

Artificial Intelligence (AI) Security Guideline (A-MLSG)

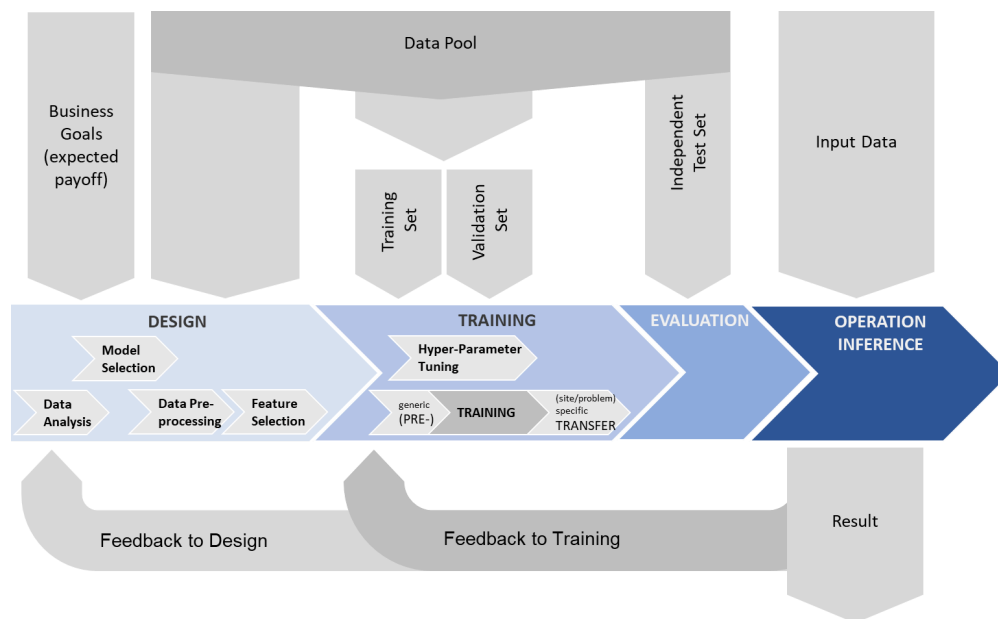
This guideline covers security goals which shall be met for applications applying artificial intelligence (AI) technologies. The guideline focusses mainly on machine learning (ML) based AI. ML and esp. deep learning algorithms have become popular over the past 5 years as availability of training data (networked sensors) and computational power even on small embedded devices (dedicated tensor processing units) increased.

The primary goal of this guideline is to increase the awareness regarding machine learning (ML) specific threats to support the threat and risk analysis for products applying ML technologies. It shall help system architects, data analysts, and software developers intending to integrate ML technology to create secure systems. Moreover, it shall support system operators to ensure the security of ML enabled systems during operation, esp. due to the frequent deployment of ML technology on cloud and edge facilities.

Disclaimer:

- *Discussing ML impacts there is a potential overlap between safety and security. Potential safety impacts induced by ML are not discussed in this guideline. However, note that security threats may result in safety related risks.*
- *ML based threat detection (e.g., intrusion detection systems) is a specific subdomain of AI and not explicitly addressed in this guideline.*

The figure below gives an overview of processes and data streams that are typically present in ML solutions and shall be considered during threat analysis.



The following [Terms & Definitions](#) may help when dealing with AI system security.

Impact due to the absence of the related control(s):

Many security controls known from classical IT/OT systems also apply for AI enabled systems, such as input data validation and integrity, use controls, need-to-know principle. This document provides guidance on how these principles apply and map to AI systems. In addition, it covers several aspects needing special considerations within AI systems. The following lists some attack types that need dedicated consideration:




