
DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD
DASH.HARVARD.EDU

HARVARD LIBRARY


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

Citation
Allen, Ryan, and Prithwiraj Choudhury. "Algorithm-Augmented Work and Domain Experience: The Countervailing Forces of Ability and Aversion." Organization Science 33, no. 1 (January–February 2022): 149–169.

Published version
https://doi.org/10.1287/orsc.2021.1554

Link
https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37373915

Terms of use
This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles (OAP), as set forth at
https://harvardwiki.atlassian.net/wiki/external/NGY5NDE4ZjgzNTc5NDQzMGIzZWZhMGFlOWI2M2EwYTg

Accessibility
https://accessibility.huit.harvard.edu/digital-accessibility-policy

Share Your Story
The Harvard community has made this article openly available.
Please share how this access benefits you. Submit a story

HARVARD | BUSINESS | SCHOOL

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

Ryan T. Allen
Prithwiraj Choudhury

Working Paper 21-073



Algorithm-Augmented Work 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

Copyright © 2020, 2021 by Ryan T. Allen and Prithwiraj Choudhury

Working papers are in draft form. This working paper is distributed for purposes of comment and discussion only. It may not be reproduced without permission of the copyright holder. Copies of working papers are available from the author.

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

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

Ryan T. Allen
Harvard Business School

Prithwiraj Choudhury
Harvard Business School

Working Paper
Conditionally Accepted at *Organization Science*
In preparation for the special issue on
Emerging Technologies and Organizing

Abstract. Past research offers mixed perspectives on whether domain experience helps or hurts algorithm-augmented work performance. To reconcile these perspectives, we theorize that domain experience affects algorithm-augmented performance via two distinct countervailing forces—ability and aversion. On one hand, workers' domain experience can complement algorithms, due to an increased *ability* to judge the accuracy of an algorithm's advice. On the other hand, workers with more domain experience tend to exhibit more *aversion* to accepting helpful algorithmic advice. Each force varies in its influence on workers with different levels of domain experience. *Ability* developed through learning-by-doing increases at a decreasing rate over the range of experience, while algorithmic *aversion* is more intense for experts. Therefore, more domain experience will increase algorithm-augmented performance for workers with low levels of domain experience, but will decrease the algorithm-augmented performance of workers with high levels of domain experience. We test this by exploiting a within-subjects experiment in which corporate Information Technology support workers were assigned to resolve problems both manually and using an algorithmic tool. We confirm that the difference between performance with the algorithmic tool vs. without the tool was characterized by an inverted U-shape over the range of domain experience. Only workers with moderate domain experience did significantly better using the algorithm than resolving tickets manually.

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

INTRODUCTION

It is increasingly common for knowledge workers in organizations to use algorithmic[1] tools to augment their work. 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 augmented. For example, there has been a notable increase in adoption of ML-trained algorithmic decision tools by managers, clinicians, and judges, to help them make decisions about hiring personnel, diagnosing diseases, and assigning bail (Miller 2015, Cowgill 2018b, 2018a, Kleinberg *et al.* 2018, Arthur and Hossein 2019). In the wake of this phenomenon, an emerging research agenda has begun to investigate the task performance of humans augmented by algorithmic tools (Shrestha, Ben-Menahem, and von Krogh 2019, Choudhury, Starr, and Agarwal 2020).

A strand of this literature compares the accuracy of algorithmic and human judgment, and examines whether humans will accept and use algorithmic advice. This line of research typically demonstrates the superior accuracy of even simple algorithms over experts (Dawes 1979, Grove *et al.* 2000, Kleinberg *et al.* 2018, Miller 2018). In many cases, people would make better decisions if they used the algorithm's recommendations. Yet people—especially experts—tend to exhibit "algorithmic aversion," and rely more on their own judgment than the advice generated by an algorithm (Dietvorst, Simmons, and Massey 2015, Logg, Minson, and Moore 2019). This research paints a bleak picture for the prospect of productively combining algorithmic recommendations with human expertise.

Yet, a puzzle emerges when we compare this view against prior literature on human capital and technological change. In contrast to the algorithm aversion literature, the human capital literature suggests that human domain experience *complements* algorithms. Human complementarity with algorithms arises from humans' relative advantages in using tacit knowledge, causal reasoning, better understanding of the context, and human values, to make judgments (Autor, 2015; Agrawal, Gans and Goldfarb, 2017, 2018; Brynjolfsson and Mitchell, 2017; Brynjolfsson, Mitchell and Rock, 2018). This allows humans to identify when algorithms make inaccurate

[1] When we use the word "algorithm" in this paper, we are specifically referring to a tool that takes information as input, then systematically parses that information to make an assessment or decision recommendation as output. 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.

predictions due to biases in training data and/or algorithms, overfit, inaccurate tuning of hyperparameters and other reasons (Shrestha, Ben-Menahem and von Krogh, 2019; Choudhury, Allen and Endres, 2020; Raisch and Krakowski, 2020). Because domain experience increases these complementary human abilities (Dane, Rockmann and Pratt, 2012), human domain experience is an essential component to productively complementing algorithms (Choudhury, Starr and Agarwal, 2020). Tensions between this view and the algorithm aversion literature motivate us to ask: under what conditions does a knowledge worker's domain experience increase algorithm-augmented performance, relative to self-performance?

To disentangle when domain experience hurts or helps algorithm-augmented performance, we theorize a framework that integrates two previously distinct countervailing forces—*ability* and *aversion*—which vary in their influence on algorithm-augmented performance for workers with different levels of domain experience. On 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. On the other hand, workers with more domain experience can exhibit more *aversion* to accepting helpful algorithmic advice, thereby decreasing performance. We posit 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 relatively more prevalent among experts (Logg, Minson and Moore, 2019), algorithm-augmented performance (relative to self-performance) will first increase with domain experience for workers with low levels of domain experience, then decline for workers with high levels of domain experience.

We test our theory in the context of IT workers using an algorithmic tool to resolve "help tickets"[2]—technical problems submitted to the IT department, which requires considerable domain experience (Mithas and Krishnan, 2008). We exploit a within-subjects experiment in which a sample of corporate IT support workers with varying levels of domain experience were assigned tickets to be resolved: (1) manually using their previous ticket resolution system; and (2) by using the new ML-trained algorithmic tool that lists the most likely solutions

[2] A help ticket is a document that is generated when someone submits an issue to an IT support team. 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 fixed.

to each ticket. This enables us to compare—across varying levels of domain experience—the proportion of resolved tickets for workers using the algorithmic tool, relative to manually resolving tickets.

We find that only workers with moderate levels of domain experience perform significantly better using the algorithm than manually resolving tickets—confirming an inverted U-shape in relative performance over the range of domain experience. To validate the mechanisms driving this relationship, we analyze individual log files for the subset of tickets resolved using the algorithm. These analyses reveal that the inverted U-shaped relationship is driven by a propensity of both the low experience and the high experience participants to ignore the algorithm's correct advice. However, the mechanisms driving this pattern appear to be different for high experience vs. low experience participants. As theorized, additional analyses and qualitative interviews suggest that low experience participants reject algorithmic advice primarily due to lack of ability to assess and use algorithmic recommendations, while high experience workers reject algorithmic advice primarily due to an aversion to the algorithm's advice. Interviews with participants suggest that the aversion of high experience workers was rooted in their belief in their own superior understanding of the complex IT systems, and in a greater sense of accountability for their actions.

Our study contributes to the human capital and technological change literature, and research on algorithm aversion. Our first contribution extends our understanding of the complementarity of human capital and algorithmic tools. Prior research emphasizes the benefits of human domain experience for algorithm-augmented work (Autor, 2015; Brynjolfsson and Mitchell, 2017; Shrestha, Ben-Menahem and von Krogh, 2019; Choudhury, Starr and Agarwal, 2020; Raisch and Krakowski, 2020), but our theory and results indicate that higher domain experience also has potential downsides. Integrating insights from the algorithm aversion literature, our framework generates a prediction that for experienced workers, marginal increases in domain experience may decrease complementarity with algorithms. Our second contribution deepens our understanding of the implications of algorithm aversion by experts. Whereas prior literature on algorithm aversion implies that expertise is a liability for algorithm-augmented judgments (Arkes, Dawes and Christensen, 1986; Logg, Minson and Moore, 2019), we counter that domain experience is, in fact, the primary means by which humans have any potential to complement algorithmic judgement. Thus, whether an expert will do better or worse with an algorithm is not merely how much domain experience they have, but rather whether (in their context) the

aversion effect overpowers the ability effect. Taken together, our theory integrates mechanisms from previously distinct theories to explain why *intermediate levels of domain experience* can provide the greatest algorithm-augmented performance.

THEORY

Domain Experience and Ability to Assess and Use Algorithmic Advice

The human capital and technology change literature highlights that (at least in the near future) algorithms cannot entirely replace humans for most tasks, and that humans are a necessary complement to algorithms. This is because many tasks performed by knowledge workers require tacit knowledge not easily codified into rules (Polanyi, 1966; Dreyfus and Dreyfus, 1986; Autor, 2015; Brynjolfsson and Mitchell, 2017), require interpretability (Shrestha, Ben-Menahem and von Krogh, 2019), or require some human value judgement in addition to prediction (Agrawal, Gans and Goldfarb, 2017, 2018). Even ML-built algorithms, which may be able to learn some tacit rules through inductive learning (Choudhury, Allen and Endres, 2020), cannot replace humans for anything but well-structured and narrow tasks (Autor, 2015). And even for well-structured and narrow tasks, ML-built algorithms are typically trained from large, noisy datasets that contain random errors or overfitting, lack any semblance of causal reasoning (Pearl, 2009; Pearl and Mackenzie, 2018), and lack contextual information. Therefore, human evaluation and intervention are often required when using algorithmic tools in practice (Lebovitz, Lifshitz-Assaf and Levina, 2020; Raisch and Krakowski, 2020).

Consulting the cognitive psychology literature suggests that humans' relative advantages over algorithms—such as tacit knowledge, causal and contextual understanding—increase with domain experience (Dane, Rockmann and Pratt, 2012). This is because domain experience—domain-specific knowledge obtained by focused practice or learning-by-doing—is a foundational driver of human knowledge and domain expertise (Chari and Hopenhayn, 1991; Ericsson, Krampe, and Tesch-Römer. 1993). Domain experience allows people to build context-dependent perceptual structures that help them frame problems, detect relevant signals, and to react intuitively and appropriately when making complex evaluations (Chase and Simon 1973, Simon 1991, Salas, Rosen, and DiazGranados 2010). This allows a fuller grasp of context, which "makes focused perception possible, understandable, and productive" (Dasgupta 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).

Accordingly, domain experience is foundational to humans' ability to complement algorithmic tools, because it allows humans to better perceive when algorithms make mistakes due to lack of context, adaptability, or simple random error. A recent experiment confirms that domain experience allows humans to better leverage algorithmic tools (Choudhury, Starr, and Agarwal 2020). Experimental subjects, who were novices in patents and intellectual property, were tasked with using an algorithmic tool to identify relevant prior art. Only participants with access to expert patent examiners' domain knowledge—which drew on contextual knowledge to highlight key information missed by the algorithm—were able to correctly interpret and leverage the useful advice generated by the ML-based algorithmic tool to identify prior art. These human advantages over algorithms are a major reason that humans are kept "in the loop" for many algorithm-augmented decisions, such as HR departments using algorithms to decide who to hire (Raisch and Krakowski, 2020).

