

Holistic Cyber Threat Modeling for Machine Learning-Based Systems: A Case Study in Healthcare

Janno Jaal^{1,2} ^a and Hayretdin Bahsi^{1,3} ^b

¹*Department of Software Science, Tallinn University of Technology, Tallinn, Estonia*

²*Cybernetica AS, Tallinn, Estonia*

³*School of Informatics, Computing, and Cyber Systems, Northern Arizona University, U.S.A.*
{jajaal, hayretdin.bahsi}@taltech.ee

Keywords: Healthcare, Machine Learning, Adversarial Attacks, Cyber Threats, Threat Modeling.

Abstract: Considering the immense pace in machine learning (ML) technology and related products, it may be difficult to imagine a software system, including healthcare systems, without any subsystem containing an ML model in the near future. However, ensuring the resiliency of these ML-based systems against cyber attacks is vital for more seamless and widespread technology usage. The secure-by-design principle, considering security from the early stages of development, is a cornerstone to achieving sufficient security at a reasonable cost. The realization of this principle starts with conducting threat modeling to understand the relevant security posture and identify cyber security requirements before system design. Although threat modeling of software systems is widely known, it is unclear how to apply it to software systems with machine learning models. Although adversarial machine learning is a widely studied research topic, it has yet to be thoroughly researched how adversarial and conventional cybersecurity attacks can be holistically considered to identify applicable cyber threats at the early stage of a software development life cycle. This paper adapts STRIDE, a widely-known threat modeling method, for the holistic cyber threat analysis of an ML-based healthcare system.

1 INTRODUCTION

Healthcare systems are responsible for a wide range of functions and services to promote and maintain the health of individuals and communities. An enormous amount of health data is generated through electronic health records, imaging, sensor data, and text (Min et al., 2017). Thus, these systems have benefited from the rapid progress in machine learning (ML), and as a result, healthcare has become one of the early adopters of this technology. For example, ML applications equipped with Internet of Things (IoT) solutions collect vast amounts of data through remote monitoring devices, send them to the cloud and run ML models on these data to enhance the management of diagnosis and treatment efforts (Kakhi et al., 2022).


Health applications are highly susceptible to malicious cyber actions, including unauthorized access, theft or manipulation of medical records, malware infections, or denial-of-service attacks. These cyber threats can lead to severe consequences, ranging from


significant safety risks to patients to disruption of critical healthcare services and substantial economic losses.

Attackers typically compromise their target systems by exploiting applications, operating systems, or network device vulnerabilities. These vulnerabilities often arise from inadequate secure development practices, which can be prevented by applying secure-by-design principles. These principles consider security throughout the development life cycle, from requirement analysis to implementation and maintenance. As the vulnerabilities are identified and mitigated in the earliest possible stages, secure-by-design reduces the cost of security-related tasks.

Threat modeling is a critical analysis task, typically done in the early stages, to identify the attack surfaces and applicable cyber threats to the target system (Xiong and Lagerström, 2019). It starts with modeling the target and systematically elaborating on cyber threats. Attack taxonomies obtained from security frameworks and the views of experts (e.g., software developers and system architects) participating in the study shape the analysis.

ML models have increasingly been deployed into

^a  <https://orcid.org/0009-0001-6122-250X>

^b  <https://orcid.org/0000-0001-8882-4095>

software systems developed in-house or procured as a COTS product. It is also possible to use cloud-based ML Services. A considerable body of research has been conducted on adversarial attacks that focus on manipulating datasets and ML methods (Vassilev et al., 2024). On the other side, the system assets where ML models are deployed can be compromised by conventional cyber security attacks (e.g., stealing the model file via network service vulnerability). These cyber attacks may be a precondition for adversarial attacks in most attack campaigns. Although the system defenders must have a holistic view regarding both attack categories, a considerable gap exists between machine learning and cyber security experts and their security practices (Apruzzese et al., 2023).

Threat modeling methods in cyber security were first developed and applied to software systems (Shostack, 2014) but were later extended for other systems (e.g., industrial control systems (Khalil et al., 2023)). However, the adaption of these methods in ML-based software systems has not been explored sufficiently due to a lack of coherent integration of software engineering, machine learning and cyber security disciplines in this problem area. Current studies (Wilhjelms and Younis, 2020; Ali Alatwi and Morisset, 2022; Mauri and Damiani, 2022) do not propose a proper system modeling nor demonstrate how adversarial and conventional cyber attacks can be reconciled for a holistic threat analysis.

This paper demonstrates how a widely utilized cyber threat modeling approach, STRIDE, can be adapted to a system with machine learning-based components. More specifically, in a healthcare system case study, we first demonstrate how ML-related assets can be represented in a data flow diagram (DFD), constituting a system model notation for STRIDE. Then, we identify the security boundaries and systematically elicit the cyber threats applicable to the target system. We put particular emphasis on covering both adversarial and conventional cyber threats during the elicitation.

