Usable Security for ML Systems in Mental Health: A Framework

Helen Jiang

Independent (affiliated with Georgia Institute of
Technology)
helen.h.jiang@gmail.com

Erwen Senge

Independent
erwen@protonmail.com

ABSTRACT

While the applications and demands of Machine learning (ML) systems in mental health are growing, there is little discussion nor consensus regarding a uniquely challenging aspect: building security methods and requirements into these ML systems, and keep the ML system usable for end-users. This question of usable security is very important, because the lack of consideration in either security or usability would hinder large-scale user adoption and active usage of ML systems in mental health applications.

In this short paper, we introduce a framework of four pillars, and a set of desired properties which can be used to systematically guide and evaluate security-related designs, implementations, and deployments of ML systems for mental health. We aim to weave together threads from different domains, incorporate existing views, and propose new principles and requirements, in an effort to lay out a clear framework where criteria and expectations are established, and are used to make security mechanisms usable for end-users of those ML systems in mental health. Together with this framework, we present several concrete scenarios where different usable security cases and profiles in ML-systems in mental health applications are examined and evaluated.

KEYWORDS

Mental Health, Machine Learning (ML), Security, Usability, Evaluation, Computer System Life Cycle, Failure Modes

1 INTRODUCTION

With a mental health crisis looming large and many ML systems being built for mental health use cases, it is challenging to trace, analyze, and compare all the designs and implementations of such systems. So far, there is a lack of well-defined framework that describes properties relating to the security of such ML systems in mental health, and even less considerations are given to how such security mechanisms can be *usable* for those systems' end users. However, without usable security, undiscovered, undisclosed, and ill-considered limitations and properties of security decisions would hold back large-scale adoption and usage[2] of ML systems in mental health use cases. For more detailed and nuanced discussions, see our treatment at section 4.3.

The goal of this framework is to establish discussions in communities of mental health, ML, and security, so we can build a

common ground for directions and expectations for usable security in ML systems used in mental health scenarios. Moreover, this framework serves to raise awareness, so that both ML and mental health communities will heed this critical aspect of usable security in ML systems for mental health. We hope that this new, interdisciplinary framework would allow researchers and practitioners to systematically compare usable security attributes across ML systems for mental health, meanwhile to identify potential limitations of particular approaches and trade-offs in different scenarios.

In this short paper, we propose that ML systems in mental health use cases, beyond the privacy and security requirements already mandated by legislation's and regulations — for example, Health Insurance Portability and Accountability Act (HIPPA)[38, 43, 64] in United States, and General Data Protection Regulation (GDPR) in European Union and its member states' national laws[11, 12] — should consider properties of usable security proposed by this framework's four pillars, and be evaluated on their (1)***context*** models, (2)***functionality*** criteria, (3)***trustworthiness*** requirements, and (4)***recovery*** principles across their life cycles.

This work presents our effort to generate discussions and consensus for a common framework in a naturally interdisciplinary area. We built our research on the foundation of computer security research, which has a rich history and long tradition of devising criteria and evaluation rubrics for system designs and implementations. We also incorporated important and recent literature from human-computer interaction (HCI), usable security, and fairness, accountability, and transparency (FAT) research of ML. Weaving these interdisciplinary threads together, we hope that our framework will benefit both researchers and practitioners working on ML systems in mental health.

2 RELATED WORK

There is a long and distinguished tradition in computer security research: presciently define evaluation criteria and structure assessment frameworks, while research communities were still in their early stages of formation. From this tradition, many remarkable security research outcomes have flourished, and guided the design and building of systems and infrastructure we rely on today[21, 22, 30, 31, 37, 51, 67]. However, while the pioneers of security research laid down "psychological acceptability" of users as a key principle for secure system design and implementations[49], this principle has not been actively researched within the security community until much later while security measures keep confusing even experts[3, 10, 54, 68, 71]. Moreover, the "psychological acceptability" principle is often doubted as incompatible with the goal of "security"[7, 15, 44, 47, 56, 63, 69, 71], and much of usable security research has traditionally been done in the HCI community, and usable security is still a small community[41, 45, 65] compared to other areas of security research.

While "psychological acceptability" principle is first identified as the meaning of "usable" in "usable security" [49, 71], there are other efforts trying to precisely define "usability" especially in HCI contexts, based on the "human-centered" attribute of interactive systems. A prominent example is ISO 9241-210 [23]: "usability" is *"the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use"*. Built on this definition, [62] of NIST proposed measurements on usability evaluations.

However, as [62, 71] both point out, measurement of usable security can be highly diverse and context-dependent, meanwhile, such measurements and evaluations focus on the *system* and its interactions with targeted users, often done with small groups in controlled environments[32, 42], with *security* as the users' top concern. While the "security is top priority" assumption can be very reasonable for use cases such as national and corporate security, the same assumption likely does not stand when we are evaluating ML systems in mental health use cases, in which users have diverse top priorities. This complicates the already fragmented landscape[26] of usable security, and while ML applications in mental health and FAT ML research are booming[53, 57], they still do not take usability and security into serious consideration.

3 USABLE SECURITY PILLARS FOR ML SYSTEMS

Our framework evaluates usable security of ML systems in mental health based on four pillars. Each pillar, in turn, serves as the top concern for each major phase of the computer system life cycle, which can be summarized as: (1)design and implementation; (2)deployment; (3)mass adoption and usage; and finally, (4)maintenance and/or disposal[39, 40, 46].

(1) Context: this pillar considers the intended operational environment of the ML system, and how it is designed and built to interact with different types of users with varying purposes, goals, and maliciousness. This pillar is most important during the design and implementation phase of ML systems for mental health.

(2) Functionality: this pillar tackles the well-known security-functionality trade-off[4, 25, 27, 35, 71]. Keeping ML systems functional while making security usable, it is imperative to ask questions about the complexity and resource-intensity of security methods within the already complex and often resource-intensive ML system, the flexibility of chosen methods to accommodate future security requirements, and how they influence user interactions with the ML system. This pillar is most crucial in the deployment phase of ML systems, especially in the initial stage, when such system is in limited use, without users' significant investment of trust and time.

