

# Generative AI and Security Operations Center Productivity: Evidence from Live Operations

James Bono  
Alec Xu  
Justin Grana  
Microsoft Corporation  
November 2024



# Abstract

We measure the association between generative AI (GAI) tool adoption and security operations center productivity.

**We find that GAI adoption is associated with a 30.13% reduction in security incident mean time to resolution.**

This result is robust to several modeling decisions. While unobserved confounders inhibit causal identification, this result is among the first to use observational data from live operations to investigate the relationship between GAI adoption and security worker productivity.

## 1. Introduction and Background

Recent developments in generative artificial intelligence (GAI) have raised questions regarding its productivity effects. While GAI-based productivity enhancements have important labor market implications [10, 19], its importance in cybersecurity extends beyond labor cost savings and has the potential to reduce costly attacks. Despite its importance, there is scant evidence detailing the potential benefits of GAI tools for security professionals in live operations.

We present the first statistical evidence of productivity enhancements from GAI tools accruing to security operations center (SOCs) analysts in live operations. Specifically, we conduct a difference-in-differences analysis of security incident mean time to resolution (MTTR) where the treatment is defined as adoption of Microsoft Security Copilot (“Copilot”), which “combines a specialized language model with security-specific capabilities” [12].

**Our results indicate Copilot is associated with a 30.13% reduction in MTTR three months post-adoption relative to a control group in the same period.**

Although selection into treatment inhibits causal identification, our estimated effect on productivity aligns in magnitude with other recent work that leverages randomization to identify GAI’s causal effects on knowledge-worker productivity [9, 15, 8]. Given that recent estimates suggest analysts spend, on average, 2.7 hours per day resolving incidents, costing \$3.3B in the US alone [1], our results suggest that GAI tools have the potential to offer SOC’s significant time and cost savings.

### 1.1 Related Work

From a productivity perspective, our work sits adjacent to recent studies that examine the impact of GAI on worker productivity utilizing randomization in either a laboratory or field setting [9, 15, 8, 14, 5]. While lab experiments provide solid causal identification, whether they generalize to real world settings is a matter of judgment about the experimental design. Field experiments make a similar trade-off to lab studies. Random assignment gives them strong causal identification, but because they are often limited to a small set of organizations, the measured effects may not generalize more broadly. Our work makes the opposite trade-off; we examine telemetry data from live operations from over 150 organizations (see [section 2](#)), so our results are likely to generalize to live operations for a broad range of organizations. However, our causal claims are weaker than lab and field experiments due to our inability to control for selection into treatment. Our work is more related to the limited studies that use observational data [4, 21] in live operations to estimate the impact of GAI on productivity. As is typical in these studies, confoundedness cannot be guaranteed and thus our results provide evidence for—though do not causally identify—GAI’s impact on security worker productivity. Despite their different shortcomings, studies have yielded surprisingly consistent estimates of GAI’s productivity effects across domains (see [table 1](#) for specific estimates) with our 30.13% estimate being similarly consistent.

| Source | Domain   | Estimated Productivity Gain  |
|--------|--|--|
| [4]    | Customer technical support                                   | <b>34% reduction</b> in task completion time for novices                               |
| [9]    | Security incident laboratory experiment                      | <b>23% decrease</b> in task completion time  |
| [15]   | Laboratory experiment implementing HTTP server in Javascript | <b>55.8% decrease</b> in time to completion  |
| [8]    | Field study of software development tasks                    | <b>26.08% increase</b> in tasks completed  |
| [3]    | Laboratory study of IT Admins                                | <b>34.53% accuracy improvement</b> and <b>30.69% reduction</b> in task completion time |

Table 1: Estimated Productivity Gains from Generative AI

From the security perspective, our study builds on research about automation of security tasks. The literature on this topic finds that risk mitigation is essential with the growth of security breaches, especially because so many security vulnerabilities operate at the gap between how systems are supposed to operate and how they actually operate [13]. Although many aspects of cybersecurity currently rely on human subject matter experts [7], researchers point out the possibility of automating error-prone and time-consuming security work. Machine learning techniques show particular promise in intelligently analyzing cybersecurity data [17]. Natural language processing, knowledge representation and reasoning, and rule-based expert systems modeling can also support AI-driven cybersecurity [18].

