A Comprehensive Artificial Intelligence Vulnerability Taxonomy

Arttu Pispa and Kimmo Halunen
National Defense University, Helsinki, Finland

arttu.pispa@iki.fi
kimmo.halunen@oulu.fi

Abstract: With the rise of artificial intelligence (AI) systems and machine learning (ML), there is a need for a comprehensive vulnerability framework that takes into account the specifics of AI systems. A review of the currently available frameworks shows that even though there have been some efforts to create AI specific frameworks, the end results have been flawed. Previous work analysed for this paper include AVID, Mitre ATLAS, Google Secure AI Framework, Attacking Artificial Intelligence, OWASP AI security and privacy guide, and ENISA Multilayer framework for good cybersecurity practices in AI. While only AVID is intended to be an AI/ML focused vulnerability framework, it has some weaknesses that are discussed further in the paper. Of the other works especially the ENISA framework has a valuable way of determining AI domains that can be affected by vulnerabilities. In our taxonomy proposal the first part of the evaluation process is determining the location in the AI system lifecycle that the vulnerability affects. The second part is determining which attributes of technical AI trustworthiness are compromised by the vulnerability. The third part is determining the possible impact of the vulnerability being exploited on a seven-step scale from the AI system functioning correctly, to it performing unintended, attacker directed actions outside the bounds it is supposed to function in. We also evaluate two known AI vulnerabilities based on our taxonomy proposal to showcase the benefits in comparison to existing frameworks.

Keywords: Artificial Intelligence, Vulnerability, Framework

1. Introduction

The rise of new and much improved AI and ML technologies and applications has led to a situation where these are used in more and more varied use cases. The proliferation of AI solutions in the everyday life of ordinary people and in the working practices of many organisations brings also the possibility of new threats and vulnerabilities, for example as a tool for finding vulnerabilities in other AI/ML systems (Carlini, 2023).

In recent years the research on these vulnerabilities has also been very active and there are already results of attacks and hacks on different applications and AI/ML techniques (Kathikar *et al.*, 2023; Qiu *et al.*, 2019). There are even some very fundamental results on how it is possible to create AI models with undetectable backdoors or weaknesses (He *et al.*, 2022).

As there are more and more issues found in AI systems, it has become evident that there is also a need for a system of classification for these vulnerabilities. In the traditional information security industry the CVE (Common Vulnerabilities Enumeration) (The MITRE Corporation, 2023a), CWE (Common Weakness Enumeration) (The MITRE Corporation, 2023b) and CVSS (Common Vulnerability Scoring System) (FIRST, 2023) are used to classify and evaluate the severity of different vulnerabilities in our digital infrastructure. Although these systems can be useful also in the AI/ML cases, there are many differences that support a more nuanced view of the vulnerabilities related to AI systems.

In this paper we evaluate previous work on building taxonomies and frameworks for AI systems' vulnerabilities. Our findings show that these taxonomies and frameworks do not adequately capture the necessary nuances of AI vulnerabilities. Thus, we also propose a new and improved taxonomy for AI/ML vulnerabilities.

2. Previous Work

There is as of yet no widely accepted vulnerability framework or taxonomy for artificial intelligence systems (Machine Learning, Generative Neural Networks, Large Language Models, etc.) unlike regular information technology vulnerabilities have with CVE (Common Vulnerability Enumeration) (The MITRE Corporation, 2023a). However, there are some contenders for a future system to be used. Most of these have grown out of a need to have some way of enumerating common types of vulnerabilities in a fast moving field, but it seems there has not been a concerted effort on creating a system suited for handling vulnerabilities in a systematic fashion across the field.

The search for existing AI/ML vulnerability frameworks was performed by queries for related keywords in both scientific search portals as well as open source tools such as Google Scholar. To ensure comprehensiveness, additional searches were performed in general internet portals, e.g. Google search, to ensure non-academic sources were also captured.

Used search portals include Scopus, EBSCO Discovery Service, Google Scholar, and regular Google search service. Searched for terms include but are not limited to "Artificial Intelligence", "Machine Learning", "Large Language Model", Vulnerability, Cybersecurity, "Cyber security", Framework, and Taxonomy, as well as most acronyms for the preceding terms.

