Building Guardrails in AI Systems with Threat Modeling

JAYATI DEV, Comcast Corporation, Philadelphia, United States
NURAY BALTACI AKHUSEYINOGLU, Comcast Corporation, Philadelphia, United States
GOLAM KAYAS, Comcast Corporation, Philadelphia, United States
BAHMAN RASHIDI, Comcast Corporation, Philadelphia, United States
VAIBHAV GARG, Comcast Corporation, Philadelphia, United States

Much like cars, AI technologies must undergo rigorous testing to ensure their safety and reliability. However, just as a 16-wheel truck's brakes are different from that of a standard hatchback, AI models too may need distinct analyses based on their risk, size, application domain, and other factors. Prior research has attempted to do this, by identifying areas of concern for AI/ML applications and tools needed to simulate the effect of adversarial actors. However, currently, a variety of frameworks exist which poses challenges due to inconsistent terminology, focus, complexity, and interoperability issues, hindering effective threat discovery. In this article, we present a meta-analysis of 14 AI threat modeling frameworks, providing a streamlined set of questions for AI/ML threat analysis. We then review this library, incorporating feedback from 10 experts to refine the questions. This refined set of questions allow practitioners to seamlessly integrate threat analysis for comprehensive manual evaluation of a wide range of AI/ML applications.

CCS Concepts: • **Security and privacy**; • **Computing methodologies** → **Machine learning**; **Artificial intelligence**;

Additional Key Words and Phrases: Threat modeling, security, privacy, artificial intelligence, AI, machine learning

ACM Reference Format:
Jayati Dev, Nuray Baltaci Akhuseyinoglu, Golam Kayas, Bahman Rashidi, and Vaibhav Garg. 2025. Building Guardrails in AI Systems with Threat Modeling. *Digit. Gov. Res. Pract.* 6, 1, Article 15 (February 2025), 18 pages. https://doi.org/10.1145/3674845

1 Introduction

The **Artificial Intelligence** (**AI**) market has grown exponentially over the past decade, with new research finding applications of **Machine Learning** (**ML**) algorithms in a variety of use cases - from recommending movies to providing healthcare solutions. However, as with all applications, AI-driven systems come with security threats and privacy risks. Some of these threats are unique to AI/ML systems like CVE-2019-20634 [2], which replicated an ML-based email classification system built to subvert the original system. AI's threat landscape is complicated by the fact that the outputs of emerging AI models is increasingly hard to explain [37] and it is unclear whether mitigations against some classes of threats are even available [24].

Authors' Contact Information: Jayati Dev, Comcast Corporation, Philadelphia, Pennsylvania, United States; e-mail: devjayati@outlook.com; Nuray Baltaci Akhuseyinoglu, Comcast Corporation, Philadelphia, Pennsylvania, United States; e-mail: nuray_baltaciakhuseyinoglu@comcast.com; Golam Kayas, Comcast Corporation, Philadelphia, Pennsylvania, United States; e-mail: golam_kayas@comcast.com; Bahman Rashidi, Comcast Corporation, Philadelphia, Pennsylvania, United States; e-mail: Bahman_Rashidi@comcast.com; Vaibhav Garg, Comcast Corporation, Philadelphia, Pennsylvania, United States; e-mail: Vaibhav_Garg@comcast.com.

ACM 2639-0175/2025/02-ART15
https://doi.org/10.1145/3674845

To address this, regulators are pushing for increased legal requirements to understand and mitigate such threats in AI applications, including efforts to codify AI protections into legislation [34]. For instance, the **European Union** (**EU**) has proposed the EU AI Act [32] to create legal guidelines for AI applications, including cybersecurity. Government agencies are also holding generative AI companies, like OpenAI, accountable [16] to enforce mitigation against the unique security threats from their generative AI tools. NIST has published a taxonomy of threats, and associated defenses for AI [31], while the **Organization for Economic Co-operation and Development** (**OECD**) curated an AI Vulnerability Database in 2022 [3].

Thus, there is a compelling need for organizations to build guardrails into both first-party and third-party AI systems to address technical risks as well as fulfill regulatory or compliance obligations. Unfortunately, many organizations may lack the resources and the subject matter expertise to do just that [26]. There are several threat lists, standards, and frameworks researchers and practitioners have been working toward to evaluate security threat posture when building an AI/ML system. However, the variety in these frameworks has been great - with differences in the types of threats covered, the scope of such threats, terminology, specificity, and classification. There is a lack of a standard mechanism for security threat discovery that can differentiate between what can be asked during a threat modeling session versus what can be tested by a red team while being comprehensive enough to cover *AI-specific* security controls.

We address this gap in this article by synthesizing 14 diverse frameworks with different scopes into one holistic AI threat controls library called Project GuardRail.[1] This library can be leveraged as a self service tool at different points in the AI development lifecycle to gain insights into the attack surface offered by a single AI model, a full AI product, or a targeted AI use case. This library helps highlight the risk exposure of your model by assessing what underlying model the application has. For example, if it is a persisted model and the data is well curated for model training, there are potentially fewer challenges with poisoning. Otherwise, data poisoning controls need to be in place over time for active learning systems. Project Guardrail helps assess the risk exposure of your AI solution by looking at the type of model, the risk of the use case, and the sensitivity of data, among others. Overall, this article makes the following contributions:

(1) We conduct an extensive literature review to narrow our search down to 14 relevant AI threat resources, and highlight areas of improvement in these works that can be addressed by a building a new, comprehensive library as we have done in the article.
(2) We then extract the list of threats in these frameworks to create a modular unified library of 63 security controls that are uniquely applicable to AI/ML systems that will guide a developer to identify potential threats caused to and by an AI application.
(3) We then interview experts on the usability of these questions, making recommended changes to improve them.

Note that our goal is specific to improving the overall security posture of AI/ML applications. We limit our scope to AI/ML threat modeling questions that can be answered by developers rather than those which require extensive adversarial testing. We also limit our focus on security threats and privacy risks to an application or its users, and do not include Offensive Machine Learning - where ML algorithms are used for malicious purposes, like phishing attacks.

The rest of the article is organized as follows. In Section 2, we discuss essential background on AI/ML security. We then talk about sources we reviewed in Section 3, followed by the process of development of the threat library in Section 4, and evaluation by expert reviewers in Section 5. We conclude the article with discussions and limitations in Section 6.

[1] https://github.com/Comcast/ProjectGuardRail

2 Related Work

There is a well-known phonemena of *overtrust* to automated systems by users. Tackling with this problem requires special approaches to securing AI systems compared to traditional software systems due to three reasons [24]. First, tests or standards for AI systems are *not comprehensive* due to the issue of not one size fits all. One standard for testing an AI system may not cover others. Second, tests for AI systems are *not concrete and exhaustive* as for traditional systems. Testing is concrete for traditional systems because a failure is clear, such as burning by a toaster or an oven. Third, testing for AI systems are *not at constituent-level*, but done in an abstract/system-level. It is challenging to test individual parts of AI systems. Also, AI/ML models encounter distinct vulnerabilities that traditional software systems do not [29].

Given the challenges of securing AI systems and threats uniquely applying to them, it is essential to pay special attention to security threat assessment for AI systems. In this section, we review related literature on AI/ML models and unique threats that apply to them.