(3) Trustworthiness: this pillar is by nature user-centered. Many non-expert, lay users are already distrustful and leery of ML, and this set of requirements show that on the matter of security and usability, ML systems may still induce users' trust in the sensitive context of mental health. This pillar is the most critical in achieving active usage and large-scale adoption[2] of secure ML systems for mental health.

(4) Recovery: this pillar handles perhaps one of the toughest challenges in both security and usability: what happens, should a security incident (e.g. a breach, a compromise, or a previously undiscovered vulnerability) happens? What are we going to do with the system and users, now and later? How do we account for the incident this time, to minimize the chance that it would happen again? This pillar is the top priority in maintenance and/or disposal phase of the computer system life cycle.

3.1 Context Models

The list can help ask the right questions for designing and building usable security[51] into ML systems for mental health: it determines what and how much "usability" to be considered in a security environment, and move from the more general security threat models, to specific cases of user interactions in mental health scenarios, and also to weigh in negative use cases. The properties below are agnostic to programming languages, software stacks, deployment platforms, and hardware specifications, so they are also flexible enough to accommodate a large class of usable security scenarios for ML systems in mental health.

C1 *Asset Audit.* ML systems in mental health almost inevitably acquire information assets while in use, for example, it may include users' locations, device types etc., as well as patients' functional status information, providers' notes, and organization's intervention plans. Understandably, existing regulations mostly focus on these *acquired* assets. However, in ML systems, "asset" is not only acquired, but also *native* to the system itself: its algorithms and models, ground truths, datasets, decision-making logic, and result evaluations, etc. Therefore, identifying both *native* and *acquired* assets of the ML system is critical for usable security.

C2 *Target User Profiling.* ML systems can be utilized by different stakeholders in mental health: from patients, providers, to government officials, they use the system to achieve different goals. Profiling the system's targeted users is the basis to make concrete observations and reasonable estimations, which are then incorporated into design and implementation requirements. Knowing the targeted users and what they use the ML system for, this is usable security's positive case: legitimate users can establish trusted paths and use the system without being hindered by its security requirements. Usable security's negative case is given in *C4*.

C3 *Behaviors Categorization.* Behaviors of targeted legitimated users described in *C2* can be either *expected* by the system, or *unexpected* and cause the system to fail, error out, or even trigger security incidents. While it is not possible to iterate through all unexpected behaviors from legitimate users, unexpected user behaviors raise two key components of usable security, and need to be addressed in design and implementation: (1) motivating users to behave in a secure manner so to minimize the systems' failures, errors, and security exposures, because users are not the enemy[3]; (2) when such motivations fail, follow the "fail-safe" principle[49], meanwhile deliver warning messages about security and failures with usability in mind [5, 50, 58]. This property is interdependent with **F5**, where we discuss *robustness*.

C4 *Threat Modeling.* Once the assets are audited and target users and behaviors profiled, threat modeling is essential for security,

as threat modeling is a well-studied and used subject in computer security[17, 33, 48, 55]. There are three main components to consider: (1) *assets* the ML system needs to protect, (2) scope of *interactions* between system and user based on **C3**; and (3) *malicious actors* and their actions the systems need to defend against. In contrast to **C2**, *malicious actors* are usable security's negative case: malicious users are stopped or slowed down by the system's security measures.

3.2 Functionality Criteria

The following properties are most useful when seen from a deployment perspective. They describe *how a ML system works with in-place security requirements while interacting with users.*

F1 *Complexity.* Most, if not all ML systems and applications have at least one of the three constraints: time, memory, and computational power. Therefore, any security measures should consider these constraints and its impact on how well the ML system serves the end users. For example, in a high concurrency event where many users are utilizing the same ML system, if a given security method uses negligible computational power resource on users' end but consumers a lot of system resources, we should consider alternatives for this security method. To measure such complexity, we can use either formal algorithmic complexity notions (e.g. Big O, little O), or empirical evaluations. For example, in 10-user, 100-user, 1,000-user concurrency scenarios, what is the average computational overhead or latency for specific sets of security requirements, with other software and hardware constraints stay the same.

F2 *Availability.* For large-scale ML systems, e.g. mental health use cases with multiple targeted user groups, security measures also need to scale. Availability evaluates how well security methods can generalize to cover a ML system's targeted users and behaviors (**C2, C3**) without hindering their access to the ML system. A quantitative heuristic for availability of security measures is estimated user adoption rates across user groups, as well as among the genera user base. Notice that the availability criteria is a trade-off to the "least common mechanism" principle for secure system design[49], and the relative importance between the two are dependent on results of *Context Modeling*, in particular **C4**. Regardless of which one of the two weighs more heavily in specific scenarios, the security mechanism in question must be carefully designed, judiciously implemented, and rigorously tested before real user runs.

F3 *Flexibility.* Retrofitting security to usability is usually a bad idea and doesn't work well[7, 69], therefore it is important to not only prioritize usable security when designing and building ML systems for mental health, but also to not let current implementations become roadblocks to additional security requirements or system capabilities. Having flexibility accommodates future changes in the system and shifting user base, and is a long-term commitment to the system's usable security traits.

F4 *Experience Validation.* To ascertain that security measures did not hold back users, it is crucial to validate real user interactions and experience with the system, regardless of the methods: ideal controlled environments, synthetic experiments, or random sampling. For positive case of usable security (**C2**) that makes the system more secure but not harder for legitimate users, conducting user studies to evaluate their experiences, interactions, effectiveness,

and satisfactions with the system[29, 60] would be indispensable evidence for the ML system's real-world usability.

