Trustworthiness for AI in Defence (TAID)

-

Risk Assessment Guideline for AI-Based Systems Annex

Submission date: 09/05/2025

Document version: 1.0

Authors: TAID Working Group (TAID WG)

TABLE OF CONTENTS

TABLE OF CONTENTS 2

TABLE OF FIGURES 3

1. SECURITY AND SAFETY RISKS OVERVIEW 4

1.1. SECURITY RISKS 4

1.1.1. State of the Art 4

1.1.2. Attack and Defence Techniques 5

1.1.3. Attack Strategies 5

1.1.4. Defence Techniques 6

1.2. SAFETY RISKS 6

1.2.1. Accuracy and Robustness 7

1.2.2. Integrity 7

1.2.3. Regulations 7

1.2.4. Reliability 7

1.2.5. Autonomy 7

1.2.6. Interoperability and Coordination 8

2. RISK ASSESSMENT GUIDELINE SECURITY AND SAFETY RISKS OVERVIEW 9

2.1. ENGINEERING/UNDERSTANDING 9

2.1.1. Wrong Test Data 9

2.1.2. Wrong Architecture 9

2.1.3. Data Distribution is not a good approximation of the real world 10

2.1.4. Unreliable Confidence Information 10

2.1.5. Inadequate Composition of test and training data 11

2.1.6. Inadequate Composition of test and training data 11

2.2. SYSTEM DEVELOPMENT 11

2.2.1. Incomprehensible Behaviour 12

2.2.2. Unknown Behaviour in rare critical situations 12

2.2.3. Define Meaningful Metrics for Validation 13

2.2.4. Adversarial AI Attacks (Risk) 13

2.2.5. Evasion Attacks 14

2.2.6. Poisoning Attacks 14

2.2.7. Oracle Attacks 15

2.3. PRODUCTION/DEPLOYMENT 15

2.3.1. Data Drift 15

2.3.2. Hardware Failures 16

2.3.3. System does not get validated 16

2.3.4. Biased Model 17

2.4. SOCIOTECHNICAL RISKS 17

2.4.1. Dataset Sovereignty *17*
2.4.2. Degradation of Skills *18*
2.5. MITIGATION STRATEGIES CLASSES 20
3. BIBLIOGRAPHY 21

TABLE OF FIGURES

Figure 1 - Sample showing that attacks can target different phases of the lifecycle of a machine learning model 14

1. Security and safety risks overview

1.1. Security risks

1.1.1. State of the Art

When discussing Artificial Intelligence, it is advisable to commence with Deep Learning, which enables the resolution of tasks with performances comparable to, and in some cases surpassing, human capabilities. However, the substantial potential demonstrated by these systems can be compromised by the utilization of adversarial attacks. This type of vulnerability can become highly precarious in applications, such as military ones, wherein the reliability of artificial intelligence systems is of paramount importance. Therefore, it is imperative to maintain a clear understanding of the vulnerabilities inherent in these models to assess their robustness and level of security.

With the simultaneous evolution of deep neural networks and high-performance hardware for training complex models, deep learning has made significant strides in traditional fields such as image classification, speech recognition, and language translation, as well as cutting-edge areas like critical security and protection environments, malware detection, autonomous driving, drones, or robotics. As deep neural networks have transitioned from laboratories to real-world applications the security and integrity of these applications represent significant concerns. Adversaries can manipulate inputs provided to the models to be imperceptible to the human eye but deceive a trained model into producing erroneous outputs.

In addition to attacks occurring during the model testing phase, an increasing number of studies are dedicated to developing attacks during the training phase. Nevertheless, after the discovery of these attacks various countermeasures have been proposed to mitigate the effects of adversarial attacks, such as the notion of employing adversarial training to safeguard the network during learning by augmenting training data with perturbations. The concept of distillation can also be discussed, utilized to propose a defence mechanism against adversarial attacks. Although each of these proposed defence mechanisms has proven efficient against particular classes of attacks, none of them can serve as a singular solution for all vulnerabilities. Moreover, the implementation of these defence strategies may lead to a degradation of the overall performance and efficiency of the model.