2.1 AI/ML Models

Before we delve into the threats from AI/ML models, we first need to understand how they are different from traditional applications, and how they work. The concept of AI originates from the 1940s and aligns with the advancements made in the field of the first electronic computers [5]. AI refers to the development of computer systems or software capable of performing tasks that typically require human intelligence [5], such as learning, reasoning, problem-solving, understanding natural language, speech recognition, and visual perception. AI is a broad domain comprising many different fields. A comprehensive review of AI techniques is presented by Sarker [35]. To provide an overview, they organized potential AI techniques under ten categories, including but not limited to ***Machine Learning (ML)***, neural networks and deep learning, data mining/analytics and knowledge discovery, rule-based modeling, and text mining and **natural language processing** (**NLP**). ***Generative AI (GAI)*** is another subfield of AI that has gained much attention in recent years. In the following, we present a brief overview of ML and GAI.

Underlying the core functionality of AI, ML algorithms facilitate enhanced decision-making procedures [29, 35]. ML systems are different than the traditional, rule-based approach of Handcrafted Knowledge Systems [5]. While the latter relies on manually inputting rules, ML systems learn from data through algorithms on training datasets, resulting in AI models. In ML, knowledge is not programmed but learned, making the process more autonomous and data-driven. In addition to learning patterns from the historical data, ML models also make predictions about unknowns on the new data using the trained predictive model (learned patterns) [35]. Based on the learning principles, capabilities, and the nature of the data they are trained on, ML models can be classified as *supervised*, *unsupervised*, *semisupervised*, and *reinforcement* learning [5, 35].

Supervised learning relies on labeled data to train algorithms for specific tasks such as classification and regression [5, 35]. Various techniques like Navies Bayes, K-nearest neighbors, Decision Trees, Ensemble learning, and Linear regression are used based on the nature of the data [35]. For instance, while classification models are beneficial for detecting cyber-attacks, regression models are effective for cyber-crime trend analysis. Despite its high performance, obtaining an adequate amount of labeled data for supervised learning can be challenging and costly[5].

Unsupervised learning works with unlabeled data to reveal patterns or structures using techniques like clustering, visualization, and anomaly detection [35]. Clustering techniques such as K-means, DBSCAN, hierarchical clustering, and association learning algorithms like Apriori and FP-Tree are commonly employed in unsupervised tasks based on the nature of the data [35]. For instance, clustering models can aid in customer segmentation problem. Despite having lower performance compared to supervised learning in some applications, unsupervised learning serves as a valuable approach for problems where labeled data is not feasible or available [5].

GAI represents an evolution within the broader field of AI, going beyond traditional ML by not only discriminating information but also generating novel content [13, 19]. In this relationship, ML serves as the foundation, uncovering patterns and providing key insights. GAI leverages the learned patterns by ML and other AI models [12], and builds on them to generate unique content and refine its own outputs. For instance, GAI technologies may leverage NLP, deep neural networks, and reinforcement learning [13].

2.2 AI/ML Threats

AI/ML systems can be subject to threats much different from traditional software systems (and on top of these usual security threats) [29]. Such threats may not be new; however, their manifestation in AI applications may be unique and distinct from non-AI applications. Let us look at some examples of such AI/ML threats that can be identified in a sample threat modeling session. This is not an exhaustive list, but a sample set of attacks discussed in the AI threat modeling frameworks we reviewed.

- *Confidentiality attacks to ML training data* [17]: This attack type refers to disclosure of sensitive data in the training set, such as those related to health, finance, or protected by law. This attack typically happens due to the *third party exposure* of the data, which could then possibly be recovered by adversaries.
- *Sensitive data leakage from output* [30]: In attacks such as CodexLeaks, prompts are engineered so that language prediction models can be tricked into leaking sensitive data used to train them. This is done by manipulating prompts in a way that the model output discloses data.
- *AI auto-hack/Reward hacking* [9, 25]: In this attack, reinforcement learning systems behave unexpectedly due to a discrepancy between the intended reward and the actual reward they receive, i.e., they "hack" the reward. This can occur when an AI system autonomously makes decisions without human involvement.
- *Model stealing/Proxy ML models* [15, 28]: Adversaries mimic the victim organization's model through representative datasets, victim inference APIs, or pre-trained models, allowing offline access and potential replication.
- *Reverse engineering* [15, 28]: This type of attack can occur if adversaries can query or interact with an ML model to discover model functionality or disclose training data.

Apart from this sample, new threats are being discovered, some of which are increasingly difficult to patch. Companies are looking at different approaches to reduce the cybersecurity risk from AI/ML applications, not only from a regulatory standpoint but also technical controls that can be determined as required early in the process. One such approach increasingly being used in the industry is threat modeling.

2.3 Threat Modeling

Threat modeling is the process of examining system representations to uncover security and privacy issues, encompassing four fundamental considerations: project focus/scope, potential risks, mitigation strategies, and overall effectiveness assessment [11]. Threat modeling literature includes an abundance of methodologies, frameworks, practical tools, and approaches to analyze and address security and privacy concerns in different systems. Yet, threat modeling in AI is a domain of increasing interest. Next, we present our review of AI threat modeling literature.

STRIDE [21] is a security threat modeling framework/methodology developed by Microsoft employees. It sheds light on the potential impact of cybersecurity threats to a system. The name is an acronoym corresponding to the first letters of various threat types, which are **Spoofing** (**S**), **Tampering** (**T**), **Repudiation** (**R**), **Information disclosure** (**I**), **Denial of Service** (**DoS**) (**D**), and **Elevation of privilege** (**E**). Banu et al. [8] proposes a methodology for using the STRIDE framework to model threats applicable to architectural components of AI-based medical chatbots and then using AISecOps techniques for automatic threat detection. STRIDE-AI [29] is another methodology adapting STRIDE approach for security assessment of AI/ML systems. It is an asset-centered methodology, which identifies the failures in ways of generating and using assests in ML life-cycle architecture, also proposed in the article. Ghosh et al. [18] introduces an approach for security threat modeling in autonomous vehicle perception systems, aligning with the ISO/SAE 21434 standard. It involves a comparative analysis using STPA-Sec and the STRIDE model to identify and address threats to the interactions and missions within the AV perception system, ensuring security at both mission and platform levels. Alatwi et al. [4] employs STRIDE and Attack Tree methods for threat modeling in ML-based **Network Intrusion Detection Systems** (**NIDS**). By analyzing system data flow with STRIDE, the researchers identify ways to exploit vulnerabilities in ML algorithms, offering insights for developing robust countermeasures against potential attacks on NIDS.

These work adapt STRIDE for AI threat modeling. Our proposed research stands out from previous work adapting STRIDE for AI/ML threat modeling by creating a novel, comprehensive threat library specific to AI/ML systems. Unlike conventional approaches, it evaluates existing frameworks, identifies limitations, and crafts a new library of security controls specific for AI/ML systems.

3 Reviewing Existing Frameworks

In this section, we describe our approach to selecting the different sources used to create the proposed threat library, their synthesis, and classification.

3.1 Selection

We followed the **Preferred Reporting Items for Systematic Reviews and Meta-Analyses** (**PRISMA**) approach to review the existing literature [20]. Since PRISMA is a widely used method for systematic and transparent literature review, it helped organize the literature review process. Our first step was to perform two kinds of searches (till June 2023). The first search was on DBLP, USENIX, and ACM digital libraries (indicated as "database" in Figure 1) to find research articles related to "AI", "ML", "security" or "privacy" "threat modeling". The second was a Google search to locate any documents, reports, archival posts, or blogs (indicated as "report" in Figure 1) which contained descriptions of potential threats to AI/ML applications. We excluded sources that were not traceable to a legitimate source from industry, academia, or government. This gave us a total of 31 initial sources. We further excluded sources which did not contain security and privacy approaches at an application level (and instead were more organization-level or country-level recommendations). This resulted in 14 sources. We listed out every threat from each of these sources for a total of 429 threats.