F5 *Robustness.* Robustness is well-researched in computer system[1, 6, 18, 36, 59], and recent interests in adversarial ML[8, 9, 16] has early roots in ML robustness[61]. In our consideration, robustness is also related to recovery principles in section 3.4, and has two angles: (1) for *security*, to tolerate and withstand certain errors and faults from the ML layer, the system layer, and user interaction layer; and (2) for *usability*, to communicates to users clearly and timely, when trusted paths cannot be established because of scenarios exceeding (1)'s robustness levels. Interdependent with this criterion is **C3** for unexpected user behavior categorization.

3.3 Trustworthiness Requirements

Many non-experts are suspicious and distrustful of ML, because of ML's "blackbox magic" reputation. Moreover, the technical nature of FAT ML methods has not endeared lay users towards machine learning either. Now, suppose that another layer of hard-to-use and hard-to-navigate security measures and designs is added to an ML system, such distrust is perhaps only going to grow more intense and open.

While the users' sentiment of distrust is understandable, the need for good mental health is agnostic about one's feelings towards machine learning and usability of security designs. Therefore, to enable active usage and large-scale adoption[2] of secure ML systems in mental health cases, it is important to first induce users' trust in the ML systems used, before their active utilization of such ML systems. The trustworthiness requirement suggests *how ML systems in mental health may still earn users' trust, through its security and usability, by well-designed user interactions and communications.*

T1 *Clarity.* Articulating relevant security mechanisms, and their intents, impacts, and implications to users, is fundamental to trust-building. We identified three clarity aspects: (1)clarity of *ML*, where certain artifacts of the ML system's decision-making logic and process (e.g. summary statistics, explanations for classification labels) are exposed and explained to user in *non-technical* manners; (2)clarity of *security*, where user-facing security mechanisms (e.g. trusted path establishment, or revocation of access delegation), and these mechanisms' intents and purposes, are disclosed before users engage in these security mechanisms and take actions, preferably in *non-technical* terms; and (3)clarity of *failure modes*, where recovery (section 3.4) plan in case of security incidents, is summarized and communicated to users in *non-technical* terminology.

T2 *Constraints.* Complementary to **T1**, whose focus is on positive cases — i.e. what can be and is done — this requirement focuses mostly on negative cases. While providing clarity, ML systems need to draw boundaries and limitations on their capabilities and responsibilities, and then communicate such information. When determining the scope of usable security and communicating to users, we suggest three main factors: (1) limitations, emphasizing what the *system* cannot do (e.g. delegating access without explicit user actions from a trusted path), is not authorized to do (e.g. sharing chatbot history with unknown third parties), or unwilling to do (e.g. exposing ML models' features and parameters) for technical and non-technical reasons; (2) boundaries, concerning what the *user*'s actions cannot accomplish; and (3) expectations, dealing with

interactions between users and systems, on what users' expectations for the systems should *not* be. This requirement may seem counterintuitive, but it is founded on the "fail-safe" principle of computer security[14, 49]: the default situation is lack of access — that is, by default, actions and operations are constrained and not allowed to execute.

T3 *Consistency and Stability.* For similar user behaviors under similar contextual conditions, ideally, usability- and security-related experience and interactions should be: (1) similar, within fixed ML systems (data, algorithm, procedure, parameters, input), and (2) comparable, across different ML systems capable to cover the same contextual conditions in their use cases. We name it "consistency" property. Conversely, for the same usability and security methods, when provided with the same user behavior inputs, should respond with similar user experience and interaction. We call this "stability" property. There properties can help users build their own mental models for how security mechanisms and the general ML system work, and align their expectations with the system's responses.

Note that we controlled the variables ("similar", "fixed", "same") while describing consistency and stability, therefore consistency is not constancy, and stability is not staleness. In fact, the dynamic nature of usable security and the user expectation-system behavior alignment model are both well-known[24]. The goal of alignment, is to motivate secure user behavior and raise user's trust level in the system, and consistency and stability are inroads to alignment.

T4 *Reciprocity.* Leveraging the human tendency to return favors, ML systems in mental health can elicit actions of trust from users, and motivate their secure behaviors and active engagements, as HCI research showed[10, 19, 20]: after users receive helpful information from a computer system, they are more likely to provide useful information or actions back to the system. For reciprocity schemes in ML systems in mental health, we identify two stages: (1)*initial exchange* of reciprocity, where after volunteering helpful information to users, the system prompts user for desirable information or behavior input; and (2)*continuous engagement*, meaning that after the initial round, if the user reciprocates, the system should aim to maintain exchanges with users, when user behaviors and other contextual conditions warrant so. Depending on specific areas where the ML system needs to induce trust and motivate behaviors (e.g. having users enable security features, or actively use ML capacities), details of the interaction mechanisms, from the initial offer of help to ongoing engagement patterns, will vary.

Because reciprocity largely depends on user interactions with the system, it naturally focuses on usability, and has different trade-off with security for different context models. Therefore, any reciprocity schemes must be designed, implemented, and validated judiciously to defend against reciprocity attacks[70].

3.4 Recovery Principles

Good security needs failure modes, and usable security is no exception. With a variety of assets to protect **C1**, many functionalities to perform, and user trust to gain and maintain, ML systems in mental health must have a concrete plan for security failures. These principles lay out a foundation to consider *the immediate and long-term aftermath of security incidents and their responses*, so ML systems

in mental health can retain usable security attributes and rebuild trust with users (**T4**).

R1 *Response.* Previous research[28, 62] surveyed security incidents such as user data leaks, but did not address more complex security challenges to ML systems in mental health, whose sensitive and diverse assets, both native and acquired, make juicy targets. Therefore, ML systems must have protocols and procedures in place, timely reviewed and revised, and ready to respond to security incidents, to achieve three goals: (1) evaluate scope and impact of incident, (2) minimize damages to impacted assets, (3) investigate and attribute sources of incident, and most importantly, (4) rebuild trust in users for the system. (1) through (3) address immediate actions, while (4) is a long-term process that ensures ML systems can maintain its stay with user bases in mental health. This principle is related to **C1**, **C4**, and trustworthiness requirements.

