Cybersecurity in Machine Learning and AI

Joel Mwaka
University of Stuttgart
Stuttgart, Germany
st175494@stud.uni-stuttgart.de

***Abstract*— Machine learning (ML) is one of the most interesting techniques being researched in recent decades and has applications in various areas. Among these applications is machine learning to identify and protect systems from potential adversarial intrusions, thus presenting promising cybersecurity solutions. However, machine learning algorithms and systems themselves are prone to a variety of security concerns. In the first part of this paper, a brief introduction to the concept of machine learning and its generic model will be given. Thereafter, some information gathered by some leading researchers on the taxonomy of attacks is presented. Some prevalent types of attacks are also discussed. Lastly, a summary of some defense mechanisms used to combat threats is drawn. The second part of this paper provides a case study of cybersecurity applications in the medical field, specifically Remote Heart Monitoring Devices. This paper aims to give an introductory understanding of machine learning threats, the defense mechanisms against them and provide a case study of how safer machine learning systems could be applied in the field of medicine.**

Keywords—Machine Learning, Cybersecurity, Defense, Adversaries, Cyberattacks, Artificial Intelligence, Medical AI.

PART 1: CYBERSECURITY IN MACHINE LEARNING

I. INTRODUCTION

Machine Learning employs a variety of algorithms to iteratively learn patterns in data and make predictions on unseen data. Although there are many advantages of applying machine learning algorithms in various fields, there are still risks involved with the use of machine learning that need to be overcome to ensure the security of data, reliability of the model, and transparency of the decision process. The main focus of this paper is on the cybersecurity risks that are associated with the application of machine learning and how these can be overcome. To have a better understanding of how these risks occur and how they can be prevented, it is important to first give a brief introduction on the general working principles of machine learning.

Machine Learning is split into three learning techniques namely – supervised learning, unsupervised learning, and reinforcement learning. In the supervised and unsupervised learning techniques, a training dataset from which the algorithm will learn the patterns in data is required. It consists of instances collected that have an input vector of attributes and in the case of supervised learning, the desired output. In case of classification problems, the desired outputs are discrete values (labels), whereas regression problems have continuous-valued outputs. The Figure I shows a generic model of an ML system as shown by McGraw et al. [1]. McGraw et al. describe the model using 9 components. The ML Pipeline starts with the raw data in the real world which is then collected and labelled in case of supervised learning. An exploratory data analysis thereafter helps find and deduce important features for the learning process. The collected data is divided into three datasets: the training dataset used to optimize the learnable model parameters like weights and biases, a validation set used to optimize model architecture parameters like number of layers and the number of training epochs in neural networks and finally, a test set used to evaluate the performance of the trained model. The optimized model with a good enough performance is then used in an ML system to make decisions on new unseen inputs from the real world. Each of the steps is vulnerable to cyberattacks.

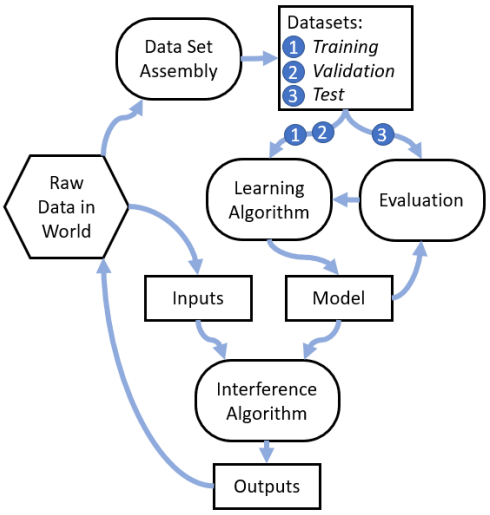

Figure I: Generic Model of an ML System [1].

