Everybody's Got ML, Tell Me What Else You Have: Practitioners' Perception of ML-Based Security Tools and Explanations

Jaron Mink*, Hadjer Benkraouda*, Limin Yang*,
Arridhana Ciptadi†, Ali Ahmadzadeh‡, Daniel Votipka||, Gang Wang*
*University of Illinois at Urbana-Champaign †Truera ‡Blue Hexagon ||Tufts University
{jaronmm2, hadjerb2, liminy2}@illinois.edu, arridhana@gmail.com, ali@bluehexagon.ai, dvotipka@cs.tufts.edu, gangw@illinois.edu

***Abstract*—Significant efforts have been investigated to develop machine learning (ML) based tools to support security operations. However, they still face key challenges in practice. A generally perceived weakness of machine learning is the lack of explanation, which motivates researchers to develop machine learning explanation techniques. However, it is not yet well understood how security practitioners perceive the benefits and pain points of machine learning and corresponding explanation methods in the context of security operations. To fill this gap and understand "what is needed", we conducted semi-structured interviews with 18 security practitioners with diverse roles, duties, and expertise. We find practitioners generally believe that ML tools should be used in conjunction with (instead of replacing) traditional rule-based methods. While ML's output is perceived as difficult to reason, surprisingly, rule-based methods are not strictly easier to interpret. We also find that only few practitioners considered security (robustness to adversarial attacks) as a key factor for the choice of tools. Regarding ML explanations, while recognizing their values in model verification and understanding security events, practitioners also identify gaps between existing explanation methods and the needs of their downstream tasks. We collect and synthesize the suggestions from practitioners regarding explanation scheme designs, and discuss how future work can help to address these needs.**

1. Introduction

In the past decades, researchers and industry practitioners have taken significant steps toward designing and developing *machine learning* based tools to support security operations. Example applications include network [1]–[4] and host [5], [6] based intrusion detection, malware classification [7], [8], provenance-based attack forensics [9], [10], and alert correlation and prioritization [11]–[13].

Despite the perceived advantages in accuracy and scalability, machine learning based tools face challenges when deployed. For example, the *explainability* problem [14]–[17] is well recognized. That is, most machine learning models (especially deep learning models) work as a "black box", the outputs of which cannot be easily interpreted by humans. A recent SANS survey found that while some security operations centers (SOCs) have deployed machine learning-based tools, they are not rated among the most effective [18].

More recently, the machine learning and the security communities have investigated *machine learning explanations* to improve machine learning based tools' usability [14]–[17], [19]–[21]. For example, an explanation method can highlight key features in a binary (e.g., specific bytecode gadgets, API calls) to explain its classification as malware. However, most existing efforts focus on designing technical solutions to generate accurate and robust explanations, without proactively engaging stakeholders who use, manage, or implement these tools in the field.

In this paper, we aim to fill this gap by exploring security practitioners' perceptions of machine learning based security tools and the corresponding explanation methods. The goal is to provide a deeper understanding of "what is needed" and facilitate further research into usable machine learning and explanation methods for security operations. We seek to answer three main research questions:

RQ1 Where and how is machine learning used in security operations centers (SOC)?

RQ2 What are the perceived benefits and challenges in using machine learning in practical security operations?

RQ3 How are existing machine learning explanation techniques perceived in practical security operations?

To answer these research questions, we conducted semi-structured interviews with 18 security practitioners with diverse roles, duties, and expertise, including front-line analysts who first respond to active security alerts, SOC and incident response team leads, academics who contract with industry, and upper-level management in organizations that produce production tools and services. During the interview, we asked participants about their experience with classification tools and use of machine learning, their perceptions of verifying and taking actions based on the tools' output, and the value and problems with explanation methods.

Key Findings. Our study has the following findings.

First, while *usability* is perceived by the research community as the primary deterrent to ML adoption in security [14]–[17], [22], [23], we find the majority of participants still regarded ML tool *effectiveness* (i.e., the ability to produce correct classifications) as a key factor preventing tool adoption. While practitioners were clear that ML brings sig-

nificant advantages as it can capture complex, subtle patterns and reduce false negatives, false positives are perceived as an overwhelmingly problematic downside in practice. As a result, practitioners seek ML tools with low false positives that can be deployed in parallel with rule-based tools to maximize effectiveness.

Second, for *usability*, not surprisingly, it's perceived difficult to reason about ML's output, creating hurdles for remediation actions. However, surprisingly, rule-based systems are not strictly seen as easier to interpret, especially when rules are complex or the analysts who need to understand the rules (or their errors) are not the rules' original creator.

Third, interestingly, few participants reported robustness against adversarial attacks (i.e., *security*) as a key factor for tool adoption. Further, evasion attacks on rule-based tools were perceived as easier (versus ML tools), while poisoning attacks were less concerning for rule-based tools.

Fourth, ML tools' effectiveness depends on "less glamorous" tasks related to *efficiency* and *adaptability* (e.g., establishing data collection infrastructure, obtaining high-quality data, labeling, model customization). Practitioners find it hard to determine how much data to collect, how long to keep historical data, and how to assess data usefulness.

Fifth, while existing explanation methods (e.g., highlighting features) are perceived as helpful to understand the ML model and verify its correctness (goal 1), participants also need explanation methods that provide *context* to understand the security events and inform further actions (goal 2). With abundant recent works devoted for goal-1 [15]–[17], [19]–[21], [24]–[27], the result suggests that more work is needed for goal-2 to by proactively engaging target users (analysts) into the design process.

Sixth, by exploring participants' perceptions of explanations, we identified additional needs not covered by existing explanations including providing context and actionable information by connecting current events of interest to previous events, providing visualization tools or natural language interfaces that support interactive (spatial/temporal) queries, methods that explain attacker behavior changes over time, and explanations that are privacy-preserving. We also identified some hesitancy toward explanation use, which must be considered in future development, as participants were concerned current explanations may cause information overload or mislead analysts.

From these findings, we provide recommendations for systematically interfacing ML and rule-based systems, addressing issues in ML-security tools' supporting tasks, and developing use-driven explanation tools.

2. Background and Related Work

Machine Learning Tools for Security Operation. Security operations broadly include detecting, analyzing, responding to security incidents, and improving the security posture of an organization. These tasks are often carried out by security professionals from a security operations center (SOC) [28] with the help of different tools. Many of such tools are machine learning based [7] such as network intrusion detection systems (NIDS) [1], [2], [4], [29], host intrusion detection systems (HIDS) [5], [6], provenance-based threat hunting tools [9], [10], attack prediction models [30]–[32], and methods for aggregating threat intelligence [33]. SOCs may use Security Information and Event Management (SIEM) systems to integrate the alerts generated from various detection methods. Due to the high volume of (false) alerts, researchers have proposed methods for alert filtering [34], [35], verification [36], correlation [11], [12], [37] and prioritization [13], [38].

Human Factors in Security Operation. In addition to tool development, researchers have studied human factors in security operations. A main direction is focused on the workflow of SOCs to explore the general challenges faced by analysts [39]. Researchers have interviewed SOC analysts to understand their perceptions of security misconfigurations [40], strategies for malware analysis [41], the "burnout" issues of SOC personnel [42], how people and tools collaborate [43] and resolve contradictions [44], and the problem of excessive security alerts (and false alerts) [28], [39]. Compared to these studies, our work has two main differences. First, these studies discussed SOC tools in a general sense whereas our study explicitly focuses on machine learning-based tools and explores their perceived differences from other tools (e.g., rule-based methods). Second, we focus on the explainability (usability) issues of machine learning tools, which are often not (or only briefly) discussed in prior works.

Focusing on usability, a recent study [45] surveyed six US Naval SOC analysts on two specific ML-based tools for network security analysis. The study revealed usability issues of these two tools (e.g., a lack of documentation, inconsistent user interface design). We expand on this line of inquiry by investigating issues with ML-based tools more broadly. That is, we do not focus on a specific tool, and we interview larger population of analysts drawn from a variety of organizations and operational settings.

Machine Learning Explanation. Related to the usability aspect discussed above, recently, researchers have investigated various methods to explain machine learning model behaviors and outcomes [46]. ML explanations can either be *intrinsic* or *post-hoc*. Intrinsic explanation means the ML model is self-explanatory in its decision-making, e.g., trees [47], linear regression [48], and rule sets [49]. Post-hoc explanation has a separate model developed for explanation purpose [19]–[21], [50]. Explanation methods can also be categorized into *global explanations* and *local explanations*. Global explanation focuses on explaining to overall model behavior while local explanation focuses on explaining individual decisions on specific inputs.

In the research community, the black-box nature of ML is commonly perceived as the major hindrance towards its adoption in security operations [14]–[17], [22], [23]. As such, security researchers have worked to tailor explanation methods for security applications [3], [15], [16], [25], [26], understand the robustness of the explanation methods [27],

Explanation	Brief Description	Example
Highlighted Features	A set of highlighted features that are most important in determining the object's classification.	File o is predicted as "malicious" due to a network-level feature, i.e., sending requests to known C&C servers.
Prediction Confidence	A numerical score describing the confidence of the classification decision.	File o is predicted as "malicious" with 90% confidence.
Similar Examples	A set of other objects that share similar features with the object of interest, and their classifications.	File o is similar to three known malware files a, b, and c.
Highlighted Object History	Highlighted historical data of the object that is important to the classification decision.	A set of selective historical occurrences of file o that were flagged in previous attacks.

TABLE 1: **Exemplar Explanations** – "*Object*" refers to an instance of interest in the attack such as a file, an IP address, a network flow, or a process. The examples will be adapted based on participants' familiar tasks/scenarios.

[51], and explore evaluation metrics [17]. However, these efforts are still focused on model development, without yet involving target users (e.g., security analysts) and real-world security operations.

More recently, human-computer interaction researchers tried to understand how explanation influences the trust between human and ML models [52], and explore ways to improve the understandability of ML explanations [53], [54]. However, most of the works focus on computer vision or natural language processing (NLP) tasks instead of security operation tasks, neglecting security-related idiosyncrasies that may affect the value or use of ML explanations. Our study fills the gap to provide a deeper understanding of the perception of machine learning tools and explanation techniques of security practitioners in their uniquely high-risk and dynamically evolving environment.

Security of Machine Learning. Another related topic is ML's security/robustness against adversarial attacks [55], [56]. Recent works have surveyed companies or developers to understand their perceptions of adversarial machine learning [57]–[60]. Their results have revealed varied levels of concerns about this threat; however, educational, technical, and organizational barriers have prevented/discouraged wide deployment of countermeasures [57]–[60]. Our study differs in several dimensions: First, prior work investigates general ML applications (e.g., computer vision, NLP), while we are interested in ML-based *security tools*. Second, they interview general data scientists, while we target SOC/security practitioners. Finally, our study does not exclusively focus on adversarial machine learning but contextualizes the security concerns of ML in a larger swath of factors that affect tool usage (e.g., effectiveness, usability).