We assume the system owner has in-house capabilities to manage the entire ML life cycle, encompassing stages such as data engineering, model development, and model operation. Consequently, our study provides comprehensive coverage of ML-related activities and systems. The unique contribution of our paper is the in-depth demonstration of system modeling for ML system assets and systematic threat elicitation with a holistic view of machine learning and cyber security disciplines.

The content of the paper is as follows: Section 2 reviews the relevant literature. Section 3 presents the methods followed in this study. Section 4 gives

the case study results. Section 6 discusses our main findings. Section 7 concludes our paper.

2 RELATED WORK

A few studies have applied STRIDE to the cyber threat modeling of ML-based systems (Wilhjelms and Younis, 2020; Ali Alatwi and Morisset, 2022; Mauri and Damiani, 2022). Wilhjelms and Younis presented a DFD in which the ML model is obtained from a third party. This study adopts an attack taxonomy for systematic threat elicitation. Although it provides a comprehensive proposal that includes a ranking and mitigation of the threats, the system modeling does not cover the whole ML life cycle, and the reasoning behind DFD choices is not discussed. More specifically, the study represents the ML model as a process without separating the application software and model repository, which may not be granular enough to elaborate on specific threats. The study elicits only adversarial-related threats.

Threat modeling of an intrusion detection system is conducted in (Ali Alatwi and Morisset, 2022). Although this study addresses both adversarial and conventional threats, it utilizes two separate modeling frameworks, attack trees for adversarial ones and STRIDE for conventional ones. Attack trees are powerful in demonstrating the attack scenarios. However, they do not provide instruments for system modeling and systematic threat elicitation. In this study, the representation of the operational system is weak. The ML model is represented by an entity, usually assigned to external actors. The applicable threats are limited for entities in STRIDE. Another study follows a similar approach with the same limitation (Cagnazzo et al., 2018).

A threat modeling framework that uses the Failure Mode and Effects Analysis (FMEA) is applied to an energy grid system that has an ML-based system (Mauri and Damiani, 2022). FMEA, which is derived from the safety domain, identifies the potential failure modes of a product or system and then determines the risks. This study uses the threat categories of STRIDE to classify the findings and attack trees of such categories for threat elicitation. The study uses a non-standard notation of DFD, and it is unclear how the threat elicitation is linked to the DFD. FMEA is a top-down approach that first identifies the failure modes and then the reasons causing them. Its implementation is complex, especially for IT and cyber security professionals unfamiliar with safety concepts.

Some studies focus on threat modeling in healthcare without addressing ML-based systems. Threat

models, including STRIDE and LINDDUN, are applied and compared for systems that process electronic health records (Holik et al., 2023). Another study conducts a device-level threat modeling for Miniaturized Wireless Biomedical devices (Vakhter et al., 2022).

As a research gap, we identify that cyber threat modeling studies addressing ML-based systems do not provide in-depth guidance about system modeling choices. They do not address conventional and adversarial threats within a unified modeling framework.

3 METHODS

Threat modeling starts with identifying the security objectives that can be derived from organizational policies, standards, regulations, and legal requirements (Khalil et al., 2023). The second stage can be named system modeling or system mapping, in which the target system is identified with the necessary details (Khalil et al., 2023). As threat modeling is conducted at early stages in development life cycles, the abstraction level in system modeling may depend on the available information. However, the main system assets and relevant data flows constitute the system modeling. Then, the threat elicitation stage starts. A systematic approach that suits the system model components should be followed to achieve optimal coverage of the applicable threats. Although a complete threat modeling consists of stages such as impact/risk assessment and identification of mitigations (Khalil et al., 2023), these stages are out of scope in this paper as our focus is threat identification for ML-based system components. In this paper, we followed the STRIDE method developed by Microsoft, which is widely known and utilized by practitioners.

3.1 Security Objectives

Our study aimed to identify cyber threats to a healthcare system with ML-based assets. We mainly covered two threat types: (1) Well-known cyber threats that address the confidentiality, integrity, and availability of the systems (i.e., we named them conventional threats), such as malware, denial of service attacks, MiTM attacks, or unauthorized access. (2) Adversarial ML threats specifically target ML models, algorithms, or training/validation data (Papernot et al., 2018). They may include threats such as poisoning, evasion, or inference attacks (Vassilev et al., 2024).

An attacker can compromise a model repository by using a network service vulnerability and then steal

the model file to violate intellectual property, which can be categorized as a conventional threat. On the other hand, an attacker can query the model several times and steal it via a model extraction attack (Chandrasekaran et al., 2020). This threat is categorized as adversarial. However, in various situations, attackers can only launch adversarial attacks if they fulfil some preconditions via conventional ones. For instance, querying the model several times in a model extraction case may require the attacker to bypass some limits enforced by network or application-level access controls, which can be typically done via conventional threats.