In section II, a taxonomy of cyberattacks on ML systems is discussed with reference to various studies done by profound researchers. A few examples of risks and cyberattacks are discussed in Section III. Section IV gives a summary of some reactive and proactive defence mechanisms against cyberattacks. In conclusion of part 1 of this paper, a personal conclusion and future outlook is drawn. Part 2 of the paper focuses then on a case study of the application of cybersecurity techniques in a medical environment, precisely, remote heart monitoring devices.

II. TAXONOMY OF ATTACKS

When attacking an ML system, an attacker may use a variety of tricks and skills to sabotage the ML model, hence leading to wrong predictions or extraction of vital data. According to Pitropakis et al. [2], regardless of what individual steps the attacker takes, the attack-process can be summed up into two phases.

A. Preparation Phase

Before the attacker can make any attack-plan, they need to gather relevant information about the model and identify the skills necessary to carry out the attack. This is the preparation phase. In [3], the knowledge that an attacker may get about a ML system include: knowledge about the training data, the feature set used, the machine learning algorithm, the cost function minimized during training and lastly, the trained parameters. The category of attacker knowledge depends on what kind of information the attacker can find about the model [2]. In case the attacker

knows information about both the learning algorithm and the ground truth (i.e., the training and test data that has already been labelled / measured before training the model) then the type of attack is called a *Whitebox attack*. Whitebox attacks are common to opensource models since information about the model architecture is often available to the public. If neither information about the algorithm used nor ground truth is available to the attacker, then we are dealing with *Blackbox attacks*. A final category known as the *Graybox attacks* defines attacks in which the attacker knows information either about the ML algorithm used or the ground truth.

Some popular machine learning algorithms that may be used to optimize the model parameters include Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs) widely used in more complex learning tasks like image recognition, classical algorithms like Naïve Bayes, Support Vector Machines (SVMs), K-Means, K-Nearest Neighbour among others.

B. Manifestation Phase

After successfully determining the necessary skillset and information needed to carry out an attack, the attacker then launches the next phase – Manifestation phase. Both Pitropakis et al. [2] and Barreno et al. [4] suggest some characteristics by which this phase may be characterized. They both however heavily focussed on attacks against supervised classification models. Barreno et al. suggests characterizing attack models based on the following characteristics:

1. *Attack Influence*: There are two ways an attacker can influence a ML system. Firstly, the attacker could make alterations the training data used in optimizing the model parameters. Such attacks are called *Causative Attacks*. Secondly, the attacker may not have access to the training data but rather use carefully designed input data on a model and observe the decisions made. From these decisions, the attacker may be able to extract some information about the training data or/and model architecture. These attacks are called *Exploratory attacks*. [3]
2. *Attack Security Violation*: Any attack on a ML system usually poses a security threat. This characteristic defines what kind of security violation the attacker does. The attacker may aim to evade the detection of certain harmful instances during classification (*Integrity Violation*). In some cases, the attacker may influence the model to classify a negative data instance as a false positive (*Availability Violation*) [4]. In some more recent research [5, 6], another security violation in which the attacker is able to obtain private information about users, model etc. by reverse engineering is discussed, i.e. the *Privacy Violation*.
3. *Attack Specificity*: This points out the range of data instances that the attacker targets. *Targeted specificity* is when the attacker wishes to have a certain class classified as a specific class other than the true class, whereas with *indiscriminate/generic Specificity* it can be classified as any other of the classes but not the true class. [3, 5, 6]

In [7], McGraw et al. from the Berryville Institute of Machine Learning further classify attacks on ML systems by the part of the model they attack. An attacker may either target the input, the model itself, or the training data used to optimize the cost function. Combining this classification with the attack influence characteristic (causative/ manipulation attacks and exploratory/extraction attacks) discussed above, we can derive a taxonomy of six categories for attacks on ML Systems. Table I shows these categories.