In conclusion the analysis of vulnerabilities to adversarial attacks in machine learning and deep learning becomes fundamental in applications where these models are utilized in contexts where security and reliability are essential. Artificial intelligence is revolutionizing the way we process and analyse data and develop automated systems, but it is imperative to be conscious of the vulnerabilities that a potential adversary could exploit to circumvent or, worse, control these systems at will. All these aspects underscore the inherent lack of security and robustness of deep learning systems. Since these are integrated into various scenarios where security is paramount (such as in automatic driving systems and facial recognition), it is crucial to enhance their robustness. Therefore, delineating a comprehensive benchmark on the robustness of deep learning models and the influence that architecture and training techniques exert on it paves the way for a critical path to understanding and improving robustness.

While this initial section has served to provide the reader with a comprehensive overview of the state of the art, henceforth, we shall focus on the objectives of this project: the study of evasion attacks that can be employed to target pre-trained artificial vision models.

1.1.2. Attack and Defence Techniques

Turning now to defence and attack techniques, due to the continuous development of AI algorithms, cyber threats have scaled their technological capabilities. Indeed, hackers have been targeting these algorithms in order to exploit their vulnerabilities. The aims of such attacks is maximizing generalization errors, corrupting training, and, possibly, altering the previsions generated by models. In spite of that, Machine Learning has gained considerable reputation in contrasting such threats among industry and academia practitioners. Its main advantage, besides malware detection, consists in expanding current knowledge about detecting newly-developed and obfuscated malwares with autonomous adaptation capabilities.

1.1.3. Attack Strategies

In this respect, threat actors employ three main attack strategies: evasion, poisoning and oracle attacks.

Evasion attacks, also known as explorative attacks, are launched during test or inference phases. Thus, attackers aim to mistake previsions calculated by automatic learning model after the training has taken place.

Poison attacks, also known as causative attacks, aim to corrupt training data and the logic of the model during the training phase in order to induce a wrong prediction by injecting malicious or manipulated data or by altering the logic of the algorithm.

Threat actors perform oracle attacks by substituting the model as a whole or by accessing the model's predictions or outputs and trying to extract information about the model or its training data by making queries to it. Moreover, the substitute model typically retains a significant portion of the functionality of the original model. These attacks can be further categorized into extraction, inversion, and inference attacks. The goal of an extraction attack is to deduce details of the model architecture, such as its parameters. Inversion attacks occur when the adversary attempts to reconstruct the training data. An inference attack allows the adversary to extract sensitive information or model properties by leveraging the model's responses to queries or the ability to access its predictions.

1.1.4. Defence Techniques

In order to defend Machine Learning models from cyber threats, strategies enhancing the robustness and the resistance of Machine Learning should be adopted. To do so, scientific literature has focused on developing remedies such as data clean, model defence, and adversarial defence. It is likely that deploying a mix of these remedies is much more effective in achieving a higher level of protection and in keeping the highest performance possible even in presence of malicious input rather than deploying them alone.

First, the data clean method aims to identify and remove malicious samples within the training or test dataset.

Secondly, another option to mitigate attacks in the training phase is implementing a training algorithm that limits the interference of malicious samples on the system. Such an approach is called robust training and it belongs to the category of model defend.

Finally, adversarial defence, the last class of defensive techniques consists of: gradient masking, where the machine learning model is obfuscated so that an attacker cannot know its gradient; adversarial training aims to increase the robustness of an automatic learning model, by training the model on a set of malicious data; gradient regularization which affects the considerable changes in output levels of the neural network in order to reduce to a minimum the loss function; feature reduction has the goal to simplify the model in order to decrease the number of independent variables and thus the attack surface; input randomization, finally, is based on adding noise or causal perturbations to the input data before the machine learning model processes them in order to increase the difficulty of generating a manipulation that can deceive the model.

1.2. Safety Risks

The increasing integration of AI on the defence sector raises challenges to make AI trustworthy, ensuring safety in critical and potentially hazardous environments.

The aim of these challenges is to reduce safety risks to the lowest level possible, helping the prevention of harm. These safety risks gather all of those risks that may affect humans, equipment or environment and can come from functionality failure of the system, human error or bad design.