It is important to note that this study does not focus on privacy threat modeling. Although security threat modeling allows us to identify various privacy violations once the assets and data flows regarding sensitive personal data are identified, it still does not provide a complete privacy analysis. Privacy threat models have a more comprehensive approach to collecting, processing, and sharing personal data, which can manifest in identifying more specific privacy threats, such as linkability or detectability (Deng et al., 2011).

3.2 System Modeling

ML-based components can be incorporated into the IT systems mainly in three ways (Estonian Information Systems Authority, 2024): (1) An external ML service that is created and maintained by a third-party cloud provider or other companies and accessed through APIs, (2) A pretrained or customized model obtained from other sources or companies can be deployed into the system. Such models can come within a specific COTS product (e.g., medical imaging or diagnostic). (3) The model is developed and deployed within an ML development life-cycle that entirely runs in the organization (in-house development)

This study considers the third option, which requires representing the whole ML life cycle in the system model. STRIDE uses a Data Flow Diagram (DFD), a semi-formal representation with specific notation, for system modeling of software systems. The notation classifies each system component into a DFD element, such as a process, entity, data store, or data flow. A process usually represents software components performing data processing, such as application servers, microservices, or authentication servers. Entities refer to external parties, such as users or external services. Data store notation is typically used for databases or other data storage forms, whereas data flows characterize the communications between system components. In STRIDE, the

applicable threats to each component are mapped as given in Table 1. As this mapping is a significant part of the systematic threat elicitation, selecting the most appropriate DFD element notation for ML-based assets is essential.

Table 1: Applicable Threats to DFD Elements (Shostack, 2014).

DFD Element	S	T	R	I	D	E
Entity	✓		✓			
Data Flow		✓		✓	✓	
Data Store		✓	✓	✓	✓	
Process	✓	✓	✓	✓	✓	✓

We reviewed the academic and grey literature to identify how ML-based assets and ML-life cycles are represented under this notation and concluded that no common understanding is applied in the case studies, and the life cycle needs to be covered comprehensively.

Cagnazzo et al. propose the simplest presentation - the AI/ML Model is just displayed as one external entity (Cagnazzo et al., 2018), which prevents the elicitation of significant threats (e.g., tampering and information disclosure). Although Alatwi et al. introduce processes for representing data preprocessing and model training stages in the DFD of the target system, an entity is chosen for the model deployed into the operational systems (Ali Alatwi and Morisset, 2022).

Based on the machine learning operations (MLOps) principles, the life cycle comprises three stages: design, model development and operations (Dr. Larysa Visengeriyeva, 2023). The design stage defined in this study covers data science activities such as gathering requirements and checking data availability. We excluded this stage in our DFD because these activities can be fulfilled with a series of human-expert actions rather than well-defined data flows. However, model development and operation stages can be represented in DFD notations. The operation stage includes model deployment, monitoring and maintenance. In ENISA's report (European Union Agency for Cybersecurity (ENISA), 2020), the model development is divided into two steps: model training and model tuning.

In our DFD, one of our critical decisions is to represent the model registry, which stores the models, including their versions and metadata (e.g., hyperparameters, update times), as a data store. We consider an entity inconvenient as it is utilized for external actors, and applicable threats are only spoofing and non-repudiation. We characterize the model registry as a data store rather than a process because model registries do not typically initiate a connection but re-

spond to queries of other system components without interacting with end users, similar to a database.

We represent the life-cycle steps, data processing, model development, and model operation in the DFD. It is important to note that each step can be represented with varying granularity levels. We divided model development into two detailed steps, model training, and model tuning, as these functions may be performed by two distinct teams and infrastructures, depending on the organization's scale. We create a process for each step (i.e., data processing, model training and model tuning).

Performance monitoring is an essential function in the model operation stage. In a real-world setting, an unexpected decrease may occur in model performance due to several reasons, such as concept drifts (i.e., deviations in the source data distribution and, thus, decision boundary) or functional errors introduced in the software updates. Thus, performance monitoring checks the accuracy of the model decisions regularly. In our DFD, the operation stage comprises a data store representing the model registry and a process embodying the performance monitoring function. We identified the data flows between these stages. The details about the proposed DFD are given in Section 4.1.

3.3 Security Boundaries

In STRIDE, security boundaries are drawn according to the attack surface. More specifically, potential malicious actors' accessibility of system assets determines the security boundaries. For instance, if it is assumed that such actors may physically penetrate the area where wireless communication occurs between the controller and mobile application, then a security boundary can be drawn between those two assets. Any data flow crossing the boundary and the endpoints of such a flow, whether an entity, process or data store, is considered in the threat elicitation. DFD elements within the same boundary can be considered trusted and excluded from the elicitation. The details of the security boundary decisions made in this study are presented in Section 4.2.