There was one additional source that was published after our initial analysis,[2] which is a holistic evaluation of large language models regarding criteria such as, accuracy, fairness, bias, and toxicity. Since our evaluation focused on AI/ML models at large - and specific to cybersecurity threats, we excluded this source.

3.2 Observations

While the frameworks we reviewed considered important security threats and privacy risks to AI/ML applications, we observed several points for improvement in these frameworks. In the following, we discuss key features to be met by an AI/ML threat modeling framework based on our review, which are the consideration of the *(i) AI/ML application type, (ii) AI/ML components/stage, (iii) language and definition of threats/threat modeling questions, (iv) specificity of threats, and (v) testing complexity of threats*. We present a summary of our observations on reviewed frameworks regarding these key features in Table 1. It shows which key features each of the reviewed frameworks meets. As seen from the table, none of the frameworks meets all the key features. Thus, we aimed at filling this gap by proposing our AI/ML threat modeling library. In this section, we note the following observations about the existing frameworks:

(1) **AI/ML application type**: As we discuss in Section 2, there can be a wide variety in the algorithms that underlie AI/ML applications, and how they operate. Thus, whether some threats would be applicable or not greatly depends on the type of AI/ML application. There should be some way to categorize threats based on the application type, so that there would not be a need for going through a large list of potential

[2] https://crfm.stanford.edu/helm/latest/

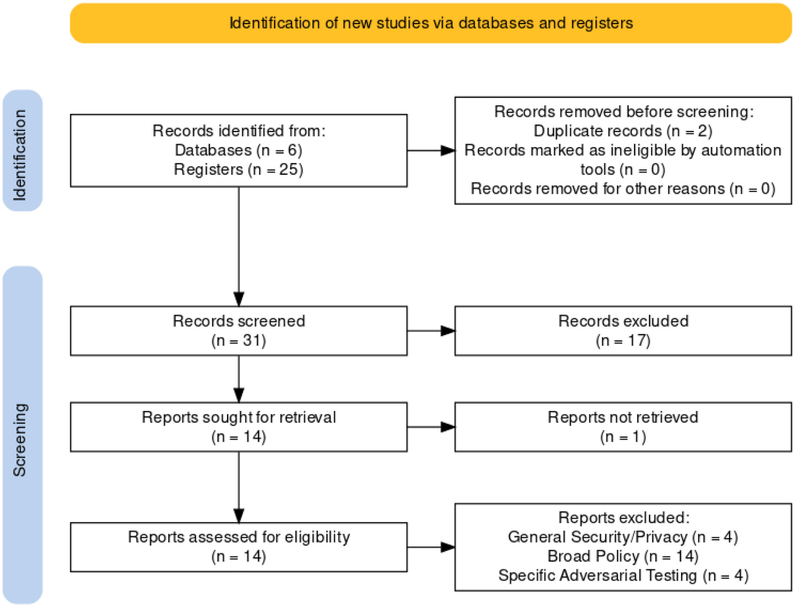
Identification of new studies via databases and registers
Identification
Screening
Records identified from:
Databases (n = 6)
Registers (n = 25)
Records removed before screening:
Duplicate records (n = 2)
Records marked as ineligible by automation tools (n = 0)
Records removed for other reasons (n = 0)
Records screened
(n = 31)
Records excluded
(n = 17)
Reports sought for retrieval
(n = 14)
Reports not retrieved
(n = 1)
Reports assessed for eligibility
(n = 14)
Reports excluded:
General Security/Privacy (n = 4)
Broad Policy (n = 14)
Specific Adversarial Testing (n = 4)

Fig. 1. Systematic analysis of different sources. Database indicates results from literature and reports indicate articles and publications found on the internet.

threats/questions during threat modeling. For example, if a developer wants to understand security threats to an AI/ML system with no user data, they should not need to go through the list of threats/questions relevant to applications with any user data. The column *Application type* in Table 1 corresponds to this feature. If a framework considers the type of AI/ML application in threat modeling, then it has a check mark.

(2) **Component/stage**: One of the key controls to mitigate a potential threat would be to focus on where in the application the threat emerges from. However, in most of the frameworks we looked at, the potential threats were contained to one stage of the application and not comprehensive enough. On other cases, these threats were reported at an application level. Ideally, if we could enumerate the questions based on whether they happen at data, model, or output - it will be easier to group the threats by category for mitigation. The column *Component* in Table 1 corresponds to this feature. If a framework considers the type of AI/ML component or the stage in AI/ML lifecycle in threat modeling, then it has a check mark.

(3) **Language and Definition**: 12 of the sources we referred to had specific terminology for threat types which we found was useful to understand what these specific threats were. However, it was not clear what questions we need to ask of the system. Hence, if this was a threat modeler analyzing an application, they would have to come up with their own questions, referring to these threats, when they are in session to consider all threat possibilities. Furthermore, if terms were defined, each source had a different definition of the different threats based on their own literature analysis and interpretation. Different definitions mean that it would require an ML expert to fully understand what each threat refers to, which might exclude other people on the product team who are not ML experts. Thus, there is a need to ask simple questions, with uniform definitions. The column *Language* in Table 1 has a check mark for a framework if it provides threat modeling questions with clear language or detailed explanations/a dictionary for threats considered.

(4) **Specificity**: Various sources exhibited differences in the specificity of threats. Some were highly specific, such as certain instances of data poisoning, while others were broad categories encompassing a wide array

Table 1. Our Observations on the 14 Initial Sources Under Consideration

Source	App Type	Component	Language	Specificity	AskorTest
Plot4AI [9]			✓	✓	
ETSI GR SAI 004 [22]		✓	✓	✓	
Mauri and Damiani, Modeling Threats to AI-ML Systems Using STRIDE [29]		✓			
Microsoft AI/ML Threat Modeling[4]	✓	✓		✓	
Microsoft + Harvard Failure Modes in Machine Learning [25]		✓	✓	✓	
Gebru et al., Datasheets for Datasets [17]	✓	✓	✓		
Amershi et al., Guidelines for Human-AI Interaction [6]		✓	✓		
Kotenko et al., Attacks Against Artificial Intelligence Systems: Classification, The Threat Model and the Approach to Protection [23]	✓	✓		✓	✓
MITRE ATLAS [15]			✓	✓	✓
Gartner MOST Framework [27]					
Sethi and Kantardzic (2017): Data Driven Exploratory Attacks on Black Box Classifiers in Adversarial Domains [1]				✓	
ENISA Cybersecurity of AI and Standardization [10]	✓		✓		
Linux ART[5]		✓	✓	✓	✓
OWASP[6]		✓	✓	✓	

of threats. For instance, the term "Evasion" could cover a diverse range of threats. Additionally, certain threats were general security issues applicable to all systems, while sources like **Holistic Evaluation of Language Models (HELM)**[3] extended beyond security and privacy to encompass various other domains. We include the column *Specificity* in Table 1 to represent this feature.

(5) **Testing Complexity**: Various threats from different sources demand varying levels of testing. For instance, certain attacks in the Linux Adversarial Robustness Toolkit (e.g., Square Attack [7], Pixel Attack [36], **Zeroth Order Optimisation (ZOO)** [14], **Geometric Decision-based Attack (GeoDA)** [33]) require automated testing to assess the feasibility of adversarial attacks. It is crucial to differentiate between tests that developers can easily conduct and those requiring automated testing. Simple checks can be incorporated into the questions posed to developers, while the more complex ones are reserved for AI security red teams to execute. The column *Ask/Test* in Table 1 denotes if a framework differentiates between tests that developers can conduct (by asking questions to them) and those requiring automated testing.