3. Methodology

To answer our research questions, we conduct 18 semi-structured interviews [61] with stakeholders who use, manage, design, or research related tools for security operations. We primarily focus on tools designed for classification tasks, i.e., classifying attack instances (e.g., malware, malicious traffic, malicious websites/URLs) from benign instances since it is the most common use case and exists within an adversarial environment unique to security [2]. To explore broad perceptions, we did not strictly define ML- or rule-based methods for classification, but instead utilized participants-provided definitions throughout the interview. Participants broadly considered rule-based methods as those that require human expertise to produce a set of explicit patterns or signatures, and ML-based methods as those that can automatically learn patterns/models from data. Our study consists of two parts: a screening survey to select qualified participants, and a one-hour semi-structured interview.

Screening Survey. Users first take a short screening survey where we collect participants' job titles, roles, industry sectors, years of experience in security, and whether they focus on offensive or defensive security, or both (see survey questions in our supplementary materials [62]). To obtain a holistic view from practitioners, we do not limit or control factors such as particular roles, organizations, industry sectors, and demographics.

Eligibility. To ensure knowledgeable participants, we only invite participants who are older than 18, with at least one year of industry experience designing, managing, or using security classification tools for interviews. We did not enforce any requirements on the type of classification tools participants used. Per IRB's recommendation, due to privacy laws (GDPR [63] and China's PIPL [64]), we did not recruit participants from European Economic Area (EEA) or mainland China. GDPR and PIPL have more complex consent requirements and data handling procedures that our study/IRB did not readily support.

Semi-Structured Interview. Each participant was interviewed about their experiences with classification tools over a 60-minute online conference call. All interviews were conducted in English. The interview consists of three primary sections designed to answer each of our three research questions (the detailed questions are provided in supplementary materials [62]). First, to understand how ML is used alongside other tools within SOCs (**RQ1**), we ask participants about their experience in using different tools. Second, to understand how ML is perceived (**RQ2**), we ask participants their perception of ML based tools (in comparison with other tools) regarding their benefits and pain points. Lastly, to understand the perception of ML explanation methods, we ask participants whether and how various explanation techniques may benefit them, and what information an ideal system would provide them. For the last part, we use exemplar explanation techniques (see Table 1) to probe participants to think aloud about the questions on this topic.

We select these examples based on existing literature [14], [46]. "Highlighted features", "prediction confidence", and "highlighted history" are commonly discussed in machine learning literature to provide explanations [24], [46], [50], [65]. "Similar examples" are often used in security contexts (e.g., code/function similarity for vulnerability analysis) to assist threat investigation [66], [67].

To maintain consistency between interviews, the interviewer followed a detailed guide of responsible procedures such as how to begin and end the interview, ask questions in a non-leading fashion, re-obtain consent, allow time for participant questions, and reaffirm compensation procedures (adapted from Rader et. al [68]; see supplementary materials [62]). To ensure question clarity and appropriate terminology use, we co-designed interview questions with three security analysts (with 6, 9, 10 years of professional experience) from the authors' personal contacts[1]. Understandability was further verified via pilot interviews with two other security analysts. As no major changes to the procedures or questions were made after the pilots, the results of these two pilots are included in our results.

Recruitment. Participants were recruited over a nearly one-year period (11/2021–9/2022) by advertising our study over various online channels including social media (e.g., Twitter) and dedicated security analyst groups and forums (e.g., Reddit, LinkedIn groups). We also reached out to our personal contacts in various organizations who then shared the study information with their security teams. Finally, we posted our recruiting message to Upwork, an online freelance marketplace, to recruit security professionals. Our recruitment methods match that of prior works [28], [39]–[41], [45] and our experience echoed theirs: it is very difficult to recruit security analysts (or developers) for research studies. Among these channels, we anecdotal found that known contacts in different companies and Upwork were most effective (Table 2). In total, we had N=18 qualified participants complete the interview.

Similar to other qualitative studies (prior studies with security analysts, which have 5–10 [40], [41], [43], [45] or 10–20 [28], [39] participants), we do not attempt to generalize our findings given our sample size, but instead use our results to highly emerging themes and concepts. To this end, we stopped recruiting once we noticed no new concepts/themes around the perceptions of ML appearing in the interviews (i.e., thematic saturation [69])[2].

Data Analysis Method All interviews were transcribed using a GDPR-compliant transcription service [70]. These transcripts were then analyzed following an inductive thematic coding approach [69]. To establish an initial codebook, two authors collaboratively analyzed 3 of the 18 interviews. The two authors then independently coded the 15 other interviews, calculating inter-rater reliability (IRR) for the codes every 4 interviews. For each code, the IRR was calculated using Cohen's-κ to account for chance agreement during coding [71]. If high-agreement was not reached for any variable (κ<0.8) during these 4 interviews, the coders met to resolve disagreements, changed the codebook as necessary, and applied changes to previously coded interviews. As two of the codes did not reach high-agreement by the end of the interviews, a third coder was brought on to recode the first four interviews; high-agreement was then reached. Ultimately, discovered disagreements were due to multiple overlapping codes leading to difficulties in proper assignments, and overlooked sections of the transcript. In total, 5 rounds of independent coding occurred to reach an IRR of κ >0.8 for reported codes. The final codebook and κ values for each variable are in Appendix (Table 4–5).

Ethics and Data Protection. The study was approved by our IRB. The informed consent was obtained during the screening survey. Our study does not collect personally identifiable information (PII) from the participants or the name of their companies/organizations. Email addresses were collected only during the study process for scheduling the interview and making payments. After that, the email addresses were not stored with the interview data. When we transcribed the interviews, we further anonymized any mentioned names of individuals or organizations. On average, participants spent 67 minutes completing both the survey and the interview (63 minutes of which were the interview) and were compensated $40 for their time ($36/hour).

Limitations. First, as a semi-structured interview, certain follow-up questions may not have been asked in some interview sessions, and the answers may not exhaustively cover all the topics in the same depth. However, all interviews have covered the main questions that are directly related to our research questions. Second, there may exist concerns of the participants not representing the entire population of security professionals who interact with classification tools. For instance, while we recruit participants from a variety of professional and demographic backgrounds (Table 2 and 3), we do not recruit from EEA or mainland China, nor do we fully represent all possible roles, industries, or demographics. To alleviate this concern, we ensure proper interpretation of our qualitative results; that is, we do not attempt to generalize our findings. Instead, we focus on finding diverse sets of views from various stakeholders to present views that exist in the community. These results can be used to inspire hypotheses in large-scale surveys or focused studies around specific professional/demographic factors in future work. Third, biases such as social desirability and confirmation biases may affect some participants' perspectives. We mitigate these by asking questions in a neutral manner and asking the participant to consider and speak to opposing viewpoints.

4. Result: Tool Usage (RQ1)

In this section, we start by describing the participants' information and then discuss their usage of security classification tools. Due to the space limit, additional discussions are presented in Appendix A.

1. These three analysts were not interview participants.

2. We observe concept saturation of participants in terms of their overall ML perceptions. While new roles/industries appeared as we interviewed more users, they did not necessarily offer new perspectives/concepts.

ID	Job Role	Yrs Exp	Sec. Knowledge	Method	Recruit.
P01	Pen. Tester	2	Offense	Rules	Contact
A02	Sec. Analyst	10	Defense + Offense	ML	Contact
A03	Sec. Analyst	2	Defense	Both	Upwork
E04	Sec. Engr.	3	Defense + Offense	Rules	Upwork
E05	Sec. Engr.	15	Defense + Offense	Both	Contact
E06	Sec. Engr.	1	Defense + Offense	Rules	Contact
D07	Developer	7	Defense	Both	Upwork
D08	Developer	2	Defense + Offense	ML	Twitter
R09	Researcher	5	Defense + Offense	ML	Upwork
R10	Researcher	5	Defense + Offense	Both	Upwork
R11	Researcher	2	Defense	Both	Contact
M12	Management	20	Defense	Both	Upwork
M13	Management	5	Defense	Both	Upwork
M14	Management	5	Defense	ML	Upwork
M15	Management	12	Defense + Offense	Both	Upwork
M16	Management	15	Defense + Offense	Both	Upwork
M17	Management	25	Defense + Offense	Both	Contact
M18	Management	32	Defense + Offense	Rules	Contact

TABLE 2: **Participants Experience** – Our sample contains a diverse set of N=18 participants holding different roles and experiences in the security industry. The "Method" column shows the tools the participant is familiar with.

Participant Background and Demographics. As shown in Table 2, we use the first letter of the participant ID to indicate their job role. For example, participant eight was a developer, so we use the ID "D08" when referencing this participant. Note that this ID does not indicate the order in time of their interviews but is grouped based roles.

Professionally, our participants consisted of a variety of roles including first-level responders to active security alerts, team leads for SOCs and incident response, academics who contract with industry, and upper management in organizations who produce tools and services for others. For the ease of referencing, we classify them into six distinct roles (Table 2) including pen tester (n=1), security analyst (n=2), security engineer (n=3), developer (n=2), researcher (n=3), and management (n=7). Demographically, participants come from a variety of ethnic backgrounds and locations (Table 3). While our sample has a disproportionate number of male participants, this is unfortunately consistent with the disproportionate number of males in the field of computer security [72]. Several prior publications have been unable to recruit many female participants [73]–[75]. Future work could investigate whether demographics have any effect on the perceptions presented in this line of work.

ML is commonly used for security classification, but needs to be in conjunction with other techniques. Most participants (n=14) noted they have used or designed machine learning (ML) tools to classify security-related events. Meanwhile, participants highlighted that other traditional methods were also used instead of, or to complement, ML. Most (n=14) reported using rule (or signature)-based methods and several (n=5) used manual analysis. Many participants (n=10) used ML and rule-based methods in conjunction with one another. In these cases, participants primarily reported that each tool has its own advantages/disadvantages that are supplemented by the other tool: "*In industry, rule-based system can cover over 90% detection and for the rest, it is the job of machine learning models. They can capture the patterns we human beings cannot see...*" (R11). However, several other participants (n=8) either solely interact with rule-based methods (n=4) or ML-based methods (n=4). While practitioners generally embrace the idea of ML-based tools, some noted ML/AI is also "overhyped" and more work is needed beyond just applying them: "*Everybody's got AI. Tell me what else you have.*" (M17)

5. Result: Perception of Tools (RQ2)

We asked participants about their perceptions of security classification tools. The discussion is centered on ML and rule-based methods. We found participants considered five properties when evaluating classification tools: *effectiveness*, *usability*, *efficiency*, *adaptability*, and *security*. An overview of participants' perceptions of ML and rule-based tools is illustrated in Figure 1. In the following, we will use this illustration to first discuss and compare the perceptions under each property (§5.1–§5.5), and then we discuss the interactions between properties (§5.6).

5.1. Effectiveness

Tool effectiveness refers to the ability to correctly classify security events under normal (non-adversarial) scenarios. Most participants reported effectiveness as a factor in their tool choice (n=10), e.g., with excessive false positives or false negatives causing participants or their customers to cease tool use. When considering ML and rule-based tools' effectiveness, participants perceived each as having advantages (n_{ML}=15;n_{rules}=9) and disadvantages (n_{ML}=17;n_{rules}=9) depending on the context of use.