Adversarial examples	Maliciously perturbed inputs for machine learning models that are designed to mislead the model during inference and create (invalid) results intended by the attacker. This can be used to cause a malfunction in a machine learning model. In contrast to apparently malformed input data adversarial examples are hard to detect by human supervision as they look legitimate to them. Note that adversarial examples are rather easy to compute and can even be transferred between different ML models. Further reading, e.g.: "Explaining and harnessing adversarial examples" ↗, Ian J. Goodfellow et al., "The Space of Transferable Adversarial Examples" ↗, Florian Tramèr et al.
Data extraction	An attacker runs queries on the ML model (using it as a backbox) and tries to reconstruct sensible data or know-how contained in the training data. Further reading, e.g.: "The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets" ↗, Nicholas Carlini et al.
Data poisoning Backdoor Attacks	An attacker tampers with the training data set in order to create an (intentionally) biased ML model or train specific triggers into the model.
Evasion Attack	An attacker tweaks the input data feed into the ML system for inference in a way to cause malfunction (e.g., failure to detect a specific pattern the ML system has been trained on). Discovery of such an attack can be additionally impeded by adversarial examples.
Reverse Engineering	An attacker runs queries on the ML model (using it as a blackbox) and tries to reconstruct the behavior of the ML system (e.g., using a GAN)

Recommended Practices:

- [A-MLSG-001 - Design, Architecture, Overall Considerations](#)
- [A-MLSG-002 - Data Handling](#)
- [A-MLSG-003 - Training](#)
- [A-MLSG-004 - Operation](#)

Further references:

Practices of this document are based on the resources:

- ["SoK: Security and Privacy in Machine Learning"](#) , N. Papernot, et. al., 2018 IEEE European Symposium on Security and Privacy (EuroS&P), London, 2018, pp. 399-414, doi: 10.1109/EuroSP.2018.00035.
- ["Datasheets for Datasets"](#)  Timnit Gebru et. al.,
- ETSI GR SAI 004 ["Securing Artificial Intelligence \(SAI\); Problem Statement"](#) , v1.1.1, 2020 ETSI
- Ongoing work in ISO/IEC SC42: TR 24028 "Trustworthiness in AI", TR 24029 "Robustness of Neural Networks", TR 24027 "BIAS in AI Systems"



Note on PSS TRA Reference

TRA question list does not properly reflect many aspects of AI systems. It has not been designed with data centric technologies and processes in mind. Especially, the TRA questions may not cover the topics to their full extent.

A-MLSG-001 - Design, Architecture, Overall Considerations

Identifier	Control	PSS TRA Reference ↗
A-MLSG-001-C001	<p>Consider the protection of data and IP in the training phase.</p> <p>Cloud or similar 3rd party resources may be used host the required data pool. 3rd party (cloud) services or frameworks are often used implement the training process. Both data and potential IP contained in the chosen ML approach need to be properly protected.</p> <p>See also:</p> <ul style="list-style-type: none"> Secure (Base) System Configuration and Hardening (A-SC) SW - Secure Data Storage / Data at Rest Security (A-SDS) 	SW-25
A-MLSG-001-C002	<p>Consider reducing the exposure of user data during deployment.</p> <p>While training of ML systems still requires a considerable amount of computational power, operational use/inference may also be done either locally (e.g., directly on an embedded device) or on dedicated edge devices. Moving ML services closer to the data source can reduce the exposure of sensitive (customer) data processed by the ML system.</p> <p>See also:</p> <ul style="list-style-type: none"> Secure (Base) System Configuration and Hardening (A-SC) CD - Container Security Guideline (A-CSG) 	CD-13 CD-14
A-MLSG-001-C003	<p>Consider which kind of AI technology will be the most compatible in terms of security considerations.</p> <p>With regards to security, traceability and comprehensibility are important criteria for the decision. Especially in (security) regulated domains explain-ability and transparency of the reasoning for the results of a system might be a challenge and thus some technologies might be preferable over others.</p> <p>Further reading, e.g.: "Interpretable Machine Learning - A Guide for Making Black Box Models Explainable" ↗, Christoph Molnar, March 2019</p>	
A-MLSG-001-N001	<p>Don't ↗ expect that using multiple (redundant) ML models or technologies in parallel will protect your system against adversarial examples.</p>	
A-MLSG-001-N002	<p>Don't ↗ use readily designed or pretrained models of unknown provenance.</p> <p>Pretrained models can contain unknown BIAS or hidden triggers. The internet provides access to readily designed and pre-trained models for common problems addressed with ML (e.g., object classification). Like with OSS software special care shall be taken before using those prefabricated models. This includes, e.g., the transparency of the model design and training data and the validation of the resulting model. If possible, test such a model against self-generated samples (see A-MLSG-002-D002).</p>	SW-39
A-MLSG-001-C004	<p>Consider using site/installation-specific training.</p> <p>This may be of special interest for scenarios where the attacker is able to influence the pattern signature (e.g., anomaly detection). There are scenarios where the attacker tries to evade or circumvent ML algorithms. E.g., in case a neural network is used for anomaly detection (the network will typically be trained to classify normal behavior), attackers will try to ingest a pattern that cannot be detected and escape detection. In this case, inclusion of site-specific training data will impede an attacker in trying to attack new targets using information gained from another attack (esp. in case access to the model is not protected).</p>	CD-10