R2 *Provenance and Chronology.* The usability of security, in its failure mode, entails that security failures can be traced, examined, analyzed, and inform future security decisions, and such need is satisfied by post-incident provenance and chronology. In ML systems for mental health, provenance and chronology should not only supply (1) a time ordering of system events, technical vulnerabilities or disadvantages, procedural limitations, uncovered edge cases, user interactions, statistics, and likely warning signals leading up to the incident, but also (2) records of any changes (e.g. content, metadata, mode, appearance) in impacted assets (e.g. manipulated ML model parameter, altered user interface, leaked health history), from when the incident *happened*, to when it is *uncovered*. Both provenance and chronology can be considered for user-facing purposes as a tool for repair (**R3**) and to rebuild trustworthiness.

R3 *Repair.* Post-security-incident repair has two aspects: (1)repairing the system itself, and (2)repairing users' trust in the system. (1) is the direct logical next step of **R1** and **R2** with immediate impact and results, while (2) tends to be long-term, and is more difficult — it needs all the building blocks of trustworthiness to repair users' trust in ML systems impacted by security incidents, especially when incidents concern user data, user-system interaction, or even users' offline behaviors. *Repairing* trust needs to address additional psychological barriers of users, hence harder than *building* trust at first, but it is still possible when **T2** and **T4** are emphasized and utilized in the repair process.

4 DISCUSSION

4.1 Form & Intent

Our framework is a suggestion, an encouragement, a proposal, and an invitation to the community to start acknowledging and researching usability and security in ML systems for mental health. While our framework is not a standardized rubric, we realize that it may become a foundation for future standards, guidelines, or recommendations by organizations such as NIST, ISO, or IEEE, for usable security in generic interactive ML systems, or specifically in mental health applications. Previously, standards were issued on transparency and autonomy in autonomous systems[52], and we are sanguine about a general consensus on usable security in ML systems, especially for mental health use cases.

4.2 Scope of Audience & Usage

We intentionally crafted this framework to be agnostic to ML techniques: hence, we can focus on providing a unified structure that is not only comprehensive enough to cover the current interdisciplinary area between traditional computer security, HCI, and ML, but is also flexible enough to accommodate future changes and progresses in these areas. We hope this framework can enable researchers and practitioners to:

(1) Identify gaps in security and usability between their theoretical capacities, design variances, actual implementations, and real-world usage patterns; and

(2) Quickly appraise properties of particular security and usability methods to decide on the most appropriate mechanism for their desired use cases.

In addition, our evaluation framework can be used as a reporting rubric targeting regulators, government officials, and policy makers, so they can quickly get all information in one place, in a clear, structured, and comparable manner.

## 4.3 A Different Kind of Usable Security

When we speak of "practitioners" in the section above, in the specific context of ML systems for mental health, there are broadly two categories that we target:

(1) Security practitioners: in general system security contexts, security mechanisms and policies are researched, designed, implemented, tested, maintained, and improved by security professionals.

(2) ML practitioners: in general ML system contexts, ML practitioners research, apply, curate, train, validate, test, maintain, improve ML models, algorithms, and date .

Yet, as we discuss usable security in *ML systems for mental health*, the matter gets more complex: there are more stakeholders, both on the system builders' side, and on the system users' side. And on each side, there are multiple considerations, interests, and mental models that come into play. Table 1 below shows the different stakeholders when we build security to be usable into ML systems for mental health. Comparing it with Figure 1, the critical differences between the building usable security into general system versus into ML systems for mental health can be clearly discerned. To summarize: there are more stakeholders on the users' side who deserve usable security for their more diverse needs of the ML system for mental health, and there are more stakeholders on the builders' side who have distinct desires for what they want do with, and how they wish such system to behave.

For example, while security and ML practitioners desire different ideal attributes from the system and those attributes are not necessarily at odds or contradict with each other, there are trade-offs to make. Between "strong defense" with implications for privacy on patient information and "collect data" for training when in general, more data is usually better, the builders within themselves need to reach a delicate balance first. On the other end, instead of the cohesive and more-or-less predictable and uniform sets of actions normally expected from user models built for general software or ML systems, we now have a diverse set of potential users, with various sets of actions and behaviors that are not usually taken into

Figure 1: From security practitioners to end-users: how security mechanisms and experiences are built and delivered in general software systems
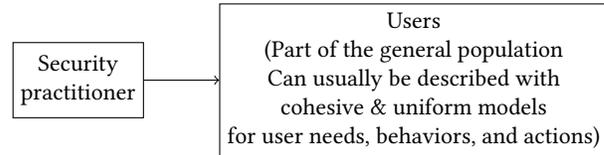
account for in those general purpose software or ML systems. In those general purpose systems, Figure 1 shows the path of how security mechanism and experience are delivered to end users. Behaviors, actions, expectations, and use scenarios of these end users would be captured in user models, and security practitioners would design, build, and deploy security measures and experiences according to those user models. But because of the diverse and varying expectations and actions from distinct groups of end users[1], such user models would be too narrow and missing out on legitimate use actions and behaviors. This is a major reason that we crafted this framework: to properly account for and appreciate the diversity and variety of users and their actions in ML systems for mental health, with the end goal to bring a usable and secure experience to all.

4.4 Dynamic Relationships

As described within sections for each of those sub-attributes, those attributes are not mutually exclusive nor completely independent from each other. Instead, there are rich and dynamic interactions between these sub-attributes, both within a single pillar and across different pillars. Four major types of interactions are list below with short examples.

(1) Inter-dependence: **F5** and **C3** are interdependent. In this case, without behavior categorization, robustness is next to impossible to plan for or implement; and without robustness measures tested and used in real-life, it would be very hard to validate if the behavior categorizations are reasonable or sufficient.

(2) Trade-offs: **F2** is a trade-off to the "least common mechanism" principle for secure system design as articulated in [49]: for security measures to generalize to diverse sets of targeted users and behaviors, commonality increases and distinctions decline.