Humans with more domain experience are even able to complement highly accurate algorithms, which are only highly accurate *on average*. Consider an example: the Google Translate algorithm translating a sentence from one language to another. This algorithm does very well on average, but it is not uncommon for native speakers (i.e., those with years of domain experience in the language and culture) to improve the translations by intuitively perceiving when word choice feels off, syntax violates subtle linguistic rules, or when the output is completely unfathomable (see Autor 2015).

Domain Experience and Aversion to Algorithmic Advice

Although workers with more domain experience may be more likely to catch an algorithm's mistakes, they may also be more averse to accepting its advice. The term "algorithmic aversion" was coined by Dietvorst et al. (2015), but the literature on noncompliance with algorithmic advice 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; Glaeser *et al.*, 2021; Yang, 2021).

Algorithm aversion is especially salient among experts—workers with high levels of domain experience. One early study on the topic found that experts[3] tended to use helpful decision rules less than those with less experience and, consequently, exhibited worse judgment (Arkes et al. 1986). More recently, Logg et al. (2019) confirmed these results in Study 4 of their paper ("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 those of fellow participants, but least confident in the algorithm.

Prior literature on expert advice-taking in general (not necessarily from algorithms) suggests that this mistrust of advice may result from experts' biased assimilation of information (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. In the end, although those with more experience tend to make more

[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."
[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.

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 for why experts' general aversion to advice 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 those rules (Grove and Meehl 1996). Experts with vast arrays of domain knowledge can mistakenly feel that they have access to important information unaccounted for by the algorithm, resulting in mistrust of the algorithmic output.

In summary, this body of work strongly suggests that aversion to algorithmic advice is more prevalent among people with more domain experience. This is partly explained by experts' aversion to advice in general, but can be especially salient for advice generated by algorithms.[8]

Hypothesis

While the literature on human capital and technology change would predict that domain experience increases a worker's ability to complement algorithms, the algorithm aversion literature shows that workers with more domain experience are apt to reject helpful algorithmic advice due to aversion. On the surface, these perspectives appear to yield contradictory results. However, our framework reconciles these perspectives by arguing that both of these countervailing forces—ability and aversion—influence workers, but at varying degrees of strength for different levels of domain experience.

First, consider how increasing domain experience affects workers' ability to complement algorithms. Some of the earliest work on the theory of human capital posited that knowledge and skills increase at a decreasing rate (Becker 1962), and later work on learning-by-doing has consistently documented diminishing

[8] Though not a salient feature in the context of our study (which we explain later in the discussion of the setting), it is worth noting that in other contexts, there may be additional reasons for resistance against algorithms, such as professional identity threats (e.g., Kellogg, Valentine, and Christin 2020).

returns to experience (Ericsson, Krampe and Tesch-Römer., 1993; Foster and Rosenzweig, 1995; Mithas and Krishnan, 2008). We should expect, then, that although workers with more domain experience do have more ability to accurately assess algorithmic advice, this relationship will have diminishing returns. The marginal increase in ability will be greater for workers with lower levels of experience.

By contrast, aversion to algorithmic advice is likely stronger for those with more domain experience (Arkes, Dawes and Christensen, 1986; Yaniv and Kleinberger, 2000; Liu, 2013; Li, 2017; Logg, Minson and Moore, 2019; Teplitskiy *et al.*, 2019). Taken together, we theorize that algorithm-augmented performance will first rise with increasing domain experience (when the marginal positive impact of ability on performance outweighs the marginal negative impact of aversion on performance), then fall (when the marginal negative impact of aversion on performance catches up to the marginal positive impact of ability on performance). For people with low domain experience, the impact of ability on overall performance increases rapidly, while the influence of aversion remains relatively low. Thus, increasing domain experience for novices will be associated with increased performance using algorithms. By contrast, for people with high domain experience, the impact of ability on performance has leveled off, while the impact of aversion has increased. Thus, increasing domain experience will be associated with decreased algorithm-augmented performance for experts. Formally, 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.*

Figure 1 illustrates the proposed framework. It highlights that we expect workers to have the best algorithm-augmented performance when the positive contribution of ability to performance (the green line on the top panel) has the biggest gap above the negative contribution of aversion to performance (the red line on the top panel). Conceptually, adding together the performance impact of these lines results in the overall impact of domain experience on algorithm-augmented performance (represented by the inverted-U blue line on the bottom panel).

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

RESEARCH CONTEXT

We test our hypothesis in the context of workers resolving "help tickets" in a corporate information technology (IT) department. This is an appropriate setting to test our hypothesis for several reasons. First, it cannot yet be fully automated by algorithms. Humans must interface with algorithms and evaluate their recommendations in a hybrid work process (Shrestha, Ben-Menahem, and von Krogh 2019). Second, IT is a sufficiently complex task that it requires significant learning-by-doing over years of experience, and there are diminishing returns to IT experience (Mithas and Krishnan, 2008). Therefore, as in many other contexts, we expect it to be a setting in which ability has a positive but diminishing effect on algorithm-augmented performance.

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 algorithmic tool that was designed to automate help ticket resolution. Using a within-subjects design, IT professionals with varying levels of IT experience were instructed to solve a set of help tickets both using the new algorithmic tool and manually. As 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 Support 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 roughly 25% of their 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, which approximately corresponds to IT-related work 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). In general, tickets 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"), "Linux," or a hybrid of both.[9]

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 possible 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 runbook among the thousands to use. According to the IT staff, it is 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, ticket resolution using the legacy manual system involves multiple windows and clicks—e.g., logging into a server, identifying a user in a directory, and granting permissions the user (for an illustrative example of resolving a ticket manually, see Appendix Figure A1).

AutomateIT: TECHCO's Algorithm for Augmenting IT Support Work

TECHCO assigned a team of machine learning (ML) engineers to build AutomateIT[10]—an IT process automation tool meant to reduce the cost of manual IT ticket resolutions. The tool automates both of the core steps of the IT ticket resolution process: runbook search and runbook execution. 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.

The tool uses natural language processing (NLP) to match ticket text to runbook solutions. It uses NLP to standardize, tokenize, and topic extract text from more than one million help tickets and the corresponding

[9] TECHCO uses systems based on both Windows/Intel and Linux
[10] AutomateIT is a pseudonym.

7,000 runbook solutions. The tickets had each been labeled 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 the text of a help ticket as input, which it uses to calculate a similarity score to each of the 7,000 possible runbooks. All runbooks above a certain similarity score threshold (e.g., 60%) are displayed as output 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. Since 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.

EXPERIMENTAL DESIGN

The experiment was executed by a team at TECHCO tasked with evaluating the effectiveness of the new AutomateIT tool. 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 any issues with the research design and how we address each issue.

Since it was not possible to randomly assign domain experience, we treat domain experience as a moderating effect on performance for tickets resolved using self-judgment (control) vs. algorithm-augmentation (treatment). We opted to implement a within-subjects design to estimate a moderating effect using a small number of participants. 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). This design 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).

There were 154 TECHCO IT support staff members who volunteered to participate in the experiment. Volunteers were told the purpose of the experiment was to test a new algorithmic tool that would assist them in their work, and were strongly encouraged to participate in the experiment by their managers. In field interviews, participants expressed that they did not feel that their jobs were threatened by the tool, but rather viewed the tool as a welcome automation of menial tasks so they could focus on other aspects of their job. Given this consensus, and the fact that these tasks represented only a small portion (about 25%) of the participants' overall job, we have no reason to think that participants were incentivized to intentionally underperform when using the algorithm.

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 the day-to-day activity of resolving tickets. Participants were not told exactly how accurate the tool would be (about 90% of top recommendations were correct). 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 eight tickets to avoid significant time pressure. In the experiment, the proctors checked to make sure that each ticket contained the correct runbook as one of the possible runbook solutions listed by the algorithm (this was not communicated to the participants). The tickets assigned in 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.

Each participant was assigned tickets to resolve using both the manual ticket resolution system and the new AutomateIT tool. In accordance with a within-subjects experimental design, each participant received four (control) "manual" self-performance tickets to resolve using the old system, and the same four (treatment) tickets to be resolved using the new 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 compare AutomateIT tickets to manual tickets while completely controlling for differences between individual participants and differences between tickets. The recurring tickets were identical 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 from repeated exposure. We took several steps to verify independence of treatment and control. First, we randomly assigned whether manual tickets or algorithmic tickets appeared first. Second, we control for recurring tickets in our models and verify that there is not a positive and significant learning effect. Third, we confirm that running between-subjects analysis on the subsample of only non-recurring tickets (i.e., tickets appearing for the first time) yields the same results (discussed later in the results; Table 3 and Appendix Table A3).

Field interviews with participants reveal why learning effects from receiving the same ticket twice did not confound the results. According to the participants and proctors, the algorithm's recommendation and solution emerged as if from a "black box." 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. This meant that getting the same ticket in manual resolution mode was not any easier after resolving it with the AutomateIT tool. Going the other direction, resolving a ticket manually did not help later performance with the AutomateIT tool because the steps they took to manually resolve the ticket did not obviously point to which algorithmic runbook label would execute the correct command.

The TECHCO proctors who ran the experiment created the tickets that were assigned to participants. These proctors were familiar with the routine work of the IT support staff, and created tickets for the experiment that were based on real tickets seen in the past. In order to simulate normal working conditions (and

[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 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 A1, column 1).

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, 40 Level 2 tickets, and 40 Level 3 tickets. 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).

Since participants were assigned tickets within their "employee level," there is a potential selection 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. 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 overall performance across participants with varying levels of experience, but rather the *difference between algorithm-augmented and manual ticket resolution performance within each participant* compared across participants with varying levels of experience. As 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 do not have varying effects on performance across the range of experience. We only need a weaker assumption: that the difference between manual and automatic tickets does not systematically vary over the range of experience.

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 (algorithmic recommendations on each set of tickets were around 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).

Third, we ran a complete subsample analysis using *only* Level 2 employees, thereby eliminating any potential confounding issues between employee levels. We present this analysis in in the robustness checks of the results section below. 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.

A final, related issue was 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, and running analyses on the subset of first-appearance tickets, we also ran our models without the participants who did not receive recurring tickets (Appendix Table A1, column 1), and the results remained unchanged. We also ran a subsample analysis using only Level 2 employees, which yielded results consistent with our main findings (see Appendix Table A1, column 2).

DATA

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

Dependent Variable

The primary measure of task performance is $Ticket\ Resolved_{ik}$, a binary variable that is set to 1 if ticket *k* was resolved by participant *i,* and 0 if it was not resolved. This is the standard measure of performance for IT support work at TECHCO.

Independent and Moderator Variables

To operationalize the comparison of algorithm-augmented to self-judgment, we compare tickets resolved using the algorithmic AutomateIT system (treatment) vs. the manual system (control). This comparison is captured for each ticket *k* by the binary variable $Is\ AutomateIT\ Ticket_k$ (1 for AutomateIT tool, 0 for the manual system).