Table I: Categories of Attacks on ML Systems. [7]

		Type of Attack	
		Manipulation Attacks	*Extraction Attacks*
Targeted Part	*Input*	Input Manipulation	Input Extraction
	Training Data	Training Data Manipulation	Training Data Extraction
	Model	Model Manipulation	Model Extraction

III. Machine Learning Attacks

The Berryville Institute of Machine Learning carried out an architectural risk analysis of ML systems [1] in which they identified 78 risks associated with using ML systems. They further go on to select and discuss the top ten risks [8] from the 78 risks identified. In this section of this Paper, a brief overview of some of the attack types will be given.

Adversarial attacks

Sadeghi et al. [9] defines Adversarial machine learning as a game between an ML System and an Adversary. The ML System will learn patterns from existing data with the goal of predicting the new data outputs, whereas the Adversary aims to alter the training data, new data, or the model parameters to cause wrong predictions.

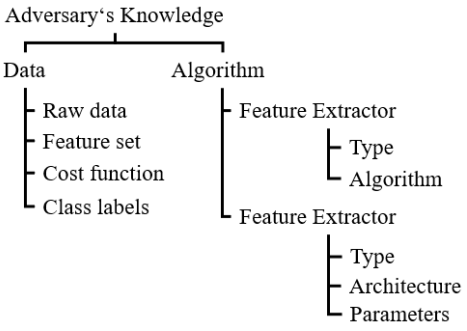

Figure II: A taxonomy of the Adversary's knowledge [9].

However, in most cases the Attacker may not have access to the training data and the model parameters of a ML system. Hence, most adversarial attacks are evasion attacks that attack an ML system by altering the new input data to be classified [1]. Assume we train a Model $F(\cdot)$ to classify an input $\boldsymbol{x} \in \mathbb{R}^d$ by a label y from a certain label space $\mathcal{Y}$. The Adversary $A(\cdot)$ will create an input $\tilde{\boldsymbol{x}} = A(\boldsymbol{x})$ that is similar to $\boldsymbol{x}$ such that the Model will do a wrong prediction for $\tilde{\boldsymbol{x}}$, i.e., $F(\tilde{\boldsymbol{x}}) \neq y$. The more information the Adversary can get about an ML system, the better equipped it is to design $A(\cdot)$ capable of fooling $F(\cdot)$. The figure II [9] shows a taxonomy of the knowledge an Adversary can possess about an ML model. This knowledge is categorized into knowledge about the data and knowledge about the algorithm. The goal of an adversarial attack is categorized by what security violation the Attacker intends to infringe [9]. In evasion attacks, the attacker does not infringe on the normal functioning of the model but rather finds inputs that

find loopholes in the model, thus violating the integrity of the model [5].

Data Poisoning

When training an ML model, training, validation, and test data plays a very important role in optimizing the model parameters with reference to the cost function. Should an attacker have the ability to alter the training or test data, there is a high chance that the proper operation of the ML system may be compromised [1]. In data poisoning attacks, the attacker intentionally alters the data in any of the three parts of the data (training, validation, and test) with the goal of influencing the optimization of the learning algorithm. It is important for ML engineers to have a good understanding of data sensitivity and what fraction of data an attacker may have access too. One particularly interesting type of the poisoning attack is the backdoor or Trojan attack. In this type of attack, the Adversary uses a backdoor key when poisoning the model such that in case the backdoor key is available, the model misclassifies the input but performs normally in absence of the backdoor key [2]. An example of this kind of attack can be shown in an ML system used in autonomous cars to detect road signs – an attacked system may correctly detect stop signs, but in case some particular landmark (backdoor key) is placed on a stop sign, it misclassifies it as a speed sign [5, 6] as represented in Figure III. This may cause accidents if the autonomous car then drives into a junction.

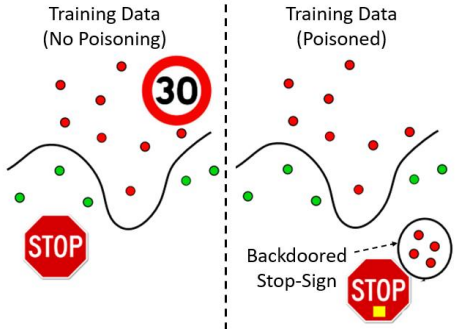

Figure III: Representation of the impact of backdoor poisoning attacks of decision function in the case of road sign detection [5].