## 1.2 Security Operations Center and Copilot

### What is Security Event Management?

Security event triage and response is a central function of an organization’s cybersecurity operations [20]. Broadly, computer network activity generates telemetry (logs, signatures, etc.). That telemetry is then collated and processed by tools such as security information and event management (SIEM) and extended detection and response (XDR) solutions. These solutions often generate security alerts based on logical rules or statistics and machine learning models. Ultimately, SIEM and XDR solutions aggregate telemetry into a set of distinct units, representing holistic depictions of suspicious activity, for human analysts to investigate. We adopt the convention of referring to these aggregate units as “security incidents” or just “incidents.”

Teams of analysts triage and resolve incidents. Some incidents may be resolved as a “false positive” where the suspicious behavior is determined to be benign. Other incidents may require more intensive remediation and response such as altering users’ privileges, disconnecting compromised systems, removing malicious files, and applying patches. Once an analyst resolves an incident, they can proceed to the next incident in their SIEM or XDR solution. However, as noted above, the incident arrival rate far surpasses what a typical SOC can effectively triage. Recent estimates suggest that as much of 67% of security incidents go unresolved [1].

### What is Security Copilot?

Microsoft Security Copilot (“Copilot”) is a GAI tool designed to enhance the operational efficiency and effectiveness of SOCs. One salient feature related to incident resolution is incident summary. Previously, analysts would need to investigate each piece of information independently to obtain the full breadth of an incident. Copilot instead aggregates that information and serves it to the user in human-readable format. The right panel in [Figure 1](#) illustrates an example of alert summarization for a Business Email Compromise incident.