3.4 Threat Elicitation

In STRIDE, threat elicitation can be conducted using two approaches: STRIDE-per-Element or STRIDE-per-Interaction (Shostack, 2014). The former traverses each element during the elicitation stage, which has a broader coverage of the threat landscape and is suitable for small or medium-scale systems. The latter only enumerates the data flows crossing

the security boundaries, which is more applicable to large-scale systems. We follow the former as our target system does not have a vast number of system assets, and that approach enables us to elaborate on each asset based on its role in ML life cycles.

In this stage, we identified conventional and adversarial ML threats that can apply to each DFD element to achieve a holistic view. It is important to note that an element that does not interact with another element belonging to a different security boundary is excluded during the elicitation stage. This means such an element resides in a trusted zone and is assumed to be secure. STRIDE provides an attack tree for the pairs of an entity and threat type (Shostack, 2014). For instance, a tampering threat against a process has an attack tree, demonstrating the possible attack scenarios within the given threat context. Although these attack trees were developed for software systems, we used them while brainstorming about the applicability of the threat and still found them helpful for threat elicitation for ML-based assets. It is important to note that such trees are less known and rarely applied in research studies regarding threat modeling. Due to the page limit in this paper, we do not provide a detailed discussion of the threats within the framework of corresponding attack trees. The identified threats are given and discussed in Section 4.3.

4 RESULTS

This section provides the results of each threat modeling step introduced in Section 3. As the necessary details about the security objectives are given in Section 3.1, we exclude that stage.

4.1 System Modeling

The DFD given in Figure 1 demonstrates the whole target system. Two external entities, ten processes, two data stores and twenty-five data flows are identified. Patients and doctors (i.e., doctors representing any medical staff) are considered entities. The remote monitoring system on the patient's network consists of five processes: three for sensors with varying functions (i.e., blood sugar, heart rate, SpO₂), one for sensor controller, and one for smartphone application. The patient entity interacts with the smartphone application to read the data obtained from sensors and make necessary configuration changes to the sensors via the sensor controller. The sensor controller process collects sensor readings from sensors, processes, aggregates and relays them to the central healthcare system (CHS) process at the medical institution's net-

work (i.e., named medical network from now on) via the smartphone application.

The CHS process acts as a core system component in the medical network. It interacts with the central database, represented as a data store, to query or store electronic medical records (EMRs). Patients can query these records via a smartphone application, whereas medical staff query them via their PCs or smartphones. CHS process is also the main interface with the ML-based system assets, so it provides raw data to the ML development life cycle, queries the model registry and receives the prediction results to utilize ML support in the medical network. The medical staff can use these predictions for better decision-making and for the patients to get recommendations or help.

A similar mobile health application architecture is given in (Latif et al., 2017).

Among the identified system assets, four processes, one data store, and nine data flows can be considered ML-based system assets. As our primary purpose in this study is to explore ML life cycles, we focus on these system assets and corresponding threats from now on.

Four processes—data engineering, model training, model tuning, and performance monitoring—represent the model development and model operation steps in the ML life cycle.

Data Engineering. This step, represented as one process, aims to analyse, clean and preprocess the raw data (e.g., removing irrelevant data rows or labeling data) to make it suitable for the model development phase. Feature engineering, identifying informative features and extracting new ones can be done to improve the model performance. This step can be investigated at a more fine-grained level by dividing the process into separate sub-processes (e.g., event-stream processing, data integration, data visualization) and adding a data store element representing a database (e.g., data lake) for raw data or extracted features. The data engineering platforms can interactively communicate with human experts to label the data, creating additional attack surfaces. However, in this study, we focus on the main incoming and outgoing data flows of data engineering tasks to simplify the case study and represent the whole step as one process.

As shown in Figure 1, the data engineering process receives the data flow named raw data from the central healthcare system and delivers labeled data, represented as another data flow, to the model training process.