For our evaluation we have also taken some frameworks, guidance documents, and similar public efforts that do not aim to be a comprehensive vulnerability taxonomy, to include them as input for our effort to create a comprehensive framework.

2.1 AVID

AVID (AI Vulnerability Database) is an open-source AI/ML vulnerability database, where anyone can in theory input vulnerabilities and they will get a generic ID assigned, as well as basic details so other people can be informed of them (AI Risk and Vulnerability Alliance, 2023). The database itself is publicly accessible at the website[1] and provides at least basic information for all vulnerabilities. Unfortunately there are some deficiencies in the effort that in our understanding limits its usefulness as a generic AI vulnerability taxonomy effort.

Firstly the largest issue with AVID is that it only considers vulnerabilities in the development phase of the AI models, so any that affect or are caused after deployment are not considered or do not have a place in the taxonomy. This leaves such vulnerabilities as might be caused by for example degradation of the used sensor feeds out of the system, and thus AVIDs usability as a common vulnerability framework is limited.

Some of the categories brought forth by AVID are good, but we also believe some of the categories given to be either too broad, such as environmental safety, while others are too narrow in scope, such as excessive queries. Some vulnerability categories are also misaligned and difficult to use as a technical platform for vulnerability taxonomy, such as the category of lacking global explanation. Even though it is known that it is beneficial for AI system developers and users to be able to explain the basis for any decisions made by the system, what is the actual vulnerability in effect of being unable to explain functionality?

2.2 MITRE ATLAS

While it is not a vulnerability framework, but a framework for attack vectors in AI, MITRE ATLAS (The MITRE Corporation, 2024) is important to evaluate when developing a vulnerability taxonomy. This is because it is a sister framework to the MITRE ATT&CK (The MITRE Corporation, 2023c), a widely used framework for regular IT system attack vectors, but for AI and ML attacks. Even though the framework doesn't cover vulnerabilities directly, many vulnerabilities have a corresponding attack vector that can be employed to take advantage of the vulnerability. Still the issues with MITRE ATLAS for vulnerability taxonomy are the same that ATT&CK has for regular vulnerabilities.

Firstly there are vulnerabilities that enable several different attack vectors to be used, and thus clear and unambiguous classification becomes more difficult, unless entries are duplicated across different vectors, which works against the intention of providing a single taxonomy enabling everyone to talk about the same thing in one place.

Secondly there are attack vectors in ATLAS that do not as such concern the AI/ML system itself, but rather how the systems are embedded in products and open APIs which are not so much a vulnerability as an intended use case of the system. Thirdly the level of detail in the ATLAS enumeration is in our opinion too detailed for a good vulnerability taxonomy fracturing the body of vulnerabilities too much and making it more difficult than necessary to see at a glance what parts of the AI system are vulnerable, and how the vulnerabilities might manifest themselves.

2.3 Google Secure AI Framework

Recently released effort by Google in providing guidance on how to implement AI and ML tools in a safe and secure manner in organisations (Hansen and Venables, 2023). At least for the time being it has little in the way of anything outside providing a small number of general guidelines in implementation and development. It does

[1] https://avidml.org/database/

not provide anything in relation to taxonomy for AI/ML vulnerabilities as of now, but it is unclear what the future plans of Google are for development of SAIF.

2.4 "Attacking Artificial Intelligence"

The report by Marcus Comiter (2019) performs some classification between different types of vulnerabilities or at least attack vectors affecting AI systems, but for a generic use case some of the delineations do not in our opinion make sense. The report is clearly good base work, but does not by itself provide a vulnerability taxonomy that could be implemented into a generic vulnerability enumeration.

The good points relate to how the intentions of the attacker relate to the severity of the attack and how different vulnerabilities are affected. This is in our opinion something that a common vulnerability taxonomy should retain at least in severity enumeration. The division of input attacks into two axis, human perceivable-invisible, and physical-digital are good enumerations for individual attacks but for vulnerabilities are not good classification methods, as the first mainly covers how stealthy the exploitation of the vulnerability can be, and the second depends heavily on where and how the AI system is implemented and used. For AI used for processing digital inputs the format is always digital, and for systems working on real world camera feeds the method will most likely be physical.

