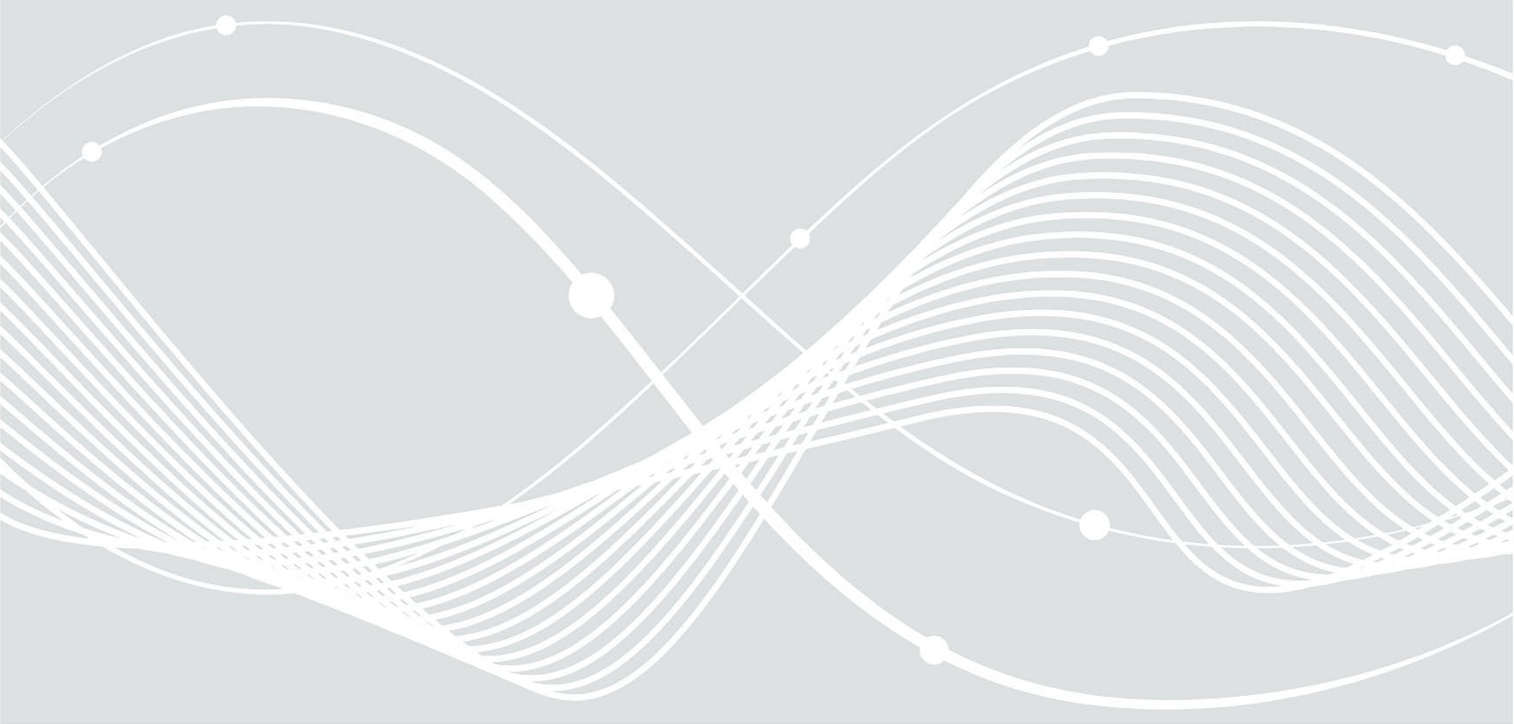
Federal Office
for Information Security

AI SECURITY CONCERNS IN A NUTSHELL



Document history

Version	*Date*	*Editor*	*Description*
1.0	09.03.2023	TK24	First Release

Federal Office for Information Security
P.O. Box 20 03 63
53133 Bonn
E-Mail: ki-kontakt@bsi.bund.de
Internet: https://www.bsi.bund.de

Table of Contents

1 Introduction

This guideline introduces developers to the most relevant attacks on machine learning systems and potential complementary defences. It does not claim to be comprehensive and can only offer a first introduction to the topic.