When discussing about prevention of harm, AI trustworthiness[1] can be related to the following aspects:

- Accuracy and robustness
- Accountability
- Privacy and Security
- Integrity

[1] Requirements for Trustworthy Artificial Intelligence – A review. Davinder Kaur, Suleyman Uslu and Arjan Durresi; Department of Computer and Information Science, Indiana University and Purdue Universaity, IN, USA {davikaur,suslu}@iu.edu, adurresi@iupui.edu)

- Reproducibility
- Regulations

From these shall be considered to ensure safety:

- Accuracy and robustness.
- Integrity and regulations.

Moreover, reliability, AI autonomy and interoperability and coordination of the systems shall also be addressed.

1.2.1. Accuracy and Robustness

Accuracy refers to the measure of the precision of an AI model in performing a specific task; robustness refers to the ability of an AI model to maintain its expected performance and behaviour under a wide variety of conditions, including those that may be different from the conditions under which it was trained.

When the AI model is highly accurate, it ensures that the system makes the right decisions when they need to be made. When combined with robustness, it helps to ensure that the decision remains correct when adversity strikes.

1.2.2. Integrity

Integrity refers to the reliability and consistency of data, models and processes used in AI systems. Guaranteeing data integrity, in training and during operations of the AI model, prevents malicious tampering or contamination of data that could compromise the Safety of the operations.

1.2.3. Regulations

Regulations are necessary to have unified laws and guidelines. In the context of AI models, such regulations are necessary to define what makes a safety risk tolerable or not, and define how to find, define, manage and reduce them.

1.2.4. Reliability

Reliability refers to the capacity of an AI model to make specific tasks on an accurate and predictable way under a given set of conditions. The system needs to be reliable on avoiding mistakes and reacting to unexpected behaviours.

1.2.5. Autonomy

Autonomy refers to the ability of the AI model to make decisions and perform actions without direct human intervention. The level of autonomy depends of the intervention of the human.

It is important to strike a balance between the autonomy of the AI model and human intervention to help prevent safety risks and guarantee safe and reliable operations.

1.2.6. Interoperability and Coordination

Interoperability and coordination between collaborative AI systems are a must. The cooperation of the different systems helps to ensure safety and avoid potentially dangerous situations.

Some important considerations on interoperability and coordination to ensure process safety would be to work with effective communication in order to exchange necessary information and coordinate actions in real time; security on the communication to protect the information exchange between the collaborative systems and avoid attacks or manipulations; and resilience to failure to make sure that the process will continue to operate safely even if some of the systems in the interaction fail.

Once AI models comply with all the properties, it only remains to address the issue of risk management and mitigation. For that matter is essential to make Safety assessments following the appropriate standards on each case and finding design or procedure mitigations to reduce risk to the lowest possible level; this requires thorough analysis of systems, risk assessment and rigorous testing, compliance with safety requirements, and the development of guidelines and regulations to define risk levels based on the autonomy of AI models, and the likelihood and severity of potential hazards.

2. Risk Assessment Guideline Security and safety risks overview

2.1. Engineering/Understanding

In this phase the system is designed, the dataset is identified, and the algorithm is ready to be trained.

2.1.1. Wrong Test Data

The wrong data got collected for testing, the test dataset is no feasible representation of the deployment domain. This can be the result of a not defined or a wrongly defined target domain.

2.1.1.1. Damage

The system is fitted to the wrong domain and there is a high chance that a mediocre model got trained.

2.1.1.2. Metrics

- Bias [TAID-44]
- Reliability [TAID-22]

2.1.1.3. Mitigation

- [M13] Implementation of a Quality Assurance process on the dataset.

2.1.2. Wrong Architecture

In the development process, a suboptimal model architecture is chosen. This can, for example, be the case if the requirements are not defined or chosen incorrectly.

2.1.2.1 Damage

The system is sub performing for its meant target domain. This could mean the performance or throughput is not up to the real-world requirements.

2.1.2.2. Metrics

- Explainability [TAID-12]
- Robustness [TAID-28]

2.1.2.3. Mitigation

- [M11] Make evaluations on the hardware to choose.
- [M13] Implementation of a Quality Assurance process on the system architecture.

2.1.3. Data Distribution is not a good approximation of the real world

Different factors could lead to this issue, for example:

- Different climatic conditions or differences between daylight and night in test data and real-world data.
- Bias (over- or under-representation of certain factors).
- Physical differences like sensor position or resolution.
- Sparse data.