(3) Prioritization: In many scenarios, prioritizing particular principles before others is the most reasonable and sensible course of action. For example, a large-scale online platform delivering automated conversational therapy may prioritize **F2** and **F5**, at the same time de-prioritize trustworthiness requirements based on the assumption that people seeking online automated services generally have greater trust in

---

[1]Many ML systems for mental health involve more than one group of stakeholders as shown in Table 1. In some cases, it might even be possible for one single ML system for mental health to encounter all the four groups of stakeholders. For example, an ML system analyzing facial and verbal expressions in online therapy sessions: the patient and the provider conduct sessions, another caregiver review the analysis after session ends to provide better care, and policy-makers may monitor some sessions for signals of large-scale mental health intervention policies.

Table 1: A sample of stakeholders and their potential needs and purposes when building and using ML systems for mental health. Left of "||": builder. Right of "||": potential user

Security practition-ers	ML practitioners	Patients	Providers	Other caregivers	Policy-makers
Strong defense	Collect data	Get treatment	Diagnose patients	Use by self	Large-scale monitoring
Up-time guarantee	Monitor & improve models	Get peer support	Treat patients	Use on behalf of patients	Decision-making (e.g. intervention)
Easy rollback	Can validate & test	Self-monitor	Collaborate with other providers	Assist patients to use	Regulate
East upgrade	Models effective for end-users	Delegate use to other caregivers	Keep records	Monitor patient status	Audit
Good maintenance & recovery	Deployed model not corrupted	...	...	...	...
...	...	...	...	...	...

ML systems and technology, and are likely to be technically proficient enough to navigate security designs built in place.

(4) Complements: **T1** and **T2** are complementary: they consider opposite sides of the same issue, and from there, create a comprehensive view and enables balanced and holistic decisions for usable security designs and implementations.

These dynamic and interactive relationships carry deep implications for usable security in ML systems for mental health, and we will explore some examples that showcase these interactive and dynamic relationships between the properties in section 5.

5 SOME EXAMPLES

We will now apply the four pillared framework, and share several tangible use cases of ML systems for mental health where we evaluate and examine their usable security needs and profiles. This way, we can concretely demonstrate the practicality of our framework, illustrate the dynamic and interactive relationships between the pillars and their corresponding sub-attributes, and showcase the complex and distinct stakeholder demands for usable security that warrant such a framework. We will elaborate on example 1 with brief comparisons and contrasts to the three other examples, and leave examples 2 to 4 for readers' exercise.

(1) Chatbot providing conversation-based therapy to young adults with mental disorders

(2) Auto-diagnosis algorithms of neuro-images for psychiatrists

(3) Personalized matching for providers & patients

(4) ML system analyzing facial and verbal expressions during tele-therapy sessions

In 1, the chatbot's context is an online automated services, and hence is more likely to experience high concurrency requests from many different users with existing mental disorders, and such online large-scale services may also be needed in times of distress (e.g. quarantine during COVID-19 global pandemic) to parts of the general population. Therefore, its usable security mechanisms need to

prioritize being *available* and *robust* enough to handle more users, and wider ranges behaviors and actions of legitimate users. At the same time, inducing users' *trust* may not be as important, because we may consider people willing to use online chatbot services are more trusting towards ML and technology in general. Although, soon we would see that if security mechanisms and designs are not usable or robust enough, such assumption of trust may not be warranted, and if there were any, would be drastically diminished.

The assets it needs to protect are not only the security of its general software infrastructure, but also its ML algorithms that generate live conversation responses to users: recall the infamous incident of Microsoft's chatbot Tay on Twitter[66], caution must be taken to ensure that legitimate users who need the therapy service could easily and readily access it without much hassle, and that malicious users could be fended off so they could not manipulate the algorithms. While this may seem simple at first, we must properly *categorize behaviors* of our potential legitimate users, who have existing mental disorders. Take attention-deficit/hyperactivity disorder (ADHD) for instance. Suppose that the chatbot implements classes of CAPTCHA or reCAPTCHA methods — which may include text, image, and sound recognition, as well as text and image matching — to defend against its *threat model* actors that include bots and malicious users trying to poison its algorithms. While these methods may be effective to defend against these threats, legitimate users with ADHD, whose attention spans are usually shorter than the general population[34], may be unlikely to complete the CAPTCHAs, especially when there are several ones that come one after another.

When security defense designs turn legitimate users away, these users may leave with the idea that such mechanisms built to trick them, and the system behind it has no genuine intention to provide them help, and hence their implied trust in the chatbot when they first approached this online automated service, may likely diminish. Hence, it would be advisable to also take *trustworthiness* requirement seriously, especially the *constraint* property. One way to demonstrate it, is using clear language or visual images to inform

users of failures. For instance, when a user's attention span is too short to successfully finish a series of reCAPTCHA challenges, a message displays: "Sorry we could not tell if you are a human or bot. Do you want to try another way?" This or similar messages could communicate to legitimate users that the system genuinely intends to provide services, the reCAPTCHAs are there because it is a security design, not a farce or trick to turn them away, and there are certain things that these reCAPTCHAs are not capable of doing.

Further thoughts bring more usable security considerations to the discussion on example 1. Should the chatbot store any records of its user interactions, so that human providers and caregivers (e.g. parents or legal guardians) of these young adults could monitor their progress, provide better diagnosis, treatment, and care, then we are onto more complex scenarios. There are now two more groups of users to consider, and how to provide usable security for them is a crucial challenge. Moreover, because now there are stored user records, there is an additional *asset* to protect, and the *recovery* principles need to be elevated to higher priorities, especially *repairing* users' trust in the service in the event of a breach or leak. This is where *flexibility* also comes into the picture: if builders of the chatbot had not originally considered this sharing services, and only later decided to add it, *flexibility* of previous security capabilities to accommodate the additional security requirements that come with sharing user records, is extremely important.