According to the study by Biggio et al. [5], poisoning attacks can violate either the integrity of a model e.g. through backdoor and trojan attacks or the normal functioning of a model (availability) by maximizing the classification error

Privacy Attacks

Popular in the category of input and training data extraction attacks is the risk of facing attacks with the goal of espionage or breach of confidentiality. An attacker can use model inversion techniques to retrieve data from a model. The aim of Model Inversion attacks is to reconstruct training samples from model parameters or model outputs [5]. A model inversion attack can either be blackbox whereby the attacker can only query the final result of the model or whitebox. With the increasing availability of model architectures available for download on the internet like through Tensorflow Hub, white box attacks are becoming more prevalent [10]. Chen et al. [10] present a study of how to carry out a white box model inversion attack using a new inversion-specific Generative Adversarial Network (GAN) that can better distil knowledge from public data to launch attacks on private models. Fredrikson et al. [11] on the other hand show how blackbox models can be attacked with just minimal information from the model. One particularly interesting example of such an attack is shown by Fredrikson et al. [11] who attack a facial recognition system with only the name of the person recognized by the model (class label) and access to the classification confidence score of the model for the given name. This poses an enormous risk to the privacy of users of facial recognition models vulnerable to this kind of attack. Figure IV [11] shows an example of the image such an attack produces when given a class label (name of person) and access to the model classification score given.

Figure IV: Image recovered by a model inversion attack on a facial recognition system (left) and the training image of the victim (right) [11].

Privacy attacks like Model Inversion violate the privacy of the model by retrieving inappropriate data from reconstructed training data.

The attacks discussed above are the most common kind of attacks found in research papers [5]. There are however other kinds or variations of these attacks on ML systems that aren't discussed. These include transfer learning attacks, model extraction/manipulation, among others.

IV. Defenses Against Attacks

In this section, a discussion about some of the popular defence mechanisms against ML attacks is done. There are two ways to react to a ML attack, and these are; either acting reactively to counter attacks that have already occurred or proactively to prevent future attacks from happening [5].

A. *Reactive Defenses*

Reactive defences have the goal of protecting a system after an attack has already happened. In some cases, reactive defences can be more effective than proactive cases [5]. The typical workflow of the reactive mechanism involves analysing the attack results to see the loopholes in the system that the attacker may have used and then proposing defence mechanisms to counter the attacks. Some examples of reactive defence strategies include; timely detection of novel attacks, training the classifier frequently and consequent comparison of the classifier decisions with reference to the training data and ground truth [3]. These mechanisms can also be used proactively.

B. *Proactive Defenses*

This type of defences try to find a secure solution to potential attacks that may occur in the future – i.e., the kind of attacks the system may face are not completely known and therefore, there needs to be an analysis of various attack models and the possible defences against them. Liu et. al [5, 6] state that the most common procedure that ML engineers follow when designing this kind of defence mechanism involves four steps: selecting potential adversarial models and modelling them, performing penetration tests on the targeted model, analysing the

impacts of the penetration tests, and finally proposing counter measures to the adversarial models. These steps are summarized in figure V.

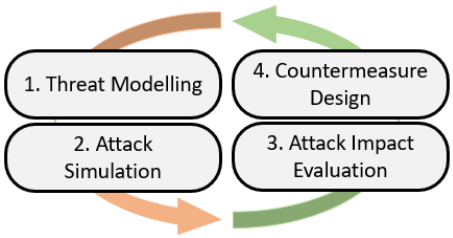

Figure V: Steps of a proactive model protection against attacks [5, 6].