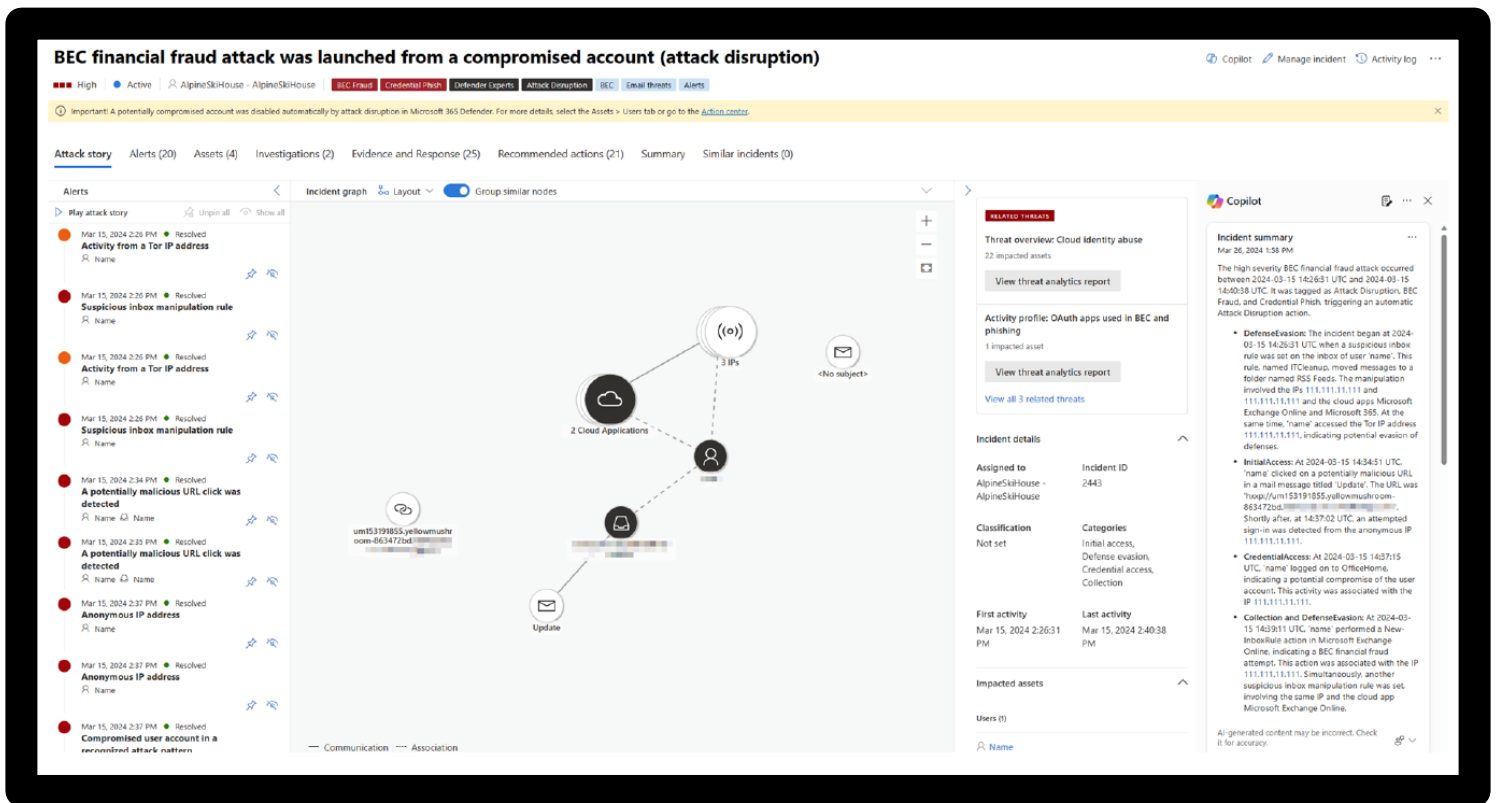


Figure 1: An example of incident summarization for a Business Email Compromise Incident

Although a full list of the product features is beyond the scope of this paper, we mention the following additional features to give the reader a better understanding of the mechanisms by which Copilot affects MTTR:

- Guiding analysts to the appropriate response actions based on the incident data
- Helping interpret malicious scripts associated with the incident
- Creating scripts to query security logs based on natural language inputs
- Retrieving relevant threat intelligence using natural language inputs

## 2. Data and Method

We leverage system metadata generated by the Microsoft Defender XDR product spanning the 180 days beginning March 3, 2024 and ending August 30, 2024. To get incident time to resolve, we compute the difference between the time an incident is first opened by an analyst and the time the incident is marked as resolved. We also leverage the following incident metadata:

### Box 1: Incident Metadata

1. Whether an incident is high severity

2. The number of alerts aggregated to define the incident

3. Microsoft Security products that triggered alerts. They can be:

- |                                     |                                 |                          |
|-------------------------------------|---------------------------------|--------------------------|
| ➤ Microsoft Defender for Endpoint   | ➤ Microsoft Defender for Office | ➤ App Governance         |
| ➤ Microsoft Defender for Identity   | ➤ Data Loss and Protection      | ➤ Microsoft Entra ID     |
| ➤ Microsoft Defender for Cloud Apps | ➤ Microsoft Sentinel            | ➤ Microsoft Defender XDR |

Organizations that have consumed Copilot resources are considered treated. To determine treatment times, we use the date the organization first consumed Copilot resources. For each organization we also include the country and industry as organization-level metadata. Finally, we use the number of licenses of the above Microsoft Security products a customer has to conduct our propensity score matching (described below).

## 2.1 Method

Our approach proceeds in two steps. First, after identifying our treatment group as those that consume Copilot resources, we use propensity score matching to construct a control group. To do this, we conduct a binary logistic regression of Copilot adoption on organization segment, industry, country, strategic category, and available seats per Microsoft security product. We then match each organization that has adopted Copilot with an organization that has the closest estimated probability of adoption.