In a base case, even when the chatbot is originally built to not only converse with patients, but also store their records and allows them to share their records with their providers and caregivers, *clarity* of usable security designs that are informed by *behavior categorization* would be integral. For instance, some young adult patients may decide to simply share their passwords with their providers and caretakers for the latter groups to look at their chat records. However, if these patients are in the U.S., this simple act may land their providers and caretakers in legal trouble: because a shared password, even a voluntarily shared one, counts as unauthorized access by the Computer Fraud and Abuse Act[13]. Preempting such behaviors would greatly inform usable security decisions when designing and building this chatbot. For example, builders may decide to use methods other than passwords to check for user authorization and authentication status; to utilize the *constraint* criteria, and outwardly warn users to not share their passwords even with trusted providers and caregivers; or to add particular terms in the end-user licensing agreement, security & privacy policies, or terms & services documents, and specify cases where the patient could share passwords; or to build a security measure so that patients can delegate access to their records to authenticated providers and caregivers. To choose the most suitable usable security mechanisms or combinations of such mechanisms, builders of the chatbot would need to deliberate on which *contexts*, and especially which *threat models* they decide to focus on.

In comparison, example 2 has a very specific and focused *profile of targeted users* (psychiatrists), so the inter-dependent properties of *behavior categorization* and *robustness* would be straightforward to analyze and design, and the *assets* and *threat models* are also relatively clearly defined and direct. Meanwhile, because such system is used for medical diagnosis, all attributes related to *trustworthiness*

need to be *prioritized*: the builders cannot assume that psychiatrists are trusting the ML system's decisions. Moreover, while the *threat models* are relatively simple compared to example 1 and 4, the system still needs to induce trust from the psychiatrists: how do they know it is *them*, instead of *malicious attackers* described in the threat models, who see the images, patients' information, and the algorithm outputs? To address these issues, some security designs may include: ML explanation options that accompany each diagnosis; a side bar that shows access activities; or a device-based two-factor authentication check. Again, it is up to both the ML and security practitioners who build this system, to decide on the specific *contexts* and *threats* they would like to prioritize.

Example 3 also has a straightforward profile for *behaviors* as example 2, but the *threat models* could be tricky: because depending on what the builders of the system decide to gather from both the patients and providers for the match, *assets* that the system needs to protect could swing a rather wide range. Meanwhile, because of the usual one-on-one nature of patient-provider relationships, in contrast to example 1, de-prioritizing the *availability* of security measures to large numbers of online users could be sensible, and the system might even be able to afford using more time-, memory-, or computationally-*complex* security mechanisms that are nonetheless usable for both providers and patients of the service. For example, incorporating *reciprocity* into the human-system interaction process, by engaging users in short Q&A games about secure behaviors — which by the way, could also induces trust from users about the system's security, and fulfill part of the *trustworthiness* requirement. But on the ML front, *trustworthiness* here is similar to the premise of 1 but the spirit of example 2. While patients and providers who choose to use a ML-powered matching service could be assumed to have a greater degree of general trust in ML and technology, the same level of trust could not be assumed in the specific matching decisions: "How and why did this black-box know that I would be a good fit for this patient/provider?" would be the question to answer for every patient or provider who uses the service.

Example 4 involves more diverse user groups (patients, providers, likely other caregivers, and potentially policy-makers). Hence, the *assets* need to be protected are more varied and diverse, the *threat models* more complex, the *recovery* scenarios more important, and the usable security mechanisms may need to make trade-offs between *availability* and *flexibility* while still being *functional* when processing live audio and video data, which is another subtle constraint on the *complexity* of usable security mechanisms. Similar to examples 2 and 3, the inevitable question of *trustworthiness* would arise on the ML system's decision rationales and explanations, and there is also an incentive on the builders' end to assure that algorithms and models powering the system are not being manipulated. Because the information being processed and analyzed by the system is largely private and sensitive, convincing different users of the effectiveness and strength of the security mechanisms is also an important task. Comparing it to example 1 where there are clearer priorities, this system poses a set of full-on challenge for usable security design, implementations, and evaluations.

6 CONCLUSION AND FUTURE WORK

In our work, we presented four categories of desired properties — based on context, functionality, trustworthiness, and recovery — to systematically frame and evaluate usable security in ML system for mental health. We discussed those properties' intents, rationales, and sources in the intersection of security, usability, ML, and mental health. We propose that ML systems in mental health be evaluated by the way of this framework for security and usability, in different phases of the computer system life cycle.

We have analyzed, structured, and presented several examples of ML systems in mental health in this framework, and for next steps, we plan to evaluate more real-life ML systems in mental health, preferably similar to the four described examples, so we can test, validate, and improve our framework and criteria. Simultaneously, we also plan to interview builders of these ML systems in mental health, to understand their awareness of, thought processes behind, and decision rationales of usable security in the systems they designed and built. Because the framework covers the computer system life cycle, while we prefer already deployed, large-scale systems, we are also happy to examine systems in early stages of the cycle. We plan to publish results on websites where this interdisciplinary community can also submit their own framework evaluation results.

In a deeper dive, our future work will explore a tiered approach to usable security for ML systems, inspired by classic security literature[37], meanwhile further examine interactions — e.g. trade-offs, enhancements, overlaps from different perspectives, complements, and interdependence — between desirable usability and security properties for ML systems in mental health.