2.5 OWASP AI Security and Privacy Guide

This related work provides a good framework and enumeration for different types of attacks that AI systems might be attacked with, but does not provide any taxonomy for AI vulnerabilities (OWASP Foundation, 2023). The main takeaway here is that the vulnerabilities underlying these attack vectors should at least be present in any comprehensive AI vulnerability taxonomy. It is mostly included for comprehensiveness of the search.

2.6 ENISA Multilayer Framework for Good Cybersecurity Practices in AI

ENISA has published a framework for general good practices in AI/ML cybersecurity (European Union Agency for Cybersecurity, 2023). While this framework itself doesn't provide any guidance or taxonomy on AI/ML vulnerabilities, it does provide for example a comprehensive version of an AI system lifecycle that is superior in our view to the one used by AVID. This comes from the fact that it contains parts of the process of creating an AI system that are not only limited to the actual development and deployment of the system, but for example starts from the business reason for creating an AI system. It also covers eventual maintenance and continuous retraining of the model as is likely to happen in any AI system with wide adoption rates.

We believe that as a related framework, it would be good if any proposed AI vulnerability taxonomy would at least in part align with the ENISA framework, so that wider and more general adoption can be achieved.

A similar effort has been recently published by the UK National Cyber Security Centre (2023), but it is mostly similar general guidance as the ENISA framework, and as such does not itself provide a good framework for a vulnerability taxonomy.

3. Taxonomy

Our taxonomy aims to address the shortcomings of the earlier taxonomies with a three stage process of evaluation, an overview of which can be seen in figure 1.

First, it should be evaluated which phase of the AI system development and use the vulnerability affects. The three phases we have chosen here are the most relevant and widespread while still being useful for actual vulnerability information sharing and for evaluating affected AI systems for existence of the vulnerability in deployed form. The first phase is the development of the AI systems, containing as examples poisoned libraries, pre-trained poisoned models, and various other software supply chain vulnerabilities that can affect the functionality of the AI model.

The second phase is the training of the model, whether it is a pre-trained model or one built specifically for the system being evaluated. Vulnerability examples in this phase include poisoning training data-sets, both public and private, using unrepresentative training sets, as well as poisoning federated training data.

The third phase is the deployment phase of the AI system including both the actual deployment to production use as well as eventual maintenance of the system in use and any modifications made to it. Vulnerability examples here are model inversion, training data inversion, instability to adversarial examples, training drift due to continuous learning, etc.