Table 1 shows each of the 14 sources and the observations applicable for each source. While these observations are not exhaustive, they formed the basis of how we structured our questionnaire. Primarily, to address these observations, we followed the following steps. First, we divided the library by applicability depending on the type of application (which are baseline, continuous learning, and user-data interaction). Second, we introduced categories to differentiate threats by which component of the application is affected (which are data, model, artefact, and system/infrastructure). Third, we used a simple binary question format to simplify the language of the threat for universal comprehension even by ML non-experts. Fourth, we scoped the list of threats to those

[3] https://hai.stanford.edu/news/language-models-are-changing-ai-we-need-understand-them
[4] https://learn.microsoft.com/en-us/security/engineering/threat-modeling-aiml
[5] https://github.com/Trusted-AI/adversarial-robustness-toolbox
[6] https://owasp.org/www-project-machine-learning-security-top-10/

specific to AI/ML applications, while leaving room for the addition of any emerging threats that might happen. Finally, we excluded questions that require automated testing as potential future work.

3.3 Synthesizing AI/ML Threats

We first identified as many threats as possible from the 14 identified sources. This gave us a total of 429 controls. We then classified them into four categories:

(1) Threat modeling (221): These controls can be asked during a threat modeling session to the product team.
(2) General security (26): These controls would be applicable to any application, and were not AI/ML specific.
(3) Adversarial testing (180): These controls can be tested by an AI/ML red team.
(4) Not relevant (2): These controls were too broad in scope to be either asked or tested.

We narrowed down our controls library to questions specifically related to "threat modeling" by excluding statements outside this category. We then grouped similar/duplicate statements into a single statement. This single statement was then modified to present it as an overall question. This reduced our list of controls from 221 to 77. We then moved to the classification stage.

3.4 Classification

To better manage the list of questions in our threat library, we classified them into two main groups. The first group was "Priority". This meant splitting the library into sections based on the application content. In this case, we divided the library into *Baseline*, *Additional-Continuous-Learning*, and *Additional-User-Data*. *Baseline* questions would be those that are relevant to controls every AI/ML application should follow. *Additional-Continuous-Learning* category involves a set of questions corresponding to controls that must be met if an AI/ML application is continuously learning from a dynamic data source, in addition to those in the baseline category. Similarly, *Additional-User-Data* category comprises questions related a set of controls to be met by an AI/ML application if it is either using user data as input to the model or interacting with end-users in some way, in addition to those in the baseline category.

The second group was based on which component of the AI/ML application is responsible for meeting the control. As shown in Figure 2, we made this classification depending on point of threat - input data, model, artefact (output), or underlying system/infrastructure that supports the application. This categorization aligns with NIST's Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations [31] so that specific components where the threat has originated from can be identified. The three prioritization categories are in blue, with 29 baseline threats, 9 continuous-learning threats, and 20 user-based threats. Each priority level is further categorized into component which is affected by specific threats. The boxes in grey indicate that there are no questions under the Artefact and System/Infrastructure for Continuous-Learning, but can be added later as needed.

3.5 Qualitative Analysis

We followed a deductive process for coding the list of threats into categories. We had three categories based on the application content and four categories for the application component (data, model, artefact, system/infrastructure) mentioned in Section 3.4. These categories and their definition formed our codebook.

Our qualitative coding process involved two rounds. For this process, we gathered three cybersecurity researchers who were also familiar with AI/ML systems. For the first round, the three researchers went over the compiled list of threats to discuss duplicates and overlapping themes. They modified the list of questions to present a list of unique AI/ML threat modeling questions without duplicates. In the second round, each of the three researchers went over the unique questions independently to classify them into appropriate categories. The initial round of coding revealed 80% agreement among the three coders. We conducted another round of discussions to resolve disagreements and reach complete consensus.

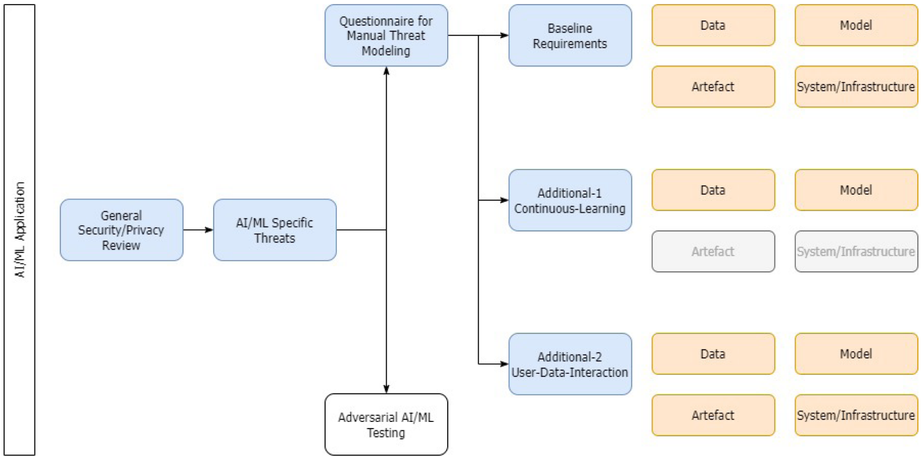

Fig. 2. Classification of the different threats in the library. For the complete list of questions, please refer to the Appendix.

Finally, for the interview round, we asked participants the pain points that they faced while going through the library. We combined these issues into themes and presented them in Section 5.

4 Unifying Controls

Following the qualitative coding process, we were able to compile the list of questions into a structured format that can be asked during a manual threat modeling process for an AI/ML application. From our analysis shown in Table 1, we found that five of the 14 sources met at least three of the observations for improved manual threat modeling. Only one of the sources met four of the five observation criteria, but it was majorly a source for automated adversarial testing. These sources contributed toward forming our list of questions, a summary of which is shown in Table 2. These threat assessment controls are categorized by relevant application type and sub-categorized by component type for easy identification of components that are susceptible to a specific threat. By taking this piece-wise approach, not all applications have to go through the same set of questions, but instead can customize the questionnaire based on their needs. For example, for a persistent ML model that contains no user data, the product team can complete only the *Baseline* assessment to evaluate their security posture. Furthermore, the questionnaire format makes it specific enough to address the threat and accessible to developers, without combining with a broad range of attacks. In the assessment, we also specify which ones require testing ("Have you tested..."). Thus, we meetall five criteria observed in Section 3.2.

4.1 Usage

Building the assessment forms a crucial component of the threat modeling process. However, it needs to be used effectively for threat modeling. Figure 3 shows the flow of decision-making in selecting the different parts of the library for AI/ML threat modeling. This library can be used as an assessment for both AI/ML applications as well as new third-party AI vendors. After an application undergoes the usual security review process and it is determined that it is not an AI/ML-driven application, the review ends. Otherwise, the application developers can take the baseline assessment. Following this, depending on whether the underlying model fits into the two additional categories outlined above, additional assessment questions can be added. This questionnaire can then be reported to the threat modeling team for review. In the review phase, developers and threat modelers can collectively determine the overall risk posture of the application and mitigation strategies to be considered.