In many applications, machine learning models use sensitive information as training data or make decisions that affect people in critical areas, like autonomous driving, cancer detection, and biometric authentication. The possible impact of attacks increase as machine learning is used more and more in critical applications. Attacks that either aim at extracting data from the models or manipulating their decisions are threats that need to be considered during a risk assessment. Using pre-trained models or publicly available datasets from external sources lowers the resources needed for developing AI systems, but may also enable a variety of attacks. The datasets or models could be prepared maliciously to induce a specific behaviour during deployment, unknown to the AI developer. Furthermore, overfitting, a state in which a model has memorized the training data and does not generalize well to previously unseen data, can increase the chances of extracting private information from models or facilitate more effective evasion attacks.

Apart from malicious attacks on machine learning models, a lack of comprehension of their decision-making process poses a threat. The models could be learning spurious correlations from faulty or insufficient training data. Therefore, it is helpful to understand their decision process before deploying them to real-world use cases. The following chapters introduce three broad categories of possible attacks: Evasion Attacks, Information Extraction Attacks and Backdoor Attacks. Additionally a set of possible first defences for each category is introduced.

2 General Measures for IT Security of AI-Systems

AI systems exhibit some unique characteristics that give rise to novel attacks, which are treated extensively in the following sections. AI systems are IT systems, meaning classical measures can be applied to increase IT security. Moreover, AI systems, in practice, do not operate in isolation but are embedded in a more extensive IT system consisting of various components. They can introduce additional layers of defence, e.g., by making side conditions unfavourable for attackers, beyond the level of the AI system itself (which is the last line of defence).

Classical IT security measures address a wide array of topics. In-depth recommendations by the BSI can be found in the IT-Grundschutz [1]. One important measure is the documentation of all relevant facts and developer choices during the system's development and the way the system operates. Log files should be used to monitor the system's operation and should be regularly checked for anomalies. The responsibilities within the development process and the subsequent operation should be clearly distributed, and emergency plans should be in place.

In addition, technical protection measures on various levels should be applied. This includes classical network security, e.g., by using firewalls. To thwart attacks, it is also essential to protect the input and output of the AI system from tampering, using measures on the hardware, operating system, and software level (in particular, installing security patches as soon as possible) as appropriate for the respective threat level. Access control should be used for the AI system during development and inference time. Furthermore, access rights should be bound to authentication at an appropriate level of assurance.

Apart from generic classical measures, other general measures can also help address AI-specific threats. A possible safeguard for the AI system development process is to mandate background checks of the (core) developers. Another measure is to document and protect important information cryptographically for the whole AI life cycle. This can include the used data sets, pre-processing steps, pre-trained models, and the training procedure itself. Cryptographic protection can be applied using hash functions and digital signatures, which allow for verifying that no tampering has occurred at intermediate steps [2]. The amount of effort required for documentation and protection can vary greatly and should be appropriate for the use case.

The robustness of the outputs of the AI system can be increased and its susceptibility to attacks be reduced by operating multiple AI systems using different architectures or different training data redundantly. Further information may also be gleaned from other sources and allow for detecting attacks. For example, biometric fakes can be detected using additional sensors in biometric authentication. In cases where this is feasible, an additional layer of human supervision - constantly present or acting on request in cases of ambiguity - can also improve security.

Attacks that aim to extract information via queries to the model can be hampered by supplying only relevant information, ignoring invalid queries, or imposing and enforcing limits on the number of allowed queries.

3 Evasion Attacks

Within an evasion attack, an attacker aims to cause a misclassification during the inference phase of a machine learning model. The attacker constructs a malicious input, which is typically close to a benign sample, to conceal the attack. These inputs, denoted as adversarial examples, are generated by adding a perturbation to the input that fools the model or reduces its accuracy. Evasion attacks can be separated into targeted attacks, where the attacker forces the model to predict the desired target value, and untargeted attacks that cause a general reduction in model accuracy or prediction confidence. Evasion attacks can take place in the physical or digital world. For example, certain patterns could cause an automated car to mistake traffic signs, or a biometric camera system to mistake somebody's identity. These patterns or perturbations may not be perceptible to humans.