ML is effective in identifying complex and subtle patterns. ML was generally found effective at learning complex patterns that are difficult for humans to discover or craft rules for. D08 explained, "*The only way you can detect that kind of attacks is you have a system execution graph, and you can find out the pattern. That's where rule-based things come up short. You cannot write rule for everything. But pattern, yes, you can define a pattern for everything. That's why I think, inherently, pattern-based solution, the statistical solutions will outperform rule-based ones.*" Participants also noted that this ability to capture complex relationships allows for subtle differences to be detected. For example, R10 described the value of this subtle classification for detecting unauthorized users: "*You can try to characterize the behavior of the user in background. How often are they clicking? What are the touch dynamics? Which pages are visited often? Etc. This is very characteristic to the actual user. Even if the attacker steals your password, the machine learning-based technique can differentiate between the [attacker and] user.*"

Rule-based methods are better when the behavior is well-defined, which is common. In contrast, rule-based methods were perceived effective at classification for static or easily described situations. Despite this limitation, participants believed many security tasks meet these restrictions. Participants noted that string matching of known malware

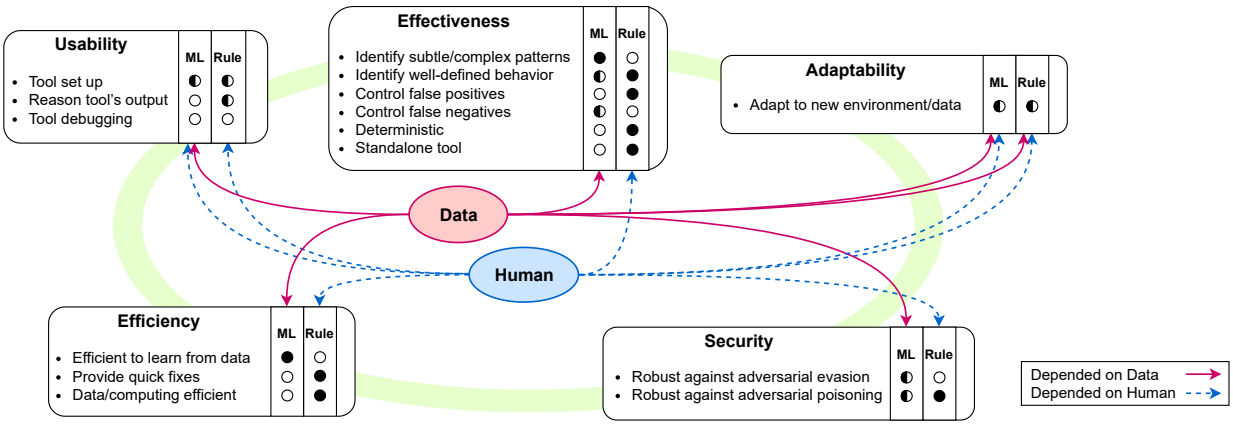

Figure 1: **Summary of Perceptions** — We present an overview of how participants perceive ML- and rule-based classification tools regarding five major properties. ●="general strength", ◐="mixed", ○="general weakness". The five properties are not independent but connected by two latent factors: "quality data" and "human expertise". A discussion of their interactions is presented in §5.6.

signatures, searching for sensitive data with regex-patterns, or finding maliciously-tampered applications with signatures to be effective. R09 provided an example: "*[If] we want to actually find out whether an application's header has being tamped, for instance, you don't need a machine learning.*" In these cases, participants often preferred rule-based methods, noting that even if they were equally effective, rule-based methods held other advantages due to their simplicity, such as efficiency: "*If you want to find evidence from a machine, you don't need machine learning... a rule-based technique would be quicker, would be more efficient.*" (R09)

ML is perceived to have more false positives, a major downside When considering errors, participants noted that ML held a tendency to produce false positives. In our context, false positives refer to benign instances incorrectly classified as malicious by the tools. Participants noted that false positives were intolerable by end-users of the system and thus essential to prevent: "*If [customers] find out your FP rates are annoyance or frustrating, [they'll] immediately throw it out...they'll bring something else on.*" (M17) For ML-based methods, as they are designed to generalize, there is tension between reducing false positives and capturing more attacks. As R11 noted: "*for machine learning system, it is harder to control the FP rate while keeping as [much] coverage as possible...if we want to keep that [false positive] rates really low sometimes we will have to lose many true positives.*" In comparison, rule-based signatures can be constructed highly specific to the intended targets, which helps to control false positives. Further, as recent work suggested, even if a rule triggers events beyond its intended use, these are benign false positives as the rule is operating as expected and it is relatively simple to handle these issues over time [28].

While presenting prediction confidence can be a useful way to offset some of this concern (which we discuss further in §6), several participants noted that this is not true in all contexts. Specifically, when the tool is used to inform high-stakes decision making, as R10 explained, "*Security people get nervous when they see this is a thing with 72% confidence. Saying that in security situation makes me nervous because you want to be right or you want to be wrong, you don't want to be 72% confident.*"

ML may have less false negatives, but this is not a primary factor for tool adoption. A key advantage of ML, noted by the participants, is the ability to generalize (to unseen data or events). Better generalizability can reduce false negatives and capture more attacks. However, several participants believed there is a limit to ML systems' ability to catch novel attacks, making its false negatives similar to rule-based systems in practice. M17 described this practical limit on ML: "*I guess that would be the biggest issue, [if] it completely misses something. When something new comes out, you need a new model, you need new features, you need retraining, you need samples, and this is the problem.*"

Interestingly, while participants stated they would discard a tool for high false positives, this was not true for false negatives. Participants explained that false negatives exist in all tools and could be discounted for two reasons. First, some participants described mitigating false negatives by using multiple tools to cover missed attacks. R11 described this strategy, saying, "*Most machine learning models are one big important part of the whole... but we cannot only rely on it alone... the best way to use them is to combine them together making use of the strength of both [ML and rule-based] methods.*" Other participants explained that the differences in false negative rate would not matter if an adversary got into the system. "*If we compare how much they fail, signature-based systems fail as much as machine learning systems, and vice versa. Because it's the same case once you get in. You have to just get in once.*" (D08)

ML is not effective enough to use alone, and cannot replace rule-based methods. A sentiment among several participants (n=6) was that ML-based systems should not be used alone. Participants reported that they were apprehensive about using ML-based classifiers alone especially in automated setups and emphasized the need for a human-in-the-loop: "*We'd still be a little cautious about an automated response that the machine learning thing would do. I've*

read enough science fiction stuff that I don't like automated things. I want a human to be the final button pusher, so to speak." (M18) In addition, participants felt that rule-based systems will still play an important role despite ML's continuing progress,: "*I think machine learning-based systems, they will not replace the traditional deterministic solutions, but they will compliment them.*" (R10)

5.2. Usability

Usability refers to the ability to easily set up the tool, understand its outputs, and use it to support downstream tasks. Usability was mentioned by many participants (n=10) as a factor for tool selection. Participants had varying perceptions of tool usability. Overall, while some indicated that ML tools were usable (n=4), the majority perceived them to have poor usability (n=13). Conversely, while several participants believed rule-based tools were usable (n=7), a few still mentioned their usability problems (n=2). Usability was discussed in two ways, i.e., tool setup/customization and responding to tool outputs, with each method offering strengths and weaknesses.

ML-based systems require both domain knowledge in security and expertise in ML to customize. For initial setup, participants considered ML and rule-based tools easy to set up. As long as the data could be provided in a way to work with standard toolkits, the procedure to set up a ML system was straightforward. R10 noted, "*[The outside of ML is] very, very simple. So, you have a box, you have the inputs here and get the out[put]. The inside is very complex, but outside, very simple.*" Similarly, rule-based systems often come with predefined rule sets: "*Most of the SIEM solutions comes with the predefined use cases that you can use and they're pretty good.*" (D07)

When *customizing* an ML or rule-based tool for a particular environment, deep expertise is needed. Domain expertise is needed to produce specific rules. However, ML introduces an additional burden. Expertise in security is necessary to tune the model, but deep ML expertise is also required—a mix that is in short supply [76], [77]. D08 stated, "*You need good people, who both understand the security perspective and machine learning perspective. And that workforce is not exactly very big.*"

Reasoning ML output is difficult, leading to hurdles in taking security remediation actions. When considering how to respond to tool output, participants noted ML is poor at "explaining" why it made an instance-specific decision. With ML tools, participants noted challenges in responding to an alert, contextualizing it within their system, and stopping a threat. M17 explained, "*ML for most people is like a check engine light, something is wrong. . . Now I got to spend time figuring it out. What [we are] looking for is actionable information, which requires the context: What problem are you in? What actions do you need to take?*"

Rule-based output has inherent usability benefits, but can also be difficult in practice Understanding rule-base tools' output was often regarded as simpler as there is an inherent explanation based on the matched rule. E06 explained, "*It is definitely easier in a rule-based system to see a flag, and backtrack to the rule that caused the flag, and to go through and check to see, okay, so this is the one thing that caused the rule to be violated, whereas in an ML scenario, it's a lot harder to see why this was flagged and this wasn't.*" However, as rule sets become more complex and when rules are created by others—both are often true in practice—, interpreting the reason the rule was triggered and any explanation provided becomes challenging. For example, M17 described often scrambling to find the rule author to determine an alert's meaning: "*What they do is they immediately resort to 'Who wrote the signature?', go find him and ask him, what the hell did he write? We usually find out from him, 'Hey, what's going on?'*"

Debugging model errors is challenging for both ML and rule-based systems. Understanding a tool's output is also important for debugging whether the output is actually correct. The ability to reason about tool behavior is important for identifying and fixing errors. Again, the lack of explanation in ML tools makes it difficult to understand its errors: "*The reason [why] statistical-algorithmic solutions are less prevalent than signature-based solutions, in the market, is people can't explain it. Security is a sensitive application, if something goes wrong and if you can't explain it — no, you don't want to use that.*" (D08)

Interestingly, rule-based systems were not perceived as inherently better in the context of debugging, especially when the rules are complex and created by people other than the users. R11 described this difficulty, noting their verification methods for their ML- and rule-based tools rely on similar processes of testing with historical data because of the complexity in both: "*I would not say rule-based is harder or machine learning-based is harder. . . for rule-based systems, if someone wrote a Yara rule before to determine a certain structure of the web page is malicious, if I see the same pattern. . . I need to check whether this Yara rule brings some false positives and I need to check historical data as well to see if there are any false positives there. . . For machine learning models, it has the same problem. Given that the machine learning model results also based on the training dataset, which is our historical data, I also need to do that - basically the same process.*"

5.3. Efficiency

Efficiency refers to the ability to avoid wasting resources (e.g., time, human efforts, data, computation power) to use and maintain the tool. Efficiency was reported by several participants (n=9) to be a factor when choosing the tools they used. Both ML and rule-based methods were perceived to be efficient by several participants (n_{ML}=6;n_{rules}=4) for certain aspects and inefficient by several participants (n_{ML}=9;n_{rules}=3) for others. For example, ML was perceived to be efficient to make use of (or learn from) large volumes of data (with computing resources). Rule-based methods are efficient for quickly resolving problems.