Biggio and Roli [5] separate the counter measures taken by proactive defences into two categories: The first category is Security-by-design. Security-by-design means building systems securely from the ground up to secure against attacks. This kind of defence is used in cases of whitebox attacks, whereby the attacker knows information regarding both the training dataset and the model architecture. Defending whitebox attacks involves techniques that detect attacks early on and/or ensure secure and robust learning. Secondly comes the category of security-by-obscurity. In this category, measures are taken to protect grey- and blackbox attacks by either hiding/randomization of information from the attacker or detecting probing attacks before they occur. Below are established countermeasure mechanisms [3, 5, 7]:

1) Randomizing data collection

This proposes the collection of training data samples at different timings and under different circumstances. Although researchers in 2008 showed that this mechanism was effective enough in hiding information from the attacker, it still remains an open research problem to understand to how far this mechanism can help prevent an attacker from learning a sufficient surrogate model and defend against model inversion attacks [5]. Furthermore, using a randomized hypotheses might reduce the importance of feedback to an attacker. Randomization does not necessarily mean information will be less available to the attacker, but rather that the attacker will have to do more work to get the information [3].

2) Input Transformation

Reconstruction of input data can help defend against attacks, mostly attacks in the input manipulation category [7]. In the raw input to an ML system, there is often a considerable amount of extra variation, that is not relevant for the classification problem [7]. As a result, some of this unnecessary data is likely to be included in the ML system's learnt hidden representations. The harmful additional information becomes mixed with the positive information in some way. This makes a model susceptible to evasion attacks that take advantage of the extra learned variations to cause misclassification. Yuan et al. [12] mentions the possibility to use variant of autoencoder with a penalty term called deep contractive penalty aimed at increasing model robustness. The autoencoder is trained to eliminate noisy distribution from adversarial samples fed into the model.

3) Robust and secure learning algorithms

The system's robustness can be improved by enhancing the learning algorithm. Learning algorithms subjected to constraints of the functions that the algorithm learns tend to have an increased robustness against causative attacks [4]. This technique is known as Regularization. It basically extends the cost function optimized during training by including a regularization term that penalizes complex distributions, that may be a result of adversaries in the training set. Ensembles of models can be used, to make it harder for the attacker to reverse engineer the model decision making process [4].

4) Noise/Anomaly/Attack detection

The input data can be compared to the typicality of the training data to detect noise and anomalies not present in the training dataset [7]. Under this category of defences that detect malicious patterns, the Reject On Negative Impact (RONI) defence mechanism proposed by Barreno et al. [3] may be included. This mechanism measures for each instance of training data, an empirical effect and eliminates instances from the training dataset that lead to negative impact on the accuracy of the model. Detecting attacks before they occur is something difficult but allows the model designer to know the capabilities of the attacker and the loopholes in the ML model. Numerous research papers studied by Yuan et al. [12] develop classifiers that detect input samples as either a clean input or an adversarial sample. Using such a model may help filter out adversarial samples beforehand.

5) Adversarial training

The aim of adversarial samples either in the test set or input, is to exploit the distribution of the training phase and find loopholes for misclassification. A solution to stop this would be to include adversarial samples to the training set during retraining [12]. This countermeasure makes the decision boundary more robust against anomalies.

6) Information hiding/limiting

Although it is nearly impossible to limit all feedback to an attacker, limiting some feedback makes it harder for the attacker [3]. The learner could also misinform the attacker with false altered information about the model or dataset [4]. This reverses the roles of the learner and attacker – i.e., the attacker faces an indiscriminate causative availability attack by the learner that provides false information. Some more sophisticated learners can be developed to trick an attacker into believing that some particular instances were not included in the dataset thus acting as an attractive point for the attacker. If the attacker then targets this attractive point, it may be easier to detect the adversaries.

V. Conclusion

This paper serves as an introduction to understanding ML attacks and how they can be detected and avoided. It covers a section of a wide range of attack variants and defence mechanisms. Defending against attacks is an open research topic. The main issue observed in research papers studied for this paper, is that each of them model attacks are based on predictable attack strategies. Although some of the defence mechanisms also detect and avoid other attack strategies, they may not be capable to do so in case of unknown, unpredictable attacks. As [5] state, ML models should be able to defend themselves even against unknown unexpected threats. This creates and interesting research field that would benefit the whole ML and AI society.