2.1.3.1. Damage

Insufficient performance or robustness when used on the field. Edge cases can be a real risk because system behaviour will not be predictable.

2.1.3.2. Metrics

- Bias [TAID-44]
- Reliability [TAID-22]
- Reproducibility [TAID-25]

2.1.3.3. Mitigation

- [M1] Collect data that represent not just the final working conditions but also possible variations in the data acquisition (light, sensor position, …). Use synthetic data for data gaps (simulated data, artificial data, ...).
- [M2] Use metrics to measure the reliability of the trained algorithm.
- [M3] Test the trained algorithm with data that represents the real-world conditions as much as possible. Use perturbations effect (blur, brightness, …) and synthetic data to test robustness.
- [M4] Analyse the test results and consider a retraining process in order to include weak or wrong predictions.
- [M5] Define and follow labelling guidelines to ensure that the data used cover as much of the real-world conditions as possible.
- [M12] OOD detection in deployment.

2.1.4. Unreliable Confidence Information

The system might not give an accurate prediction because the training data were poorly distributed or because of uncertainty in the data itself (motion blur, rain, …)

2.1.4.1. Damage

The system might give high confidence to predictions that are not correct.

2.1.4.2. Metrics

- Reliability [TAID-22]

- Reproducibility [TAID-25]

2.1.4.3. Mitigation

- [M2] Use metrics to measure the reliability of the trained algorithm.
- [M3] Test the trained algorithm with data that represents as much as possible the real world conditions.
- [M7] Force the algorithm to provide more meaningful results in order to better understand the reasons that led to a certain prediction.

2.1.5. Inadequate Composition of test and training data

Training data and test data should be different but equally representative of real-world conditions. Using test data collected in the same way as the training data might lead to an overestimation of the predictions' accuracy and can result in a “leakage”.

2.1.5.1. Damage

The accuracy of predictions is lower when starting to use the trained algorithm on real-world data.

2.1.5.2. Metrics

- Bias [TAID-44]
- Reliability [TAID-22]

2.1.5.3. Mitigation

[M9] Define and follow guidelines for data partitioning, then test for data leakage.

2.1.6. Inadequate Composition of test and training data

For supervised learning, the quality of predictions is directly dependent on the labelling quality performed on the training data set.

2.1.6.1. Damage

Poor or wrong algorithm predictions.

2.1.6.2. Metrics

- Bias [TAID-44]
- Explainability [TAID-12]
- Reliability [TAID-22]

2.1.6.3. Mitigation

- [M5] Define and follow guidelines for data labelling.

2.2. System Development

The system is trained and then validated.

2.2.1. Incomprehensible Behaviour

It is unclear what is causing wrong algorithm predictions because of the algorithm's complexity.

2.2.1.1 Damage

The system outputs are not as expected.

2.2.1.2 Metrics

- Controllability [TAID-11]
- Explainability [TAID-12]
- Transparency [TAID-36]

2.2.1.3 Mitigation

- [M7] Force the algorithm to provide more meaningful results in order to better understand the reasons that led to a certain prediction.

2.2.2. Unknown Behaviour in rare critical situations

It is difficult to collect data (training and testing) for conditions that are very unlikely to happen or reproduce. Then, it is difficult to monitor the prediction behaviour in these situations.

2.2.2.1 Damage

The system might follow a wrong or unpredictable behaviour when a rare situation happens.

2.2.2.2 Metrics

- Explainability [TAID-12]
- Robustness [TAID-28]
- Testability [TAID-33]
- Resilience [TAID-03]

2.2.2.3 Mitigation

- [M1] Collect data that represents all the possible situations despite their probability to happen or use synthetic data for recreating and testing rare critical situations.
- [M2] Implement model monitoring for models in deployment.
- [M3] Test the algorithm with as much complete data as possible, even if they have a low probability of being encountered.
- [M4] Analyse the test results and consider a retraining process to include weak or wrong predictions and rare situations.
- [M6] Update or retrain the model when finding new input data.

2.2.3. Define Meaningful Metrics for Validation

It might be hard to define metrics that are meaningful for validating an AI system.

2.2.3.1 Damage

Loose metrics can lead to poor safety of the system, while metrics that are too strict might lead to an inefficient system.