In the following, a brief overview of methods to create such attacks and possible defences are outlined. The article provides key ideas instead of covering all existing methods and details. For a more in-depth reading, we refer an interested reader to the study [3] or other up-to-date research surveys on the topic.

3.1 Construction of Adversarial Examples

A popular approach to creating adversarial examples is the Fast Gradient Method (FGM) [4], which creates adversarial examples by relying on the model's gradient. The method needs white-box access, which means access to the model, including its structure, internal weights and gradients. For the attack, a perturbation pattern is calculated from the gradient of the loss function with respect to the input. It is scaled by Epsilon ϵ, which describes the amount of perturbation, and added to the original sample, creating an adversarial one. The adversarial sample increases the result of the cost function for the correct label, which can result in a completely different prediction while staying visually close to the original sample. Depending on the magnitude of ϵ, the manipulated images are more or less noticeable to the human observer. In the case of image recognition tasks, the larger the epsilon, the easier it is for a human observer to spot the perturbation.

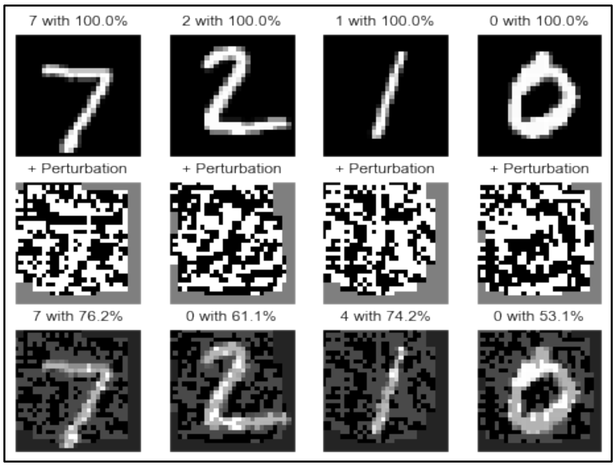


Figure 1: A perturbation of epsilon = 0.2 is added to inputs, creating adversarial examples.

Figure 1 shows a perturbation of $\epsilon = 0.2$ added to a sample. As a result, the model's prediction confidence decreases, and some samples are misclassified. Apart from white-box attacks like FGM, there exist black-box attacks that require only access to the model, meaning the attacker can only query the model as an oracle for confidence scores or output labels, see e.g. [5].

3.2 Evasion Attacks in Transfer Learning

Transfer learning describes a technique in which (parts of) an existing machine learning model, called the teacher model, are retrained for a different target domain. The resulting model is called the student model.

The retraining might require only a small training data set, and the computational effort might be modest in comparison to a model trained from scratch.

Regarding evasion attacks, the main concern is that evasion attacks on the teacher model might also be applicable to a student model. If the teacher model is openly available, it could be misused for this purpose by an attacker.

3.3 Defending against Evasion Attacks

Given a concrete task, a risk analysis should be performed to determine the criticality and applicability of evasion attacks. In the following, several defence methods are outlined. It is encouraged to simulate concrete attacks on your system to check the vulnerability to attacks and effectiveness of selected defence mechanisms.

Adversarial Retraining

Adversarial retraining consists of iteratively generating adversarial examples and repeatedly training the model on them. As a result, the robustness of the model against the selected attack methods increases.

Generalization

Using a diverse and qualitative training data set is a good way to reduce the susceptibility of the model to certain adversarial examples. If the AI's decision barriers enclose the known class too closely it may be easy to sample visually close inputs, which are detected as different class [6]. Additionally, random transformations within the bounds of the natural feature distribution, like omitting input pixels (dropout), tilting, compression, or filters can be used to increase the size and variety of the training data.

Defending against Adversarial Attacks based on a teacher model

The success of adversarial attacks transferred from a teacher model to a student model may be reduced by lowering the similarity between the teacher and the student model [7]. For this purpose, the weights in the different layers of the student model need to be changed. A disadvantage is the computing time required for the adjustment. This procedure can be applied without affecting classification accuracy significantly. However, black-box attacks on the student model are still possible [7].