Our moderator variable of interest, domain experience, is measured by the total years of IT experience for participant *i* ($Years\ of\ IT\ Experience_i$).[12] Since the participants in our sample were relatively 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) validate this operationalization of domain experience by demonstrating that IT competencies are acquired through learning-by-doing, and thus that "technical

[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.

competencies of IT professionals are reflected in their on-the-job IT experience" (Mithas and Krishnan 2008, pp. 417). They also point out that while firm-specific IT experience (i.e., familiarity with the IT systems of the firm) is valuable, overall 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.

Controls

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 $Ticket\ Matches\ OS\ Track_{ik}$ 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 $Ticket\ Order_{ik}$ in which each ticket was opened (from first through eighth) and for whether each ticket was a $Recurring\ Ticket_{ik}$ (1 if the participant had already seen the same ticket using the other resolution system; otherwise 0).[13]

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

STATISTICAL ESTIMATION

We obtain OLS[14] estimates using the following model:

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

where *i* indexes individual-level attributes, *k* indexes ticket-level attributes, and Y_{ik} captures whether the ticket was resolved by a participant. The effect of interest is captured by the terms β_4 and β_5, the quadratic fit of the

[13] As 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.
[14] Logistic regression models yield nearly identical results

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. The controls also include interactions between $Ticket\ Order_{ik}$ and $Recurring\ Ticket_{ik}$ with $Years\ of\ IT\ Experience_i + Years\ of\ IT\ Experience_i^2$.[15] All controls are included twice in the model—once alone and once interacted with $Is\ AutomateIT\ Ticket_k$. Interacting $Is\ AutomateIT\ Ticket_k$ with each variable allows the equation to estimate different effects for tickets that were resolved manually vs. using the AutomateIT system. 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, 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 an 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. Comparing the differences between the two, moderately experienced workers perform significantly better when using the algorithm vs. manually resolving tickets. High experience workers perform about the same when using the algorithm vs. manually, and surprisingly, low experience workers perform worse using the algorithm (an unexpected finding we explore in qualitative interviews with participants, which we present below).

==========

[15] This is because, in addition to the balance test for AutomateIT vs. manual tickets in the summary statistics in Table 1, we also checked for balance across levels of IT experience in 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 potential issue with the experiment: The ticket order of AutomateIT tickets are not evenly distributed across participants with varying levels of domain experience. To address this, we control for $Ticket\ Order_{ik}$ and $Recurring\ Ticket_{ik}$ in our models, as well as interact them with $Is\ AutomateIT\ Ticket_k$. We also interact both $Recurring\ Ticket_{ik}$ and $Ticket\ Order_{ik}$ 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. Additionally, in a robustness check, we ensure that the same patterns hold, even if just using tickets' first appearance with no recurring tickets (discussed later in Tables 3 and Appendix Table A3).

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

In columns 1–3 of Table 2, we formally test differences in performance resolving tickets using AutomateIT vs. manually, across the range of experience. Regression results confirm that the relationships displayed in Figure 2 hold after adding relevant controls and fixed effects. The model displayed in column 1 confirms that participants with more years of IT experience were more likely to accurately resolve AutomateIT tickets relative to manual tickets, but this reverses so that the effect disappears for higher experienced workers. Column 2 confirms that the relationship is robust to problem fixed effects, ruling out alternative explanations due to unobserved problem heterogeneity (though the relevant squared term is only marginally significant, presumably due to the many fixed effect dummy variables included in the model). Column 3 adds participant fixed effects, which explicitly models a quadratic relationship between $Ticket\ Resolved_{ik}$ and $Years\ of\ IT\ Experience_i$ for AutomateIT tickets *relative to* manual tickets *within* each participant. This column shows that, accounting for differences between individuals, there is an inverted U-shaped difference in performance between AutomateIT tickets and manual tickets, across the range of domain experience.

To facilitate interpretation of the estimates, Figure 3 displays the predictions of ticket resolution conditional on experience, as predicted by the regression model in Table 2, column 3. Panel A displays the predicted percentage of tickets resolved for both manual and AutomateIT. Panel B displays the relevant comparison as the difference between predicted percentage of tickets resolved for AutomateIT (relative to manual). The figure 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 Checks

Subsample Analysis: Buckets of Domain Experience

We conducted several additional analyses to check the robustness of the main findings. First, we ran subsample analyses to confirm that our results were not driven by misinterpreted interaction terms, unjustified

functional form assumptions, or repeated exposure learning effects. In columns 1–4 of Table 3, we break up the sample into four roughly equal-sized subsamples across the range of domain experience (0–3 years, 4–6 years, 7–9 years, 10+ years), comparing performance with vs. without the AutomateIT tool. These analyses confirm the main finding: only moderately experienced workers did significantly better with the algorithmic tool. High experience workers did about the same with the algorithmic tool, and the lowest experience workers did worse (we explore why in the "Qualitative Interviews with Participants" section below). In columns 5–8 we conduct a similar analysis comparing between subjects, using only tickets appearing for the first time (i.e., no recurring tickets were included). This analysis confirms that results were not driven by repeated-exposure learning effects.

==========
INSERT TABLE 3 ABOUT HERE
==========

Subsample Analysis: Buckets of AutomateIT and Manual tickets

Second, we conducted a similar analysis on separate subsamples of manual and AutomateIT tickets. For each subsample, we estimated how performance varied across four buckets of experience (again 0–3 years, 4–6 years, 7–9 years, and 10+ years). The results, displayed in Appendix Table A3, indicate a constant upward (though not statistically significant) upward trend for manual tickets over the range of domain experience, and a marginally significant bump in relative algorithm-augmented performance only for moderately experienced workers.

Subsample analysis: Level 2 employees

Third, we briefly return to the potential identification issue of assigning a different set of questions to each employee level. To address this issue, we ran a complete subsample analysis using *only* Level 2 employees. Though this narrows the data to a much smaller sample size, analyses using only these employees eliminates 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. The analysis provides another confirmation that there was a statistically significant inverted U-shape in relative performance across the range of experience, which was not driven by the ticket assignment to

the different employee levels. Results from the regression model are included in Appendix Table A1, column 2. For visualization of the predicted effects of the regression model, see Appendix Figure A4.

Tests of Inverted-U functional form

Finally, we ensure that our estimation is not a spurious result of estimating an OLS model with a forced quadratic functional form. First, 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. This test confirms a sign change in probability of resolving an AutomateIT ticket (relative to a manual ticket) for low vs. high levels of experience (see Appendix Figure A5). Second, we confirmed the functional form using random forest models. Since 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, rationale, and cautions for implementing this approach 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. The random forest algorithm independently found the same inverted U-shaped relationship further validating that we fit a reasonable model to the data (see partial dependence plots in Appendix Figure A6).

Mechanisms

Next, we confirm that our proposed mechanisms—ability and aversion—drive performance algorithm-augmented performance differences across workers with varying levels of domain experience. Whereas the study's primary comparison is *within*-participant performance with vs. without the algorithm, these mechanism analyses make comparisons *between* participants for the subset of tickets that were resolved using the algorithm. Therefore, these analyses no longer rely on the study's within-participant randomization, and are best viewed as exploratory analyses rather than experimental evidence (though it gives us confidence that the main results hold with between-subjects comparisons; see Table 3 and Appendix Table A3).

In this subset of tickets that were resolved using AutomateIT, we observe intermediate outcomes in addition to whether the ticket was resolved. First, we observe whether the algorithm's top recommendation was correct. Second, we observe whether a ticket was "attempted" by a participant. Before opening an AutomateIT ticket, participants can view the list of tickets they are assigned, which includes a preview of the description of

the problem (see Appendix Figure A2 for visualization). 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 to be resolved manually later, resulting in an unresolved ticket. Third, we observe is if the participant selected the correct runbook. 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. So finally, we observe whether the participant actually resolved the ticket.

To get a sense of the mechanisms driving ticket resolution, we first compare percentages of tickets that successfully reached each intermediate outcome, over the range of domain experience. Figure 4 displays the raw percentage of AutomateIT tickets at each step in the ticket resolution process. First, it shows the percentage for which the algorithm gave the correct runbook as the top recommendation (red/circle/solid line); second, tickets that were "attempted" (green/short-long dash/triangle line); third, the percentage of tickets for which the human participant selected the correct runbook (blue/dotted/square line); and finally, the percentage for which the ticket was resolved (purple/long dash/plus line). This visual confirms that the rate of the algorithm's accuracy was similar for all groups, so this was not a driving factor of the results (i.e., our results are driven by individuals' behaviors rather than differences in the rate of algorithmic accuracy). It also shows that most (79%) of the errors of commission were due to "releasing" tickets (i.e., not selecting any runbook), both for the low and the high domain experience participants (for comparisons of errors of omission and commission, see Appendix Figure A7).

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 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 4 ABOUT HERE
==========

Evidence Consistent with Proposed Mechanisms: High Experience Participants Do Not Spend More Time Before Releasing More Difficult Tickets

According to our framework, high experience workers would not release tickets because they are too difficult, but rather because they were more averse to algorithmic advice. If that were true, we would expect high experience participants to be more likely to release tickets independent of their ability to solve the ticket. Therefore, we would expect them to release tickets relatively quickly, regardless of the ticket's level of difficulty. Conversely, we would expect low experience participants to spend more time puzzling out the problem before they release it.

Although we do not directly observe how much time a participant spends on a ticket before releasing it, we can indirectly test this assumption by observing the average amount of time it took others of the same employee level to resolve the ticket. Figure 5 displays the difference in the average amount of time participants spent on tickets that were released vs. attempted. 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 zero). This pattern is consistent with the claim that high experience workers are not releasing tickets because they are difficult for them.

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

Evidence Consistent with Proposed Mechanisms: Domain Experience Positively Affects Ticket Resolution for the Subset of 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 framework, we would expect that if we could somehow remove algorithm aversion for high experience participants, more domain experience would lead to a linear relationship between domain experience and better performance with the AutomateIT tool. As a proxy for completely removing aversion, we consider the subset of tickets that were attempted (i.e., not released). Our regression models confirm that more IT experience is

positively and linearly related to a greater predicted probability of resolving the ticket.[16] The predicted marginal effects of the regression model are displayed in Figure 6. This evidence is consistent with our claim that removing the aversion effect among high experience workers, domain experience positively affects performance via the ability mechanism for all experience levels.

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

Alternative Explanations

We briefly consider three alternative mechanisms that could explain the patterns 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). Considering the pattern of both high and low experience participants releasing tickets (displayed in Figure 4), this seems unlikely. If the story were overconfidence, we would expect relatively more attempted tickets and fewer correctly resolved tickets. We also found no evidence of overconfidence in the qualitative interviews.