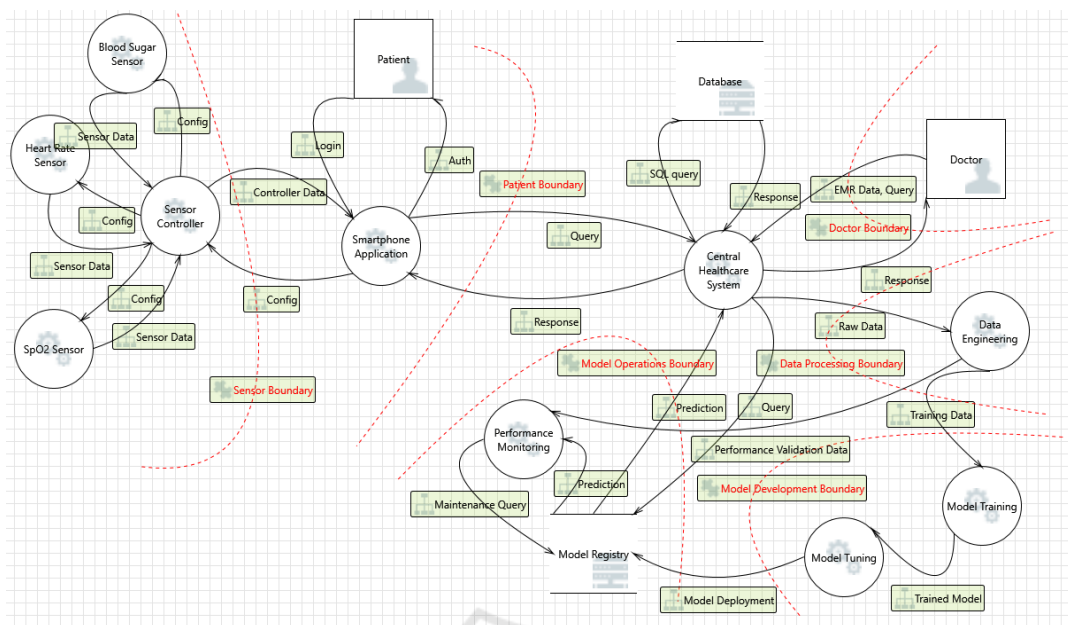


Figure 1: Data Flow Diagram.

Model Training & Model Tuning. We address model development through two sequential processes: model training and model tuning. The model development step aims to create, test, and deliver a stable model to an operational setting. An initial model is created by utilizing a ratio of labeled data as training data in the model training process. This process typically benchmarks various ML methods to minimize the classification error on validation data. The model tuning process obtains the created model, optimizes it (e.g., hyperparameter optimization) and prepares it for operation.

As shown in Figure 1, the model training process obtains labeled data from the data engineering process and sends the trained model to the model tuning stage in a dedicated data flow. The model tuning process delivers the final model (i.e., model deployment data flow) to be deployed to the operational setting.

Similar to data engineering, the model development step of the ML life cycle can be more detailed with model repositories, version control or data visualization processes; however, we prefer a more coarse-grained approach for simplicity.

Performance Monitoring. The operational stage of the ML life cycle is represented by two DFD elements, a process for performance monitoring and a data store for the model repository, as shown in Figure 1. This process checks the model’s performance periodically. The models deployed to the operational systems are stored in the model repository with relevant metadata (e.g., model versions). The queries of the

central healthcare system regarding the decisions of the ML model are directed to the model repository in a dedicated data flow. The prediction data flow returns the model decisions to the central healthcare system process. The performance monitoring process periodically queries the model repository in the maintenance query data flow and retrieves maintenance prediction in a separate data flow. As checking the model performance can be realized by reference labeled data, this process retrieves it in a data flow (i.e., performance validation data) from the data engineering process.

4.2 Security Boundaries

Our DFD is divided into six security boundaries. Three of them are dedicated to ML-based assets. We created one boundary for data engineering, one for model development, and one for model operation, as demonstrated in Figure 1.

Model development boundary comprises two processes: model training and model tuning. We assume that the data flow, trained model, that occurs between these two processes can be protected by organizational procedures. Thus, it stays within the security boundary, meaning threat elicitation for that DFD element is not required. However, we consider that data flow that transfers labeled data from data engineering to model training processes should not be trusted as different teams in an organizational setting can handle them. More importantly, attackers may pay particular interest to the integrity of labeled data to launch various attacks, including poisoning.

Another boundary is drawn for the assets of model operation, namely the performance monitoring process and model repository data store. We contemplate that the final model (i.e., model deployment in Figure 1) obtained from the model tuning process should be checked as the attackers may aim to manipulate or copy that model before it reaches the operational system. It is possible that the central healthcare system process can be compromised as it resides at the attack surface facing end users. Thus, the data flow between that process and the model repository should be secured. However, the data flows, namely maintenance query and prediction, between performance monitoring and mode registry, stay within the security boundary, and there is no need for threat elicitation for them as long as the performance monitoring process and model registry are protected. The data flow, named performance validation data, should be secured between the data engineering and performance monitoring process. The attackers may aim to disturb the performance monitoring function to reduce the operational benefit.

4.3 Threat Elicitation

By analyzing ML-based system assets, we identified 18 threats: 10 for data flows, three for data stores, and five for processes. Eight of these threats are in the adversarial category, whereas 10 are conventional ones. Table 2 presents a selected subset of those threats. Due to the page limit, we could not present all threats in this paper. This subsection explains how we elicit threats for different DFD elements, one data flow, one data store and one process, regarding ML-based system assets.