In the second step, we estimate a difference-in-differences model with the log of incident resolution times as the independent variable. We only consider incidents that are resolved within 12 hours after opening. We limit the time to resolve because longer incidents can be driven by a variety of factors that are beyond what we expect our treatment variable, Copilot, to affect, e.g., shift changes and new priorities. We also limit our treatment group to organizations that have conducted at least 200 Copilot executions and have at least 15 observations in the periods before and after treatment. Nevertheless, we show our results are robust to these key modeling decisions. After filtering, our analysis considers 89 Copilot adopters in the treatment group and 88 non-adopters (two adopters matched to the same non-adopter). There was a total of 95,522 incidents with 52,698 of those observations for the treatment group and 42,824 of those observations for the control group.

The “before” period includes incidents that were closed before the organization first uses Copilot and opened at least 12 hours before their first Copilot use. The “after” period includes incidents that started between the time of the first Copilot use and 12 hours before the end of the data. The before and after periods for the control group are determined by the before and after periods of their treatment group counterparts. We exclude incidents that start between 12 hours pre-adoption and adoption as well as incidents 12 hours before the end of data to control for effects due to censoring as well as incidents that overlap the treatment periods.

The difference-in-differences regression specification is standard with two exceptions. First, we include weekly fixed effects, organization industry and country, and the incident metadata given in [Box 1](#). Second, we disaggregate the after period by month to discern the relationship between Copilot adoption and MTTR one, two, and three or more months after adoption.

Following [\[2, 6\]](#), we model the logarithm of time to resolve because a multiplicative common trends assumption makes more sense than an additive one. That is, common trends should reflect the relative change in difficulty to resolve an incident due to the evolving threat landscape. For example, take two organizations, the first with an MTTR of 10 and the second with an MTTR of 100, both in the before period. Then, if the first has an MTTR of 12 in the after period, multiplicative common trends says the second would have an MTTR of 120 in the after period, not 102, as additive common trends would imply. Time to resolve also roughly follows a lognormal distribution, so the log specification offers a much better model fit the alternative specification in levels.

Formally, we estimate the following model:

$$\log(TTR_{it}) = \alpha + \eta_1 I[Treatment_i] + \sum_{\tau=1}^3 [\gamma_{\tau} I[\tau = t] + \beta_{\tau} I[\tau = t] \times I [Treatment_i] + \text{controls}] \quad (1)$$

**Where:**

- >  $i$  indexes incidents
- >  $t$  indexes months after adoption
- >  $I[Treatment_i]$  indicates if the incident originated from an organization that adopted Copilot

The  $\beta_t$  parameters capture the difference-in-differences between the MTTR of the treatment and control group  $t$  months after adoption. When reporting results, we present  $1 - e^{\beta t}$  which approximates the percentage reduction of MTTR associated with treatment. See [appendix](#) for details of this approximation. This model is what [16] refers to as the “dynamic specification” of a two-way fixed effects model but removes the organization level fixed effects. [Section 3.1](#) discusses extending the model to include such fixed effects.

### 3. Results

Our main result is that Copilot adopters experienced a statistically significant 30.13% reduction in mean time to resolution three months post adoption relative to the control group. The results are summarized in table 2 below.

| Quantity          | Estimate | S.E. (clustered) | p-value |
|-------------------|----------|------------------|---------|
| $1 - e^{\beta_1}$ | -.0301   | .1295            | .8191   |
| $1 - e^{\beta_2}$ | .1293    | .1427            | .3956   |
| $1 - e^{\beta_3}$ | .3013    | .1806            | .0487   |

Table 2: Reduction in MTTR for treatment group. Positive numbers represent a reduction.

Although we do not observe statistically significant differences in the first and second months post adoption, the estimated coefficients trend toward increasing gains over the three-month period. The delay in effect suggests that customers need time to learn about how to best integrate Copilot in their workflows. However, this analysis does not rule out other reasons for the delay such as the time it takes for analysts to adopt Copilot across an organization. The estimates for all three months and their 95% (blue) and 90% (yellow) confidence intervals are given in figure 2.