A-MLSG-002 - Data Handling

Identifier	Control	PSS TRA Reference ↗
A-MLSG-002-D001	<p>Do make sure to comply with regulatory requirements regarding the handling of sensitive data (e.g., privacy regulations such as the General Data Protection Regulation).</p>	SW-25 SW-26
A-MLSG-002-D002	<p>Do use validation data from different data sources than the ones you select your training data from.</p>	
A-MLSG-002-D003	<p>Do assess which kind of bias may be beneficial from the point of view of potential attackers.</p> <p>Use proven statistical methods (e.g., confusion matrix, equality of odds, equality of opportunity) to verify that your training and validation data is free of such bias and does not adversely affect the fairness of your model.</p>	SW-39
A-MLSG-002-N001	<p>Don't use training/validation datasets of unknown provenance.</p> <p>The internet provides access to public/opensource datasets for ML (e.g., Google Dataset Search, ImageNet). Like with OSS software special care shall be taken before using those datasets. Ask yourself questions like:</p> <ul style="list-style-type: none"> Who is the institution hosting the dataset? Who contributed to the dataset? Are there datasets from independent sources that could be used for validation (see A-MLSG-002-D001)? 	SW-39
A-MLSG-002-D004	<p>Do rule out any manipulation of your own training and validation data like changing of existing or injection of additional (malicious) samples.</p> <p>Apply measures (e.g., access control, digital signatures or MACs) to ensure authenticity/integrity of your training data wherever possible.</p>	SW-25

A-MLSG-002-D005	Do keep records of all data used during training and validation. Tracking of data used for ML shall be an integral part of the configuration management for ML system development.	CD-20

A-MLSG-003 - Training

Identifier	Control	PSS TRA Reference ↗
A-MLSG-003-D001	Do clear the training dataset from data features not related to your problem to defend against data extraction. In case the data and the ML model are not protected equally and privacy or confidentiality of the data is an issue there is the question, how much does a ML model tell about the data? There is a wide range of still visible data for all kinds of ML models (ranging from tree-based to ANN). Only data features actually needed to train the system shall be used for training (need-to-know principle towards AI system).	CD-13
A-MLSG-003-C001	Consider using techniques like feature elimination, distillation, or model reduction to reduce the information contained in the ML model (e.g., ANN) to reduce the attack surface (cf. A-MLSG-003-D001).	
A-MLSG-003-C002	Consider using techniques likes anonymization/pseudonymization to preprocess training datasets. Attackers may be able to extract (even detailed) sensitive (personal) data from the resulting model even in case the model has not explicitly been trained to remember those data.	CD-13
A-MLSG-003-D002	Do provide unit tests for the software used to calculate derived variables or extract statistical features used for training. Certain variables may not properly be calculated depending on the input data. For example, an integer overflow during feature calculation will lead to a wrong model or wrong classification results. Identify potential corner stones and boundary values of the input data for model training and model deployment.	
A-MLSG-003-D003	Do validate any kind of (3 rd party) preprocessing software used to prepare the data. Using preprocessing software which is not suitable for your scenario or that has been intentionally tweaked may induce bias or other incorrect results in your ML model.	SW-39