Data Flow. Based on the mapping between DFD elements and threat categories (see Figure 1), data flow is subject to tampering, information disclosure and denial of service. We selected the data flow named query between the central healthcare system process and model registry. The purpose of the query data flow is to send patient data to the model to get predictions about a health decision.

Tampering in this stage could lead to input manipulation attacks, also called evasion attacks, in which the attacker changes the inputs with small perturbations to cause the model to make wrong predictions. Sundas et al. presented various ways in which input manipulation can lead to the accuracy downfall of the model in the healthcare domain (Sundas and Badotra, 2022). Adversarial examples are the first risk listed in the top ten ML security issues provided by BIML (Gary McGraw, 2020). Rahman et al. successfully

conducted evasion attacks for COVID-19 deep learning systems in medical IoT devices (Rahman et al., 2021).

As this data flow carries sensitive patient data, the attacker can eavesdrop on the communication channel using attack techniques (e.g., sniffing the network), indicating an information disclosure threat. It is important to note that although the realization of this threat is a significant achievement for an attacker who aims to steal patient data, it can be also a precondition for an evasion attack as the attacker may need to collect some queries and predictions to create a shadow model and use it to generate samples to evade the model.

An attacker may conduct a denial-of-service attack by overwhelming the capacity of the model repository with unnecessary queries and, thus, disrupting critical medical decision-making (Ruan et al., 2024). Depending on the network access control implemented at the model operation boundary, spoofing the central healthcare system process may be a precondition for this threat.

Information disclosure and denial of service threats identified in this data flow can be categorized as conventional, whereas evasion threats are typical ML-based threats.

Data Store. Data store elements are susceptible to tampering, repudiation, information disclosure and denial of service (see Figure 1). Our DFD has one data store, model registry, which can be categorized as an ML-based asset.

The tampering threat may apply when the attacker can replace the model with a malicious one by bypassing the authorization function of the model repository. This threat can cause severe consequences, ranging from performance degradation to intentionally misleading medical decision-making (e.g., changing the diagnosis from critical disease to normal). The attacker could change the functionalities of the existing one, thus conducting model reprogramming. Model reprogramming involves altering healthcare AI/ML models to produce incorrect or biased outputs. Authorization problems can also lead to information disclosure threats when they enable the attacker to steal the model, causing significant intellectual property violations.

Process. A process is susceptible to spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privileges. Here, we address the data engineering process to demonstrate the elicitation for this DFD element type. Spoofing the data engineering process may enable attackers to replace

the labeled data sent to the model training process, a type of data poisoning attack. A similar situation is with tampering. If the attackers can access the data engineering process, they can carry out model reprogramming attacks that result in a malfunctioning AI/ML model. Microsoft also introduced the neural net reprogramming threat in their AI/ML-specific threats report (Andrew Marshall, 2022).

5 THE VALIDATION OF THE THREAT MODEL RESULTS

We followed a qualitative validation approach to validate our results due to the lack of ground-truth data about all relevant threats. A semi-structured survey was sent to four experts (represented as X1-4) with cyber security and ML backgrounds. The experts, on average, with more than five years of professional experience, were identified from the authors' professional social networks.

All experts agreed that the design choices in DFD make sense in general. They also agreed that using a data store element for the "Model Registry" is valid. EX3 suggested splitting the model registry for the deployed model and the registry for older models and metadata into two separate elements. The experts also agreed to represent the model development phase and boundaries. All experts agreed with the threats' relevancy but provided additional comments. EX1 brought out that, ideally, the threats would have an impact tied to them as well. Our scope is only threat identification, not ranking, so we will consider this feedback in future studies. EX1 mentioned that more privilege escalation, human errors, rogue employees and physical attacks could be mentioned, and EX2 pointed out insider threats. EX3 also mentioned backdoor attacks. While these threats are present in the system, they are not ML-specific, so they are not identified in the analysis. However, privilege escalation and insider threats can be reconsidered for ML-based assets. The backdoor attacks can also target some assets in the model development stage.

6 DISCUSSION

In this paper, we conduct threat modeling of a system with in-house ML model development capability. Although this option comprehensively covers ML-based system assets, other options, such as COTS products with ML models or cloud ML services, may have different cyber threats worth exploring in dedicated

studies. Threats originating from supply chain issues or the security of the model updates in online mode may be considered for COTS products. Semi-trusted cloud admins, access control weaknesses in multi-tenant cloud environments, or availability issues due to sharing hardware and platform infrastructures can be analyzed for cloud ML services.

The existing studies that introduce threat modeling to ML-based systems have weaknesses in system modeling. The details of ML model representations in DFDs and design choices are not elaborated (Wilhelm and Younis, 2020; Ali Alatwi and Morisset, 2022; Mauri and Damiani, 2022). ML life cycles, in general, and operational steps (e.g., performance monitoring), in particular, have not been completely captured. The design considerations about system boundaries are not usually justified. One of our main contributions revolves around the detailed system modeling and discussion of security boundaries.

