-0.083 (0.035)***	
Is AutomateIT Ticket	-0.152 (0.138)	-1.677 (1.152)	0.050 (0.117)
Is AutomateIT Ticket * Controls	Yes	Yes	Yes
Years of IT Experience		0.091 (0.311)	
Years of IT Experience Squared		-0.008 (0.024)	
Controls	Yes	Yes	Yes
Ticket-level Fixed Effects	N/A	Yes	N/A
Individual Fixed Effects	Yes	N/A	Yes
Adj. R^2	0.230	0.074	0.151
Num. obs.	856	340	1085

Notes: This table displays the main OLS regression results, using several different subsamples: participants that were assigned 4 recurring tickets, i.e. 4 unique tickets that recurred twice each (column 1); Level 2 employees (column 2); and tickets that were attempted, i.e. any runbook was selected (column 3). The dependent variable is *Ticket Resolved* (whether a ticket was resolved or not). The model includes fixed effects for the experimental session or individual participant fixed effects. Each column also includes controls for all the control variables listed in Table 1 in the paper, which include *OS Track, Ticket Matches OS Track,* and *Years at Company*. The variable *Ticket Order* were also included, interacted with the *Years of IT Experience* + *Years of IT Experience*^{2}. Asterisks indicate statistical significance at p-value cutoffs: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$, $^{+}p<0.1$. Columns 1 and 3 cluster standard errors at the participant level, while Column 2 (which had a much smaller sample) clustered at the experimental session level. All regressions use "CR2" bias-reduced clustered standard errors from the "estimatr" package in R, which are used to correct any bias from small sample clustering (Pustejovsky and Tipton, 2018).

Figure A1. Example illustration of manual ticket resolution process

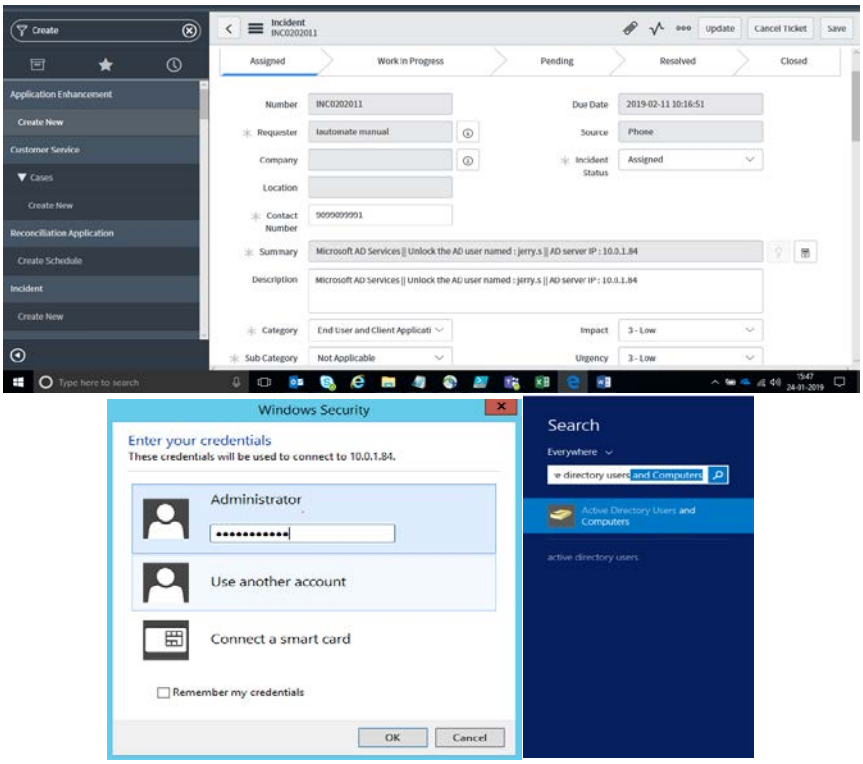
Example: granting permissions

Step 1. Logging in to Console, selecting from list of pending tickets to view the ticket's problem statement