A-MLSG-004 - Operation

Identifier	Control	PSS TRA Reference ↗
A-MLSG-004-D001	Do protect the data passed into the ML system for inference against tampering. Attackers can construct adversarial examples to mislead ML classification systems or trigger hidden features. Protecting the integrity and authenticity of data intended to be processed by the ML system prevents the attacker from inserting adversarial examples.	SW-25 SW-26 SW-27
A-MLSG-004-D002	Do protect the integrity and authenticity of your ML model and configuration (algorithm and weights). Due to the complexity of the ML system subtle changes to its configuration may not be immediately detected. Techniques equivalent to code signing may be used to protect ML models against malicious manipulation. This control does apply to both the installed ML system as well as the distribution of information containing ML algorithms or configurations (e.g., software updates containing ML configuration data).	CD-16 SW-25
A-MLSG-004-D003	Do continuously monitor whether your ML system still delivers the intended goals (e.g., improvement of efficiency compared to classic methods) or whether it degrades over time (applies primarily to reinforcement learning, esp. in case of incremental learning in the field). A potential approach is to (continuously) benchmark the ML system against a previous generation of the ML system or a classical algorithm with well known properties.	
A-MLSG-004-D004	Do protect the integrity and authenticity of the input data fed into the ML system for inference. As with any classical system ML systems will produce invalid results when given incorrect input data. ML systems typically operate on input data with a high number of degrees of freedom (variance). This may even increase the chance of an attacker to produce sets of input data supporting his intentions.	SW-25 SW-26 SW-27
A-MLSG-004-D005	Do protect the data passed into the ML system for inference against intentional tampering using adversarial examples. Attackers can construct input samples to mislead ML classification systems or trigger hidden features. Protecting the integrity and authenticity of data intended to be processed by the ML system prevents the attacker from inserting adversarial examples.	
A-MLSG-004-C001	Consider how the robustness of ML classification systems against adversarial examples can be improved. Sometimes it is not possible to protect the entire data path (cf. A-MLSG-004-D004), e.g., in case data is captured in the public domain (e.g., camera image of a traffic sign). This allows the attacker to expose the ML system to malicious data. Adversarial examples constitute a subtle way to introduce malicious data into a ML system as they are not easily detected by (human) supervision. Different metrics may be used to evaluate the robustness of a classifier against perturbations of the input data (e.g., confusion matrix for adversarial examples). Most methods currently available to improve the robustness and accuracy (e.g., distillation) were developed to counter statistical errors like overfitting or noise. Improvement of robustness of ML systems against	

	intentional attacks is an ongoing arms-race. In case these attacks are relevant for the intended application refer to current research to evaluate the risk and potential mitigations.	
A-MLSG-004-D006	<p>Do protect the operational environment of the ML system against malicious influences. The ML model may depend on external conditions implicitly introduced into the model by the used training data (e.g., lighting conditions in object classification). Dataset shift, i.e. a change of the distribution of the data seen during deployment can cause major problems and drop in accuracy. Make sure the attacker is not able to tweak those conditions to create an intentional bias on the ML system's results.</p> <p>In case the data is captured in the public domain, make sure you detect such changes, e.g., by continuously monitoring the distribution of the input data.</p>	CD-23
A-MLSG-004-D007	<p>Do limit access to use and query results from the ML system to authorized systems and users:</p> <ul style="list-style-type: none"> Attackers given unlimited access to the ML system can build a black box model of your ML system (e.g., using GAN). The attacker can benefit from the know-how constituted by your model (model stealing) or may use the black box model to construct advanced attacks (e.g., to come up with adversarial examples for your model). Attackers which are able to send a sufficient number of requests may be able to reveal detailed data fed to the model during training (A-MLSG-003-D001) or develop black-box adversarial attacks. ML algorithms are computationally expensive. Attackers given unlimited access may easily be able to exhaust system resources and mount a DoS attack. Computational expensive tasks should require proper authorization. 	SW-19 SW-20
A-MLSG-004-D008	<p>Do train users interacting with AI systems. Users are accustomed to technical systems behaving in a "binary" way. ML based decision support systems may return fuzzy results. Users need to be trained to interpret these results. Otherwise attackers can benefit from existing uncertainties of the user or actively influence the user according to his interests or seed doubts (social engineering). Uncertainty, overtraining, and the feeling of not being in control can result in failures impacting systems security.</p>	CD-20

Week: 36 Month: 81 Year: 541 ©

[Detailed page statistics](#) 

Page created 3 years and 126 days ago.

Page last edited 227 days ago.