Second, it is possible that the inverted U-shape could arise from an adverse selection effect between employee Levels 1, 2, and 3. "Competent" employees could be promoted to the next level despite having less experience. Therefore if "Incompetent" employees with more experience outperform "Competent" employees, it may not be because they have more domain experience, but rather because they were assigned an easier set of tickets. However, there are several observations that make this an unlikely explanation for our results. First, though there were some exceptions at the margins, promotion to the next level was overwhelmingly driven by tenure. Second, this alternative explanation can explain differences in overall performance, but it is much harder to explain why selection would be relevant to *differences* between algorithmic and manual performance (which is the primary focus of our theorizing and empirical measurement). Finally, we observe the same inverted U-shaped pattern for the subset of only Level 2 employees (see Appendix Table A1 column 2; and Figure A4). Within this

[16] 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 = 1.82). This is partly due to the decreased sample size of the subset of attempted tickets.

subsample, if less experienced participants are included because of relatively high competence, then they should perform relatively well. But we observe that they perform worse than more experienced peers.

Third, it is possible that participants with different levels of experience learned more quickly to use or trust the algorithm (a 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 (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.

Qualitative Interviews with Participants

We conducted field interviews with participants to elaborate on our mechanisms and explore *why* high experience workers exhibited algorithm aversion and why low experience workers did *worse* using the algorithm than resolving tickets manually. Responses generally supported our theorizing, but also highlighted relatively novel explanations of algorithmic aversion and noncompliance with algorithmic advice.

Why low experience workers did worse using the algorithm

Our interviews explored a counterintuitive finding of our study—that low experience workers actually did *worse* using the algorithm than resolving tickets manually. As expected, low experience employees agreed that they released tickets because they simply did not have enough experience to know what to do with the ticket. One participant with two years of IT experience said, "It might not be possible for us to [resolve the ticket] because we were unaware of it." Another worker with three years of experience emphasized the role of (lack 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." Because they thought it would be easier to manually resolve the ticket correctly the first time, rather than having to go back to fix it if the algorithm got it wrong, they released tickets they couldn't evaluate. This was exacerbated by the fact that they didn't have a precise idea of the algorithm's baseline performance compared to their own baseline performance, making the quality of the algorithmic advice more difficult to assess, and the outcome of its recommendation more uncertain. Based on these observations, we propose that in contexts where the costs of accepting false algorithmic advice are higher, and the baseline quality of the algorithm more uncertain, less experienced workers will be more likely to ignore algorithmic advice. This inability to judge when the algorithm is correct, and

therefore fail to implement it due to the cost of making an error, is a new explanation for noncompliance with algorithmic advice.

Why high experience workers had worse aversion: advice discounting and accountability

Highly experienced employees agreed that IT experience helped in leveraging the tool, yet they were unsurprised when we told them that the high experience employees were relatively likely to release correct tickets. Compared to low experience workers, they were more likely to recognize a runbook and know what to do with it. However, it was difficult for them to trust that 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."

They perceived themselves as possessing a much deeper understanding of the intricacies and interconnectedness of the back-end systems than lower-level employees. One participant emphasized their 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]." These highly experienced employees released tickets not because they did not know what was going on, but rather because they trusted their own ability to cleanly resolve the ticket more than the algorithm.

Yet, this explanation (which aligns with prior work on algorithm aversion) did not completely account for high experience workers' high rate of ticket release. We observed that high experienced workers felt a greater *accountability* for the result of their actions. One explained this sentiment: "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 (e.g., a key server) goes down, so what will be the impact? An inexperienced employee will never think like that." Another participant with 12 years of experience agreed: "We cannot go blindly before seeing anything and just execute the runbooks." These conversations illustrated the prevailing sense that, unlike inexperienced

employees, the more experienced had a sense of accountability to ensure that there were no unintended consequences of accepting algorithmic advice—the buck stopped with them. This sense of accountability seemed to be a meaningful driver of algorithm aversion for the most experienced workers.

Taken together, we suggest that high experience workers will exhibit less aversion in contexts where there is greater algorithmic transparency, or where workers have greater control over the algorithm-augmented process.

DISCUSSION

We began with the question: Under what conditions does a knowledge worker's domain experience increase algorithm-augmented performance? We combined insights from two distinct literatures to highlight that domain experience moderates algorithm-augmented performance via two countervailing mechanisms—ability and aversion. We argued that domain experience can increase performance via the ability to assess the quality of algorithmic advice (e.g., identify inaccurate predictions), but aversion may decrease performance via rejecting accurate algorithmic advice. Integrating these perspectives, we argued that because ability developed through learning-by-doing increases at a decreasing rate and algorithmic aversion is more prevalent among experts, algorithm-augmented performance (relative to self-performance) will first rise with domain experience, then fall—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 of IT workers to compare their performance resolving tickets with an algorithmic tool vs. resolving tickets manually. We confirmed the hypothesis, finding that only moderately experienced workers performed significantly better when using the algorithm.

Exploring mechanisms suggested that 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"), but for different reasons. Low experience workers' low performance is driven by lack of ability to assess algorithmic advice, and high experience workers' failure to improve performance using algorithms is driven by their relatively high algorithmic aversion. As evidence, we document that low experience workers were relatively likely to release more difficult (time consuming) algorithm-augmented problems, while high experience workers released algorithm-augmented tickets indiscriminately. We also documented that for the algorithm-

augmented tickets that were attempted, the relationship between performance and domain experience was positive and linear. These observations were consistent with the mechanisms underlying our predictions. We also interviewed participants, which confirmed our proposed mechanisms, and also abductively shed light on reasonable explanations for *why* highly experienced workers exhibited greater algorithm aversion: they discounted advice based on a sense that they better understood the nuances of the IT systems, and had greater accountability for unintended consequences of accepting inaccurate algorithmic advice.

Next, we pursue the theoretical implications of our theory and findings, highlighting contributions to research on human capital and technological change, research on algorithm aversion, and managerial practice. We also outline limitations and scope conditions of our study.

Implications for human capital and technological change

Though prior literature on human capital and technological change emphasizes the benefits of human domain experience for algorithm-augmented work (Autor, 2015; Brynjolfsson and Mitchell, 2017; Shrestha, Ben-Menahem and von Krogh, 2019; Choudhury, Starr and Agarwal, 2020; Raisch and Krakowski, 2020), our theoretical framework and results indicate that higher domain experience also has potential downsides for algorithm-augmented performance. Our study draws on the algorithm aversion literature to question this literature's prevailing view that more domain experience is better in algorithm-augmented work. We highlight that, although domain experience may always lead to increases in ability, domain experience may also trigger other mechanisms that inhibit algorithm-augmented performance. In other words, theoretically, there could be a limit to the extent experience positively affects algorithm-augmented work performance.

A primary contribution to this literature is a framework that incorporates a countervailing force—algorithmic aversion. We integrate previous literatures by pointing out that the countervailing forces of ability and aversion exhibit varying levels of relative strength for different levels of domain experience. This framework generates a new prediction that intermediate levels of domain experience provide the greatest increases in algorithm-augmented performance (relative to self-performance). These theoretical implications suggest a need for increased sensitivity to the multiplicity of mechanisms at play when workers augment their judgment using algorithms. Whether human domain experience complements algorithms depends heavily on whether the ability

effect or the aversion affect is stronger for a worker with a given level of domain experience in a given context (see scope conditions and limitations section below).

Our contribution to this literature echoes, but is distinct from, the broad literature on the adoption of new technologies in general. Prior work highlights that workers with more domain experience may have decreased motivation to adopt new technologies due to a vested stake in the status quo (Barley, 1986; Henderson, 1993; Edmondson, Bohmer and Pisano, 2001; Helfat and Peteraf, 2003; Kellogg, 2014; Eggers and Kaul, 2018; Greenwood *et al.*, 2019). Unlike these prior studies, the new algorithmic tool in our context did not represent a major threat to the status quo—i.e., the livelihood, status, or economically significant human capital investments of workers. For the participants in our study sample, resolving help tickets represents just a small portion of overall work, and in field interviews they reported that the ticket resolution task was considered "low status and menial". In short, the algorithmic tool was a "welcome relief" to make their work "faster and easier." Yet there was still aversion from more experienced workers, for reasons we discuss below.

Implications for algorithm aversion

Whereas prior literature on algorithm aversion implies that expertise is a liability for algorithm-augmented judgments (Arkes, Dawes and Christensen, 1986; Logg, Minson and Moore, 2019), we counter that domain experience is, in fact, the primary means by which humans have any potential to complement algorithmic judgement. Our theory and results highlight that, because domain experience increases complementarity via increased ability, increasing domain experience actually increases algorithm-augmented performance for low experience workers (or others who do not exhibit high levels of algorithm aversion). Thus, the primary question in whether an expert will do better or worse with an algorithm is not merely how much domain experience they have, but rather whether the aversion effect overpowers the ability effect in a given context (see scope conditions and limitations section below).

Our study also highlights that algorithm aversion is present despite the technological shift to algorithms built using machine learning. Experts have long been averse to algorithms of past technology vintages, such as expert rule-based systems (Dreyfus and Dreyfus, 1986) or decision rules (Arkes, Dawes and Christensen, 1986). But it is not obvious that experts would have the same aversion to algorithms built using machine learning, which rely on an inductive learning approach rather than hardwired codifiable knowledge (Choudhury, Allen and

Endres, 2020). When AI meant rule-based logic, observers of expert systems observed that experts don't think by rules, so AI would have limited usefulness to them in most contexts (Dreyfus and Dreyfus, 1986). Yet, we observe that even when AI more closely resembles the case-based inductive learning processes of human experts, aversion persists for other reasons—such as egocentric advice discounting and accountability. In other words, experts may not be averse to codified decision rules, but have a more general aversion toward algorithms, broadly conceived.

Our theoretical contributions were driven by two key empirical contributions to this literature. First, we explored the forces at play across a *gradient* of experience—which allowed us to uncover the inverted U-shaped relationship perhaps hidden by the binary classifications (e.g., "expert" vs. "layperson") used in previous lab studies (Logg, Minson and Moore, 2019). Our supplementary between-subjects analysis is partially consistent with previous lab experiments that indicate that experts may reject advice more often than nonexperts (Arkes, Dawes, and Christensen 1986, Logg, Minson, and Moore 2019). Comparing moderately experienced to highly experienced participants in our study, there was evidence that the highly experienced rejected good algorithmic advice more frequently. However, comparing the least experienced to the most experienced participants, there was not much of a difference. We reconcile differences between studies below in the scope conditions and limitations section.

Second, because we empirically observed aversion outside the lab, we were able to notice a relatively novel mechanism for explaining *why* higher experience workers have greater algorithm aversion: greater accountability for possible unintended consequences of accepting inaccurate algorithmic advice. In addition to the usual explanations for aversion, our interviews revealed that high experience workers were more aware of the potential consequences of their actions than their lower experience counterparts. These high experience workers expressed greater accountability for the smooth operation of the firm's IT systems. If the system crashed, it would be their fault and they would have to explain what went wrong. This greater accountability, along with the belief that they had a deeper understanding of the systems than the algorithm, prompted the high experience workers to release tickets to resolve manually—at a far greater rate than necessary based on the underlying accuracy of the algorithm. This observation echoes recent observations that experienced radiologists who bear financial and legal accountability for diagnoses ignore algorithmic advice when they are unable to interrogate the

reasoning behind the algorithmic recommendation (Lebovitz, Lifshitz-Assaf and Levina, 2020). We suggest that accountability as an explanation of aversion is not easily observable in a lab, but will be readily observed in real-world organizational settings.