REFERENCES

[1] 1990. IEEE Standard Glossary of Software Engineering Terminology. *IEEE Std 610.12-1990* (1990), 1–84.
[2] 2017. Inclusive persuasion for security software adoption. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*.
[3] Anne Adams and Martina Angela Sasse. 1999. Users are not the enemy. *Commun. ACM* 42, 12 (1999).
[4] Eirik Albrechtsen and Jan Hovden. 2010. Improving information security awareness and behaviour through dialogue, participation and collective reflection. An intervention study. *Computers & Security* 29, 4 (2010).
[5] Ammar Amran, Zarul Fitri Zaaba, Manmeet Mahinderjit Singh, and Abdalla Wasef Marashdih. 2017. Usable security: Revealing end-users comprehensions on security warnings. *Procedia Computer Science* 124 (2017).
[6] Jack W Baker, Matthias Schubert, and Michael H Faber. 2008. On the assessment of robustness. *Structural Safety* 30, 3 (2008).
[7] Dirk Balfanz, Glenn Durfee, Diana K Smetters, and Rebecca E Grinter. 2004. In search of usable security: Five lessons from the field. *IEEE Security & Privacy* 2, 5 (2004).
[8] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. 2019. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705* (2019).
[9] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*. IEEE.
[10] Sonia Chiasson, PC van Oorschot, and Robert Biddle. 2007. Even experts deserve usable security: Design guidelines for security management systems. In *SOUPS Workshop on Usable IT Security Management (USM)*. Citeseer.
[11] European Commission. 2016. *Overview of the national laws on electronic health records in the EU Member States*. https://ec.europa.eu/health/ehealth/projects/nationallaws_electronichealthrecords_en
[12] European Commission. 2019. *Data protection in the EU*. https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en
[13] Raymundo Cornejo, Robin Brewer, Caroline Edasis, and Anne Marie Piper. 2016. Vulnerability, sharing, and privacy: Analyzing art therapy for older adults with dementia. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. 1572–1583.

[14] Cybersecurity and United States Infrastructure Security Agency (CISA), Department of Homeland Security. 2013. *Failing Securely*. https://www.us-cert.gov/bsi/articles/knowledge/principles/failing-securely#footnote1_lppmt3d
[15] Alexander J DeWitt and Jasna Kuljis. 2006. Is usable security an oxymoron? *interactions* 13, 3 (2006), 41–44.
[16] Diego Didona, Francesco Quaglia, Paolo Romano, and Ennio Torre. 2015. Enhancing performance prediction robustness by combining analytical modeling and machine learning. In *Proceedings of the 6th ACM/SPEC international conference on performance engineering*. 145–156.
[17] Microsoft Security Engineering. [n.d.]. *Microsoft Security Development Lifecycle (SDL)*. https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling
[18] Jean-Claude Fernandez, Laurent Mounier, and Cyril Pachon. 2005. A model-based approach for robustness testing. In *IFIP International Conference on Testing of Communicating Systems*. Springer, 333–348.
[19] BJ Fogg and Clifford Nass. 1997. How users reciprocate to computers: an experiment that demonstrates behavior change. In *CHI'97 extended abstracts on Human factors in computing systems*.
[20] Brian J Fogg. 2002. Persuasive technology: using computers to change what we think and do. *Ubiquity* 2002 (2002).
[21] European Union Agency for Cybersecurity. 2008. *ISO/IEC Standard 15408*. https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/laws-regulation/rm-ra-standards/iso-iec-standard-15408
[22] International Organization for Standardization. 2016. *ISO/IEC 11889-1:2015 Information technology — Trusted platform module library — Part 1: Architecture*. https://www.iso.org/standard/66510.html
[23] International Organization for Standardization. 2019. *ISO 9241-210:2019 Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*. https://www.iso.org/standard/77520.html
[24] Simson Garfinkel, Gene Spafford, and Alan Schwartz. 2003. *Practical UNIX and Internet security*. " O'Reilly Media, Inc.". 6 pages.
[25] Janne Merete Hagen and Eirik Albrechtsen. 2009. Effects on employees' information security abilities by e-learning. *Information Management & Computer Security* (2009).
[26] Luigi Lo Iacono, Matthew Smith, Emanuel von Zezschwitz, Peter Leo Gorski, and Peter Nehren. 2018. Consolidating Principles and Patterns for Human-centred Usable Security Research and Development. In *European Workshop on Usable Security, London*.
[27] Ronald Kainda, Ivan Flechais, and AW Roscoe. 2010. Security and usability: Analysis and evaluation. In *2010 International Conference on Availability, Reliability and Security*. IEEE.
[28] Sowmya Karunakaran, Kurt Thomas, Elie Bursztein, and Oxana Comanescu. 2018. Data Breaches: User Comprehension, Expectations, and Concerns with Handling Exposed Data. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*.
[29] Kat Krol, Jonathan M Spring, Simon Parkin, and M Angela Sasse. 2016. Towards robust experimental design for user studies in security and privacy. In *The {LASER} Workshop: Learning from Authoritative Security Experiment Results ({LASER} 2016)*.
[30] Carl E Landwehr. 1981. Formal models for computer security. *ACM Computing Surveys (CSUR)* 13, 3 (1981), 247–278.
[31] Carl E Landwehr. 2001. Computer security. *International Journal of Information Security* 1, 1 (2001), 3–13.
[32] Reham Ebada Mohamed and Sonia Chiasson. 2018. Online Privacy and Aging of Digital Artifacts. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*.
[33] Suvda Myagmar, Adam J Lee, and William Yurcik. 2005. Threat modeling as a basis for security requirements. In *Symposium on requirements engineering for information security (SREIS)*, Vol. 2005. Citeseer.
[34] NIH National Institute of Mental Health. 2019. *Attention-Deficit/Hyperactivity Disorder*. https://www.nimh.nih.gov/health/topics/attention-deficit-hyperactivity-disorder-adhd/index.shtml
[35] Guillermo Navarro and Simon N Foley. 2005. Approximating SAML using similarity based imprecision. In *International Conference on Intelligence in Communication Systems*. Springer, 191–200.
[36] NIST. [n.d.]. *Network security & robustness*. https://www.nist.gov/topics/network-security-robustness
[37] Department of Defense. 1985. *Trusted Computer System Evaluation Criteria*. https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/early-cs-papers/dod85.pdf
[38] United States Department of Health and Human Services. 2014. *HIPAA Security Rule Crosswalk to NIST Cybersecurity Framework*. https://www.hhs.gov/sites/default/files/nist-csf-to-hipaa-security-rule-crosswalk-02-22-2016-final.pdf
[39] The Department of Justice Systems Development Life Cycle Guidance Document. 2003. *THE SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)*. https://www.justice.gov/archive/jmd/irm/lifecycle/table.htm