ML can efficiently make use of large amounts of data. With an increase in security events, some participants noted they have moved toward ML models due to their efficiency in handling large amounts of data. Rule-based systems, however, often require manual data analysis to produce heuristics, which does not scale. For example, A02 explained, "*The amount of data that we see has exceeded the capability of humans. So 20 years ago, maybe it is possible to do that manually with the order of hundreds of alerts or megabyte of data. But now it has grown exponentially to the order of terabyte and hundred of thousand alerts per day. So manual processing and based on heuristic is no longer tenable... we found machine learning is a good tool for that.*"

Rule-based systems can be fixed quicker. Participants suggested errors in rule-based systems (both false-positive and -negative) could be patched quicker. They might not provide a long-lasting solution in more complex cases, but participants found this efficiency particularly important for urgent problems. R11 described using short-term rule fixes to prevent novel threats before developing long-term solutions: "*Just that a simple rule may not be able to detect a similar attacks, but as a fast solution, a rule-based system is good enough.*" Conversely, ML tools require significant effort to retrain the model and resolve these errors. E06 described this problem saying, "*When it does come to patching those errors, you can always add another clause to the rule. But if you want to patch an error in machine learning, you're going to need to come up with a lot of data of that specific edge case and then add that to the training pool and retrain in order to have that accounted for.*"

Training ML models requires extensive computing resources, which are not available to all users. Participants noted that one of the factors allowing them to use ML tools is the recent advancement and availability in computational resources: "*You have the humongous processing power in order to process that data in no time.*" (R09) But these computational resources are not readily available to everyone. This is especially true for training large models from scratch. D08 described this limitation saying, "*we don't have the compute to create sufficiently big models easily. For example, Google, they can just dish out a couple of million dollars for training something huge. As a smaller company... [we] can't give you $1 million for just computing something... that's financial issues.*"

5.4. Adaptability

Adaptability refers to the ability to easily adjust the tool to the current environment. This property is important to support specific infrastructure or data and meet business requirements. Additionally, participants explained the threat landscape changes rapidly as adversaries adapt techniques to circumvent mitigation, requiring regular adaptation of tools over time. Several participants indicated that adaptability impacts their choice of tool (n=5). Most participants who mentioned adaptability perceived ML as adaptable (n=6), while some participants also perceived rule-based systems were adaptable (n=3). Only one participant mentioned a lack of adaptability, specifically for ML tools (n_{ML}=1,n_{rules}=0).

ML can be adapted to a new environment with sufficient additional data. ML can be adapted to a particular environment (e.g., an organization's network) by retraining the model with additional data from that environment, a process known as "fine-tuning". As long as data is available, ML-systems can be quite readily adapted at a large-scale. When asked whether their ML tool can detect malicious network actions within different types of network configurations, D08 responded, "*Different kind of networks, can we adopt them? Even at industrial scale? Yes.*" Fine tuning is even more pertinent for anomaly detection systems where a baseline of normal behavior from an environment is required for the tool to function. One participant noted that, in practice, data availability can be a challenge: "*Models, you start looking into, no, I need more samples. There's not enough features. I need 1000 samples to do the retraining. You start having concerns about fixing things.*" (M17)

Rules can be adapted but require expertise and manual efforts. In comparison, rules-based systems need to be adapted to a new environment (or new attacks) manually. This could be a challenge especially when the rules need to be regularly updated. M17 estimated the lifetime of rules saying, "*Even the best people I've hired, their signature would not last for more than 30 days to 90 days.*" While M17 did not expect an ML model would last much longer—"*at least 90 days to maybe 120 [days], to even 6 months*"—, they explained the key difference was that rule-based tools require an "*army of analysts*" to update the rules.

5.5. Security

We discuss security primarily in the context of robustness against adversarial attacks. Surprisingly, this was the least discussed among the five properties. Few participants reported classifier robustness against attacks as important factor in tool choice (n=4), and they were concerned about the lack of robustness in both ML and rule-based classifiers (n_{ML}=3;n_{rules}=5). Few participants considered neither tool type robust against adversarial attacks (n_{ML}=2;n_{rules}=0).

Evasion attacks against rule-based tools were perceived as easier. While adversarial evasion was considered a risk for both ML and rule-based systems, participants believed the level of expertise needed to evade ML-based classifiers was higher. R09 explained "*Even script kiddies can bypass a rule-based web attack detection technique, but if we talk about machine learning techniques, it is smart enough to thwart even advanced attacks, at times zero-day attacks as well.*" To evade ML tools, M14 noted that hackers need to "*understand the assumptions of neural networks and other machine learning models... these are the mathematical assumptions and here's how you can go in and play around so you can get around it.*" While at least one participant indicated they had heard of such an attack in carried out in the wild, this level of expertise was perceived as more rare. This echoes the hypothesis from a recent position paper [56]

that adversarial ML is not the most economically viable evasion option for attackers.

Conversely, rule-based systems were considered often less complex and easier for an attacker to infer the rule scheme. R11 explained, "*I think the biggest negative side is that if attackers can send a lot of traffic and they can easily find out what the rules we are using and can bypass it.*"

Poisoning attacks are less of a concern for rule-based systems. Participants recognized ML tools are vulnerable to training-stage attacks such as poisoning. R09 said, "*If contamination happens right at the data preparation or data training phase, then that's even more dangerous, because you're not in the right fashion. So I guess that is one fundamental aspect as well.*" In contrast, rules were perceived as less vulnerable without the added entry point for poisoning as they rely on analysts manually crafting rules.

5.6. Interaction Between Properties

The five properties discussed above are not independent– instead, as illustrated in Figure 1, we observed underlying, connecting latent factors. Specifically, we observed ML tools were more dependent on data, while rules are more dependent on human expertise.

ML tools are more dependent on quality data. Participants regularly reported that ML tools' performance depended on the data's representativeness and availability of sufficient data to train models. Data quality was primarily seen as affecting tool *effectiveness* by impacting ML tools' ability to identify subtle differences in complex data. However, gathering sufficient data was seen as *inefficient*. As reported by M18: "*It took us a couple of years to get the collection engines you know built up.*" Participants also noted that determining the right balance in this trade-off between effectiveness and efficiency can be challenging. D07 demonstrated the uncertainty of this decision when discussing determining the right amount of initial data to collect to establish a baseline for anomaly detection saying, "*The baseline creation was quite difficult. Even though we set it for 30 days and that was more than what we thought that it should be. Still maybe they could have been improved.*" Further, the challenge of data collection is exacerbated when considering *adaptability* to changes to the environment over time, as organizations must regularly collect and retrain ML models. Perceptions of ML tools' *usability* were also related to this data dependence. Because of the focus on large data sets for defining tool behavior over human expertise, participants perceived ML tools as more challenging to reason about and therefore, less *usable*. Finally, when considering *security*, ML tools were perceived as uniquely susceptible to data poisoning attacks due to their reliance on training data.

As depicted in Figure 1, certain properties of ML are also dependent on human expertise. For example, domain expertise is necessary to fine-tune/customize ML models (adaptability), and interpret ML outputs (usability).

Rule-based tools were dependent on human expertise. Rule-based systems were perceived to be dependent on knowledgeable analysts (i.e., knowledgeable about security and knowledgeable about the target environment). D08 noted, "*Signature based algorithm are basically relying on domain expertise. You're writing, 'if this happens, then do this or that' . . . because you have seen those things happening in the past.*" To produce *effective* rules, analysts must first invest time to learn about the environment and attacker tactics, techniques, and procedures. E04 described this learning process saying, "*I got to know different things. Initially, I had the knowledge what are sensitive ports, which are attacked the most, or which ports are vulnerable. . . There was a learning experience.*" This was viewed negatively with respect to *usability* by our participants due to the high demand for expertise during setup. However, once sufficient expertise is achieved, the rules can be easily understood and *adapted* quickly—at least for short-term fixes—, to address errors and *security* issues related to evasion attacks. While these changes were perceived as relatively simple and *efficient* generally, the reliance on skilled practitioners to make these changes and the rapid shifting of the environment makes rule-based systems inefficient overall.

Rules are still dependent on *data* but require strong human interventions (Figure 1). For example, when updating rules for a new environment (adaptability), analysts still need to collect and analyze some data from the new environment to craft and validate the rules.

6. Result: ML Explanation (RQ3)

After discussing the perceptions of ML and other tools, we asked participants whether recently proposed ideas for ML explanations would alleviate their held concerns, in particular those related to usability (§5.2). For this discussion, we utilized exemplar explanation methods including highlighted features, prediction confidence, similar examples, and highlighted object history as probes to elicit their opinions on the subject (see §3, Table 1).

We observed participants' perceptions were partially informed by experience. Several participants noted they previously encountered some form of explanation method providing additional context when using security classification tools[3], including highlighted features (n=3), prediction confidence (n=7), similar examples (n=3), and highlighted object history (n=4). Nearly all participants' initial perceptions of each explanation method were positive, i.e., most believed showing highlighted features (n=15), similar examples (n=16), prediction confidence (n=14), and highlighted object history (n=16) would be helpful. However, even for these positive participants, their perceptions were dependent on their goals, the availability of addition information, and the ability to overcome some common concerns. As such, there were also participants (n=2) who believed none of these methods could sufficiently address their need.

3. The explanation method can be discussed in the context of both ML- and rule-based systems, depending on participants' familiar tools. For example, ML models usually provide prediction confidence as part of the model. For rule-based systems, there are similar forms of explanation such as "risk scores" (e.g., how much a value goes beyond a threshold).

In this section, we first discuss the tasks for which participants envisioned using explanations (§6.1). Then, we discuss how participants achieved these goals through different types of explanation schemes (§6.2 to §6.4). Finally, we note participants' potential concerns about explanations (§6.5) as well as their suggestions (§6.6).

6.1. Goal of Explanation

Determine model correctness. Explanation methods are designed to increase model transparency, allowing humans to understand and verify model correctness. This was reported by many participants as a goal of explanation (n=10), with confidence and highlighted features thought to be used by several participants ($n_{highlightF}$=7, $n_{confidence}$=6), and highlighted object history and similar examples being used by a few participants ($n_{highlightH}$=2,$n_{similar}$=3).

Developers noted they would use the ability to determine if their model correctly learned the intended features, *security analysts* noted that explanations would help in their ability to determine how to follow-up with a particular classification, or whether to follow up at all, and *model providers* may find use in providing evidence of model decisions and reasoning to customers: "*When we show the evidence, we can make our results more convincing.*" (R11)

Understand security events. Many participants (n=9) noted explanation methods can be used as an investigative tool, supporting additional inferences about the security event. Contrary to using an explanation method to inspect the *internals* of a model, participants described using the information given to further their understanding of the *external* event detected. That is, the ML tool's ability to identify subtle differences would help point out patterns the practitioners themselves might not notice, and provide insight into important characteristics of the event if well explained. M13 noted, "*That would build my own mental heuristic model. Because if the model is telling me that this certain characteristic you need to be on the lookout for, that will shape my actions.*" Participants indicated that highlighting features/object history (n=5) and similar examples (n=5) would be useful when seeking this further understanding, but no participant mentioned this goal when discussing prediction confidence.

Participants also noted several use cases. *Security analysts* noted that they would use such knowledge to inform an investigation or update their perceptions of attacks. *Developers* noted that they would use a better understanding of attacks to build more effective detection systems.

6.2. Highlighting Important Information

Focusing on participant perceptions to specific explanation types, we start by jointly discussing *highlighted features* and *highlighted object history*. We group these two schemes because both schemes involve highlighting important instance information, allowing users to narrow their focus to items that influenced classification decisions. Also, during our interviews, participants often discussed the two schemes in the same terms, and we thus present them together.