Scope Conditions, Other Limitations and Future Research Directions

We expect that the forces of ability and aversion will be present in a wide variety of contexts, but contextual factors will influence the relative intensity of the forces, and thus sharpen or flatten the inverted U-shape of overall performance. Here we highlight a few contextual factors that we expect to be particularly salient. First, in some contexts, experts may feel more threatened by algorithms if the algorithms have a significant impact on their professional identity or livelihood (Kellogg, Valentine and Christin, 2020). We expect this would increase the relative intensity of the aversion effect for higher experience workers, causing a more precipitous decline in the inverted-U shape. Second, based on our own observations and prior literature (Dietvorst, Simmons and Massey, 2016), we expect that greater control over the algorithm in both designing the algorithm and while overriding algorithmic advice in production settings, and/or greater algorithmic transparency, would decrease aversion and thus flatten the U-shape. Third, we expect that the accuracy of the algorithm (overall and relative to the human), should have significant impact on algorithm-augmented performance relative to self-performance. Though we are not aware of work on this topic, we expect that—assuming humans have an accurate perception of the algorithm's accuracy—more accurate algorithms will flatten the U-shape.

Another contextual factor may help reconcile our findings with lab studies conducted in other contexts. Domain experience gleaned from learning-by-doing may have a limited impact on ability in contexts with high causal ambiguity (Kahneman and Klein, 2009)—such as geopolitical forecasting (Tetlock, 2009). In these cases, we expect the influence of aversion to more quickly outweigh ability over the range of domain experience. Accordingly, algorithm-augmented performance could actually become *worse* than self-performance (which may help explain the worse performance of geopolitical forecasting experts in Logg, Minson and Moore, 2019).

Due to such contextual dependencies, our study takes no general stance on whether algorithm-augmented performance will be worse than (or significantly better than) self-performance. It merely predicts that the best algorithm-augmented performance (relative to self-performance) will be achieved by those with

moderate levels of domain experience—a result we expect to hold to varying degrees in a variety of contexts. Future studies can test these expectations by varying some of the contextual factors listed above.

Our study has other limitations, suggesting a rich agenda for future research. First, 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, and how quickly workers with varying levels of experience learn to trust (or mistrust) an algorithm. Second, 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. Third, 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). 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.

Implications for Practice

Our research guides managers seeking to design effective hybrid human/algorithm decision processes. A key insight is that effective interventions may look quite different for employees with different levels of experience.

High experience workers, who tend have high ability but also high aversion, will benefit most from efforts to decrease aversion. One intervention could be to design algorithmic tools and processes with greater transparency, and with greater human control over the algorithmic actions. Greater transparency and control would allow high experience workers to interrogate the algorithm's reasoning, and feel more at ease with a sense of control over the end result (Dietvorst, Simmons and Massey, 2016). This would also give high experience workers, who feel a greater sense of accountability for the final result, a way to responsibly make use of an algorithm without having to trust their responsibility to a black box.[17] For this reason, we predict the emerging stream of research related to algorithmic accountability (e.g., the 'fairness, accountability and transparency' or FAT stream of the literature, see Shin and Park 2019) will be increasingly relevant as it is applied to policy

[17] To quote Garfinkel et al. 2017, "Accountability rejects the common deflection of blame to an automated system by ensuring those who deploy an algorithm cannot eschew responsibility for its actions."

discussions on algorithmic accountability outside the lab (e.g., discussions hosted by the ACM U.S. Public Policy Council and ACM Europe Council Policy Committee).

Low experience workers, who tend to have low aversion but low ability, will likely benefit more from a different set of interventions. Although it's not possible to instantly grant a worker additional domain experience (which takes years of learning-by-doing), it is possible to design training, process, and incentives so that workers are more inclined to accept an accurate algorithm's advice. Based on our analyses, we suggest two ways to reduce algorithmic noncompliance among low experience workers. First, we suggest lowering the cost of accepting inaccurate algorithmic advice. In our study, participants who could not assess the algorithm's accuracy felt it would be too costly to get it wrong, and instead preferred "releasing" the ticket for someone to get it right the first time. Second, we suggest training workers to understand the baseline quality of algorithmic recommendations. In our context, if low experience workers had understood that the algorithm was accurate 90% of the time, and their own baseline performance was lower, they might have been more inclined to defer to the algorithm's advice. These interventions will be applicable in settings where greater algorithmic compliance is actually desirable—i.e., the algorithm tends to outperform workers with low experience.

CONCLUSION

If current trends are any guide, work at the intersection of humans and algorithms will grow as an important topic for organizational scholars. Existing work has focused on the role of domain experience in achieving human-algorithm complementarities, with mixed perspectives on whether domain experience increases or inhibits algorithm-augmented performance. By proposing a unifying framework of how domain experience affects performance via countervailing mechanisms—ability and aversion—we help reconcile these perspectives. We hope to inspire future work on the multiplicity of competing mechanisms at play when workers use algorithms in their work.

REFERENCES

Agrawal, A., Gans, J. and Goldfarb, A. (2017) 'Introduction', in *Economics of Artificial Intelligence*. Toronto.

Agrawal, A. K., Gans, J. S. and Goldfarb, A. (2018) 'Prediction, Judgment and Complexity', *NBER Working Paper Series*, p. 27. Available at: http://www.nber.org/papers/w24243.

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. and Mitchell, T. (2017) 'What can machine learning do? Workforce implications', *Science*, 358(6370), pp. 1530–1534. doi: 10.1126/science.aap8062.

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.

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*.

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

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'.

Dane, E., Rockmann, K. W. and Pratt, M. G. (2012) 'When should I trust my gut? Linking domain expertise to intuitive decision-making effectiveness', *Organizational Behavior and Human Decision Processes*. Elsevier Inc., 119(2), pp. 187–194. doi: 10.1016/j.obhdp.2012.07.009.

Dasgupta, P. and David, P. A. (1994) 'Toward a new economics of science', *Research policy*. Elsevier, 23(5), pp. 487–521.

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.

Dreyfus, H. L. and Dreyfus, S. E. (1986) 'From Socrates to expert systems: The limits of calculative rationality', in *Philosophy and technology II*. Springer, pp. 111–130.

Edmondson, A., Bohmer, R. and Pisano, G. (2001) 'Disrupted Routines: Team Learning and New Technology Implementation in Hospitals', *Administrative Science Quarterly*, 46(4), p. 685. doi: 10.2307/3094828.

Eggers, J. P. and Kaul, A. (2018) 'Motivation and ability? A behavioral perspective on the pursuit of radical

invention in multi-technology incumbents', *Academy of Management Journal*, 61(1), pp. 67–93. doi: 10.5465/amj.2015.1123.
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.
Garfinkel, S. *et al.* (2017) 'Toward algorithmic transparency and accountability'. ACM New York, NY, USA.
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.
Glaeser, E. L. *et al.* (2021) 'Decision Authority and the Returns to Algorithms'.
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.
Helfat, C. E. and Peteraf, M. A. (2003) 'The dynamic resource-based view: Capability lifecycles', *Strategic Management Journal*, 24(10 SPEC ISS.), pp. 997–1010. doi: 10.1002/smj.332.
Henderson, R. (1993) 'Underinvestment and incompetence as responses to radical innovation: evidence from the photolithographic alignment equipment industry', *The RAND Journal of Economics*. WileyRAND Corporation, 24(2), pp. 248–270. doi: 10.2307/2555761.
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.
Kahneman, D. and Klein, G. (2009) 'Conditions for Intuitive Expertise: A Failure to Disagree', *American Psychologist*, 64(6), pp. 515–526. doi: 10.1037/a0016755.
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.
Lebovitz, S., Lifshitz-Assaf, H. and Levina, N. (2020) 'To incorporate or not to incorporate AI for critical judgments: The importance of ambiguity in professionals' judgment process ', *NYU Stern School of Business*.
Lesgold, A. *et al.* (1988) 'Expertise in a complex skill: Diagnosing x-ray pictures', *American Psychological Association*.
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. (2013) *The expertise paradox: Examining the role of different aspects of expertise in biased evaluation of scientific information*. University of California, Irvine.
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. (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.
Pearl, J. (2009) *Causality*. Cambridge university press.
Pearl, J. and Mackenzie, D. (2018) *The book of why: the new science of cause and effect*. Basic books.
Polanyi, M. (1966) 'The Logic of Tacit Inference', *Philosophy*, 41(155), pp. 1–18.
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.
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.
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.
Shin, D. and Park, Y. J. (2019) 'Role of fairness, accountability, and transparency in algorithmic affordance', *Computers in Human Behavior*. Elsevier, 98, pp. 277–284.
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.
Soll, J. B. and Mannes, A. E. (2011) 'Judgmental aggregation strategies depend on whether the self is involved', *International Journal of Forecasting*. Elsevier, 27(1), pp. 81–102.
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.
Tetlock, P. E. (2009) *Expert political judgment: How good is it? How can we know?* Princeton University Press.
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.
Yang, H. (2021) *Cognitive Reflection and Algorithmic Aversion*.
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.

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 modeling domain experience as quadratic term

	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		
Experiment Session Fixed Effects	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. Each column 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 CR2 standard errors clustered at the participant level (Pustejovsky and Tipton, 2018).

Table 3. OLS Regressions on domain experience subsamples

	All Tickets				First Appearance Tickets (No Recurring Tickets)			
Dependent Variable: Ticket Resolved	**(1) ≤3 years**	**(2) >3 years, ≤6 years**	**(3) >6 years, ≤9 years**	**(4) >9 years**	**(5) ≤3 years**	**(6) >3 years, ≤6 years**	**(7) >6 years, ≤9 years**	**(8) >9 years**
Is AutomateIT Ticket	**-0.212 (0.047)*** **	**0.162 (0.050)** **	**0.198 (0.054)*** **	**-0.051 (0.051)**	**-0.294 (0.055)*** **	**-0.006 (0.094)**	**0.211 (0.084)* **	**0.048 (0.068)**
Controls	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Employee Experience Level Fixed Effects					Yes	Yes	Yes	Yes
Experiment Session Fixed Effects					Yes	Yes	Yes	Yes
Participant Fixed Effects	Yes	Yes	Yes	Yes				
First Appearance Tickets Subsample					Yes	Yes	Yes	Yes
Adj. R^2	0.318	0.180	0.228	0.148	0.173	-0.021	0.076	0.034
Num. obs.	360	352	208	304	186	182	123	185

Notes. Each column includes controls for all the control variables listed in Table 1 (columns 5-8 do not include *Recurring Ticket* because they use the subsample of first-appearance tickets). Asterisks indicate statistical significance at p-value cutoffs: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$. All regressions use CR2 standard errors clustered at the participant level (Pustejovsky and Tipton, 2018).