This study uses STRIDE as the baseline method for threat modeling. It is widely known and easy to understand. Although more complex methods are applied in the literature (e.g., business process notations, FMAE), we contemplate that usability is a significant success factor. Threat modeling is a collaborative brainstorming activity between various stakeholders (e.g., system owners, system architects and security analysts). Instead of seeing the problem from a method-centric lens, we suggest practitioners and researchers see it from a process-centric perspective. An approach that facilitates a common understanding among people with different backgrounds and systematizes the security knowledge around the target system may give more useful results.

Accumulating and incorporating the existing security knowledge into the modeling process as a systematic knowledge base (e.g., attack taxonomies, libraries) would be an essential part of that process view (Khalil et al., 2023). Although we use the content of some documents as an attack taxonomy for adversarial attacks (Vassilev et al., 2024; Andrew Marshall, 2022), we do not put additional effort into creating a well-developed attack knowledge base. Creating a knowledge base is addressed in (Wilhelm and Younis, 2020). We will address this issue in our future studies.

7 CONCLUSION & FUTURE WORK

In this paper, we conducted cyber threat modeling of a healthcare system that collects data from remote mon-

Table 2: Identified threats (selected samples).

Type - Conv: Conventional, Adv: Adversarial, DFD Elements - DF: Data Flow, P: Process, DS: Data Store.

Type	DFD Element	Identified threats	Threat description
Adv	Training Data (DF)	Data poisoning	Data poisoning in healthcare means intentionally manipulating medical data to influence AI/ML model training. This could be achieved through MiTM or spoofing attacks and modification of data features or labels. This could result in inaccurate diagnoses and compromised patient care (Jagielski et al., 2018; Mozaffari-Kermani and Sur-Kolay, 2015; Vassilev et al., 2024; Gary McGraw, 2020)
Conv	Raw Data (DF), Training Data (DF), Performance Validation Data (DF)	Data confidentiality threats	Data confidentiality threats in healthcare involve unauthorized access or disclosure of sensitive medical data. This could be achieved by means like packet sniffing. This could lead to potential privacy breaches and misuse of personal information (Gary McGraw, 2020)
Conv	Query (DF)	Denial of service	Denial of Service disrupts healthcare AI/ML systems by overwhelming them with requests. Spoofing the source of the request could be done for this. This can potentially disrupt critical medical decision-making (Ruan et al., 2024)
Adv	Prediction (DF)	Membership inference	During membership inference, the attacker tries to find out if a particular record or sample was part of the training by querying the model. Spoofing the processes and sending several queries can be one way to achieve this. This could uncover sensitive patient data, posing risks to privacy and confidentiality (Shokri et al., 2017; van Breugel et al., 2023)
Adv	Model Registry (DS)	Model reprogramming	Model reprogramming involves altering healthcare AI/ML models to produce incorrect or biased outputs. Attackers could access the model from weak access control mechanisms in the model registry. This could pose risks to patient safety and treatment effectiveness (Andrew Marshall, 2022)
Conv	Model Deployment (DS)	Model replacement	Model replacement involves replacing whole healthcare AI/ML models with malicious ones. Attackers could access the model from weak access control mechanisms in the model registry. This could pose risks to patient safety and treatment effectiveness
Adv	Data Engineering (P)	Data poisoning	Data poisoning in healthcare means intentionally manipulating medical data to influence AI/ML model training. Attackers could use backdoors, privilege escalation, malware, or software vulnerabilities to access the process of carrying out the poisoning. This could result in inaccurate diagnoses and compromised patient care (Jagielski et al., 2018; Mozaffari-Kermani and Sur-Kolay, 2015; Vassilev et al., 2024; Gary McGraw, 2020)

Continued on next page

Table 2 – continued from previous page.

Type	Data flow	Identified threats	Threat description
Conv	Data Engineering (P), Performance Monitoring (P), Central Healthcare System (P)	Data confidentiality threats	Data confidentiality threats in healthcare involve unauthorized access or disclosure of sensitive medical data. Attackers could use backdoors, malware or software vulnerabilities to compromise the process. This could lead to potential privacy breaches and misuse of personal information (Gary McGraw, 2020)
Adv	Model Training (P), Model Tuning (P)	Model reprogramming	Model reprogramming involves altering healthcare AI/ML models to produce incorrect or biased outputs. Attackers could achieve access to the model from weak access control policies and maliciously fine-tune the model. This could pose risks to patient safety and treatment effectiveness (Andrew Marshall, 2022)

monitoring devices, sends it to central servers, and facilitates its analysis with ML models. Our particular emphasis is on ML-based system assets. We created a system model covering the entire ML model development life cycle and systematically elicited threats. In the future, we aim to extend this study by prioritizing threats and identifying relevant countermeasures. Privacy threat modeling of such systems is another research direction.