In general, participants found it helpful to be able to attribute important features and history that influenced or contributed to a classification decision. These explanations were seen as helpful for both goals from §6.1 as they help practitioners leverage domain expertise by focusing on the most important elements of the data.

Compare expert reasoning with model reasoning. Using highlighted information, participants reported they compare model reasoning with their own mental models informed by domain expertise. This comparison can help establish trust in a model or flag potential issues. D08 noted, "*[A] domain expert and machine learning expert, if we put them together, they can say, 'Okay, your network is learning this, but it should not. Maybe we should focus on this.' So machine learning expert can say, 'Okay, these are the wrong features, what are the right features?'*"

Other participants noted this comparison of their perceptions with model reasoning would help refine their knowledge about the attack. R09 explained, "*... it is actually going to further apprise me about what's happening in the network and what's happening on the Android application.*"

Identifying and analyzing repeated patterns. Participants also noted that the highlighted information could help them identify connections between events over time. A03 described the utility of this higher-level view of the data saying, "*If we continuously highlight a specific portion of a specific kind of patterns, it means that we can analyze that someone is trying repeatedly to do some bad thing[s] to our organization. It will help us to detect it.*"

6.3. Retrieval of Similar Examples

The ability to retrieve similar existing examples to the target instance was found beneficial as they help place the results in a broader context, allowing practitioners to leverage prior experience to interpret ML outputs.

Compare against known/trusted historical data. Participants noted that similar examples from a known historical event can help to evaluate the ML model's accuracy. For instance, showing how a new example related to a well-known *trusted example* helps practitioners verify whether the model is working properly. P01, who was skeptical of ML, noted these explanations could help increase their trust of the model: "*I think if they are similar, correct examples, and it's showing this is the same thing... I could quickly assess information that says like, 'this is how it works', then I'd be more comfortable.*"

Participants also noted that similar examples can help them quickly interpret new security events, by connecting the results to a context already familiar to the practitioners. M17 explained, "*The guy [that] tried to get in once, he was successful, he [may do] it to you twice. Chances are those attacks are going to look similar.*" This may further speed up threat response as practitioners can find related historical data indicating potential attacker follow-up actions

and previous, successful mitigation. However, this requires a good understanding of the prior events and the ability to contextualize the knowledge for the new event.

Compare different decisions for similar examples. Participants found it beneficial to compare a model's decisions on similar examples to identify potential inconsistencies. On one hand, the inconsistent decisions may indicate incorrect functionality of the model. On the other, participants noted that it may indicate previous events being misclassified or previously undetected compromise. E04 gave the example, "*If the packets are the same, I would have to analyze it, why was it bypassed earlier and why it hasn't been bypassed now? Is it related to the refinement of the model?... and now, over a period of 15 days, it has enough data that the [model] is classifying it as malicious. So, I would investigate the earlier packet, as well, and the current packet.*"

Filter by time and environment to show the most relevant examples. Participants pointed out that not all similar examples would be relevant to the practitioner. Instead, they suggested *time-bounding* and/or *environmentally-bounding* provided examples to ensure relevance. For time-bounding, M17 noted that having this option would present examples most relevant to ongoing attacker campaigns and recent security events: "*If you can time bound it, that would definitely help because again, new campaigns, they're probably going to be exhibiting behavior to campaigns within the last 90 to 120 days.*" Environmental bounds would select examples from a practitioner-determined environment, such as a segment of a network, a set of computers, or an organization: "*The most practical approach is nearest neighbors within their context versus nearest neighbors of globally what's out there because that ended up being a larger population and confused people.*" (D07)

6.4. Confidence of Inference

Participants reported that having a metric expressing the model's decision confidence was helpful to understand the correctness of the model and triage further actions. However, it is not noted helpful to understand the security event taking place (as it does not provide contextual information).

Understand model behavior and triaging responses. Prediction confidence was used by participants to gain insights into the model behaviors and prioritize their response. Participants noted the confidence score could help set thresholds for manually evaluating tool outputs. M13 noted, "*That would definitely help because then you could come up with a policy that says, 'anything below 70% certainty, we need to do a manual investigation.*'" M13 further explained that this was particularly important to have when justifying decisions to management, saying "*that's [prediction confidence] something that's pretty easy to communicate to business people and other stakeholders who are usually the ones that are responsible for risk management or owning the risk.*" Participants also indicated prediction confidence could help assess model performance: "*[Prediction confidence] is another dimension to detect or to understand whether my results [are] great or not.*" (R10)

6.5. Concerns of Explanations

While participants found explanations valuable, some noted concerns about introducing the proposed explanations into their workflow. Ultimately, the concern was whether explanation would save time or just create more work.

Unhelpful explanations. Participants noted explanations may be too disconnected from practitioners' workflows. A02 cautioned, "*There's still a disconnect between what the decision such an algorithm output and what the analyst would do.*" This is particularly important because analysts do not generally act without first performing some manual verification to verify results. M12 explained, "*I have yet to find a tool that you don't need to investigate after something [an alert]. I don't know anyone who doesn't investigate before reacting.*" D08 further pointed out that the explanation's presentation is important. Explanations should be crafted to help analysts without overloading them with information.

Wrong explanations can cause harm, consider adding warnings. Another concern is incorrect explanations that would lead analysts down the wrong path, wasting their limited time and causing missed true attacks. E06 notes, "*If [the explanation] is wrong and causes them to say, 'Oh, okay, so this was a false positive', and then something happened as a result...*" A02 suggested that warning the user for potential tool error may help mitigate this: "*The user needs to acknowledge that the machine can be wrong, and the reasoning needs to be made transparent so the user can see all of the inconsistencies in the signal that the machine has to take in when it gives that reasoning.*"

6.6. Participant Suggestions

Participants also made suggestions for current and future explanation methods to meet their practical needs.

Guide actions to take in response to classification. Once a security event is classified as malicious, one participant noted that providing remedial advice via actionable next steps would help in responding to threats. These remedial steps could both help address long-term threats by assessing whether previously used vulnerabilities are present, as well as immediate threats, such as how to contain an ongoing intrusion: "*What are the IPs I need to block?... What ports do I need to block so I can stop it laterally moving from one machine to another?... Most people are looking for immediate actionable information based on their situation or stage of the kill chain.*" (M17) However, this prescriptive guidance should be provided with care. While explanations provide additional context and leave decision-making to practitioners, incorrect suggested actions could move the system in a more harmful state. M17 explained this dilemma saying, "*If it's a generalized explanation, that is kind of okay to give, but you start giving actionable information... that's where it tends to be risky.*"

Similarly, participants suggested including relevant context about the identified vulnerability or threat to help guide remedial actions. This could include malware capabilities, potential motivations of associated actors, and any potential or expected damage the attack might cause. M17 noted, "*So [analysts] are just looking for 'tell me why.' Explain to me, again in that context of attack surface, who is attacking me? Why is he attacking me? Is there a pattern? Is it random? Is there somebody specific I should be worried about?*"

Participants also suggested explanations might inform subsequent actions by collecting, analyzing, and presenting prior practitioner responses to similar events. E04 described the value of this additional context saying, "*[There are] 50 analysts are working in this environment, they are not prioritizing these alerts, and they're prioritizing these [other] alerts. So you can set up three different levels for that parameter, analyst preference, and you can set it up as 'low', 'medium', and 'high'. And if I'm working, I would see that this alert is coming, these are 50 experienced analysts and they weren't prioritizing this. So I would prefer going something which is rated medium.*"

However, one participant (M15) had some skepticism regarding the addition of suggested remedial actions as they may be redundant with existing playbooks. Playbooks lay out predefined, and often pre-approved, actions for analysts in response to a particular security event [78]. M15 explained that their SOC maintains a set of playbooks based on the organization's collective prior experience that is regularly reviewed by leadership, limiting the value of, and potentially conflicting with, suggestions provided by a tool.

Summary graphs/reports for business people – Putting events together, not just instance by instance. Other participants find value in summarized graphs and reports. In addition to conveying information to less technically savvy stakeholders such as new analysts or business executives, graphics that help put attacks and affected users (spatial aspect) on a visual timeline (temporal aspect) would help inform their understanding: "*Give them better visualization tools. Any information they want to see, put it in context with the respect of temporal line, which is timeline, what happened, time difference and everything ... let's call it user space, timeline and user space ... And visualization will be the most powerful aspect of making sense of that. If we can't make them visualize the information with proper context, pretty much nothing will work.*" (D08)

Provide high-level view of model history to understand attacker behavior changes over time. One participant noted that they would appreciate a global explanation on how models are changed over time to understand on how attackers are changing. For example, R11 explained: "*the malicious campaigns change from time to time... So for machine [learning] models, what we have to do is to retrain the model from time to time. But in the meanwhile, if we can understand what has been changed, why has changed, I think they will save our time a lot instead of just retraining the model with the most recent dataset... that will help us understand the chain of the malicious attack.*"

Additional interface (and redaction) considerations. Participants also suggested improvements for *how* explanations could be presented to improve usability and also privacy. This included changes that match typical design best practices, including using natural language when specific syntax can be hard to recall [79, pg. 332-337], allowing interaction with the output (e.g., supporting data filtering) [80], and showing explanations on-demand [81]. Participants also suggested using distinctive visual summaries for common information to reduce the amount of information presented to users, avoiding information overload.

Participants also noted that privacy may be impacted negatively through the presentation of internal system information. M13, for instance, was concerned for possible intellectual property (IP) leakage when presenting explanations to customers. They noted that the ability to intentionally withhold information in these explanations would improve privacy: "*The ability to redact certain things [would be useful]...you're not going to show your source code to the customer. But you could show conceptually and allow differentiated levels of access depending on what your trust level is with that customer.*"

7. Discussion and Conclusion

7.1. Interfacing ML and Rules

A main finding in §5.1 is that practitioners generally believe ML tools are *not effective enough* to be used alone and cannot replace rule-based methods. A main perceived downside of ML is the difficulty to control false positives. While rule-based methods are perceived to better control false positives (after careful expert tuning), they also sacrifice on false negatives (§5.1), are easily evaded (§ 5.5), require frequent updates (§5.4), and are highly dependent on human expertise (§5.2). As a result, practitioners use multiple tools in parallel to improve overall effectiveness.

Recommendation. Given this major need for effectively using both ML and rule-based methods together in security operations, we recognize a key gap in current research, which is, *how to do it*. Figure 1 illustrates that ML and rule-based methods' strengths and weaknesses complement each other, but it is unclear how to systematically integrate the two approaches to maximize their collective strengths.

Instead of studying ML or rule-based methods in isolation, research is needed to explore *interfacing* ML and rules. For example, researchers have started to look at using statistical methods (e.g., ML) to automatically generate rules [82]–[84]. This is likely a fruitful area for future work as rules could offer a useful interface with ML that analysts are already familiar with. Specifically, analysts could provide initial rule sets based on their expertise to guide the ML's search of the possible classification space. The ML model would then take these rules and historical data as input to identify additional rules that would be provided back to the analyst for review. This process is similar to recent work that has shown promise in the related area of

program synthesis [83], [85], [86]. Also, by using rules as the foundation for this approach, analysts maintain the power of rules (e.g., the ability to quickly remedy urgent issues, reason about output, etc.), while, at the same time, helping the ML learn and reaping its benefits.