[40] National Institute of Standards and Technology. [n.d.]. *THE SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)*. https://csrc.nist.gov/csrc/media/publications/shared/documents/itl-bulletin/itlbul2009-04.pdf

[41] National Institute of Standards and Technology. 2017. *Usable Cybersecurity*. https://csrc.nist.gov/Projects/Usable-Cybersecurity

[42] National Institute of Standards and Technology. 2017. *Usable Cybersecurity: Behavior*. https://csrc.nist.gov/Topics/Security-and-Privacy/security-and-behavior/behavior

[43] National Institute of Standards and United States Technology. 2014. *Framework for Improving Critical Infrastructure Cybersecurity*. https://www.nist.gov/system/files/documents/cyberframework/cybersecurity-framework-021214.pdf

[44] Andrew S Patrick, A Chris Long, and Scott Flinn. 2003. HCI and security systems. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*.

[45] Bryan D Payne and W Keith Edwards. 2008. A brief introduction to usable security. *IEEE Internet Computing* 12, 3 (2008), 13–21.

[46] Parag C Pendharkar, James A Rodger, and Girish H Subramanian. 2008. An empirical study of the Cobb–Douglas production function properties of software development effort. *Information and Software Technology* 50, 12 (2008), 1181–1188.

[47] Lucy Qin, Andrei Lapets, Frederick Jansen, Peter Flockhart, Kinan Dak Albab, Ira Globus-Harris, Shannon Roberts, and Mayank Varia. 2019. From Usability to Secure Computing and Back Again. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*.

[48] Chris Salter, O Sami Saydjari, Bruce Schneier, and Jim Wallner. 1998. Toward a secure system engineering methodolgy. In *Proceedings of the 1998 workshop on New security paradigms*.

[49] Jerome H Saltzer and Michael D Schroeder. 1975. The protection of information in computer systems. *Proc. IEEE* (1975).

[50] Martina Angela Sasse, Sacha Brostoff, and Dirk Weirich. 2001. Transforming the 'weakest link'—a human/computer interaction approach to usable and effective security. *BT technology journal* 19, 3 (2001).

[51] M. Schaefer. 2004. If A1 is the answer, what was the question? An Edgy Naif's retrospective on promulgating the trusted computer systems evaluation criteria. In *20th Annual Computer Security Applications Conference*. 204–228.

[52] Kyarash Shahriari and Mana Shahriari. 2017. IEEE standard review—Ethically aligned design: A vision for prioritizing human wellbeing with artificial intelligence and autonomous systems. In *2017 IEEE Canada International Humanitarian Technology Conference (IHTC)*. IEEE, 197–201.

[53] Adrian BR Shatte, Delyse M Hutchinson, and Samantha J Teague. 2019. Machine learning in mental health: a scoping review of methods and applications. *Psychological medicine* 49, 9 (2019), 1426–1448.

[54] Steve Sheng, Levi Broderick, Colleen Alison Koranda, and Jeremy J Hyland. 2006. Why johnny still can't encrypt: evaluating the usability of email encryption software. In *Symposium On Usable Privacy and Security*. ACM.

[55] Adam Shostack. 2014. *Threat modeling: Designing for security*. John Wiley & Sons.

[56] D Smetters. 2007. Usable security: Oxymoron or challenge.

[57] Kacper Sokol and Peter Flach. 2020. Explainability Fact Sheets: A Framework for Systematic Assessment of Explainable Approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT* '20)*. https://doi.org/10.1145/3351095.3372870

[58] Andreas Sotirakopoulos, Kirstie Hawkey, and Konstantin Beznosov. 2011. On the challenges in usable security lab studies: lessons learned from replicating a study on SSL warnings. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*.

[59] Gerald Jay Sussman. 2007. Building robust systems an essay. *Citeseer* 113 (2007), 1324.

[60] Marian Sweeney, Martin Maguire, and Brian Shackel. 1993. Evaluating user-computer interaction: a framework. (1993).

[61] David Tcheng, Bruce Lambert, Stephen CY Lu, and Larry Rendell. 1989. Building robust learning systems by combining induction and optimization. *Urbana* 100 (1989), 61801.

[62] Mary Theofanos. 2015. *Usable security*. https://csrc.nist.gov/CSRC/media/Presentations/Usable-Security/images-media/day2_research_430-530pt1.pdf

[63] Mary Theofanos. 2020. Is Usable Security an Oxymoron? *Computer* 53, 2 (2020).

[64] United States Department of Health and Human Services. 2017. Health Information Privacy, HIPPA for Professionals. https://www.hhs.gov/hipaa/for-professionals/security/guidance/cybersecurity/index.html.

[65] USENIX. 2015. *USENIX SOUPS 2020 conference page*. https://www.usenix.org/conference/soups2019

[66] Daniel Victor. 2016. *Microsoft Created a Twitter Bot to Learn From Users. It Quickly Became a Racist Jerk*. https://www.nytimes.com/2016/03/25/technology/microsoft-created-a-twitter-bot-to-learn-from-users-it-quickly-became-a-racist-jerk.html

[67] Willis Ware. 1970. *Security Controls for Computer Systems: Report of Defense Science Board Task Force on Computer Security*. https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/early-cs-papers/ware70.pdf

[68] Alma Whitten and J Doug Tygar. [n.d.]. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0.

[69] Ka-Ping Yee. 2004. Aligning security and usability. *IEEE Security & Privacy* 2, 5 (2004).

[70] Feng Zhu, Sandra Carpenter, Ajinkya Kulkarni, and Swapna Kolimi. 2011. Reciprocity attacks. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*.

[71] Mary Ellen Zurko and Richard T. Simon. 1996. User-Centered Security. In *Proceedings of the 1996 Workshop on New Security Paradigms (NSPW '96)*. Association for Computing Machinery.