Figure 1. Theoretical Framework Visualization

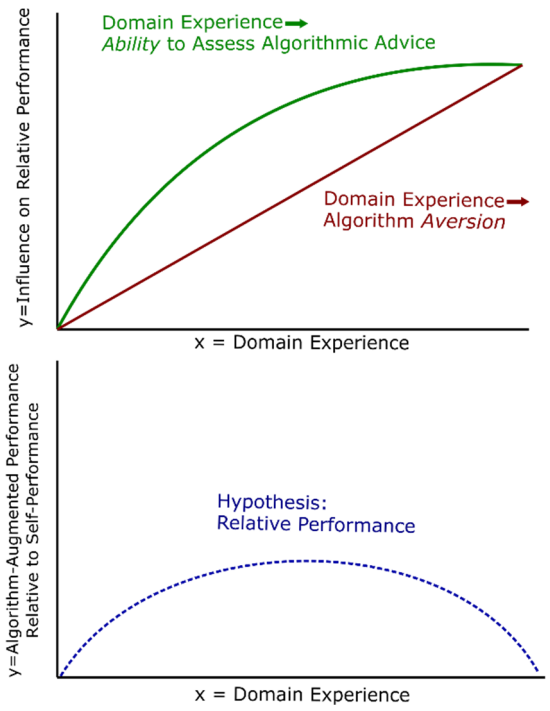
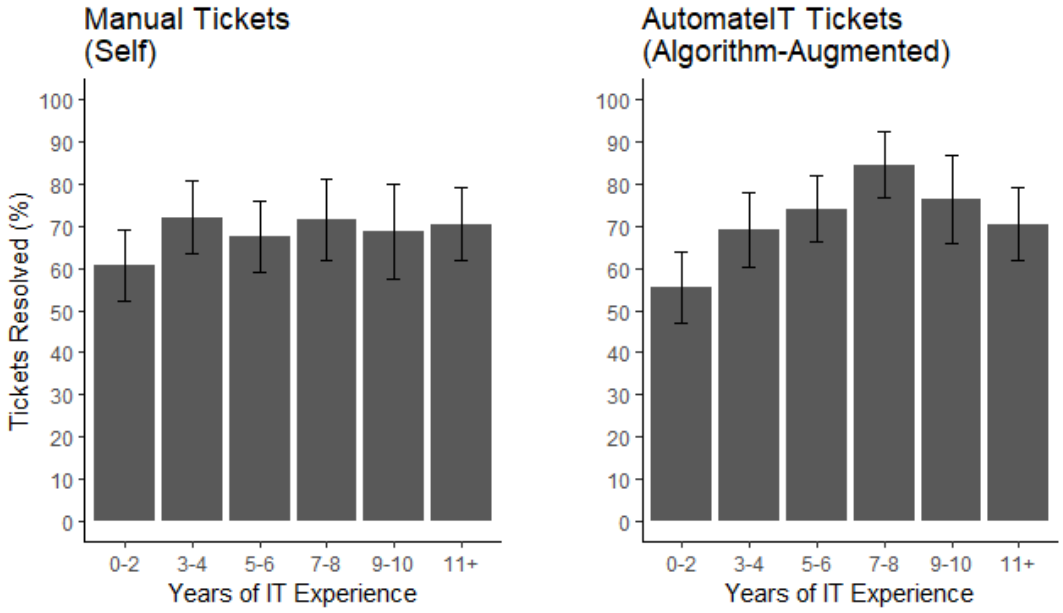
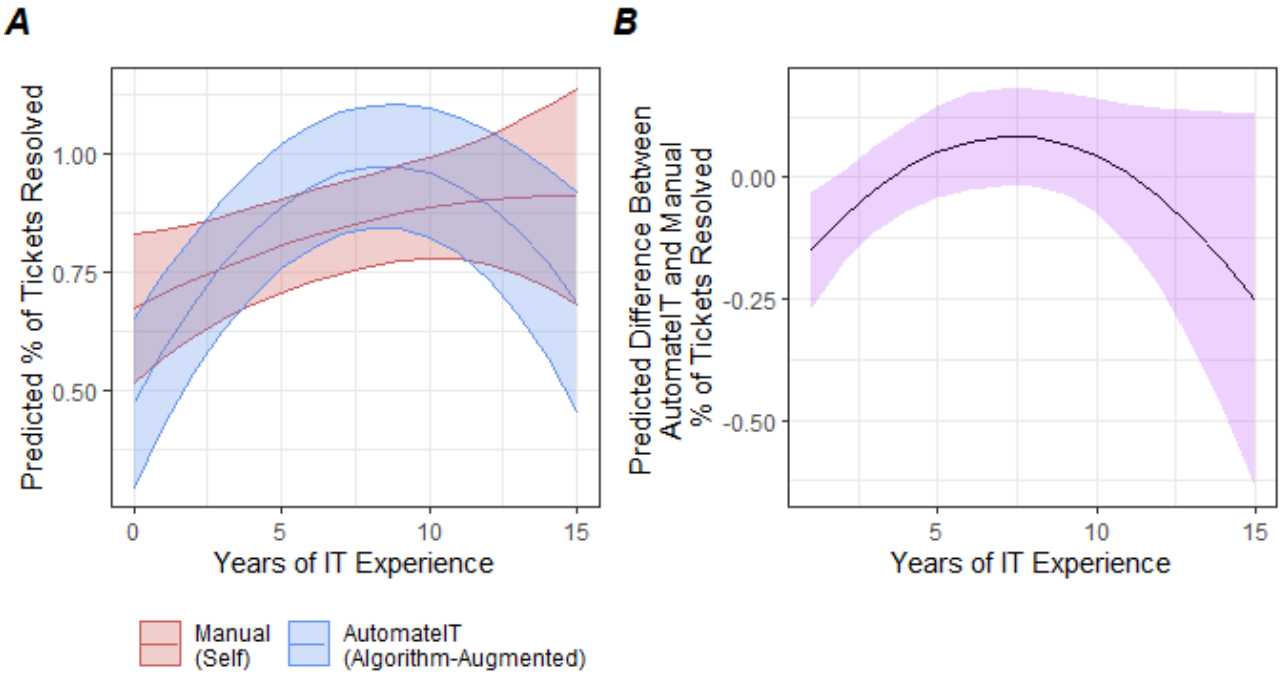


Figure 2. Raw Percentage of Tickets Resolved Across Range of Experience for Manual and AutomateIT Tickets



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. Panel A displays predicted percentage of tickets resolved for AutomateIT and Manual tickets, conditional on Years of IT Experience. Panel B displays the predicted difference in percentage of tickets resolved for AutomateIT vs. manual tickets. Predicted values were obtained using the model in Table 2, column 3.

Figure 4. Rates of Correct Algorithmic Predictions, Runbook "Attempts," Correct Participant Runbook Selection, and Ticket Resolution

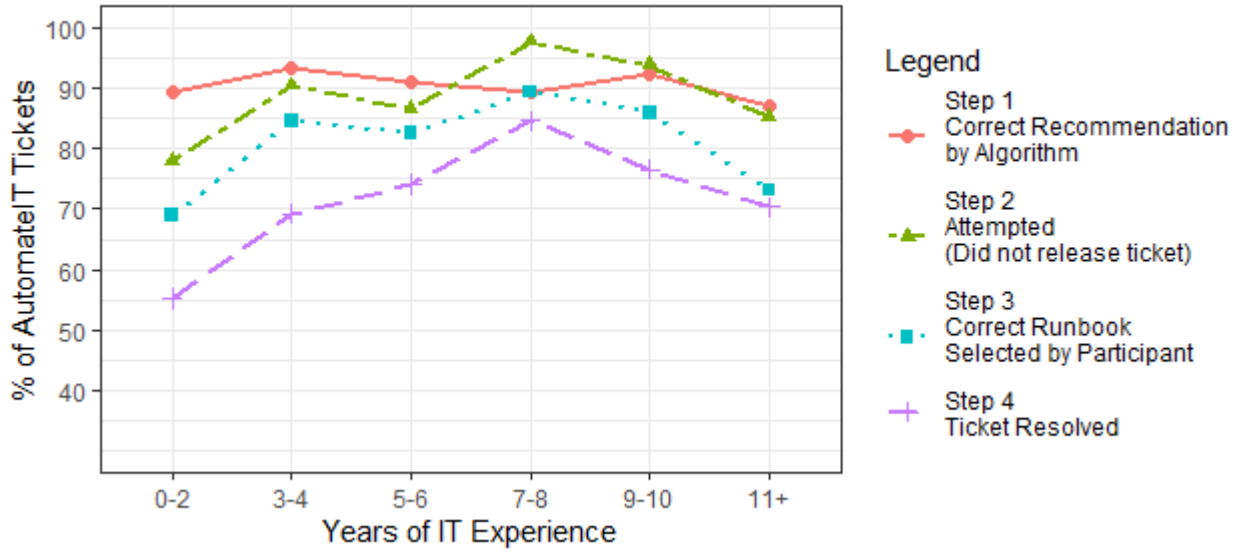
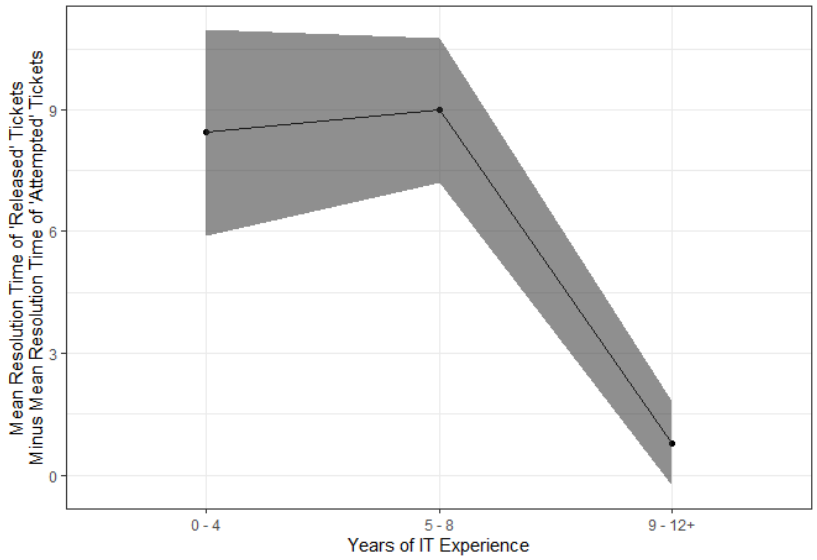
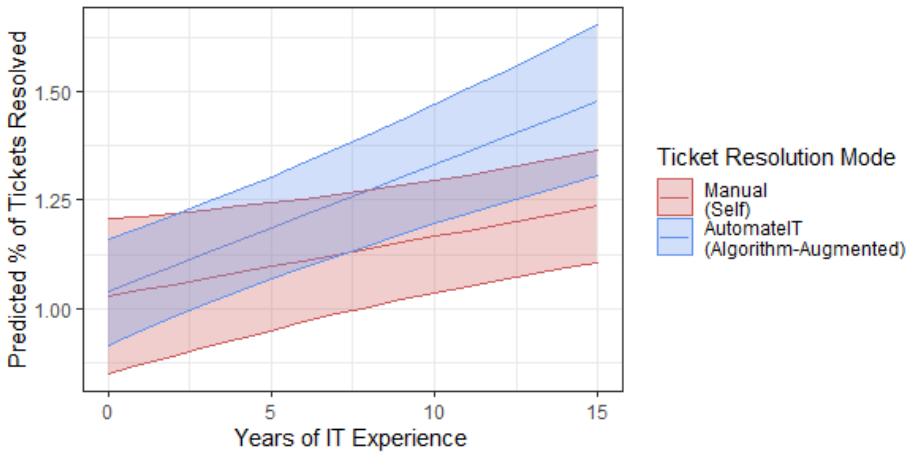


Figure 5. Difference in Time Spent on Released vs. Attempted Tickets for Different Levels of Domain Experience



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 zero). We were not able to observe how long participants spent on 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 6. 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 coefficient estimates and model used to produce this figure are included in Appendix Table A1, column 3.