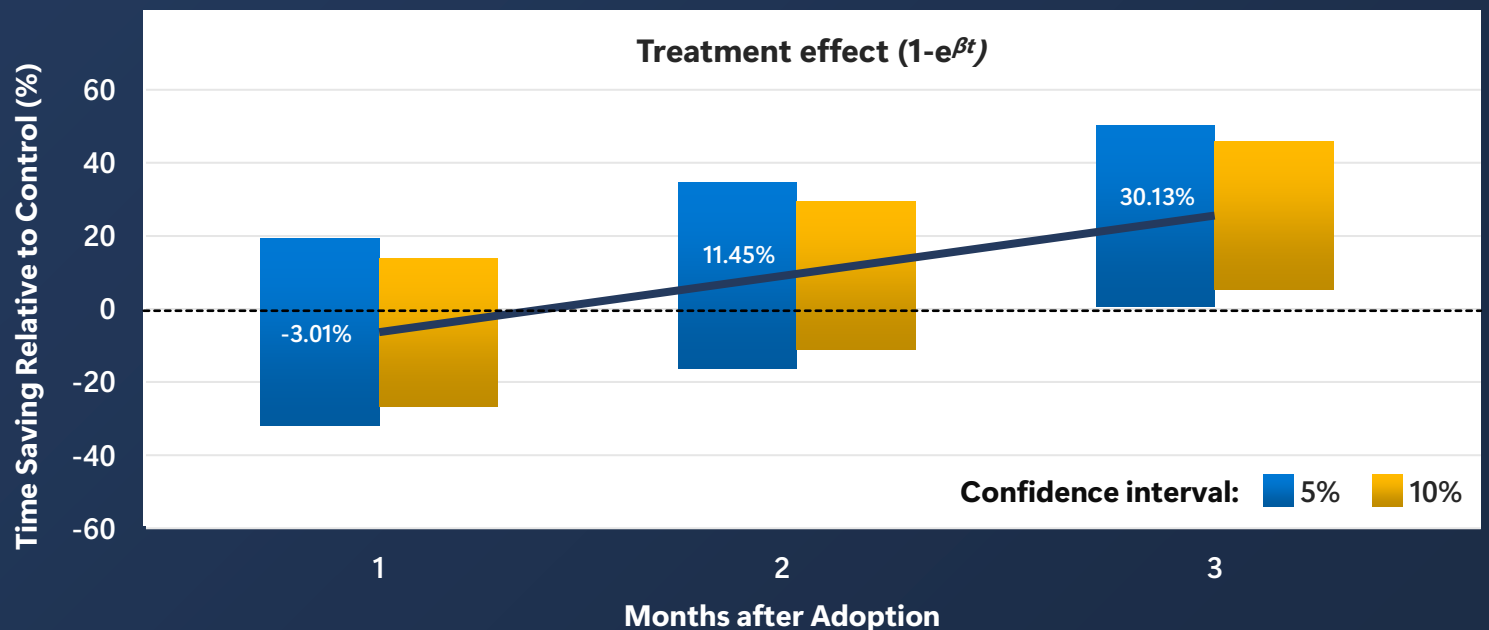


Figure 2: Parameter estimates for treatment effect one, two, and three months after treatment.



## 3.1 Robustness

In general, our results are robust at the 10% significance level for many alternative specifications. As Table 3 shows, the results are largely robust at the 10% significance level with estimates ranging from 22.60% to 31.84%. The only specification that loses statistical significance is when we remove the minimum incident time. However, even there, we easily regain statistical significance by restricting the dataset to organizations that have 20 (compared to 15) observations before and after treatment (effectively removing customers that provide little signal due to lack of observations), with an estimated effect of 30.20% and a p-value of .0625. Furthermore, if we limit the study to large customers (defined by those that have closed at least 100 incidents in the before and after period), we get a similar estimate of the treatment effect (30.78%) but unsurprisingly lose robust statistical significance ( $p=.265$ ) resulting from the reduction in degrees of freedom. Controlling for organization-specific heterogeneity is a promising direction of future analysis as increased Copilot usage allows for greater statistical power.