PART 2: CASE-STUDY: CYBERSECURITY IN REMOTE HEART MONITORING DEVICES

I. MOTIVATION

In the western world, good quality medical service is something that almost everyone can easily access. This however is not the case in many third world countries that do not have the health care infrastructure or expertise necessary to provide even the basic services necessary to their population. A solution for these people might be the use of decentralized medical devices that can gather health information from the patient and either make a diagnosis with help of some ML model or transmit collected data to a professional health worker, who makes the diagnosis. Such a system can also be used in regions with good medical services to save time and reduce costs incurred by both the patient and the medical service provider. Several papers have been written about wearable health monitoring devices. For example, Fahim Faisal and Syed Hossain [13] from Bangladesh present a remote medical diagnosis system that collects heartbeat and temperature data from the patient and is rendered over the internet in real time. The data can then be viewed through a web browser remotely. However, in the paper [13] the security aspect of the data collected is not discussed explicitly. Part 2 of paper aims to point out risks and threats associated with remote heart monitoring devices (RHMD) and highlight some concerns that require special attention. It also presents a concept for such a device based on emerging technologies that can be used to ensure a better safety of remote RHMDs from cyberattacks.

II. ASSOCIATED RISKS AND THREATS

Since the intention is for this device to be able to diagnose heart complications with help of a ML model, it is critical to put emphasis on the security of the model used. As shown in part 1 of this paper, a lot of the measures against cyberattacks involve hiding as much data as possible from the attacker. This means we need to design our system architecture such that there is minimal to no possibility for an attacker to access data and disrupt the normal functioning of the device. In a 2017 study [14], Piggin lists some generic threats specific to medical devices. Some of these threats include database injection used to gain access and steal data, escalation of privileges to perform actions that would otherwise not be permitted, denial of service by affected availability of computing resources, communication disruption, among others. The U.S Food and Drug Administration (FDA) that oversees the medical devices recommends that a process of cybersecurity risk assessment of the device's clinical performance be implemented by considering the exploitability of the vulnerabilities and the seriousness of the health impact to the patient in case a vulnerability is exploited [14]. The European Medicine Agency (EMA) also recommends the same urgency when investigating the safety of new medical devices [15]. The heart being one of the most vital parts of our bodies definitely poses a higher risk than other parts like the leg or arm. The risks associated may range from 'negligible' for example a misclassification of a normal state of the heart as a heart attack or 'catastrophic' in case the ML model doesn't detect a crucial problem with the proper heart functioning. Catastrophic threats may lead to death in a patient and call therefore for special care. The Figure VI below shows how the FDA suggests one should assess the vulnerability of a device and whether the risk is acceptable or not.

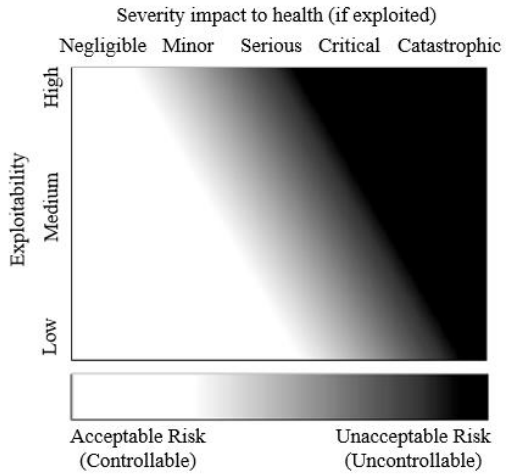

Figure VI: Vulnerability risk assessment by the FDA [14].

III. CONCEPT OF REMOTE HEART MONITORING SYSTEM AND CONCERNS INVOLVED

In this chapter a brief concept of a remote heart monitoring device with the main peripherals is presented. The figure VII below shows the architecture of the remote heart monitoring system.