REFERENCES

- Ali Alatwi, H. and Morisset, C. (2022). Threat modeling for machine learning-based network intrusion detection systems. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 4226–4235.
- Andrew Marshall, J. P. (2022). Threat modeling ai/ml systems and dependencies.
- Apruzzese, G., Anderson, H. S., Dambra, S., Freeman, D., Pierazzi, F., and Roundy, K. (2023). “real attackers don’t compute gradients”: bridging the gap between adversarial ml research and practice. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 339–364. IEEE.
- Cagnazzo, M., Hertlein, M., Holz, T., and Pohlmann, N. (2018). Threat modeling for mobile health systems. In *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 314–319.
- Chandrasekaran, V., Chaudhuri, K., Giacomelli, I., Jha, S., and Yan, S. (2020). Exploring connections between active learning and model extraction. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1309–1326.
- Deng, M., Wuyts, K., Scandariato, R., Preneel, B., and Joosen, W. (2011). A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1):3–32.
- Dr. Larysa Visengeriyeva, Anja Kammer, I. B. (2023). Mlops principles. Last accessed 12.05.2024.
- Estonian Information Systems Authority (2024). Tehisintellekti ja masinõppe tehnoloogia riskide ja nende leevendamise võimaluste uuring.
- European Union Agency for Cybersecurity (ENISA) (2020). Artificial intelligence cybersecurity challenges.
- Gary McGraw, H. F. (2020). An architectural risk analysis of machine learning systems: Toward more secure machine learning.
- Holik, F., Yeng, P., and Fauzi, M. A. (2023). A comparative assessment of threat identification methods in ehr systems. In *Proceedings of the 8th International Conference on Sustainable Information Engineering and Technology*, pages 529–537.
- Jagielski, M., Oprea, A., and Biggio (2018). Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35.
- Kakhi, K., Alizadehsani, R., Kabir, H. D., Khosravi, A., Nahavandi, S., and Acharya, U. R. (2022). The internet of medical things and artificial intelligence: trends, challenges, and opportunities. *Biocybernetics and Biomedical Engineering*, 42(3):749–771.
- Khalil, S. M., Bahsi, H., and Korötko, T. (2023). Threat modeling of industrial control systems: A systematic literature review. *Computers & Security*, page 103543.
- Latif, S., Rana, R., Qadir, J., Ali, A., Imran, M., and Younis, S. (2017). Mobile health in the developing world: Review of literature and lessons from a case study. *IEEE Access*, PP:1–1.
- Mauri, L. and Damiani, E. (2022). Modeling threats to ai-ml systems using stride. *Sensors*, 22(17).
- Min, S., Lee, B., and Yoon, S. (2017). Deep learning in bioinformatics. *Briefings in bioinformatics*, 18(5):851–869.
- Mozaffari-Kermani, M. and Sur-Kolay (2015). Systematic poisoning attacks on and defenses for machine learn-

- ing in healthcare. *IEEE Journal of Biomedical and Health Informatics*, 19(6):1893–1905.
- Papernot, N., McDaniel, P., Sinha, A., and Wellman, M. P. (2018). Sok: Security and privacy in machine learning. In *2018 IEEE European symposium on security and privacy (EuroS&P)*, pages 399–414. IEEE.
- Rahman, A., Hossain, M. S., Alrajeh, N. A., and Alsolami, F. (2021). Adversarial examples—security threats to covid-19 deep learning systems in medical iot devices. *IEEE Internet of Things Journal*, 8(12):9603–9610.
- Ruan, J., Liang, G., Zhao, H., Liu, G., Sun, X., Qiu, J., Xu, Z., Wen, F., and Dong, Z. Y. (2024). Applying large language models to power systems: Potential security threats. *IEEE Transactions on Smart Grid*.
- Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models.
- Shostack, A. (2014). *Threat modeling: Designing for security*. John Wiley & Sons.
- Sundas, A. and Badotra (2022). Recurring threats to smart healthcare systems based on machine learning. In *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 1–8.
- Vakhter, V., Soysal, B., Schaumont, P., and Guler, U. (2022). Threat modeling and risk analysis for miniaturized wireless biomedical devices. *IEEE Internet of Things Journal*, 9(15):13338–13352.
- van Breugel, B., Sun, H., Qian, Z., and van der Schaar, M. (2023). Membership inference attacks against synthetic data through overfitting detection.
- Vassilev, A., Oprea, A., Fordyce, A., and Andersen, H. (2024). Adversarial machine learning: A taxonomy and terminology of attacks and mitigations.
- Wilhjelm, C. and Younis, A. A. (2020). A threat analysis methodology for security requirements elicitation in machine learning based systems. In *2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 426–433.
- Xiong, W. and Lagerström, R. (2019). Threat modeling—a systematic literature review. *Computers & security*, 84:53–69.