| Specification  | Weekly Fixed Effects  |         | Monthly Fixed Effects |         |
|--|-----------------------|---------|-----------------------|---------|
|  | Relative Time Savings | p-value | Relative Time Savings | p-value |
| Additionally control for Alert Categories in Incidents | 28.57%                | .061*   | 29.28%                | .0558*  |
| Remove incident minimum time                           | 22.60%                | .164    | 23.71%                | .1427   |
| Increase Usage Threshold To 400 Executions             | 30.23%                | .055*   | 31.93%                | .051*   |
| Reduce Copilot Usage Threshold to 100 Executions       | 30.44%                | .046**  | 31.15%                | .040**  |
| Increase Incident Maximum time to two days             | 31.84%                | .064*   | 33.11%                | .054*   |

Table 3: Estimate of time savings associated with the treatment group three months after adoption ( $1 - \beta_3$ ) for alternative modeling specifications. \*\* indicates significance at the 5% level where \* indicates significance at the 10% level.

## 4. Discussion

Our main result is a 30.13% reduction in MTTR associated with Copilot adoption. This 30.13% is similar to other studies that causally estimate productivity enhancements stemming from GAI adoption. Our statistical model is robust to alternative modeling decisions.

Although we use propensity score matching to control for selection into treatment based on observables, it is impossible to rule out the possibility that factors other than Copilot are generating the increase in productivity. For example, an organization may adopt Copilot because the SOC received a budget expansion. So, even though Copilot may contribute to productivity gains, the co-occurrence of unobserved factors like an increased number of analysts, other new software licenses, and additional trainings make it impossible to isolate the impact of Copilot on productivity using observed telemetry alone.

Moreover, it is even likely that those that are willing to pay for Copilot are the ones that would most benefit from Copilot. In other words, when organizations perform a cost/benefit analysis on the decision to adopt Copilot, only those with the highest benefit would adopt Copilot, since all customers face the same sticker price. This means that, even assuming our estimates are causal, our estimates would overstate the average productivity impact of Copilot for non-adopters.

Considering these limitations, our results provide evidence for the causal impact of Copilot on productivity as measured by MTTR.

However, short of performing a randomized control trial on live customer operations or discovering a natural experiment that mimics randomized selection, the associations uncovered in this study are among the most promising signals supporting the positive effect of GAI tools (like Copilot) on the productivity of security operations. Furthermore, the results are statistically significant and robust, so it is clear that those that adopted Copilot observed a productivity gain.

Future research can help control for some of the unobserved confounders. Specifically, moving from a basic difference-in-differences specification to a two-way fixed effects model can help address selection issues by leveraging heterogeneous adoption times and organization fixed effects. The current volume of available data does not provide sufficient power for a two-way fixed effects model. Despite this, the fact that our point estimate when controlling organizational fixed effects shows a similar magnitude of effect as our main result (though not statistically significant) is an encouraging signal for future work.