To illustrate usage further, let us consider a weather prediction application as an example. Assuming it does not contain user data, they would take the Baseline assessment, and potentially Additional-1 (Continuous Learning),

Table 2. Sample of Questions from each Category and Sub-Category Described

Category	Sub-category	Sample Questions
Baseline	Data	Is the entire dataset developed within the organization? If no, can you document the source of the third party dataset? If no, is the dataset open source? Are there restrictions to using the open source dataset (e.g., requiring owner approval)?
	Model	Are all third-party supply chain dependencies in code documented for security? *This includes external packages used to import algorithms, online data repositories, or external code snippets.*
	Artefact	Have you tested if the model output shows information beyond its intended goal/scope?
	System/Infrastructure	Was any part of the AI/ML application created or modified by a team within the organization? If yes, do you have correct license and documentation in place for attribution? *This can be important if the application can potentially be used to create a proxy model.*
Additional-1	Data	Have you tested for data drift? *Data drift occurs when new data causes a change in distribution and results in inaccurate predictions. Data drift can be of two types: feature drift (when the input data changes) or label drift (when the labels change causing prediction shifts).*
	Model	Have you tested for concept drift? *Concept drift happens when the relationship between the feature set and the predicted outcome changes. For example, let's consider income was a predictor of home purchasing power. However, due to concept drift, income no longer significantly determines home purchasing power.*
Additional-2	Data	Is user behavioral data being collected/used as input to the model without their knowledge?
	Model	**Does the AI system automatically label users? Do you check for correctness in the labels?** *Incorrect labeling can lead to incorrect predictions.*
	Artefact	Can users ask questions or provide feedback about the output if it is incorrect?
	System/Infrastructure	Are adequate security and privacy controls provided to users when they are interacting with the AI/ML model?

The complete questionnaire is available in the Appendix. Text in italics is an extended description of the sample question. Text in bold indicates that the specific question is for generative AI systems.

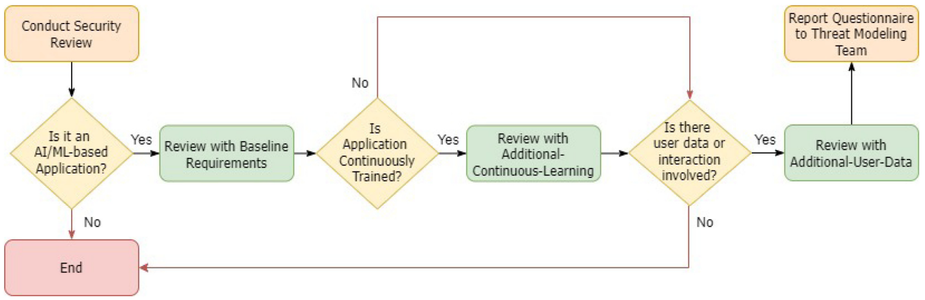
Conduct Security Review
Is it an AI/ML-based Application?
Yes
No
End
Review with Baseline Requirements
Is Application Continuously Trained?
No
Yes
Review with Additional-Continuous-Learning
Is there user data or interaction involved?
Yes
No
Review with Additional-User-Data
Report Questionnaire to Threat Modeling Team

Fig. 3. Workflow showing how the type of application would determine which components of the questionnaire is applicable (and how the additional priority levels can be chosen).

depending on which model is used. The development team would take this assessment and respond yes or no to each question. When these responses are transferred to the threat modeling team, they can determine which responses do not match with expected responses, and work with the development team to mitigate those threats.

4.2 Evaluation

At the end of the coding process, we had 77 threats. We subsequently tailored the following three questions to gather specific feedback on the questionnaire:

(1) What are some questions that might seem unnecessary to AI practitioners? These could be redundant questions that they may come across in other assessments or may be out of scope for developers to answer.

(2) Have we phrased our questions in a manner that is easy to understand? If not, what changes can we make to make it more comprehensible?
(3) What are some of the sources practitioners have been looking at to improve their understanding of security threats in their AI/ML application?

To answer these questions, we conducted one-on-one interviews with nine industry experts to obtain their feedback on the library and the general tools they use for detecting cybersecurity threats. These professionals possess a high level of expertise and background knowledge in the cybersecurity, privacy, and AI/ML areas. They have worked in the cybersecurity field for at least five years at a large organization or research institution, with the exception of one participant who was in the field for 3 years. These interviews were an hour each, and were then coded following a thematic analysis approach. Since these were expert interviews at a large organization where we obtain their views on our questionnaire, an **Institutional Review Board** (**IRB**) requisition was not required.

5 Integrating Feedback

Once the library was completed, our next goal was to ascertain whether such a library would be beneficial to developers who wanted to determine their security and privacy technical obligations. Our approach was two-fold: (1) to understand where ML experts currently get their security controls from, and (2) If they were to use a literature-based questionnaire like ours, how can we align it better to industry needs? The following section discusses our evaluation results in greater detail.

5.1 "Authoritative" Websites are Preferred

Our first objective was to understand the various sources experts consulted to identify security threats for their applications. This would help us ensure that there were not any sources that we may have missed. All participants indicated that they would search the internet first to see if there was a framework to evaluate threats to AI/ML applications. However, they could not immediately recall any frameworks specifically tailored to AI/ML security threats. Nevertheless, when asked about general security threats, two participants could immediately think of STRIDE [21].

Four participants mentioned that they would look at authoritative government websites first for any guidance, especially if there were publications done by the **National Institute of Standards and Technology** (**NIST**). One participant specifically mentioned the NIST AI **Risk Management Framework** (**RMF**),[7] which is something that they have started reading through to get a broad understanding of risks to AI applications. While we excluded NIST AI RMF from our sources due to its applicability at an organization level rather than application level, we did include Microsoft's AI threat modeling questionnaire as one of the sources mentioned by P3.

5.2 Feedback on the Assessment

All participants found the library to be helpful in understanding the different aspects of AI/ML cybersecurity threats. Despite the comprehensiveness of the library and participants' willingness to use it for their applications, when participants were asked about what they would like to see as improvements, there were several themes came up during feedback on the finalized library.

Who is Answering the Assessment Matters: About half participants brought up that it is important to incorporate the stakeholder who will be responding to the assessment. For instance, P2 mentioned that if there are specific legal questions like "What potential legal issues can happen if the model fails?", it might be harder for developers to answer since there is a low probability they would be aware of all legal ramifications of model failure. Similarly, P6 mentioned that it is better to phrase some questions as "Have you tested for concept drift

[7] https://www.nist.gov/itl/ai-risk-management-framework

(the variable projected changes)?" instead of "Is there concept drift happening (the variable projected changes)?" because it is more practical to test and confirm rather than make assumptions about the possibility of a risk.

Need for Description: While reviewing the questions, all participants noted a few questions that they found unclear. Some mentioned that certain questions were challenging to understand, especially for individuals without expertise in machine learning. They suggested that providing better descriptions for some questions would enhance clarity. For instance, we added a description to the question on "data drift" to better define the question and make it accessible to more people.

Order and Number of Questions: After the first iteration of the assessment, the library had 79 questions across the three priority levels. However, a common feedback across all participants was the need for a brevity in the assessment, especially the baseline controls which would be applicable for all applications. We re-reviewed our list of questions, removing those which could combined with another question or which would be covered by another question. This reduced our set of questions to 63 - with 29 baseline, 9 continuous learning, and 20 user data specific questions. While one participant mentioned that automatic threat detection would be desirable to find some of these threats, that was out of scope for our current AI/ML threat modeling questionnaire.

Furthermore, P6 mentioned that not only the questions but their order is also important - hence we moved the data questions to the top - so that they are asked before asking any questions about the model or the artefact (output).

Separating Elements of Supervised Learning: While we had separated questions as baseline, continuous learning, and user data, depending on the type of AI/ML application - two participants noted that there were some questions specific to generative AI applications, that would not be applicable for usual ML applications. We noted these questions and separated them out into respective subcategories for both *Baseline* and *User-Data-Interaction* categories. *Continuous-Learning* did not have any questions in this category.