2.2.3.2 Metrics

- Controllability [TAID-11]
- Explainability [TAID-12]
- Testability [TAID-33]
- Transparency [TAID-36]

2.2.3.3 Mitigation

- [M10] Performance and metrics should be adapted to the context. It might not be possible to apply a validation metric as it is, but it may change regarding the context, or more than one metric can be combined to achieve the validation performance (e.g. if validating an image recognition algorithm, it is different if the system takes a single picture or a video, because the misclassification of an object in a single frame of the video might not compromise the application, while misclassifying the only one taken picture can lead to a wrong prediction).

2.2.4. Adversarial AI Attacks (Risk)

An adversarial example is where an extremely small change made to the input to a neural network produces an unexpected (and wrong) large change in the output (i.e. a completely different result in respect to the unchanged inputs).[2]

[2] From ISO/IEC TR 29119-11:2020

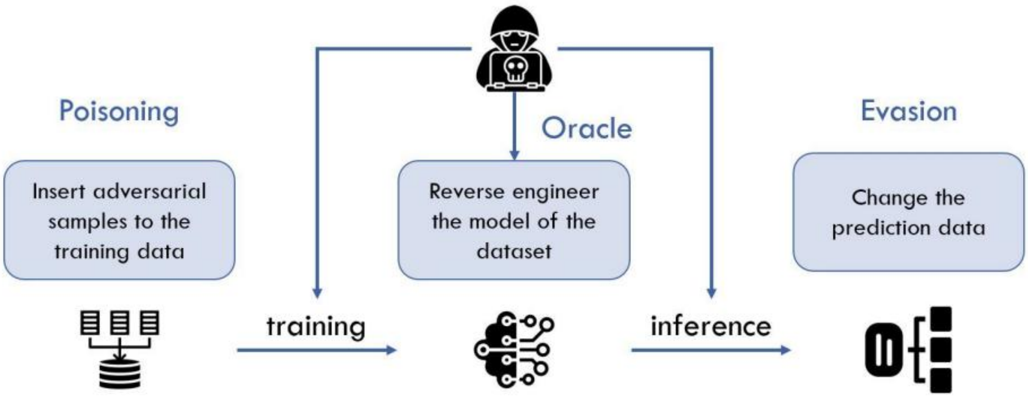
Poisoning
Insert adversarial samples to the training data
Oracle
Reverse engineer the model of the dataset
Evasion
Change the prediction data
training
inference

Figure 1 - Sample showing that attacks can target different phases of the lifecycle of a machine learning model

2.2.5. Evasion Attacks

Also known as *explorative attacks*, they are launched during the test or inference phases. Thus, attackers aim to mistake previsions calculated by automatic learning model after the training has taken place. This kind of attack consists of making small changes to an input to keep it normal for a human but causing misclassification to the AI.

2.2.5.1 Damage

Input data can be misclassified, making it hard for a human to understand the issue.

2.2.5.2 Metrics

- Explainability [TAID-12]
- Reliability [TAID-22]

2.2.5.3 Mitigation

- [M2] Implement model monitoring for models in deployment.
- [M3] Include manipulated data in the training set with proper classification.
- [M8] Specify the adversarial threat model and incorporate defence mechanisms like including adversarial samples in the training or detecting connections in the algorithm that are not commonly activated.

2.2.6. Poisoning Attacks

Also known as causative *attacks*, these aim to corrupt training data and the logic of the model during the training phase in order to induce a wrong prediction by injecting malicious or manipulated data or by altering the logic of the algorithm.

2.2.6.1 Damage

Poisoned training led to higher chances of misclassification at deployment time.

2.2.6.2 Metrics

- Explainability [TAID-12]
- Reliability [TAID-22]

2.2.6.3 Mitigation

- [M1] Verify data origin and use data clean methods aimed to identify and remove malicious samples within the training or test dataset.

2.2.7. Oracle Attacks

These attacks consist in substituting the model as a whole or by accessing the model's predictions or outputs and trying to extract information about the model or its training data by making queries to it. Moreover, the substitute model typically retains a significant portion of the functionality of the original model.

2.2.7.1 Damage

Enemy may have the possibility to degrade performance at a chosen point in time.