Another example is a recent paper that uses "common sense" (i.e., rules) to reduce misclassifications of ML [87]–[89]. Using rules such as "a car moves faster than a pedestrian" or " a person is usually vertically thinner than a car" to regulate ML behaviors can help to reduce misclassifications of ML, even in an adversarial environment [87]. While these general ideas show the potential to deeply integrate ML and rules, more work is needed to understand how to systematically integrate them within the SOC context.

7.2. "Less Glamorous" ML Problems

ML tools' effectiveness not only depends on the system design for the main task (e.g., detecting attacks) but also depends on a range of "less glamorous" supporting tasks around *data* (§5.6). While data scarcity is a well-recognized problem in general machine learning [90], [91], recent work [7] reviewed this problem (from researchers' perspectives) specifically for security-oriented ML applications. They pointed out the common data sampling biases and label inaccuracy problems in existing model development and evaluation pipelines. Contrasting with this work, our study provides the industrial practitioners' perspectives which concern making *operation-level* decisions. For example, instead of worrying about data scarcity, practitioners are often overwhelmed by the large volume of data (e.g., network traces, system logs, alerts). They have expressed challenges to determine how much data to collect, how long to keep historical data (in what representation), and how to assess the usefulness of data to current and future models. It is unrealistic to keep all the historical (raw) data over time, but historical data is valuable for a variety of reasons, e.g., debug models (§5.2), generate explanations (§6.3), and inform actions (§6.6). Meanwhile, as the attack landscape is evolving, there is a need for continually collecting new data to support model updating (§5.4).

Recommendation. Our recommendation is to further investigate data issues in the context of SOC and IT security by qualitatively and quantitatively studying practitioners' perspectives. For example, a particular topic of interest is *why* existing solutions (e.g., active learning [90], concept drift detection [3], [92]) are not adopted to (or cannot sufficiently) address data problems manifested in day-to-day operations. These insights can help to inform the needed properties or capabilities of new tools to support security practitioners to make *operation-level* decisions. This may involve innovating ways to recognize and reason (attacker or benign) behavior changes to inform the *timing* and *methods* for model updating, to identify *important data* to collect, store, or label with minimized overhead, or to design new training/updating methods to reduce the amount of data needed. It may also be fruitful to solve the problem with effective collaboration between human experts (who know the problem space) and ML models (which can learn what data and how much is needed for training/updating).

7.3. Use-driven Explanation

One of our main findings in §6 is that explanations need to be tailored to specific security use cases and requirements that differ from generic ML explanations. Specifically, explanation methods are needed by practitioners not only to (1) understand the classification model itself (e.g., verifying correctness) but also (2) to provide context to understand the detected security event (e.g., to inform further actions). Existing research has been primarily focused on developing explanation techniques for goal-1 (understanding the model) [19]–[21], [24], [50], [65], while the ML capability to support goal-2 (providing context for security events) has not been sufficiently explored. Further, participants demonstrated concerns regarding ML tools' utility when integrated into their established workflows (e.g., conflicting with SOC playbook procedures), and suggested additional information and interaction needs not supported by current explanations. Together, these results indicate a divide between general ML/HCI academic (improving for goal-1) and security practitioner perspectives of explanation needs (both goal-1 and goal-2, with a focus on integration with existing workflows).

Recommendation. To better align explanation development to practitioner needs, our recommendation is that researchers should proactively engage target users (e.g., security analysts) when designing explanation methods for security tasks. The evaluation of explanation methods should consider users' downstream tasks and evaluate whether the explanation can save users' time end-to-end.

Our work provides an initial step toward this goal, synthesizing several practitioner explanation needs: (1) An explanation scheme can provide helpful context for the current event by identifying and highlighting similarities with previously happened (known) events or/and explain how similar previous events were handled. (2) It is desirable if explanation schemes can provide *actionable* information to support further actions for analysts (e.g., deciding which IP/port to block, which server to shut down). (3) The presentation and interface of explanation schemes should be customized to the local environment and support aggregation of events (e.g., using time-bound graphical or natural language interfaces). (4) It is desirable if the explanation engines can be interacted with and support queries from analysts. Future work should build on this, leveraging existing expert interface design literature [81], [93]–[96], as well as formative and summative interface assessments [97, pg. 271-275] with practitioners.

Acknowledgments

This work was supported in part by NSF grants CNS-2030521, CNS-2055233, CNS-1955719, an Amazon Research Award, C3.AI Research, IBM-Illinois Discovery Accelerator Institute, and the Graduate Research Fellowship Program (DGE-1746047).

References

[1] D. J. Chaboya, R. A. Raines, R. O. Baldwin, and B. E. Mullins, "Network intrusion detection: Automated and manual methods prone to attack and evasion," *IEEE Security and Privacy*, 2006.

[2] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. of IEEE S&P*, 2010.

[3] L. Yang, W. Guo, Q. Hao, A. Ciptadi, A. Ahmadzadeh, X. Xing, and G. Wang, "CADE: Detecting and explaining concept drift samples for security applications," in *Proc. of USENIX Security*, 2021.

[4] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. of BICT*, 2016.

[5] X. Han, T. F. J. Pasquier, A. Bates, J. Mickens, and M. I. Seltzer, "Unicorn: Runtime provenance-based detector for advanced persistent threats," in *Proc. of NDSS*, 2020.

[6] W. U. Hassan, A. Bates, and D. Marino, "Tactical provenance analysis for endpoint detection and response systems," in *Proc. of IEEE S&P*, 2020.

[7] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *Proc. of USENIX Security*, 2022.

[8] H. S. Anderson and P. Roth, "EMBER: an open dataset for training static PE malware machine learning models," *CoRR*, vol. abs/1804.04637, 2018.

[9] A. Alsaheel, Y. Nan, S. Ma, L. Yu, G. Walkup, Z. B. Celik, X. Zhang, and D. Xu, "ATLAS: A sequence-based learning approach for attack investigation," in *Proc. of USENIX Security*, 2021.

[10] S. M. Milajerdi, B. Eshete, R. Gjomemo, and V. Venkatakrishnan, "Poirot: Aligning attack behavior with kernel audit records for cyber threat hunting," in *Proc. of CCS*, 2019.

[11] T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda, "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," in *Proc. of ACSAC*, 2013.

[12] T. van Ede, H. Aghakhani, N. Spahn, R. Bortolameotti, M. Cova, A. Continella, M. van Steen, A. Peter, C. Kruegel, and G. Vigna, "DeepCASE: Semi-Supervised Contextual Analysis of Security Events," in *Proc. of IEEE S&P*, 2022.

[13] J. Lee, F. Tang, P. M. Thet, D. Yeoh, M. Rybczynski, and D. M. Divakaran, "Sierra: Ranking anomalous activities in enterprise networks," in *Proc. of Euro S&P*, 2022.

[14] A. Nadeem, D. Vos, C. Cao, L. Pajola, S. Dieck, R. Baumgartner, and S. Verwer, "Sok: Explainable machine learning for computer security applications," in *Proc. of Euro S&P*, 2023.

[15] A. S. Jacobs, R. Beltiukov, W. Willinger, R. A. Ferreira, A. Gupta, and L. Z. Granville, "AI/ML for Network Security: The Emperor has no Clothes," in *Proc. of CCS*, 2022.

[16] W. Guo, D. Mu, J. Xu, P. Su, G. Wang, and X. Xing, "Lemna: Explaining deep learning based security applications," in *Proc. of CCS*, 2018.

[17] A. Warnecke, D. Arp, C. Wressnegger, and K. Rieck, "Evaluating explanation methods for deep learning in security," in *Proc. of Euro S&P*, 2020.

[18] C. Crowley and B. Filkins, "Sans 2022 soc survey," https://www.sans.org/white-papers/sans-2022-soc-survey/, 2022.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?" Explaining the predictions of any classifier," in *Proc. of KDD*, 2016.

[20] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. of NeurIPS*, 2017.

[21] R. C. Fong, M. Patrick, and A. Vedaldi, "Understanding deep networks via extremal perturbations and smooth masks," in *Proc. of ICCV*, 2019.

[22] N. Capuano, G. Fenza, V. Loia, and C. Stanzione, "Explainable artificial intelligence in cybersecurity: A survey," *IEEE Access*, 2022.

[23] D. Gunning, E. Vorm, Y. Wang, and M. Turek, "Darpa's explainable ai (xai) program: A retrospective," *Authorea Preprints*, 2021.

[24] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," in *Proc. of NeurIPS*, 2019.

[25] D. Han, Z. Wang, W. Chen, Y. Zhong, S. Wang, H. Zhang, J. Yang, X. Shi, and X. Yin, "Deepaid: Interpreting and improving deep learning-based anomaly detection in security applications," in *Proc. of CCS*, 2021.

[26] M. Melis, M. Scalas, A. Demontis, D. Maiorca, B. Biggio, G. Giacinto, and F. Roli, "Do gradient-based explanations tell anything about adversarial robustness to android malware?" *Int. J. Mach. Learn. Cybern.*, vol. 13, no. 1, pp. 217–232, 2022.

[27] Y. Gan, Y. Mao, X. Zhang, S. Ji, Y. Pu, M. Han, J. Yin, and T. Wang, ""Is Your Explanation Stable?": A Robustness Evaluation Framework for Feature Attribution," in *Proc. of CCS*, 2022.

[28] B. A. Alahmadi, L. Axon, and I. Martinovic, "99% false positives: A qualitative study of SOC analysts' perspectives on security alarms," in *Proc. of USENIX Security*, 2022.

[29] A. Pecchia, A. Sharma, Z. Kalbarczyk, D. Cotroneo, and R. K. Iyer, "Identifying compromised users in shared computing infrastructures: A data-driven bayesian network approach," in *Proc. of SRDS*, 2011.

[30] Y. Shen, E. Mariconti, P. A. Vervier, and G. Stringhini, "Tiresias: Predicting security events through deep learning," in *Proc. of CCS*, 2018.

[31] L. Allodi and F. Massacci, "Security events and vulnerability data for cybersecurity risk estimation," *Risk Analysis*, 2017.

[32] C. Feng, S. Wu, and N. Liu, "A user-centric machine learning framework for cyber security operations center," in *Proc. of ISI*, 2017.

[33] V. G. Li, M. Dunn, P. Pearce, D. McCoy, G. M. Voelker, S. Savage, and K. Levchenko, "Reading the tea leaves: A comparative analysis of threat intelligence," in *Proc. of USENIX Security*, 2019.

[34] A. AlEroud and G. Karabatis, "Beyond data: Contextual information fusion for cyber security analytics," in *Proc. of SAC*, 2016.

[35] T. Ban, N. Samuel, T. Takahashi, and D. Inoue, "Combat security alert fatigue with ai-assisted techniques," in *Proc. of CSET Workshop*, 2021.

[36] C. Krügel and W. K. Robertson, "Alert verification determining the success of intrusion attempts," in *Proc. of DIMVA*, 2004.