Potential for Scoring: One of the questions we asked was around the helpfulness of having a scoring system, since almost all of our initial questions were binary in nature. Seven of ten experts agreed that a binary system that allows scoring might be helpful in gathering an overall picture of how secure the application is against AI/ML threats. Two participants mentioned that the scoring will need to be supplemented with comments to get some additional contextual information, especially when some of them are marked as not applicable for some applications.

6 Discussion

In this section, we discuss our alternative interpretations to our findings, the reasons behind incorporating feedback received from the industry experts, timing of launching the assessment in developmental lifecycle, and some of the limitations to our library.

Adjustments to Library: After gathering feedback from the experts on our library, we made the following changes. First, we added description to certain questions while were unclear, jargon, or open to interpretation. Second, after reviewing the questions, we decided to remove some of the duplicate and similar questions resulting in a final set comprising 63 distinct questions. Third, considering the three stages defined in our threat library, we changed the order of questions to align with the chronological developmental phases of an AI/ML application by asking data-related questions first, followed by model, artifact, and system/infrastructure to maintain linearity. Fourth, we separated questions that were more specific to generative AI applications into a new table (also found in Appendix – Tables 3, 4, and 5). Fifth, moving away from our initial questions which were a mix of open-ended and binary questions, we changed all questions to a binary format that can potentially be assigned true/false values and scored. This facilitates ease of comprehension and efficient analysis where quantifiable and categorical output are sought. Finally, we compiled the evaluated version of the library of controls that can be asked during threat modeling, leaving systematic automated adversarial testing for future work.

Completeness: We offered a comprehensive AI/ML library of controls that enables technologists to perform manual threat analysis of a wide range of AI/ML applications. Our library accounts for applicability of threats and particular developmental stages of AI system. This aids in minimizing any ambiguity arising from the broad categorization of AI/ML applications and threats and assists technologists in adopting a more targeted and organized methodology for AI threat modeling. The use of precise language/terminology coupled with specificity of threats in our library contributes to a more effective procedure (where users experience less confusion and more clarity), and generation of actionable output. Our library also suggests that responses may come from a technologist or an automated test rather than leaving users to navigate on their own to figure out the right source/method of collecting answers. As a result, this helps to achieve a more optimized and time-efficient process.

Assessment Timing: Timing is crucial when it comes to conducting security threat assessment. Initiating the threat assessment process early in the development lifecycle is a best practice recommended by many threat modeling frameworks. Early assessment helps to proactively identify potential threats at the early stages and allows technologists to integrate security measures with the least amount of friction. Due to the nature of systems and the extensive data collection that happens in initial phases, early-stage assessment is imperative to AI applications. Therefore, we suggest leveraging our library of controls from the outset of developmental lifecycle (e.g., as early as solution initiation and design). Additionally, depending on the core model of the AI system and the training methods (e.g., active learning or one-time training) and how often the model is updated, the assessment should be revisited.

Limitations: One thing to note for the AI/ML threat library is that while comprehensive of existing literature, it is not an exhaustive list of threats and additional considerations maybe added as new threats emerge. The categories allow expansion of the library and addition of new threats as they emerge in the form of questions to be asked. We plan to open-source the questions so that we can receive continuous update from the research community. Furthermore, the questions are intentionally binary in format. This allows for a scoring system if required, as future work may or may not include different weights assigned to each question response and generate an overall score for an application. In practice, there could also be a comment section for developers to fill out that accompanies each question - in case they wish to expand upon their response.

7 Conclusion

The expansion of AI is instigating a shift in perspectives, fostering the creation of inventive solutions by harnessing ML algorithms across a spectrum of applications. Nevertheless, the intrinsic security and privacy threats linked with AI/ML systems emphasize the urgent necessity for robust defenses in this dynamic landscape. The complex nature of AI models and their distinct functions makes these risks even more complicated. Stakeholders globally, acknowledging these challenges, are advocating for legal frameworks to confront security and privacy issues in AI applications.

Efforts such as the EU AI Act and government actions to hold AI companies responsible show a joint commitment to protect against the specific risks posed by generative AI tools. Initiatives from organizations like NIST and OECD to offer frameworks, databases, and guidelines for AI threats are positive steps. However, as more organizations adopt AI applications, there’s an increasing need for practical, actionable safeguards. Unfortunately, many organizations may face challenges in implementing these guardrails due to resource constraints and a shortage of subject matter expertise.

Our article addresses this pressing need by consolidating diverse AI threat frameworks into a comprehensive AI threat library. This library, consisting of 63 security/privacy questions applicable to AI/ML systems, serves as a practical and accessible tool for developers, product teams, and security and privacy engineers. By synthesizing existing knowledge and making improvements through expert input, our work aims at empowering organizations to navigate the security landscape of AI with confidence.

Finally, despite being thorough, the limitations of the presented AI/ML threat library show a difficulty in handling new threats not covered in the current framework. The intention to open-source the questions and invite continuous updates from the research community reflects a proactive approach to tackling this challenge. In summary, future challenges include being flexible to handle changing threats, finding the right balance between details and coverage, smoothly integrating threat assessments, and having a dynamic process to address new threats. These challenges underline the dynamic nature of the AI security landscape, requiring ongoing collaboration and innovation to ensure the effectiveness of threat assessment and modeling practices.

Availability

Project GuardRail is available open source at https://github.com/Comcast/ProjectGuardRail.

Appendix

A GuardRail Questionnaire

Table 3. List of Baseline Controls

Category	Description
Data	Have all features been tested for data leakage before being used as training data? *The set of features developed for a model should not contain data unnecessary for predicting an outcome. To test for data leakage, check if your original training data attributes match the ones input to the ML model.*
	Is the training data to the model verified for completeness? If there are variables we cannot collect due to dataset restrictions, are they documented? *Ensure that there are no missing variables required for prediction or incomplete data with missing values. Missing values should be handled using proper statistical approaches. Furthermore, changes to the data must be reviewed before direct addition. This is important so that malicious values cannot be inserted into the training data.*
	Have required permissions been obtained to use the training dataset? Does the training data have any copyright or legal restrictions?
	Is the entire dataset developed within the organization? If no, can you document the source of the third party dataset? If no, is the dataset open source? Are there restrictions to using the open source dataset (e.g., requiring owner approval)?
	Can the training dataset be recovered from a secondary location within the organization that is not your application? *This secondary location can be another application or a third-party customer/vendor. If external to the company, the location must be verified to be secure.*
	Is the training data maintained? (to correct labeling errors and attributes)
	Is the training dataset well-curated during pre-processing to not introduce any intentional or unintentional errors? *For example, if pre-processing is done using code and not manually, the chances of errors being introduced are less.*
	Is there a recovery plan in case training data is contaminated? *This would ideally mean having a process to recover the training data and retrain a model in case it does not behave as intended.*
	Has the model been tested with realistic data (similar to what is being predicted) and conditions before deployment? Are anomalies recorded?
Model	Is the model explainable? *Though explainability is broad, for now, we mean documenting the (i) the purpose, (ii) model functionality (how the model works), including type of algorithm used, and, (iii) accuracy with test cases.*
	In case the model is using confidential information (personal information or proprietary information), is there a warning/notification in place to limit unchecked predictions? *For instance, if the AI/ML model is recommending a higher broadband plan, it should come with a warning that recommends consulting with a representative before making a change based on AI/ML output alone. If not, it can potentially have a wrong prediction in case the user received a manipulated output.*

(Continued)