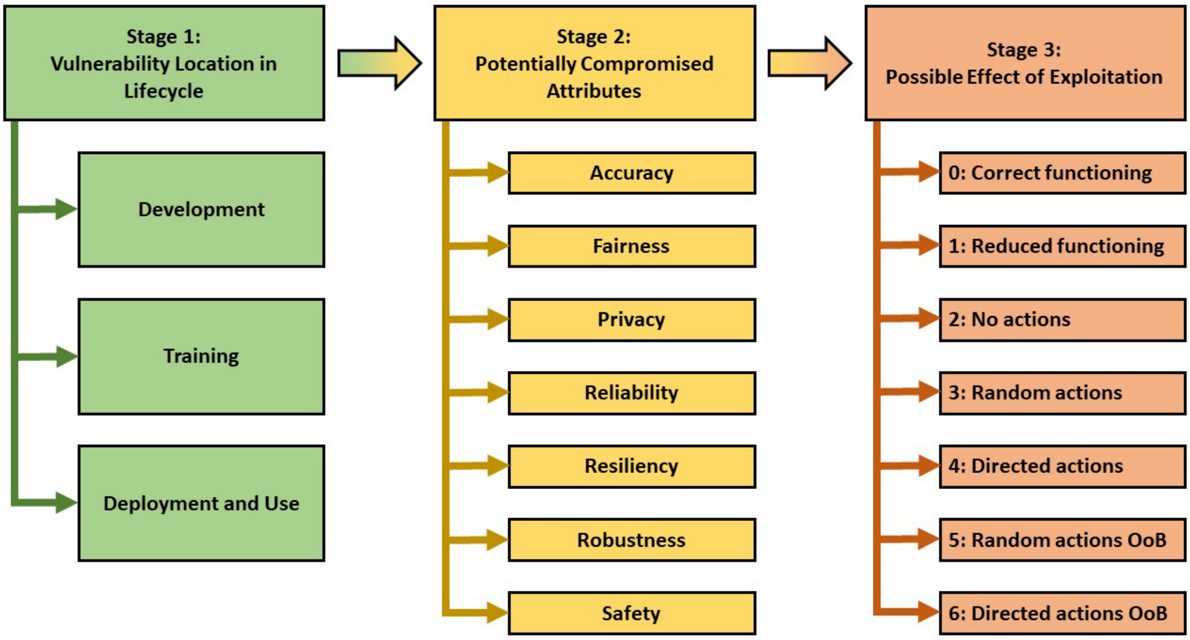

Figure 1: Overview of the proposed three stage evaluation process for AI system vulnerabilities

In the second stage the attributes of secure and trustworthy AI that are undermined by the vulnerability are evaluated to bring context to the use and focus of the vulnerability. Here we propose to use a framework of AI trustworthiness closely aligned with the ENISA framework to enable closer cooperation between actors in the AI security sphere. Some examples of the attributes that can be undermined are accuracy, fairness, reliability, and safety.

The third stage of evaluation is performed by assessing the level of degradation of functionality of the AI system in regards to the original intention of the creator. This metric is used as it allows us to have more detailed descriptions of the vulnerability effects, as well as takes into account the fact that in normal use the system is intended to perform autonomous actions and thus not being under control of the owner is standard operating procedure instead of a malfunction. The full devised scale is detailed later, but runs from the AI performing correctly (no impact), through the AI performing undirected unintended actions within the bounds of the AI system parameters, to the AI system performing unintended actions under attacker direction out of bounds of the AI system parameters.

3.1 Vulnerability Location in AI System Lifecycle

The first part of the evaluation is to determine which part of the AI system lifecycle the vulnerability affects. For this purpose, we suggest dividing the AI system development and deployment into three distinct phases to help narrow which parts of the development-deployment pipeline potential users of the vulnerability disclosures need to scrutinise to ensure their own systems are or are not affected by the disclosed vulnerability.

The first phase is the Development of the AI system where the architecture and design of the system itself are developed. The vulnerabilities in this phase affect systems used in creating the AI system itself, but not the learning process for the AI being used. Common examples of vulnerabilities in this phase are poisoned libraries (Ladisa *et al.*, 2023), pre-trained poisoned models being re-used (Kathikar *et al.*, 2023), as well as any vulnerabilities in the development pipeline that may compromise the security and functioning of the AI system once trained and deployed.

The second phase is Training of the AI where the AI system is taught to transform a certain input into a desired output. Vulnerabilities in this phase have to do with the training of the AI models being used in the system, but not the tools and design of the system itself. Common examples of vulnerabilities here are poisoned training sets (He *et al.*, 2022), both publicly available ones as well as private versions or sets, unrepresentative training sets, either due to accidental or intentional omissions or other errors, and poisoning federated learning schemes (Sun *et al.*, 2022), where unrepresentative or adversarial training data is fed into an AI system to affect the functioning of versions in use by other actors.

The third phase is deployment of the AI system, both including the act of deploying the system into real use as well as eventual use and maintenance of the system during its lifecycle. Vulnerabilities here have to do with how the AI models and systems handle user input, are susceptible to misdirection and sabotage through regular use channels of the system, and other channels available during the deployment and use of the system. Common examples of vulnerabilities affecting this phase of the lifecycle are model inversions, training data inversion, and adversarial example instability.

3.2 Potentially Compromised Attributes of Trustworthy AI

Secondly, to evaluate how the vulnerability affects the use and security of the AI system, it should be determined which attributes of a trustworthy AI system are broken or compromised by exploiting the vulnerability. The attributes of AI trustworthiness we propose to use for this in a general taxonomy are aligned with the ENISA Multilayer framework for good cybersecurity practices in AI to allow the taxonomy to be used in a wider setting and to maintain cross-compatibility across the industry. Some exceptions to attributes presented in the ENISA framework not included here are detailed later.