[37] E. Raftopoulos, M. Egli, and X. Dimitropoulos, "Shedding light on log correlation in network forensics analysis," in *Proc. of DIMVA*, U. Flegel, E. Markatos, and W. Robertson, Eds., 2013.

[38] G. Ho, A. Sharma, M. Javed, V. Paxson, and D. Wagner, "Detecting credential spearphishing in enterprise settings," in *Proc. of USENIX Security*, 2017.

[39] F. B. Kokulu, A. Soneji, T. Bao, Y. Shoshitaishvili, Z. Zhao, A. Doupé, and G.-J. Ahn, "Matched and mismatched socs: A qualitative study on security operations center issues," in *Proc. of CCS*, 2019.

[40] C. Dietrich, K. Krombholz, K. Borgolte, and T. Fiebig, "Investigating system operators' perspective on security misconfigurations," in *Proc. of CCS*, 2018.

[41] O. Akinrolabu, I. Agrafiotis, and A. Erola, "The challenge of detecting sophisticated attacks: Insights from soc analysts," in *Proc. of ARES*, 2018.

[42] S. C. Sundaramurthy, A. G. Bardas, J. Case, X. Ou, M. Wesch, J. McHugh, and S. R. Rajagopalan, "A human capital model for mitigating security analyst burnout," in *Proc. of SOUPS*, 2015.

[43] J. Goodall, W. Lutters, and A. Komlodi, "I know my network: Collaboration and expertise in intrusion detection," in *Proc. of CSCW*, 2004.

[44] S. C. Sundaramurthy, J. McHugh, X. Ou, M. Wesch, A. G. Bardas, and S. R. Rajagopalan, "Turning contradictions into innovations or: How we learned to stop whining and improve security operations," in *Proc. of SOUPS*, 2016.

[45] S. Oesch, R. Bridges, J. Smith, J. Beaver, J. Goodall, K. Huffer, C. Miles, and D. Scofield, "An assessment of the usability of machine learning based tools for the security operations center," in *Proc. of iThings*, 2020.

[46] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, p. 68–77, 2019.

[47] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 18, no. 6, pp. 275–285, 2004.

[48] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[49] H. Lakkaraju, S. H. Bach, and J. Leskovec, "Interpretable decision sets: A joint framework for description and prediction," *Proc. of KDD*, 2016.

[50] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proc. of ICCV*, 2017.

[51] X. Zhang, N. Wang, H. Shen, S. Ji, X. Luo, and T. Wang, "Interpretable deep learning under fire," in *Proc. of USENIX Security*, 2020.

[52] A. Papenmeier, D. Kern, D. Hienert, Y. Kammerer, and C. Seifert, "How accurate does it feel? – human perception of different types of classification mistakes," in *Proc. of CHI*, 2022.

[53] W. Zhang and B. Y. Lim, "Towards relatable explainable ai with the perceptual process," in *Proc. of CHI*, 2022.

[54] Q. V. Liao, D. Gruen, and S. Miller, "Questioning the ai: informing design practices for explainable ai user experiences," in *Proc. of CHI*, 2020.

[55] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proc. of AISec*, 2011.

[56] G. Apruzzese, H. Anderson, S. Dambra, D. Freeman, F. Pierazzi, and K. Roundy, "Position: "Real Attackers Don't Compute Gradients": Bridging the Gap Between Adversarial ML Research and Practice," in *Proc. of SaTML*, 2023.

[57] K. Grosse, L. Bieringer, T. R. Besold, B. Biggio, and K. Krombholz, ""Why do so?" - A Practical Perspective on Machine Learning Security," *IEEE TIFS*, 2023.

[58] R. S. S. Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, and S. Xia, "Adversarial machine learning-industry perspectives," in *Proc. of IEEE SPW*, 2020.

[59] L. Bieringer, K. Grosse, M. Backes, B. Biggio, and K. Krombholz, "Industrial practitioners' mental models of adversarial machine learning," in *Proc. of SOUPS*, 2022.

[60] J. Mink, H. Kaur, J. Schmüser, S. Fahl, and Y. Acar, ""Security is not my field, I'm a stats guy": A Qualitative Root Cause Analysis of Barriers to Adversarial Machine Learning Defenses in Industry," in *Proc. of USENIX Security*, 2023.

[61] B. Berg, *Qualitative Research Methods for the Social Sciences*, 2001.

[62] J. Mink, H. Benkraouda, L. Yang, A. Ciptadi, A. Ahmadzadeh, D. Votipka, and G. Wang, "Everybody's Got ML, Tell Me What Else You Have: Practitioners' Perception of ML-Based Security Tools and Explanations (Supplementary Materials)," 2023, https://osf.io/ykt9c/?view_only=44692b1120134fd2888b6be217326987.

[63] "General data protection regulation (gdpr)," https://gdpr-info.eu/, 2023.

[64] "Personal information protection law (pipl) of the people's republic of china," http://en.npc.gov.cn.cdurl.cn/2021-12/29/c_694559.htm, 2021.

[65] C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud, "Explaining image classifiers by counterfactual generation," in *Proc. of ICLR*, 2019.

[66] X. Xu, C. Liu, Q. Feng, H. Yin, L. Song, and D. Song, "Neural network-based graph embedding for cross-platform binary code similarity detection," in *Proc. of CCS*, 2017.

[67] A. Marcelli, M. Graziano, X. Ugarte-Pedrero, Y. Fratantonio, M. Mansouri, and D. Balzarotti, "How machine learning is solving the binary function similarity problem," in *Proc. of USENIX Security*, 2022.

[68] E. Rader, S. Hautea, and A. Munasinghe, ""I Have a Narrow Thought Process": Constraints on Explanations Connecting Inferences and Self-Perceptions," in *Proc. of SOUPS*, 2020.

[69] J. Corbin and A. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 4th ed. Thousand Oaks, CA: Sage Publications, 2015.

[70] Rev, https://rev.com, 2022.

[71] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[72] K. R. Fulton, S. Katcher, K. Song, M. Chetty, M. L. Mazurek, C. Messdaghi, and D. Votipka, "Vulnerability discovery for all: Experiences of marginalization in vulnerability discovery," in *Proc. of IEEE S&P*, 2023.

[73] D. Votipka, R. Stevens, E. M. Redmiles, J. Hu, and M. L. Mazurek, "Hackers vs. testers: A comparison of software vulnerability discovery processes," in *Proc. of IEEE S&P*, 2018.

[74] O. Akgul, T. Eghtesad, A. Elazari, O. Gnawali, J. Grossklags, M. L. Mazurek, A. Laszka, and D. Votipka, "Bug hunters' perspectives on the challenges and benefits of the bug bounty ecosystem," in *Proc. of USENIX Security*, 2022.

[75] D. Votipka, S. Rabin, K. Micinski, J. S. Foster, and M. L. Mazurek, "An observational investigation of reverse engineers' processes," in *Proc. of USENIX Security*, 2020.

[76] $(ICS2)^2$, "Cybersecurity workforce study," $(ICS2)^2$, Tech. Rep., 2022. [Online]. Available: https://www.isc2.org/Research/Workforce-Study

[77] R. Crawford and M. Smith, "How to solve the data science skills shortage," SAS Institute, Tech. Rep., 2022. [Online]. Available: https://www.sas.com/content/dam/SAS/documents/technical/education/en/solve-data-science-skills-shortage-us-113038.pdf

[78] R. Stevens, D. Votipka, J. Dykstra, F. Tomlinson, E. Quartararo, C. Ahern, and M. L. Mazurek, "How ready is your ready? assessing the usability of incident response playbook frameworks," in *Proc. of CHI*, 2022.

[79] B. Shneiderman, C. Plaisant, M. S. Cohen, S. Jacobs, N. Elmqvist, and N. Diakopoulos, *Designing the user interface: strategies for effective human-computer interaction*. Pearson, 2016.

[80] C. Ahlberg, C. Williamson, and B. Shneiderman, "Dynamic queries for information exploration: An implementation and evaluation," in *Proc. of CHI*, 1992.

[81] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *The craft of information visualization*. Elsevier, 2003.

[82] H. Lakkaraju, S. H. Bach, and J. Leskovec, "Interpretable decision sets: A joint framework for description and prediction," in *Proc. of KDD*, 2016.

[83] H. Hajipour, M. Malinowski, and M. Fritz, "Ireen: Reverse-engineering of black-box functions via iterative neural program synthesis," in *Proc. of ECML PKDD*, 2021.

[84] L. Shi, Y. Li, B. T. Loo, and R. Alur, "Network traffic classification by program synthesis," in *Proc. of TACAS*, J. F. Groote and K. G. Larsen, Eds., 2021.

[85] H. Peleg, S. Shoham, and E. Yahav, "Programming not only by example," in *Proc. of ICSE*, 2018.

[86] J. Hu, P. Vaithilingam, S. Chong, M. Seltzer, and E. L. Glassman, "Assuage: Assembly synthesis using a guided exploration," in *Proc. of UIST*, 2021.

[87] Y. Man, R. Muller, M. Li, Z. B. Celik, and R. Gerdes, "That person moves like a car: Misclassification attack detection for autonomous systems using spatiotemporal consistency," in *Proc. of USENIX Security*, 2023.

[88] Z. Yang, Z. Zhao, B. Wang, J. Zhang, L. Li, H. Pei, B. Karlas, J. Liu, H. Guo, C. Zhang, and B. Li, "Improving certified robustness via statistical learning with logical reasoning," in *Proc. of NeurIPS*, 2022.

[89] T. Zhang, L. Lowmanstone, X. Wang, and E. L. Glassman, "Interactive program synthesis by augmented examples," in *Proc. of UIST*, 2020.

[90] B. Settles, "Active learning literature survey," 2009.

[91] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," *IEEE TKDE*, 2019.

[92] R. Jordaney, K. Sharad, S. K. Dash, Z. Wang, D. Papini, I. Nouretdinov, and L. Cavallaro, "Transcend: Detecting concept drift in malware classification models," in *Proc. of USENIX Security*, 2017.

[93] L. Battle and J. Heer, "Characterizing exploratory visual analysis: A literature review and evaluation of analytic provenance in tableau," *Computer Graphics Forum*, 2019.

[94] J. Heer and B. Shneiderman, "Interactive dynamics for visual analysis," *Communications of the ACM*, vol. 55, no. 4, pp. 45–54, 2012.

[95] A. Perer and B. Shneiderman, "Systematic yet flexible discovery: Guiding domain experts through exploratory data analysis," in *Proc. of IUI*, 2008.

[96] K. Dimitriadou, O. Papaemmanouil, and Y. Diao, "Explore-by-example: An automatic query steering framework for interactive data exploration," in *Proc. of SIGMOD*, 2014.

[97] J. Lazar, J. H. Feng, and H. Hochheiser, *Research Methods in Human-Computer Interaction*. San Francisco: Elsevier Science & Technology, 2017.

Appendix A. Additional Discussion and Concepts

A.1. Participants and Tool Usage

Defense vs. offense expertise. As described in §3, we used a screening survey to identify qualified participants and prioritized those of different roles and experiences to add to the diversity of perspectives. While we initially recruited both offensive and defense-oriented participants, we later prioritized defense-oriented ones. This is due to several interviews noting that classification tools were primarily considered and used in defensive scenarios.