## References

- [1] Vectra AI. 2023 state of threat detection, 2023.
- [2] Joshua D. Angrist and Jörn-Steffen Pischke. Mostly harmless econometrics. *Mostly Harmless Econometrics*, 12 2009.
- [3] James Bono and Alec Xu. Randomized controlled trials for security copilot for econometrics' administrators. *Forthcoming Unpublished Working Paper*, 2024.
- [4] Erik Brynjolfsson, Danielle Li, and Lindsey R. Raymond. Generative ai at work: Measuring the clinical and economic outcomes associated with delivery systems. *NBER Working Papers*, 31161:1–34, 2023.
- [5] Jonathan H. Choi. Ai assistance in legal analysis: An empirical study. *SSRN*, September 2023.
- [6] Emanuele Ciani and Paul Fisher. Dif-in-dif estimators of multiplicative treatment effects. *Journal of Econometric Methods*, 8, 1 2020.
- [7] Rafael Costa and Shuyuan Mary Yu. Towards integrating human subject matter experts and autonomous systems. *In Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pages 2110–2112, 2018.
- [8] Z. Cui, M. Demirer, S. Jaffe, L. Musolff, S. Peng, and T. Salz. The effects of generative ai on high skilled work: Evidence from three field experiments with software developers. *SSRN*, March 2024.
- [9] Benjamin G. Edelman, James Bono, Sida Peng, Roberto Rodriguez, and Sandra Ho. Randomized controlled trials for microsoft copilot for security. *SSRN*, March 2024.
- [10] Dona Ghosh, Rajarshi Ghosh, Sahana Roy Chowdhury, and Boudhayan Ganguly. Ai-exposure and labour market: a systematic literature review on estimations, validations, and perceptions. *Management Review Quarterly*, 70(1):1644–1658, 2024.
- [11] Brendon McConnell. What's logs got to do with it: On the perils of log dependent variables and difference-in-differences. *arXiv preprint arXiv: 2308.00167*, 2023.
- [12] Microsoft. Microsoft copilot for security frequently asked questions, 2024.
- [13] Philip L. Morgan, Emily I.M. Collins, Tasos Spiliotopoulos, David J. Greeno, and Dylan M. Jones. Reducing risk to security and privacy in the selection of trigger-action rules: Implicit vs. explicit priming for domestic smart devices. *International Journal of Human-Computer Studies*, 168:102902, 2022.
- [14] Shakked Noy and Whitney Zhang. Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, March 2024.
- [15] Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. The impact of AI on developer productivity: Evidence from Github Copilot. *arXiv preprint arXiv: 2302.06590*, February 2023.
- [16] Jonathan Roth, Pedro HC Sant'Anna, Alyssa Bilinski, and John Poe. What's trending in difference-in-differences? a synthesis of the recent econometrics literature. *Journal of Econometrics*, 235(2):2218–2244, 2023.
- [17] I. H. Sarker. Machine learning for intelligent data analysis and automation in cybersecurity: Current and future prospects. *Ann. Data. Sci.*, 10:1473–1498, 2022.
- [18] I.H. Sarker, Furhad M. Hasan, and Ra Nowrozy. Ai-driven cybersecurity: an overview, security intelligence modeling and research directions. *SN Computer Science*, 2(3): 1–18, 2021.
- [19] Robert Seamans and Manav Raj. Ai, labor, productivity and the need for firm-level data. *NBER Working Papers*, 24239:1–34, 2018.
- [20] Manfred Vielberth, Fabian Böhm, Ines Fichtinger, and Günther Pernul. Security operations center: A systematic study and open challenges. *Ieee Access*, 8:227756–227779, 2020.
- [21] Doron Yeverechyahu, Raveesh Mayya, and Gal Oestreicher-Singer. The impact of large language models on open-source innovation: Evidence from Github Copilot. *arXiv preprint arXiv: 2409.08379*, 2024.



## Appendix

Consider a standard two period, two group difference-in-differences regression where  $y_{td}$  is the response variable indexed by time period ( $t$ ) and whether an individual was treated ( $d$ ). Let  $\beta$  be the parameter on the  $Treatment \times Time$  interaction variable. As [11] notes, in a log-linear difference in difference regression

$$e^\beta = \frac{y_{11}}{\frac{y_{01}}{\frac{y_{10}}{y_{00}}}} \quad (2)$$

However, when the response of the treatment and control group are of similar magnitudes in the before period so that  $y_{00} \approx y_{01}$ , we get

$$1 - e^\beta \approx \frac{y_{11} - y_{10}}{y_{01}} \quad (3)$$

which represents the percent reduction in the response associated with treatment. Furthermore, if the control group does not change over time ( $y_{10} \approx y_{00}$ ), then

$$1 - e^\beta \approx \frac{y_{11} - y_{01}}{y_{01}} \quad (4)$$

which again represents the percent reduction associated with treatment.

In our data, the treatment and control group have a similar MTTR before treatment (58 minutes for treated, 53 minutes for control) and the control group's MTTR changes minimally in the third month in the post treatment period (52 minutes). Hence, we use  $1 - e^\beta$  as an approximation for percent reduction in MTTR due to treatment.