2.2.7.2 Metrics

- Robustness [TAID-28]
- Testability [TAID-33]
- Transparency [TAID-36]
- Resilience [TAID-03]

2.2.7.3 Mitigation

- [M2] Implement model monitoring for models in deployment.
- [M3] Implement techniques in data understanding to detect possible data injections.
- [M6] Increase update frequency for newly identified possible attacks.
- [M8] Specify the adversarial threat model and incorporate defence mechanisms like including adversarial samples in the training or detecting connections in the algorithm that are not commonly activated.

2.3. PRODUCTION/DEPLOYMENT

The system has been trained and it is now released in production.

2.3.1. Data Drift

Since the real world is constantly changing, even after “perfectly” training a model, it is possible that the input data will change over time.

2.3.1.1 Damage

The system, over time, starts to deteriorate, misrecognizing new objects or unrecognizing them.

2.3.1.2 Metrics

- Explainability [TAID-12]
- Maintainability [TAID-19]
- Reliability [TAID-22]
- Robustness [TAID-28]

2.3.1.3 Mitigation

- [M2] Implement model monitoring for models in deployment.
- [M6] Retraining or updating of the model. This is sometimes possible in the machine itself (paying attention to un-supervision issues or computational power) or by sending the newly collected data to a central station that will retrain and update the model.

2.3.2. Hardware Failures

There can be a problem in the hardware or in the data network that leads to the repetition, delay or corruption of data.

2.3.2.1 Damage

Data repetition can lead to a repeated output and consequently an incorrect behaviour of the system, while delays or network disturb can cause failure in the data processing or acquisition.

2.3.2.2 Metrics

- Maintainability [TAID-19]
- Availability [TAID-07]

2.3.2.3 Mitigation

- [M11] Hardware redundancy and additional control in the logic when using an AI system must be taken into account in order to minimize possible negative effects caused by hardware failures.

2.3.3. System does not get validated

The AI system might not be suitable for the chosen application or the implementation is not on point.

2.3.3.1 Damage

Systems detections or performances are not up to expectations.

2.3.3.2 Metrics

- Explainability [TAID-12]
- Reliability [TAID-22]
- Transparency [TAID-36]

2.3.3.3 Mitigation

- [M13] Review the project and the QA process.

2.3.4. Biased Model

Potentially unintended distance between the predicted value provided by an AI model and a desired fair prediction.

2.3.4.1 Damage

An AI-based system that demonstrates systematic discrimination against an individual or group of individuals is considered to be showing unfair bias. Bias is normally caused by the machine learning picking up unwanted patterns in the training data. Training data can be compromised by both explicit and implicit bias. Implicit bias is created unintentionally when unknown unwanted patterns in the training data exist. Explicit bias is created when known unwanted patterns in training data influence the derived model.

2.3.4.2 Metrics

- Bias [TAID-44]

2.3.4.3 Mitigation

- [M1] Testing for bias in an AI-based system can be performed at two stages. First, bias can be detected (and subsequently removed) in the training data through reviews, but this requires expert reviewers who can identify possible features that create bias. Second, a system can be tested for bias by the use of independent testing using bias-free testing sets. When we know that training data is biased, it may be possible to remove the source of the bias (e.g. we could remove all information that provided clues to the sex or race of the subjects). Alternatively, we could accept that a system includes bias (either implicit or explicit) but provide transparency by publishing the training data.

2.4. Sociotechnical Risks

2.4.1. Dataset Sovereignty[3]

Control data access with login or biometrics data to prevent unwanted data access or poisoning (also to the storage)

Dataset sovereignty ensures that entities, including organisations, governments, and individuals, retain control over their data, thereby determining the parameters of its collection, storage, dissemination, and utilisation.

[3] Hummel, P., Braun, M., Tretter, M., & Dabrock, P. (2021). Data sovereignty: A review. Big Data & Society, 8(1).

2.4.1.1 Damage

Whereas the sovereignty can be associated to more protection the concept of dataset sovereignty, when implemented, may amplify cybersecurity vulnerabilities especially when data is centralized within a single location. Such concentration facilitates the targeting and potential compromise of data by cybercriminals, thereby exposing security plans or information.