APPENDIX

Appendix Table A1. 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**	**0.867**	**0.016**
	(0.037)*	**(0.410)***	**(0.0085)+**
Is AutomateIT Ticket * Years of IT Experience Squared	**-0.008**	**-0.083**	
	(0.003)*	**(0.035)***	
Is AutomateIT Ticket	-0.152	-1.677	0.063
	(0.138)	(1.152)	(0.078)
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
Experimental Session Fixed Effects		Yes	
Participant Fixed Effects	Yes		Yes
Adj. R^2	0.230	0.074	0.151
Num. obs.	856	340	1085

Notes: This table displays 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* (1 if ticket was resolved, else 0). Models include fixed effects for experimental session or individual participant fixed effects. Columns 1 and 2 included the same set of controls as the models in Table 2. Column 3 uses a minimal set of controls: *OS Track, Ticket Matches OS Track,* and *Years at Company* and *Ticket Order* (interactions with the quadratics were not included because we were testing a linear effect). All models used clustered standard errors, clustered at the participant level. Asterisks indicate statistical significance at p-value cutoffs: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$, $^{+}p<0.1$.

Appendix 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}.

Appendix Table A3. OLS Regressions on ticket type subsamples

	All Tickets		First Appearance Tickets (No Recurring Tickets)	
Dependent Variable: Ticket Resolved	**(1) Tickets Resolved Manually**	**(2) Tickets Resolved with AutomateIT**	**(3) Tickets Resolved Manually**	**(4) Tickets Resolved with AutomateIT**
Intercept (Baseline 0-3 Years of IT Experience)	0.874 (0.066)***	0.696 (0.069)***	1.064 (0.075)***	0.792 (0.082)***
3-6 Years of IT Experience	0.023 (0.104)	0.187 (0.099)+	-0.072 (0.279)	0.226 (0.100)*
6-9 Years of IT Experience	0.143 (0.128)	0.165 (0.119)	0.042 (0.289)	0.179 (0.178)
10+ Years of IT Experience	0.233 (0.144)	0.006 (0.140)	0.173 (0.296)	0.030 (0.192)
Controls	Yes	Yes	Yes	Yes
Employee Experience Level Fixed Effects	Yes	Yes	Yes	Yes
Experiment Session Fixed Effects	Yes	Yes	Yes	Yes
First Appearance Tickets Subsample			Yes	Yes
Adj. R^2	0.092	0.072	0.038	0.087
Num. obs.	612	612	326	350

Notes. Each column includes controls for all the control variables listed in Table 1 (columns 3-4 do not include *Recurring Ticket* because they use the subsample of first-appearance tickets). Asterisks indicate statistical significance at p-value cutoffs: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$. All regressions use CR2 standard errors clustered at the participant level (Pustejovsky and Tipton 2018).

Appendix Figure A1. Example illustration of manual ticket resolution process

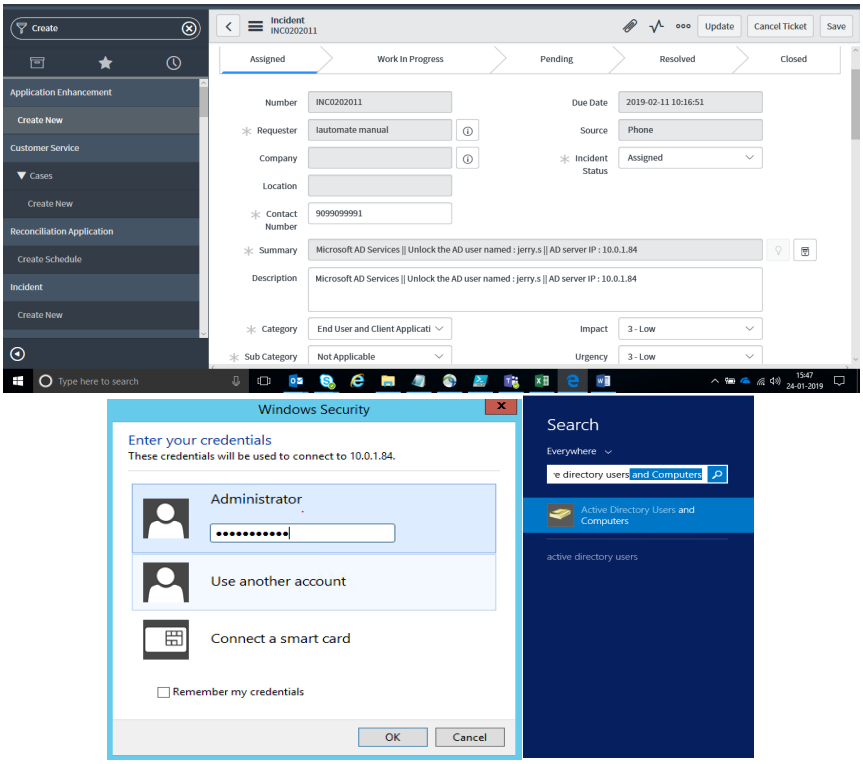
Example: granting permissions

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

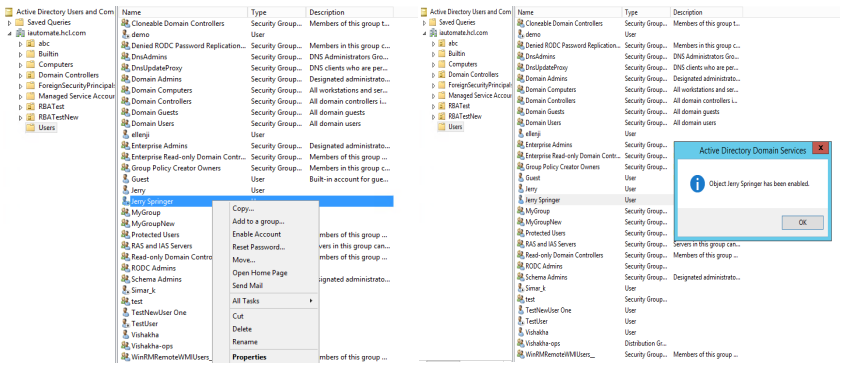
Step 2. (not pictured) if needed, consult the runbooks for how to resolve the problem

Step 3. Logging in as a system administrator

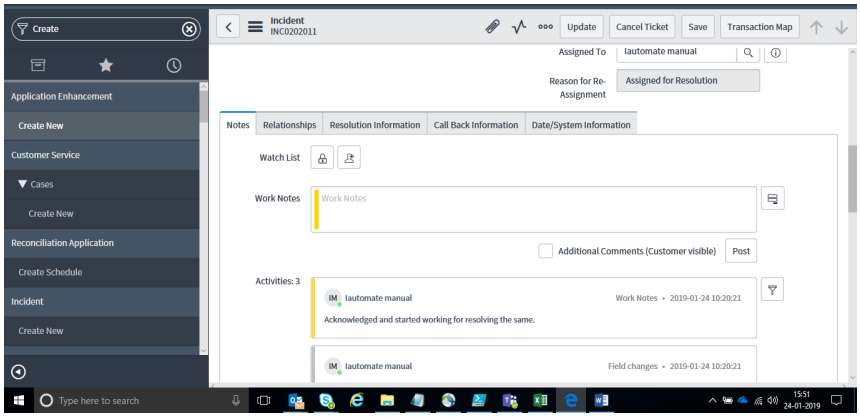
Step 4. Finding the correct directory of users



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



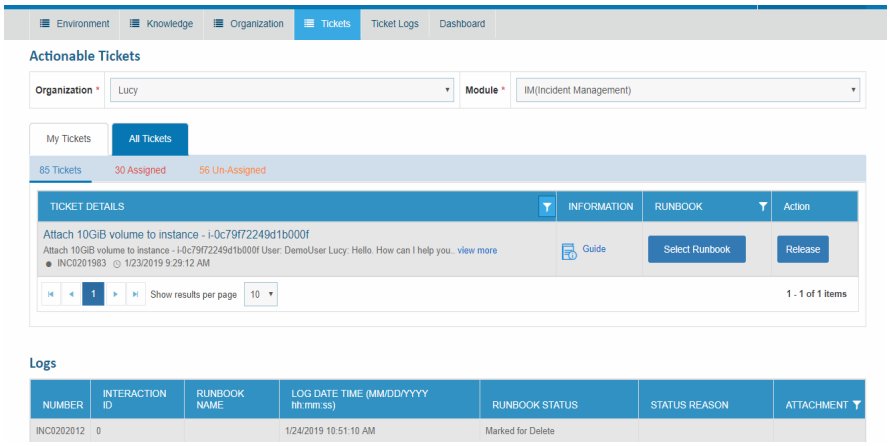
Step 6. Navigating back to console, updating the ticket and marking as resolved (if unable to resolve they would mark as unresolved)



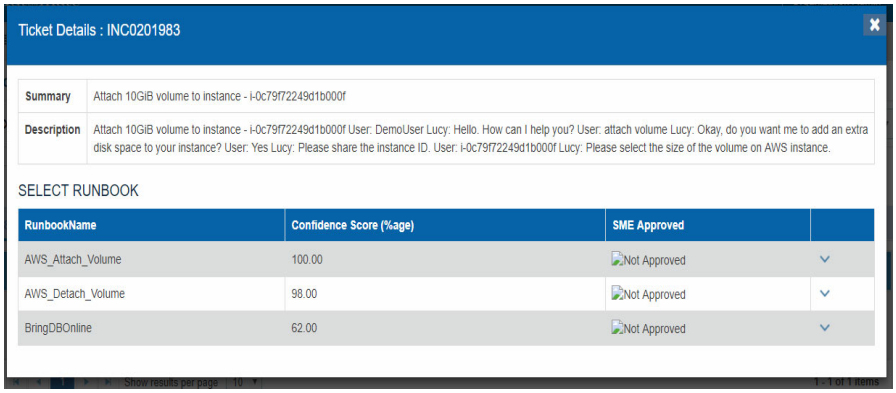
Appendix Figure A2. Example illustration of AutomateIT ticket resolution process