- **Accuracy:** Accuracy of the prediction made by the model as opposed to reality. While many vulnerabilities can affect the accuracy of the AI system, vulnerabilities affecting mainly the accuracy of the system are those where there is a mismatch between the AI model and causality in the real world, for example due to improper training of the model, or unaccounted for deviations in the sensor feeds given to the AI system, thus distorting it's understanding of reality.
- **Fairness:** Fairness represents the equitable distribution of actions and decisions made by the system, such that evidence, and not biases, opinions, emotions, or other limitations either intentionally or unintentionally placed in the system affect the decision making process and results of the system. Vulnerabilities affecting mainly Fairness of the AI system introduce bias either intentionally or unintentionally while being covert from the users of the AI system such that the system makes somehow biased decision not based only on the evidence or input it has received to make the decision.
- **Privacy:** Privacy of the data and models used in the creation of the AI system, as well as respecting the personal privacy of the users and subjects of the AI system are gathered under the attribute of privacy. Privacy vulnerabilities cause unintentional leaking of private data, either personal data for users of the AI system, or some details of the AI system or its input data that is not generally available, such as model weights, training data, or system usage data as examples.
- **Reliability:** The ability of the AI system to consistently produce similar results within statistical error margins for similar inputs are main factors of reliability. Vulnerabilities affecting mainly the reliability of the AI system introduce enough random noise into the system such that the results of the system cannot be relied upon to be repeatable for similar data inputs or cause it to be vulnerable to non-obvious edge cases where the output of the system unforeseeably flips into other than expected states.
- **Resiliency:** The ability of the AI system to function even under attack by an adversary, as well as to recover to a regular operating state after an attack has been suffered. Resiliency vulnerabilities hamper the AI systems ability to function properly while under attack or make it so that the system requires significant effort to recover into a normal operational state after such an attack.
- **Robustness:** The ability of the AI system to function at the minimum acceptable operational efficiency even while under attack. Prime example of a vulnerability affecting the robustness of the AI system is a vulnerability that allows easy Denial of Service attacks against the system or otherwise allows degrading the performance metrics of the system to unacceptable levels.
- **Safety:** Prevention of harm to humans, environment, or society in general. Generally vulnerabilities affecting the safety attribute are such that make existing safety interlocks and other safety systems embedded in the AI system to fail to function as intended, allowing the system to perform actions generally not allowed in its regular operational envelope.

Attributes not presented here are Accountability, which is not used as any technical vulnerability causes the originating entity to lose the ability to ensure proper functioning of the system, Explainability, as any system with a vulnerability does not function as intended, so it cannot be explained, Security, as all security vulnerabilities by definition affect the security of the system, and Transparency, as a system acting not in accordance with the intention of its creator cannot really be transparent about its functioning.

3.3 Possible Effect of the Vulnerability Being Exploited

Third and last step in evaluating the vulnerability should be assessing the possible level of control or effect that exploiting the vulnerability allows the attacker. Here we propose a seven step scale for evaluating the effect the vulnerability can have, based on how far outside the bounds of the original design and how well under the control of the attacker/exploiter of the vulnerability the AI system functions. The proposed scale starts at zero to allow informational level effects to also be monitored within the framework.