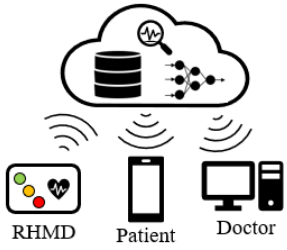

Figure VII: Concept architecture of system.

The idea is to have a RHMD that is either wearable or implanted in the patient. With help of sensors, the RHMD can collect vital information necessary for diagnosis. The RHMD can then either detect a heart issue with an inbuilt ML model or notify medical personnel in case an unknown state of the heart is detected. In case the RHMD is wearable, it may be built with indicators that let the patient know the state of their heart. For example, an LED that shows green for normal heart operation, orange to provide caution to the user and red to seek professional help. The information gathered by the RHMD can then be broadcast to a safe database in a cloud. Integrated data analysis and detection applications can also be included in the cloud to analyse and monitor more complex heart issues that may either need more computational power or more data. The patient may have access via a secure mobile application to monitor their heart operation and receive notification from the doctor or analysis application if dangerous patterns like increased stress levels or high and low blood pressure are noticed. This may then allow the patient to act early enough before any heart complications occur. The doctor is also allowed access to the information in the cloud. Apart from the patient and the doctor, no one else should be able to access this information. In [16], they develop an intelligent pill dispenser with a similar system architecture. Some issues that need to be given particular attention are discussed below.

a) RHMD safety using processors:

This poses a question to the ability of the RHMD device to provide security against attacks on the software running on the remote device. Applying the recently introduced processors with Trusted Execution Environments (TEEs) like the Intel SGX in the RHMD provides a solution to secure the data, code and models used by restricting access to the trusted part of the processor [17, 18]. This enhances the security of the ML model by hiding it from adversaries.

b) Secure cloud computing:

Secure cloud computing calls for procedures and technology to secure cloud environments against threats. Some security solutions for secure cloud computing include user access control to ensure only authorized users, device access control to prevent unknown devices, malware prevention, data loss prevention, data encryption to prevent unauthorized access to data even when it is stolen, data visibility to define who has access to which data and who doesn't. These solutions limit adversarial access to the model or data used for training thus aid in implementing the security solutions in Chapter IV of Part 1.

c) Data transmission:

For heart problems to be detected in real-time, data must be broadcast to the cloud constantly. I.e., even without internet connection, the device should still transmit data to the cloud - might be challenging to implement. Frequency of data transmission should also be determined such that the data saving structure is not overloaded but still has enough real-time data to make a diagnosis. Transmission should be encrypted to avoid unwanted access. The cloud computing concerns and data transmission concerns are similar to that of the intelligent pill dispenser [16].

d) Acceptance of medical AI in society:

AI has been depicted in various ways by the media and movies. These depictions often build a fear of AI. This tends to make society weary of AI applications. However, the fear is not fully unjustified. There are still many short comings that AI has, and the fear only increases the need for proper test protocols and regulations to monitor and limit the application of AI. For example, AI in medicine is today only seen as an assistant to professional doctors [14]. The RHMD suggested in this paper however aims to allow ML models make decisions on their own and only seek professional help when needed. As shown in Chapter III of Part 1, ML models are however vulnerable and require very special attention in securing them.

IV. Conclusion

As discussed in Part 1, limiting the access of an adversary to the model architecture and data contributes highly to securing ML Systems. However, there are still some loopholes even in the safest systems that can be exploited. For example, some researchers have been able to find vulnerabilities in some TEEs, which are actually considered to be very safe [18]. There are still many research areas open to ensure absolute safety, and thereby acceptance of medical systems even in cases of catastrophic severity levels that demand very low system exploitability as shown in Figure VI.

References

[1] G. McGraw, H. Figueroa, V. Shepardson, and R. Bonett, "An architectural risk analysis of machine learning systems: Toward more secure machine learning.," [Online]. Available: https://berryvilleiml.com/docs/ara.pdf

[2] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, and G. Loukas, "A taxonomy and survey of attacks against machine learning," *Computer Science Review*, vol. 34, p. 100199, 2019, doi: 10.1016/j.cosrev.2019.100199.