2.4.1.2 Metrics

- Confidentiality [TAID-09]
- Data Integrity [TAID-17]

2.4.1.3 Mitigation

- [M1] Conduce a comprehensive data audit. Organizations should undertake a thorough examination of their data ecosystem, encompassing its storage, processing, and transmission facets. This audit should align with pertinent data protection statutes and regulations, aiding in the identification of potential dataset sovereignty vulnerabilities and ensuring compliance with applicable legal frameworks.
- [M14] Employing dataset localisation strategies: dataset localisation entails the practice of housing data within the jurisdiction of its origin. This approach serves to subject data to the legislative and regulatory frameworks of the respective country of origin. By embracing data localisation strategies, organisations can safeguard sensitive data within the legal confines of their originating jurisdiction.
- [M15] Deploy robust dataset protection measures: organizations ought to implement robust data protection mechanisms, encompassing encryption, access controls, and vigilant monitoring. These measures are instrumental in safeguarding sensitive data against unauthorized access and misuse, encompassing both data in transit and data stored.

2.4.2. Degradation of Skills

The integration of artificial intelligence within the defence sector holds promise for automating tasks traditionally undertaken by human operators or alleviating cognitive burdens. While this transition may yield heightened productivity and operational efficiency, the rapid pace of change entails notable implications for both organizations and personnel, as AI implementation may evoke concerns regarding potential job displacement. Effectively navigating this paradigm shift necessitates the adoption of measures and strategies aimed at enhancing the skill sets of the workforce through upskilling or reskilling initiatives. Consequently, organizations are confronted with significant challenges, including the retention and continuous training of their personnel. This can lead to a skills deficit, which becomes problematic when AI-driven systems cease to be present or function due to any type of error.

2.4.2.1. Damage

After relying too much on an AI system a skilled user can lose its preparation (skills).

2.4.2.2. Metrics

- Transparency [TAID-36]

2.4.2.3. Mitigation

- [M16] Continuous training, both with and without systems, may always be necessary.

2.5. Mitigation Strategies Classes

There are many different mitigation strategies that are detailed in every risk, but they can be regrouped in general concepts as follows:

- [M1] Use solid strategies for data acquisition/creation not only regarding the variety of data but also the possible perturbations (external attack, different weather and light conditions, physical sensors positions, ...). Use strategies for synthetic data creation for filling data gaps.
- [M2] Implement adequate metrics for monitoring the system after its deployment.
- [M3] Test the trained data with even low probability data or artificially perturbated data if it needs to better reflect the application conditions.
- [M4] Retrain and retest the model for weak or wrong predictions.
- [M5] Define and follow guidelines for data labelling.
- [M6] Retrain or update the model after deployment including newly collected data.
- [M7] Force the algorithm to give meaningful results in order to better understand what led to certain predictions.
- [M8] Include self-defence strategies in the algorithm to prevent or respond to adversarial attacks.
- [M9] Define and follow guidelines for data partitioning in order to identify meaningful data for both training and testing. Afterwards, test for data leakage between training set and testing set.
- [M10] Adapt performance metrics and validation rules to the context. AI algorithms do not always give 100% accuracy, it is important to identify whenever a performance is adequate to the application.
- [M11] When using the algorithm, the possibility of issues caused by the hardware must be considered; these negative effects need to be reduced by the use of redundant hardware or additional logic controls in the software.
- [M12] Detect OOD (out of distribution) at deployment time.
- [M13] In machine learning Quality Assurance (QA) isn’t just about the system as a whole but there are many individual aspects that needs to be tested: the data, the model, the algorithm, ...
- [M14] Physically store data in countries where they are protected by the country's jurisdiction.
- [M15] Protect both the physical and logical environment of both the data and the algorithm.
- [M16] Continuous training of the personnel that might operate the system in case of AI malfunction.

3. Bibliography

(1) Willers Oliver, Sudholt Sebastian, Raafatnia Shervin, and Abrecht Stephanie. 2020. Safety concerns and mitigation approaches regarding the use of deep learning in safety-critical perception tasks. In Proceedings of the International Conference on Computer Safety, Reliability, and Security. Springer, 336–350.

(2) Steimers, A.; Schneider, M. Sources of Risk of AI Systems. Int. J. Environ. Res. Public Health 2022, 19, 3641. https://doi.org/10.3390/ijerph19063641.