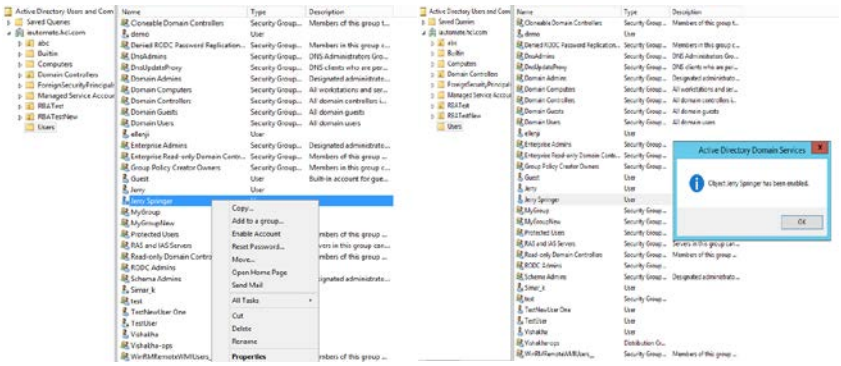
Step 2. (not pictured) if needed, consult the runbooks for how to resolve the problem

Step 3. Logging in as a system administrator

Step 4. Finding the correct directory of users



Step 5. Navigating to the correct user, opening properties, granting correct permissions to unlock the user. Checking that it has been done correctly.



Step 6. Navigating back to console, updating the ticket and marking as resolved (if unable to resolve they would mark as unresolved)

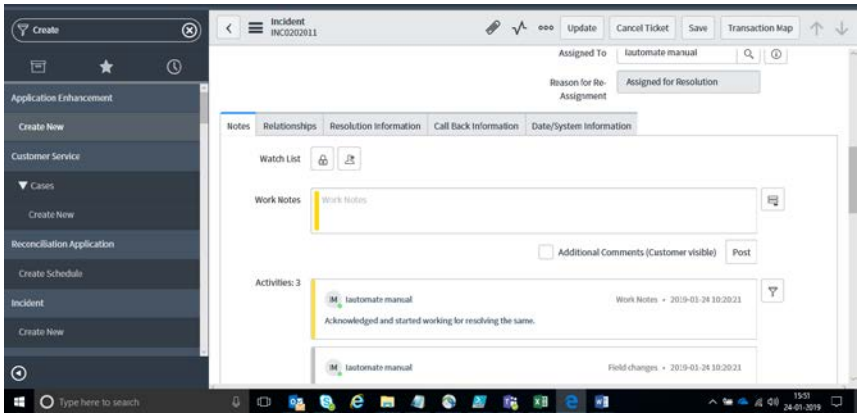
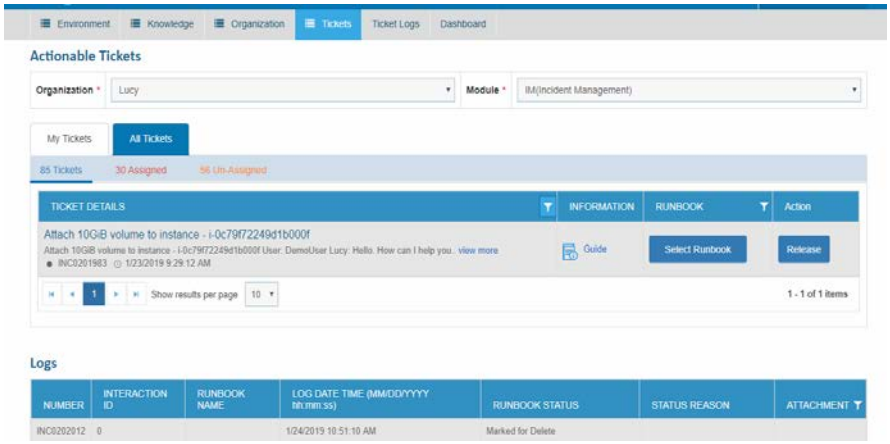