Table 3. Continued

Category	Description
	Has the model been tested with different test datasets to emulate actual prediction conditions before production?
	Can your model code be deployed for secondary use? *For example, can it be applied to a different use case/context or wrapped as a single package that can be hidden under another frontend.*
	Are all third-party supply chain dependencies in code documented for security? *This includes external packages used to import algorithms, online data repositories, or external code snippets.*
	Did you test if someone without access to the source code, can learn information about the training data or algorithm?
	Did you test if the training data can be mislabeled without notifying the team?
	Do you document if the model behavior (including accuracy) change during migration from one environment to another? Do you have version control in place when the model is updated? *For example, when the AI/ML application is moved from development to production, could there potentially be significant changes in the number of false positives or false negatives? Environment in this case can also mean changing platforms or the programming language used in the source code.*
	Do you keep track of how the model predictions have changed over time? Do you document the cause if there is significant model deviation? *This should also include documentation if the underlying model being used has also changed.*
	Has the model been tested to check if it can be modified to prevent legitimate users from accessing? *This is because attackers can modify a query to get a desired response from an AI/ML application. If this compromises the model's ability to classify accurately, it might classify legimitate user systems incorrectly and prevent them from access.*
	Are changes to the model tracked and logged? Are access controls in place regulating who has access to change the model? Are these used to audit correctness? Is the audit performed either annually or on a risk-informed basis?
	Did you test if the model can make automated incorrect decisions if incorect input (prompt) is provided when it self learns? *To prevent this, applications typically limit the scope of the data provided to the model as input to a specific purpose. There are guardrails built to reject unexpected data.*
Artefact	Do you have a process in place for a human to correct any incorrect output from the application?
	Can the model output carry any sensitive information about a person? Can this output be saved outside the application? Are there access controls in place for who can access this sensitive data? *Sensitive information can be either direct identifiers or inferences about a person, like their shopping preferences.*
	Have you tested if the model output shows information beyond its intended goal/scope?
	Are consequences (unexpected user output, degradation of service, DoS, etc.) of failure modes of the output documented? *A 'failure mode' happens when a model does not behave as expected. This can either be no output, wrong output, or problematic output (like usage of derogatory terms for instance).*
System/Infrastructure	Does your application use separate hardware to support the AI/ML system? If yes, have you tested if an attacker can observe any information about the model? Has a penetration testing assessment been conducted on the hardware components of the model?
	Are there resources to maintain the model source code?
	Was any part of the AI/ML application created or modified by a team within the organization? If yes, do you have correct license and documentation in place for attribution? *This can be important if the application can potentially be used to create a proxy model*
	Are there geolocation conditions (local laws, regulations, restrictions, etc.) that need to be met for multinational deployment, if applicable?

Question highlighted in bold is more applicable for generative AI systems.

Table 4. List of Additional-Continuous-Learning Controls

Category	Description
Data	Have you tested for data drift? *Data drift occurs when new data causes a change in distribution and results in inaccurate predictions. Data drift can be of two types: feature drift (when the input data changes) or label drift (when the labels change causing prediction shifts).* Are there measures in place to prevent real-time data collection channels from failing? *Data collection channels could be APIs and links that connect to an updating training source or it could also be the feedback loop.* When datasets from external sources are updated, is there a system in place to process the new data on time? Can the updated training data be randomized, sliced, or modified in any way that could distort the model's inference? If the model uses randomness, is the randomness generator recorded and protected? Can there be real input data to the model that is not present in the training data? Do you have a way for the model to handle this data?
Model	Have you tested for concept drift? *Concept drift happens when the relationship between the feature set and the predicted outcome changes. For example, let's consider income was a predictor of home purchasing power. However, due to concept drift, income no longer significantly determines home purchasing power.*

These are in addition to *Baseline*.

Table 5. List of Additional-User-Data-Interaction Controls

Category	Description
Data	Is user behavioral data being collected/used as input to the model without their knowledge? Is the training user data representative enough of different user population groups? Does the input to the AI model contain any sensitive and often privileged information like health, legal, financial, biometric, etc.? Do you have processes in place to prevent loss or misconfiguration of this data? Is your application collecting data from specific vulnerable or protected groups, like children, for example? Is user data properly pre-processed (use recommended data splits, attribute information is made available, reduce biases in sampling, correct formatting, and metadata recording)? Are users notified about collection and use of their data for training the AI/ML system? Do users have choice in revoking consent to use their data for AI/ML input at any time? Have retention policies been put in place for AI/ML training data that contains user information?
Model	Are users affected adversely when the system fails? Have these failure modes that could affect users documented? **Does the AI system automatically label users? Do you check for correctness in the labels?** *Incorrect labeling can lead to incorrect predictions.*
Artefact	Can users ask questions or provide feedback about the output if it is incorrect? Do you provide users with information on how the AI/ML application works? Have you tested to check if the output/actions of the model can be misinterpreted? Have checks been performed to test the model output for different user groups? Is the correctness of the results documented? Can one or more user groups be uniquely identified by the features of this model either directly or indirectly? *You might need to consult with the de-identification team to answer this question.* Does the output allowing logging recent interactions? Can the predicted output contain user data? *Note that this user data need not be personal information - it can also be behavioral inferences, like shopping habits, or last movie watched, for instance.* Are users notified that they are interacting with an AI/ML system? Are users provided with enough information about potential benefits and risks of using the AI/ML system? Do users know the consequences of their input? *Such consequences can include but not limited to, the input being used as feedback to the model, or any other purposes. For example, notifying the user that hiding an advertisement could prevent them from receiving further tailored advertisements.*

(Continued)

Table 5. Continued

Category	Description
	Do you check the scope of the output of the model? Can there be cases where the output can be counterproductive for users? *To prevent this, typically models limit the scope of what they produce as output. For example, a retail customer service chatbot should not provide healthcare information.* **Can the AI/ML system automatically make decisions for a user without involving the user in the process?** *For example, if there is an AI/ML system processing that some customers use more data than their original plan, can it automatically enroll them into a higher data plan without consulting the customer?*
System/Infrastructure	Are adequate security and privacy controls provided to users when they are interacting with the AI/ML model?

These are in addition to *Baseline*. Questions highlighted in bold is more applicable for generative AI systems.

Acknowledgments

We thank everyone within Comcast who have contributed to this effort. We further thank the following people for reviewing initial drafts of this work, including, Chujiao Ma, Sandhya Baskaran, Jacob Jones, Nitish Uplavikar, Yasmine Abdillahi, Bill Sweeney, Roy Mathen, Nancy Miller, Namita Madhira, Keith Pifko, Shilla Saebi, and Chan Voong.

References

[1] Tegjyot Singh Sethi and Mehmed Kantardzic. 2018. Data driven exploratory attacks on black box classifiers in adversarial domains. *Neurocomput.* 289, C (May 2018), 129–143.

[2] 2020. NVD - CVE-2019-20634. Retrieved January 13, 2024 from https://nvd.nist.gov/vuln/detail/CVE-2019-20634

[3] 2022. OECD AI Policy Observatory Portal. Retrieved January 13, 2024 from https://oecd.ai/en/catalogue/tools/ai-vulnerability-database

[4] Huda Ali Alatwi and Charles Morisset. 2022. Threat modeling for machine learning-based network intrusion detection systems. In *Proceedings of the 2022 IEEE International Conference on Big Data*. IEEE, 4226–4235.

[5] Greg Allen. 2020. Understanding AI technology. *Joint Artificial Intelligence Center (JAIC) The Pentagon United States* (2020).

[6] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI'19)*. Association for Computing Machinery, New York, NY, USA, Paper 3, 1–13.