Use of security tools. Among the 18 participants, 16 self-reported as users/operators of security tools while 12 self-reported as developers/producers of such tools (10 hold both roles). There is a range of tasks performed using classification tools such as intrusion detection (n=7), threat hunting (n=3), and vulnerability assessment (n=8). In addition to security operations, participants also noted other operational tasks needed to manage or develop tools. Several participants discussed developing the system (n=12), (re)implementing tools for their environments (n=6), collecting data for training (n=4), and maintaining the service for others (n=4).

Number of Participants	**18**
Gender:	
Male	18
Age:	
Mean	36
Standard Deviation	11.8
Education:	
Secondary School (High school, German Gymnasium)	1
Bachelor's Degree (B.A., B.S., B.Eng.)	3
Master's Degree (M.A., M.S., M.Eng., MBA)	10
Other Doctoral Degrees (Ph.D., Ed.D.)	4
Location:	
USA (8), Pakistan (4), Japan (2), Turkey (1), Qatar (1), Canada (1), Malaysia (1)	
Ethnicity*:	
East Asian (5), South Asian (5), White or of European descent (5), Southeast Asian (3), Middle Eastern (2), Black or of African descent (1), Hispanic or Latino/a/x (1), Prefer not to disclose (1)	

TABLE 3: **Participant demographics** — Aggregated demographics of our participants (*multiple answers allowed).

Tool opacity is a general concern. There were general concerns about tool opacity which comes from two main sources. First, opacity is caused by the use of third-party tools. Among the n=16 self-reported users of the classification tools, most of them (n=13) have used at least one tool developed by others (regardless ML or rules), and there is a lack of visibility into the tool's internals to understand how they work. Second, opacity is related to the difficulty to reason/understand certain tools' classification results, especially for ML. This aspect has been discussed in detail in §5.2 and §6.

A.2. Perceptions of Tools

Relationship between the five properties of ML/rules. Not too surprisingly, the five properties also influence one another. For example, when updating the tool to handle new threats (related to *security* and *adaptability*), the *efficiency* of the tool affects its *effectiveness*. As mentioned in §5.3 and §5.4, practitioners prefer to write a new rule for a quick, short-term fix, before developing long-term solutions. Similarly, *usability* could affect *efficiency*, as E04 notes that a tool that is harder to understand can lead to slower responses: "*If you have 20 alerts coming in an hour but it wouldn't specify what the issue is, you have to investigate the packet every time... it would obviously be a headache for the analyst.*" Since we did not explicitly ask participants about the relationships among properties, we keep this discussion brief.

Primary Code	Subcode	Freq.	Description
Domain (κ=1.00)	**Finance**	3	Organization specializes in finance-based domain.
	Industrial Control Systems	1	Organization specializes in industrial control systems-based domain.
	Education	4	Organization specializes in education-based domain.
	Healthcare	1	Organization specializes in healthcare-based domain.
	Government Research	1	Performs research in a governmental organization.
	Managed IT	4	Manages IT services for other organizations.
	Tool Development	2	Performs tool development for cybersecurity in other organizations.
	Managed IT & Tool Development	2	Performs IT services & tool development for cybersecurity in other organizations.
Technical Tasks (κ=0.89)	**Compliance Audit**	1	Performs audits of systems to ensure standards (GDPR, HIPAA, etc) compliance.
	Hardware Management	1	Manages and monitors hardware devices (routers, links, etc).
	Incident Response	8	Responding and mitigating a security event.
	Reverse Engineering	3	Analyzing a suspected malware to determine its traits/behaviors.
	Digital Forensics	3	Analyzing digital evidence such as logs or memory, to investigate a security event.
	Threat Hunting	3	Preemptively searching for on-going security threats.
	Integrate Security System	6	Integrating a security system with an existing infrastructure.
	Manage IT Policies	3	Generating, updating, or enforcing IT policies.
	Intrusion Detection	7	Automated monitoring of a network or a set of hosts to find security threats.
	Vulnerability Assessment	8	Evaluating a system for vulnerabilities.
	Threat Research	1	Understanding patterns in security threats and predicting new trends.
	Curate Threat Database	1	Curating databases of security threats for future use.
	Collect Data for Training	4	Collecting malware or exploits for training a model.
	Maintain Model Service	4	Running, maintaining, and keeping a model up to date for clients to use.
	Record Logs	3	Recording network or host-based logs for compliance, model training, etc.
	Analyze Alerts	4	Verifying, assessing, and responding to alerts from classification tools.
	Develop Tool	12	Developing a framework, model, or a ruleset for a security classification tool.
Non-Technical Tasks (κ=0.83)	**Client-Interaction/Sales**	5	Holds a client-facing position that requires collaboration or sales.
	Management	6	Manages a set of employees.
	Pedagogy	4	Lectures/runs an academic course.
	Cyber Security Awareness	4	Efforts to encourage better awareness of security threats and safe practices.
	Write Reports	8	Creating reports, papers, or blogs to teach people about cybersecurity subject matter.
	Report Incidents	1	Reporting cyber security incidents to law enforcement or government institutions.
Task Output Recipient (κ=1.00)	**General Public**	4	An output to the general public.
	Clients/Customers	6	An output to the clients/customers of a company.
	Technical Position	4	An output to an information technology, SOC, or other technologically trained person.
	Project Manager	1	An output to a direct project manager.
	ISP	1	An output to an internet service provider.
	Law/Government	1	An output to a division of law enforcement or local/federal government.
Classification Methods (κ=0.86)	**Rule-based Engine**	14	Uses a manually crafted set of rules, patterns, or signatures to determine classification.
	Machine Learning Engine	14	Uses data to train a machine learning model to determine classification.
	Composite Engine	5	Uses rules-based and machine learning approaches to determine classification.
	Manual Analysis	5	Uses learned knowledge and manual efforts to determine a classification.
Classification Output (κ=1.00)	**Malicious/Benign**	14	Classifies a file, packet, behavior, etc as malicious or benign.
	Anomalous	7	Classifies behavior as expected or anomalous.
	Vulnerable	5	Classifies software as vulnerable.
	Exploitation Likelihood	1	Classifies likelihood of exploitation for specific vulnerabilities.
	Changed Integrity	1	Classifies files/software/data as having been modified.
	High-risk/Sensitive Data	4	Classifies data as sensitive/high-risk.
Tool Output Recipient (κ=1.00)	**Information Technology**	16	An IT, SOC, or generic security analysts receives the tool's output.
	Other Tool	1	Another tool receives the tool's output.
Developer-Recipient Relation (κ=1.00)	**Internal**	9	The tool was developed (significantly modified) by the same organization that uses it.
	External	13	The tool was developed (significantly modified) by a different organization that uses it.
Positive Tool Perceptions (κ=0.80)	**Adaptable**	8	Easy to be adjust to specific environments, or updates over time.
	Secure	2	Works correctly under adversarial conditions or follows privacy/security compliance.
	Effective	17	Classifies well based on some metric, can generalize, and can succeed in normal tasks.
	Efficient	8	Requires few resources (time, data, computation) during implementation or production.
	Usable	9	Requires little cognitive overhead and effort. Easy to understand and train/deploy.
Negative Tool Perceptions (κ=0.81)	**Not Adaptable**	1	Difficult to adjust to specific environments, or updates over time.
	Not Secure	7	Easy to bypass under adversarial conditions.
	Not Effective	18	Does not work well under normal conditions due to FPs/FNs, poor generalization, etc.
	Not Efficient	13	Requires many resources (time, data, computation) during implementation or production.
	Not Usable	13	Requires much cognitive overhead and effort. Hard to understand and implement/deploy.
Classification Method Factors (κ=0.83)	**Adaptability**	5	Ability to be updated temporally or to a specific environment.
	Ease of Implementation	9	Ability to integrate the system into existing architecture, human expertise, and environment.
	Security	4	Ability to withstand adversarial environments.
	Effectiveness	10	Ability to classify well according under normal conditions.
	Usability	10	Ability to understand, develop, and deploy classifier.
	Efficiency	9	Required resources (money, time, data, computing resource).
	Construction of Report	1	Ability to construct reports for managers and auditors.

TABLE 4: **Interview codebook** — We show the coded and related descriptions for our interview coding.

Primary Code	Subcode	Freq.	Description
Which Method was Verified (κ=1.00)	**Machine Learning**	15	Discussed a verification method for a machine learning-based classifier.
	Rule-based	11	Discussed a verification method for a rule-based classifier.
	Composite	2	Discussed a verification method for a composite classifier.
Verification Method (κ=0.85)	**Method in Composite**	1	View which method(s) in the composite tool contribute to the detection.
	Verify on Other Data	7	Provide data for other known classification or expected patterns.
	Prediction Confidence	1	Use tool-provided prediction confidence.
	Consistency of Multi-Methods	7	Correlate the results to multiple tools to determine classification.
	Threat Intelligence Feed	5	Compare with reported malicious activity from other organizations.
	Contact Tool Developers	1	Contact the tool developers to discuss the reason behind certain classifications.
	Static/Dynamic Analysis	1	Perform static/dynamic analysis on suspicious files.
	Undefined Manual Verification	10	Unspecific/vague notion of a manual analysis.
	Search Online	1	Look up classification result online to determine meaning and likely accuracy.
	Host-based Forensics	5	Analyze host-based forensics such as logs or memory.
	Network-based Forensics	7	Analyze IP sources, traffic patterns/volume, or other packet info.
	Matched Rule Analysis	3	Verify whether the matched rule is valid or what it implies.
	Trust in Tool Developer	2	Determine validity based on placed trust in the tool developer.
Effect of Explanations on ML Opinion (κ=1.00)	**More Likely to Adopt**	7	Explanations positively affect the adoption of machine learning-based classifiers.
	Does Not Affect Adoption	3	Explanations have no effect on the adoption of machine learning-based classifiers.
Explanation Utility (κ=0.83)	**Helpful**	18	Perceiving a particular explanation as helpful, in some or all cases.
	Not Helpful	12	Perceiving a particular explanation as not helpful, in some or all cases.
	Uncertain	4	Unsure whether an explanation would be helpful.
Used Explanation (κ=1.00)	-	10	Has used an explanation in practice or has incorporated one into their tool.
Explanation: Positive Perception (κ=0.85)	**Understand Classified Object**	10	Used to understand the classified object and related scenario.
	Understand Model	9	Used to understand the model's reliability, functionality, and patterns.
	General Understanding	1	Used to provide an unspecified understanding of the situation .
	Efficiency	10	Saves time, often by directing focus and presenting relevant info.
	Security	1	Used to increase the robustness of the classifier or account for vulnerabilities.
	Usability	2	Used to help simplify or reduce the effort required by analysts.
Explanation: Negative Perception (κ=0.85)	**No Understanding Provided**	6	Doesn't help provide an understanding of the scenario or the model.
	Hard to Implement	2	The technique is difficult to add to the system.
	Not Trustworthy	3	Lack of trust in the resulting explanation.
	Not Efficient	3	The explanation would take a long time to parse by an end-user.
	Not Applicable	2	Does not address the needs of the task.
	Not Secure	1	Is not robust to adversarial attacks on the explanation.
	Redundant	1	Information is provided via another vector already.

TABLE 5: **Interview codebook** — We show the coded and related descriptions for our interview coding.