1. **AI performs correctly:** At this level there is little to no effect on the functioning of the AI system in general, or other safety controls make sure that any deviation from regular use is corrected before any meaningful change from normal operation is able to manifest. This level is to be used for mostly informational vulnerabilities so they can be monitored in case some other vulnerability or change in environment makes actual exploitation of the vulnerability more feasible or impactful.
2. **AI functions within normal operational envelope, but at reduced capacity:** Here the normal functioning of the AI system has been affected in some way, but it still manages to function as it is intended to function, just in some way reduced capacity. Examples are degradation of prediction accuracy, increase in noise of the results, slowness in response times, excessive data/compute usage running up abnormal operating expenses, etc.
3. **AI performs no action, or becomes unavailable:** At this level the normal operation of the AI system is denied to it's users and the organisation utilising it, either by making the system unresponsive in totality by overloading the AI system or its functions, or affecting the underlying infrastructure in a way that creates a Denial of Service situation for the system as a whole. Examples include logic bombs, DoS attacks on infrastructure and open APIs, inability to discard malformed data causing ingestion methods to clog up, degrading analysis enough that no meaningful decision on data can be made, etc.
4. **AI performs unintended, undirected actions within the bounds of the system:** At this point the attacker is able to cause the AI system to take actions, but is unable to cause it to take actions that it wouldn't normally be expected to take, while also not being able to direct which actions precisely the system takes. Examples include affecting self-driving systems in such a way that the vehicle acts erratically, but being unable to cause the system to take specific actions, or causing an AI system to erroneously evaluate loan applications, but not being able to cause certain loans to be accepted, or others to be rejected.
5. **AI performs unintended, directed actions within the bounds of the system:** Here the attacker is able to direct what the AI system does, or how it acts, but is unable to break out of the bounds of the regular operating environment of the AI system. Examples include being able to cause a self-driving system to brake or accelerate on command, or causing an AI system processing loan application to process certain loan applications such that the attacker is able to always cause certain applications to be approved or disapproved if they wish.
6. **AI performs unintended, undirected actions out of bounds of the system:** At this level the attacker is able to break out of the AI system itself through the AI system and perform actions outside the AI system itself, although not being able to fully control the actions. Examples include for example being able to corrupt databases connected to the AI system, or causing an AI powered robot to flail around uncontrollably without obeying regular limitations on joint movement or movement envelope.
7. **AI performs unintended, directed actions out of bounds of the system:** Here the attacker is able to essentially gain full control of the AI system, both to remove regular limitations on actions the AI system is able to take, as well as possibly breaking out of the system itself to run commands on the machines running the system. Examples include being able to install additional software on the machines running the AI system, or gaining full control of the system while being able to disable all safety systems that would limit system usage.

3.4 Completed Evaluation

The above three step procedure will give us a result that has three parts: Attack vector (i.e. where in the lifecycle the vulnerability presents itself), impacted attributes and level of impact (on a seven-step scale). It is important to note that the exact value of the severity is not necessarily the level on the impact scale. For example, in a real-time scenario an attack that causes a denial of service in the collision avoidance system of an autonomous vehicle can be much more severe than an attack that gives the attacker some control on the underlying infotainment system in the vehicle. In our taxonomy these would be on levels 3 and 7 respectively on the impact scale.

However, the taxonomy will help put new and old vulnerabilities in a systematic form and can help decision makers in determining what are the implications in their context. It is also meant as a functional framework that discussion and data collection on AI vulnerabilities can be done in a structured and organised fashion.

3.5 Examples of Application

To provide a few examples on how the proposed taxonomy can be used to evaluate and enumerate vulnerabilities, two existing articles providing detailed examples of AI/ML vulnerabilities were chosen and evaluated. The examples are not vulnerabilities in specific systems but possible ways to attack them, but they should still serve as a good example on applying the proposed framework. For exact vulnerabilities we would either need gain access to information not currently available to the public on existing vulnerabilities or rely on information available in public databases that do not contain all the relevant information for classification as detailed above in explaining why a new framework is needed.

3.5.1 Hardware Trojan Attack by Hou et al. (2024)

In their recent article published in Micromachines, Hou et al proceed to demonstrate a possible attack on Convolutional Neural Networks (CNN) through trojanized FPGA (Field Programmable Gate Array) hardware (Hou *et al.*, 2024).

In the first stage the vulnerability would be considered to be located in the development phase of the AI system, as it affects the physical hardware being used to run the CNN, and is done by creating poisoned (trojanized) hardware that could be inserted into the supply chain of the system development organization.

In the second stage the attributes being compromised include Accuracy, Fairness, Reliability, and Robustness of the system as all these attributes can be compromised with the proposed attack. This is caused by the way the demonstrated attack changes the decision-making process of the system as the calculations that are performed are different from the ones defined in the original code and design of the original implementer on the trojanized hardware.

In the third stage the vulnerability would be evaluated to be at level 1 or 2, depending on the amount of change the trojanized FPGA is able to cause in the calculations and how the CNN is being used in the system. The specific level of effect enabled by the vulnerability depends here on the actual implementation of the system, so that a more definitive determination is not possible on such a generic attack.