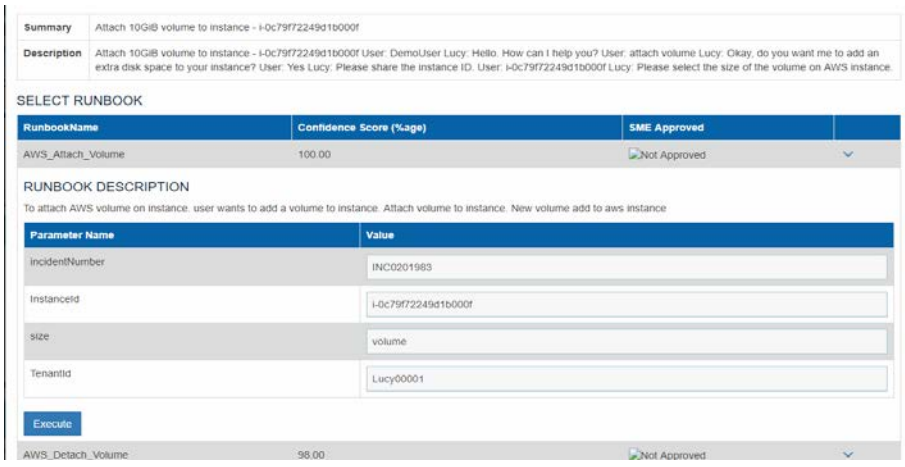
Figure A2. Example illustration of AutomateIT ticket resolution process

Example: Attaching AWS Instance



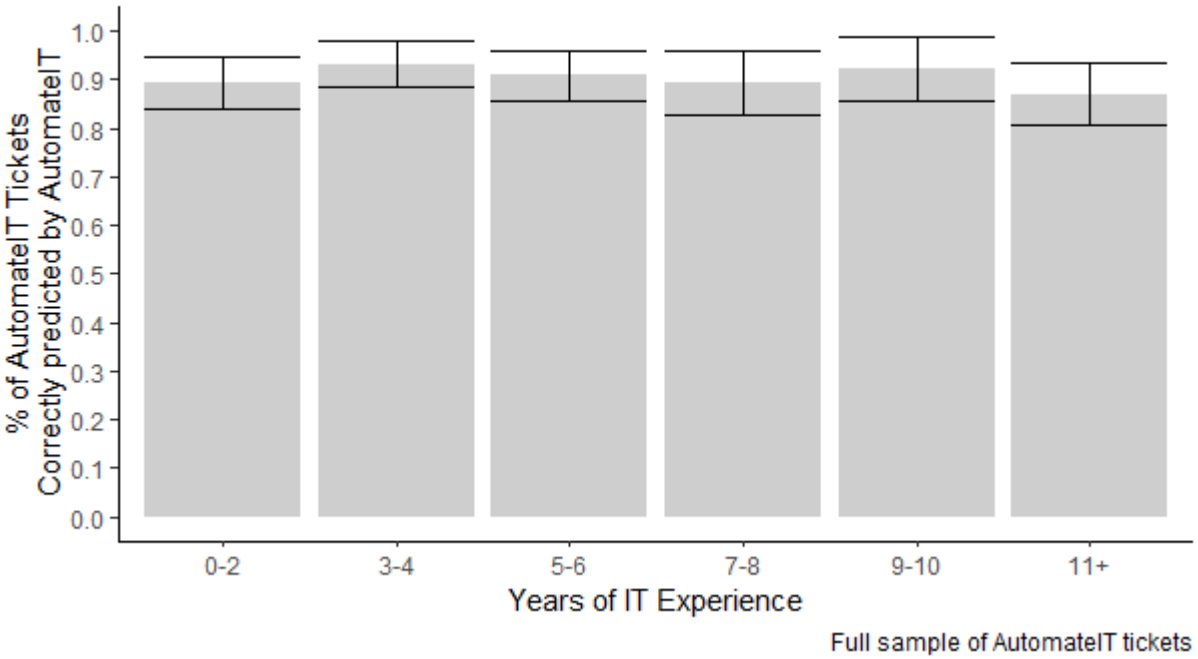
Step 1. Logging in to AutomateIT console and selecting from list of pending tickets

Step 2. Viewing the problem statement, description, and the list of recommended runbook solutions.



Step 3. Selecting the AWS_Attach_Volume runbook (correct for this ticket), adjusting the default parameter values as needed, and executing the runbook.

Figure A3. Percentage of correct algorithmic recommendations across range of experience



Notes: Gray bars represent the raw % of correct algorithmic recomendations (95% confidence interval error bars) for participants with varying levels of IT experience.