Example: Attaching AWS Instance

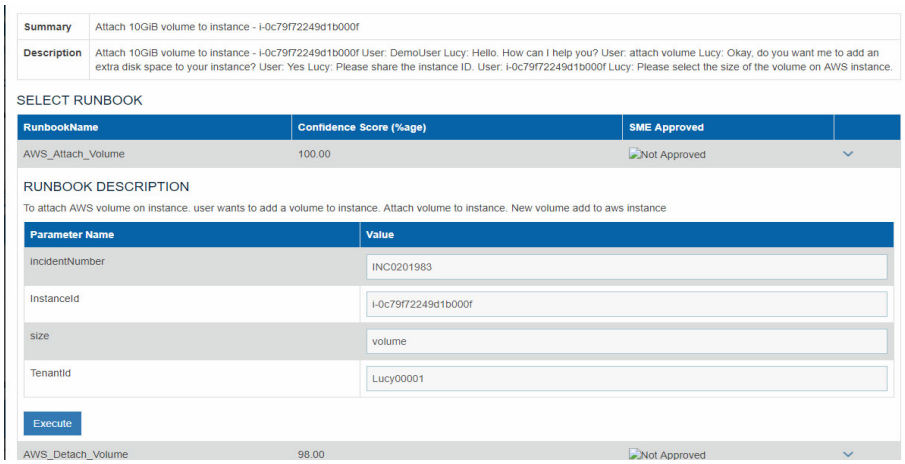
Step 1. Logging in to AutomateIT console and selecting from list of pending tickets



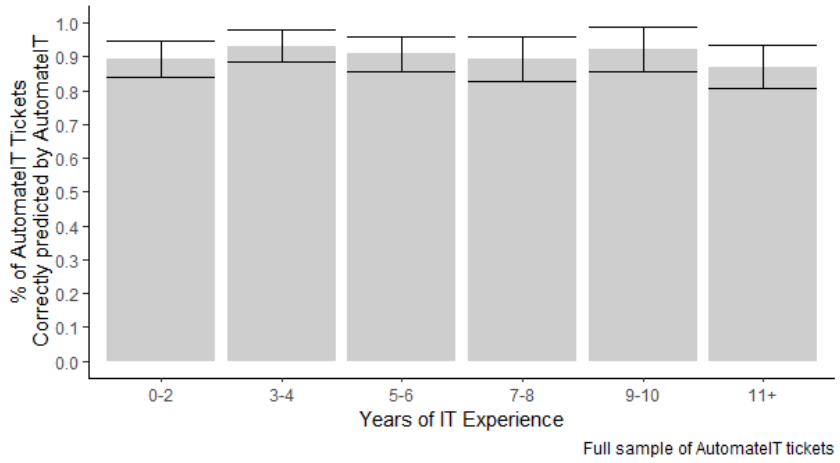
Step 2. Viewing the problem statement, description, and the list of recommended runbook solutions.



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

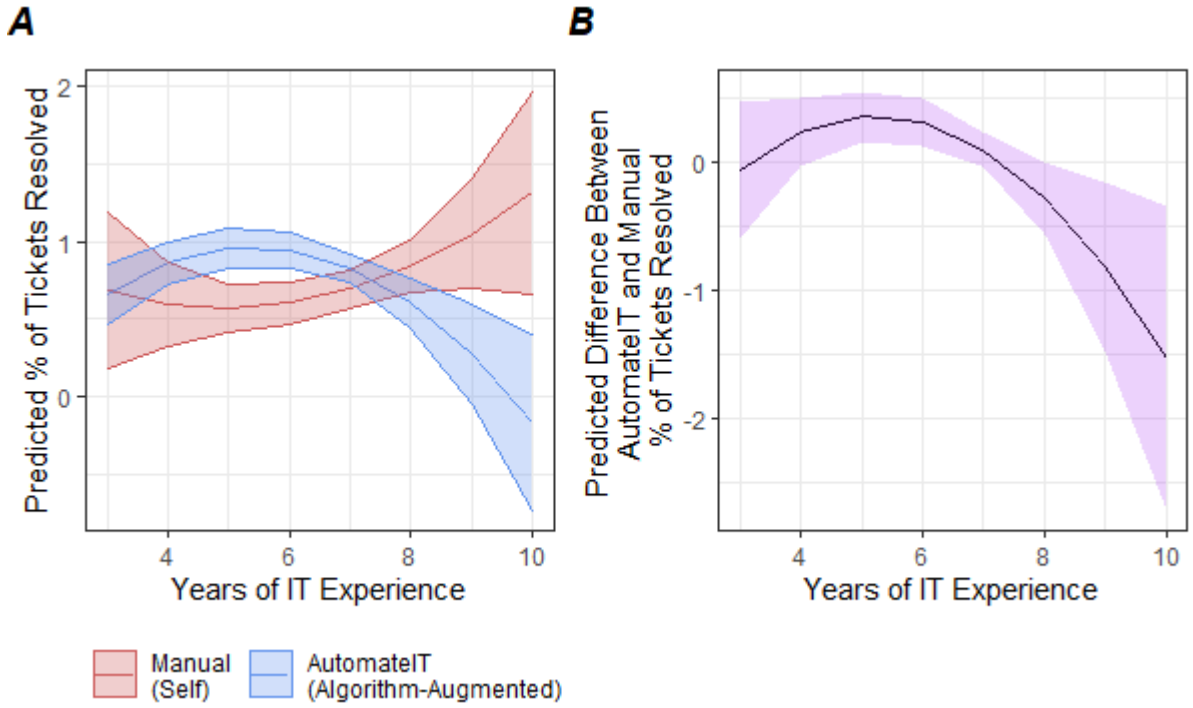


Appendix Figure A3. Percentage of correct algorithmic recommendations across range of experience



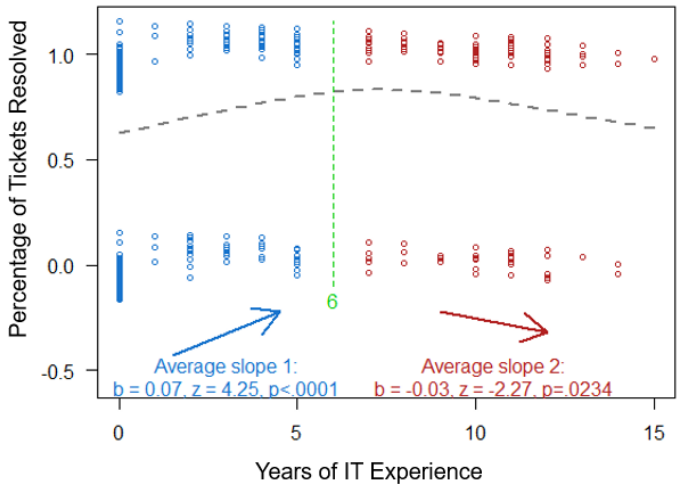
Notes: Gray bars represent the raw % of correct algorithmic recommendations (95% confidence interval error bars) for participants with varying levels of IT experience.

Appendix Figure A4. Level 2 Subsample Analysis: Predicted Effects for Ticket Resolutions and Years of IT Experience



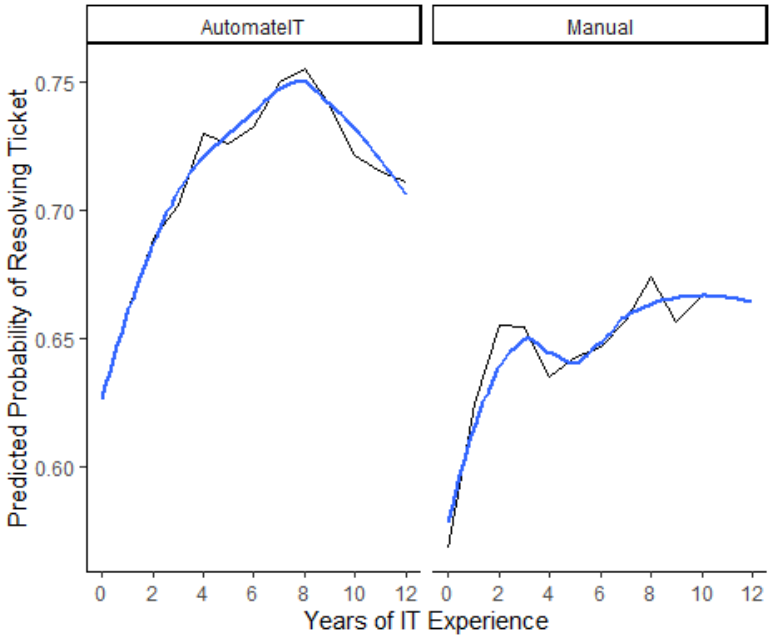
Notes. The figure displays predicted effects for the subsample of Level 2 employees. Panel A displays predicted percentage of tickets resolved for AutomateIT and Manual tickets, conditional on *Years of IT Experience*. Panel B displays the predicted difference in percentage of tickets resolved for AutomateIT vs. manual tickets. 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 A1, column 2.

Appendix Figure A5. Two-Lines Test for AutomateIT Ticket Resolutions and Years of IT Experience



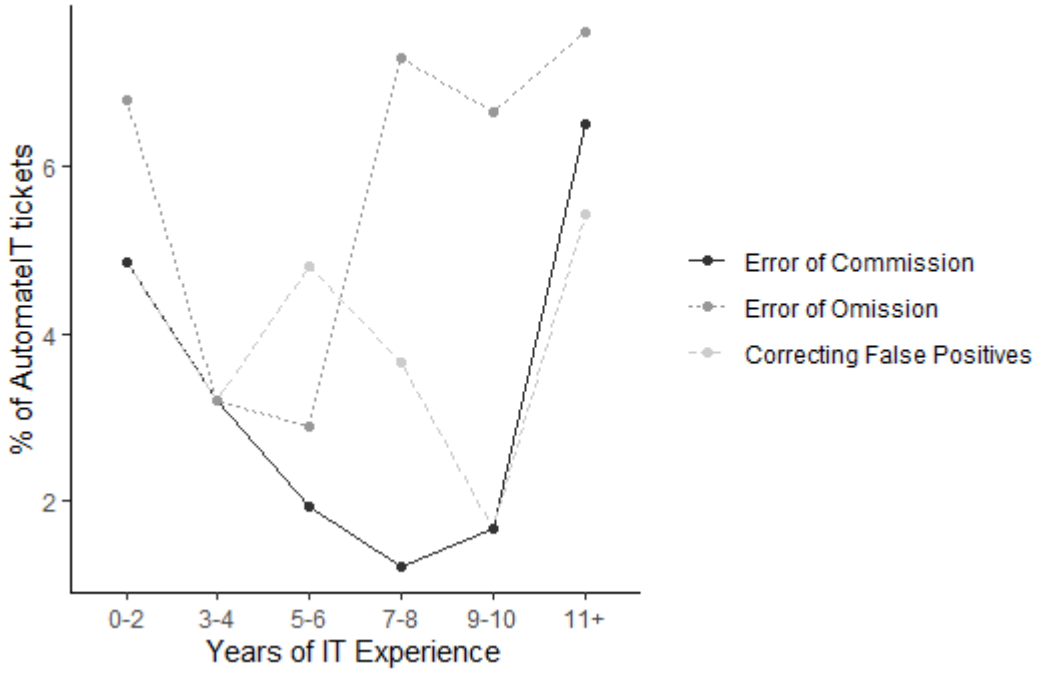
Notes. This figure displays results from Simonsohn's (2018) "two-lines" test applied to the variables *Ticket Resolved* and *Years of IT Experience*Is AutomateIT Ticket*. We ran the test including all the same variables and controls as in the model in Table 2, column 1. The test finds whether there is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

Appendix Figure A6. 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.

Appendix Figure A7. Rates of Error for 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.