[3] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Mach Learn*, vol. 81, no. 2, pp. 121–148, 2010, doi: 10.1007/s10994-010-5188-5.

[4] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, J. D. Tygar, "Can Machine Learning Be Secure?," 2006, doi: 10.1145/1128817.

[5] B. Biggio and F. Roli, "Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning," *Pattern Recognition*, vol. 84, no. 3, pp. 317–331, 2018, doi: 10.1016/j.patcog.2018.07.023.

[6] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung, "A Survey on Security Threats and Defensive Techniques of Machine Learning: A Data Driven View," *IEEE Access*, vol. 6, pp. 12103–12117, 2018, doi: 10.1109/ACCESS.2018.2805680.

[7] G. Mcgraw, R. Bonett, H. Figueroa, and V. Shepardson, "Security Engineering for Machine Learning," *Computer*, vol. 52, no. 8, pp. 54–57, 2019, doi: 10.1109/MC.2019.2909955.

[8] G. Mcgraw, R. Bonett, V. Shepardson, and H. Figueroa, "The Top 10 Risks of Machine Learning Security," *Computer*, vol. 53, no. 6, pp. 57–61, 2020, doi: 10.1109/MC.2020.2984868.

[9] K. Sadeghi, A. Banerjee, and S. K. S. Gupta, "A System-Driven Taxonomy of Attacks and Defenses in Adversarial Machine Learning," *IEEE transactions on emerging topics in computational intelligence*, vol. 4, no. 4, pp. 450–467, 2020, doi: 10.1109/tetci.2020.2968933.

[10] S. Chen, M. Kahla, R. Jia, and G.-J. Qi, "Knowledge-Enriched Distributional Model Inversion Attacks," Oct. 2020. [Online]. Available: http://arxiv.org/pdf/2010.04092v4

[11] M. Fredrikson, S. Jha, and T. Ristenpart, "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, Denver Colorado USA, 2015, pp. 1322–1333.

[12] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," Dec. 2017. [Online]. Available: http://arxiv.org/pdf/1712.07107v3

[13] F. Faisal and S. A. Hossain, "IoT Based Remote Medical Diagnosis System Using NodeMCU," in *2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, Island of Ulkulhas, Maldives, 2019, pp. 1–7.

[14] R. Piggin, "Cybersecurity of medical devices: Addressing patient safety and the security of patient health information," 2017.

[15] P. Office, "REGULATION (EU) 2017/ 745 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL - of 5 April 2017 - on medical devices, amending Directive 2001/ 83/ EC, Regulation (EC) No 178/ 2002 and Regulation (EC) No 1223/ 2009 and repealing Council Directives 90/ 385/ EEC and 93/ 42/ EEC,"

[16] N. Sahlab *et al.*, "Development of an Intelligent Pill Dispenser Based on an IoT-Approach," in *Advances in Intelligent Systems and Computing, Human Systems Engineering and Design II*, T. Ahram, W. Karwowski, S. Pickl, and R. Taiar, Eds., Cham: Springer International Publishing, 2020, pp. 33–39.

[17] C. Segarra, R. Delgado-Gonzalo, M. Lemay, P.-L. Aublin, P. Pietzuch, and V. Schiavoni, "Using Trusted Execution Environments for Secure Stream Processing of Medical Data," in *Lecture Notes in Computer Science, Distributed Applications and Interoperable Systems*, J. Pereira and L. Ricci, Eds., Cham: Springer International Publishing, 2019, pp. 91–107.

[18] Sergio Prado, *Introduction to Trusted Execution Environment and ARM's TrustZone.* [Online]. Available: https://embeddedbits.org/introduction-to-trusted-execution-environment-tee-arm-trustzone/ (accessed: Jan. 10 2022).