[7] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. Square attack: A query-efficient black-box adversarial attack via random search. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII* (Glasgow, United Kingdom). Springer-Verlag, Berlin, 484–501. DOI: https://doi.org/10.1007/978-3-030-58592-1_29

[8] Ruby Annette, Aisha Banu, Sharon Priya S, Subash Chandran. 2023. Taxonomy of AISecOps threat modeling for cloud based medical chatbots. Retrieved January 13, 2024 from https://arxiv.org/abs/2305.11189

[9] Isabel Barberá. 2020. Privacy Library Of Threats for AI (PLOT4AI). Retrieved January 13, 2024 from https://plot4.ai/

[10] P. Bezombes, S. Brunessaux, and S. Cadzow. 2023. Cybersecurity of AI and Standardisation. Retrieved January 13, 2024 from https://www.enisa.europa.eu/publications/cybersecurity-of-ai-and-standardisation

[11] Zoe Braiterman, Adam Shostack, Jonathan Marcil, Stephen de Vries, Irene Michlin, Kim Wuyts, Robert Hurlbut, Brook S. E. Schoenfield, Fraser Scott, Matthew Coles, Chris Romeo, Alyssa Miller, Izar Tarandach, Avi Douglen, and Marc French. 2021. *Threat Modeling Manifesto*. Retrieved from http://www.threatmodelingmanifesto.org/

[12] Erik Brynjolfsson, Danielle Li, and Lindsey R. Raymond. 2023. *Generative AI at work*. Technical Report. National Bureau of Economic Research.

[13] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun. 2023. A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt. arXiv:2303.04226. Retrieved from https://arxiv.org/abs/2303.04226

[14] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (Dallas, Texas, USA). Association for Computing Machinery, New York, NY, USA, 15–26. DOI: https://doi.org/10.1145/3128572.3140448

[15] MITRE Corporation. 2020. *MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems)*. Retrieved January 13, 2024 from https://atlas.mitre.org/

[16] Lauren Feiner. 2023. OpenAI faces complaint to FTC that seeks investigation and suspension of ChatGPT releases. Retrieved January 13, 2024 from https://www.cnbc.com/2023/03/30/openai-faces-complaint-to-ftc-that-seeks-suspension-of-chatgpt-updates.html

[17] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM* 64, 12 (2021), 86–92.

[18] Subhadip Ghosh, Aydin Zaboli, Junho Hong, and Jaerock Kwon. 2023. An integrated approach of threat analysis for autonomous vehicles perception system. *IEEE Access* 11 (2023), 14752–14777.

[19] Roberto Gozalo-Brizuela and Eduardo C. Garrido-Merchan. 2023. ChatGPT is not all you need. A state of the art review of large generative AI models. arXiv:2301.04655. Retrieved from https://arxiv.org/abs/2301.04655

[20] Neal R. Haddaway, Matthew J. Page, Chris C. Pritchard, and Luke A. McGuinness. 2022. PRISMA2020: An R package and shiny app for producing PRISMA 2020-compliant flow diagrams, with interactivity for optimised digital transparency and Open Synthesis. *Campbell Systematic Reviews* 18, 2 (2022), e1230.

[21] Michael Howard and Steve Lipner. 2006. *The Security Development Lifecycle*. Microsoft Press Redmond.

[22] Secure AI (SAI) ETSI Industry Specification Group (ISG). 2020. *Securing Artificial Intelligence (SAI); Problem Statement*. Retrieved January 13, 2024 from https://www.etsi.org/deliver/etsi_gr/SAI/001_099/004/01.01.01_60/gr_SAI004v010101p.pdf

[23] Igor Kotenko, Igor Saenko, Oleg Lauta, Nikita Vasiliev, and Ksenia Kribel. 2022. Attacks against artificial intelligence systems: Classification, the threat model and the approach to protection. In *Proceedings of the International Conference on Intelligent Information Technologies for Industry*. Springer, 293–302.

[24] Ram Shankar Siva Kumar and Hyrum Anderson. 2023. *Not with a Bug, But with a Sticker: Attacks on Machine Learning Systems and What To Do About Them*. John Wiley and Sons.

[25] Ram Shankar Siva Kumar, David O Brien, Kendra Albert, Salomé Viljöen, and Jeffrey Snover. 2019. Failure Modes in Machine Learning Systems. arXiv:1911.11034. Retrieved from https://arxiv.org/abs/1911.11034

[26] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. 2020. Adversarial machine learning-industry perspectives. In *Proceedings of the 2020 IEEE Security and Privacy Workshops*. IEEE, 69–75.

[27] Avivah Litan. 2021. Use Gartner's MOST Framework for AI Trust and Risk Management. Retrieved January 13, 2024 from https://www.gartner.com/en/documents/4001144

[28] Andrew Marshall, Jugal Parikh, E. Kiciman, and R. S. S. Kumar. 2019. Threat modeling AI/ML systems and dependencies. *Security and Documentation*. (2019). Retrieved January 13, 2024 from https://learn.microsoft.com/en-us/security/engineering/threat-modeling-aiml

[29] Lara Mauri and Ernesto Damiani. 2022. Modeling threats to AI-ML systems using STRIDE. *Sensors* 22, 17 (2022), 6662.

[30] Liang Niu, Shujaat Mirza, Zayd Maradni, and Christina Pöpper. 2023. CodexLeaks: Privacy leaks from code generation language models in GitHub copilot. In *Proceedings of the 32nd USENIX Security Symposium*. USENIX Association, Anaheim, CA, 2133–2150. Retrieved from https://www.usenix.org/conference/usenixsecurity23/presentation/niu

[31] Alina Oprea and Apostol Vassilev. 2023. *Adversarial machine learning: A taxonomy and terminology of attacks and mitigations*. Technical Report. National Institute of Standards and Technology.

[32] Lisa O'Carroll. 2023. EU "in touching distance" of world's first laws regulating artificial intelligence. *The Guardian* (Oct 2023). Retrieved January 13, 2024 from https://www.theguardian.com/technology/2023/oct/24/eu-touching-distance-world-first-law-regulating-artificial-intelligence-dragos-tudorache

[33] Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. 2020. Geoda: A geometric framework for black-box adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8446–8455.

[34] Reuters. 2023. Governments race to regulate AI tools. *Reuters* (Oct 2023). Retrieved January 13, 2024 from https://www.reuters.com/technology/governments-race-regulate-ai-tools-2023-10-13/

[35] Iqbal H. Sarker. 2022. Ai-based modeling: Techniques, applications and research issues towards automation, intelligent and smart systems. *SN Computer Science* 3, 2 (2022), 158.

[36] Danilo Vasconcellos Vargas and Jiawei Su. 2020. Understanding the One pixel attack: Propagation maps and locality analysis. In *Proceedings of the Workshop on Artificial Intelligence Safety 2020 co-located with the 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI 2020), Yokohama, Japan, January, 2021.* Huáscar Espinoza, John A. McDermid, Xiaowei Huang, Mauricio Castillo-Effen, Xin Cynthia Chen, José Hernández-Orallo, Seán Ó hÉigeartaigh, and Richard Mallah (Eds.), CEUR-WS.org. Retrieved from https://ceur-ws.org/Vol-2640/paper_4.pdf

[37] Chloe Xiang. 2022. Scientists Increasingly Can't Explain How AI Works. Retrieved January 13, 2024 from https://www.vice.com/en/article/y3pezm/scientists-increasingly-cant-explain-how-ai-works

Received 16 January 2024; revised 29 March 2024; accepted 11 June 2024