3.5.2 Robust Physical-World Attack by Eykholt et al. (2018)

In their article from 2018 in IEEE/CVF Conference on Computer Vision and Pattern Recognition, Eykholt et al proceed to demonstrate several different attacks on traffic signs in the physical world that cause Deep Neural Net (DNN) image classifiers trained on the LISA and GTSRB training sets to misclassify the attacked traffic signs in similar situations as would be encountered by self-driving vehicles in regular road conditions (Eykholt *et al.*, 2018).

In the first stage of the evaluation the phase of the vulnerability would be deployment, as the proposed attack method is clear intended to be applied into traffic signs in the world at large to be encountered by self-driving systems during their normal operation.

In the second stage the attributes under attack are Accuracy and Safety as both the accuracy of the image detection is being attacked, as well as the end result of making traffic signs unreadable by autonomous vehicles is to make them ignore traffic rules and thus compromise safety of both the self-driving vehicle as well as all other users on the road.

In the third stage the effect of the attack is undirected actions within the normal operational envelope, or level 3, as even though stop signs are shown to be misclassified as speed limit 45 or 80 km/h signs, the fact itself doesn’t yet mean that the self-driving vehicle would accelerate to that speed on seeing the sign. Then again the same attack method could be used to make speed limit signs to be misclassified as stop signs, which would then place the vulnerability on level 4, as the most likely effect of a self-driving vehicle on seeing a stop sign is to brake to come to a stop before it.

4. Discussion

Although the above taxonomy gives a great tool for better understanding the AI/ML vulnerabilities, there are still many areas for improvement and further research. As mentioned in the previous section, our taxonomy has impact levels, but these cannot be directly translated into severity of the vulnerability. In traditional vulnerability classification the CVSS (Common Vulnerability Scoring System) is used to give a numerical value for the severity of vulnerability. This helps system administrators and other decision makers to prioritise mitigations and make decisions related to patching the vulnerability in their organisations. CVSS is far from perfect and has received some criticism. It has also been updated over the years (current CVSS is version 4, with the latest update coming out in the beginning of November of 2023) to better reflect the developments in vulnerabilities.

One further development area is to build a similar system for scoring AI/ML vulnerabilities. This could be done in the context of our new taxonomy. One difficulty for this is the fact that the context in which the vulnerability is manifested can be very different and the implications can also vary greatly. On the other hand, in traditional vulnerabilities the context (e.g. the affected software or hardware components) is part of the Common Vulnerability Enumeration entry and the severity can be evaluated in that context.

Another interesting venue for further research is developing systems and platforms for testing AI/ML systems for vulnerabilities. Again, in the traditional vulnerability research domain there are many different tools and platforms (even whole OSes) dedicated to testing systems for vulnerabilities. In the AI/ML domain these tools and platforms are still developing. In order to improve the security of these systems there is a need for not only taxonomies and scoring methods to evaluate newly found vulnerabilities but also tools and platforms to test new AI/ML systems against known vulnerabilities.

5. Conclusion

In this paper we have reviewed the present efforts in creating a framework for security vulnerabilities in AI/MLL use cases and found that all current options have deficiencies. We've proposed a new taxonomy to use as a basis for a new effort into creating an encompassing vulnerability taxonomy for artificial intelligence systems and provided two short examples on application of the evaluation. This work can best be continued by evaluating existing vulnerabilities with the proposed taxonomy in a systemic fashion to show the improvements over existing efforts, as well as developing a vulnerability scoring system based on the proposed taxonomy.

Acknowledgements

We would like to thank the Matine research group at Oulu University for their ideas and conversations surrounding this topic. The research has received funding from the 2023 grant application of the Scientific Advisory Board for Defence.

References

AI Risk and Vulnerability Alliance (2023) *AVID*. Available at: https://avidml.org/ (Accessed: 14 January 2024).

Carlini, N. (2023) 'A LLM Assisted Exploitation of AI-Guardian'. arXiv. Available at: http://arxiv.org/abs/2307.15008 (Accessed: 17 August 2023).

Comiter, M. (2019) 'Attacking Artificial Intelligence'. Available at: https://www.belfercenter.org/publication/AttackingAI (Accessed: 17 August 2023).

European Union Agency for Cybersecurity (2023) *A multilayer framework for good cybersecurity practices for AI :: security and resilience for smart health services and infrastructures.* LU: Publications Office. Available at: https://data.europa.eu/doi/10.2824/588830 (Accessed: 28 September 2023).

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T. and Song, D. (2018) 'Robust Physical-World Attacks on Deep Learning Visual Classification', in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA: IEEE, pp. 1625–1634. Available at: https://doi.org/10.1109/CVPR.2018.00175.

FIRST (2023) 'Common Vulnerability Scoring System version 4.0 Specification'. Available at: https://www.first.org/cvss/v4.0/specification-document (Accessed: 14 January 2024).

Hansen, R. and Venables, P. (2023) *Introducing Google's Secure AI Framework*, *Google*. Available at: https://blog.google/technology/safety-security/introducing-googles-secure-ai-framework/ (Accessed: 17 August 2023).

He, Y., Shen, Z., Xia, C., Hua, J., Tong, W. and Zhong, S. (2022) 'SGBA: A Stealthy Scapegoat Backdoor Attack against Deep Neural Networks'. arXiv. Available at: http://arxiv.org/abs/2104.01026 (Accessed: 15 January 2024).

Hou, J., Liu, Z., Yang, Z. and Yang, C. (2024) 'Hardware Trojan Attacks on the Reconfigurable Interconnections of Field-Programmable Gate Array-Based Convolutional Neural Network Accelerators and a Physically Unclonable Function-Based Countermeasure Detection Technique', *Micromachines*, 15(1), p. 149. Available at: https://doi.org/10.3390/mi15010149.

Kathikar, A., Nair, A., Lazarine, B., Sachdeva, A. and Samtani, S. (2023) 'Assessing the Vulnerabilities of the Open-Source Artificial Intelligence (AI) Landscape: A Large-Scale Analysis of the Hugging Face Platform', in *2023 IEEE International Conference on Intelligence and Security Informatics (ISI)*. *2023 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Charlotte, NC, USA: IEEE, pp. 1–6. Available at: https://doi.org/10.1109/ISI58743.2023.10297271.

Ladisa, P., Ponta, S.E., Ronzoni, N., Martinez, M. and Barais, O. (2023) 'On the Feasibility of Cross-Language Detection of Malicious Packages in npm and PyPI', in *Proceedings of the 39th Annual Computer Security Applications Conference*. New York, NY, USA: Association for Computing Machinery (ACSAC '23), pp. 71–82. Available at: https://doi.org/10.1145/3627106.3627138.

National Cyber Security Centre (2023) 'Guidelines for secure AI system development'. Available at: https://www.ncsc.gov.uk/collection/guidelines-secure-ai-system-development (Accessed: 14 January 2024).

OWASP Foundation (2023) *OWASP AI Security and Privacy Guide*. Available at: https://owasp.org/www-project-ai-security-and-privacy-guide/ (Accessed: 17 August 2023).

Qiu, S., Liu, Q., Zhou, S. and Wu, C. (2019) 'Review of Artificial Intelligence Adversarial Attack and Defense Technologies', *Applied Sciences*, 9(5), p. 909. Available at: https://doi.org/10.3390/app9050909.

Sun, G., Cong, Y., Dong, J., Wang, Q., Lyu, L. and Liu, J. (2022) 'Data Poisoning Attacks on Federated Machine Learning', *IEEE Internet of Things Journal*, 9(13), pp. 11365–11375. Available at: https://doi.org/10.1109/JIOT.2021.3128646.

The MITRE Corporation (2023a) *CVE - Common Vulnerability Enumeration*. Available at: https://cve.mitre.org/ (Accessed: 17 August 2023).

The MITRE Corporation (2023b) *CWE - Common Weakness Enumeration*. Available at: https://cwe.mitre.org/ (Accessed: 17 August 2023).

The MITRE Corporation (2023c) *MITRE ATT&CK®*. Available at: https://attack.mitre.org/ (Accessed: 23 January 2024).

The MITRE Corporation (2024) *MITRE | ATLAS^{TM}*. Available at: https://atlas.mitre.org/ (Accessed: 17 August 2023).