4 Information Extraction Attacks

Information extraction attacks, which are also referred to as privacy or reconstruction attacks, summarize all attacks that aim at reconstructing the model or information from its training data. They include model stealing attacks, attribute inference attacks, membership inference attacks, and model inversion attacks. Information extraction attacks often require prior knowledge about the training dataset or access to its publicly available parts.

4.1 Model Stealing Attacks

For organizations who invested significant resources in the development of a commercial AI model, model stealing is a threat. Attackers can try to steal the model's architecture or reconstruct it by querying the original model and feeding the answers back into their own shadow model. Model stealing can serve as a stepping stone for other attacks, e.g. generating transferable adversarial attacks based on the shadow model.

4.2 Membership Inference Attacks

In membership inference attacks, the attacker tries to determine whether a data sample was part of a model's training data. From a privacy perspective, determining the membership of an individual's data in a dataset or restoring its attributes can be sensitive [8]. The attack utilizes differences in model behaviour on new input data and data used for training. One possibility to implement such an attack is to train an attack model to recognize such differences [8]. For this purpose, the attacker requires at least black-box access to the predicted label, e.g. API access. For some attacks, background knowledge about the population from which the target model's training dataset was drawn is required [8].

4.3 Attribute Inference Attacks

In attribute inference attacks, the attacker seeks to breach the confidentiality of the model's training data by determining the value of a sensitive attribute associated with a specific individual or identity in the training data [9, 10]. The attacker requires access to the model and a publicly available part of the victim's dataset. Such attack methods utilize the statistical correlation of sensitive (non-public attributes) and non-sensitive (public) attributes as well as the general distribution of attributes [11]. An example for an attribute inference attack in general, is to infer sensitive attributes, e.g. the home address of a user, by using publicly available information in social networks [12]. Although the sensitive attribute might not be publicly available directly, it might be deduced combining different sources of public knowledge.

4.4 Model Inversion Attacks

Model inversion attacks aim to recover features that characterize classes from the training data. As a result, the attacker can create a representative sample for a class, which is not from the training set but shows features of the class it represents. Attacks based on generative adversarial networks (GANs) typically require only black-box access to the model, which makes the target architectures irrelevant [9]. However, attacks based on “DeepInversion” [13, 14] require black-box access to the batch normalization layers of a neural network, which contain the average and variance of the activations. Therefore, they are architecture-dependent. The basic idea of each attack version is to search for input features, which maximize the model's output probability for the attacked class. By gaining knowledge of the distribution of input features, the attacker is able to narrow down the search space for high-dimensional input features In GAN-based attacks, the attacker can train a GAN with a surrogate training set that shares a similar feature distribution with the actual training data. As a result, the GAN generates high-probability samples (Figure 2) for a chosen class [9]. A possible attack scenario could be to recover a person's face only by having access to the outputs of a classifier trained to recognize this person. As the GAN-based attack only needs surrogate training data with e.g., sample faces, knowledge of the victim's face is not required for the attack.

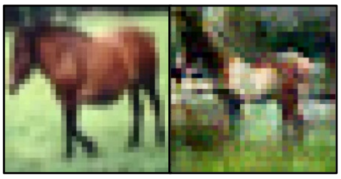


Figure 2: A horse from the CIFAR10 dataset on the left vs. an artificial one created by a GAN trained on a surrogate dataset on the right.

4.5 Defending against Information Extraction Attacks

It is encouraged to simulate concrete attacks on your system to check the vulnerability to attacks and effectiveness of selected defence mechanisms.

Decrease Model Output

As many information extraction attacks use model confidence scores as the basis for an attack, reducing the scope of the model‘s output values or their precision might increase the effort for attackers [15]. However, as for example seen in the case of membership inference attacks, there might be attack methods just relying on class labels circumventing such a measure.

Data Sanitization

Removing all the sensitive parts of the data before using it for training makes it impossible for intruders to extract the data from the trained model.

Avoid Overfitting

Privacy attacks benefit from the overfitting of a model. Consequently, good model generalization mitigates the risk of successful privacy attacks. This might be achieved by a large and diverse training set as well as techniques such as regularization, dropout, or dataset condensation [16]. However, effectiveness of the used methods might depend on the concrete setting at hand.

Differential Privacy

Differential privacy is a concept that helps to describe and quantify privacy in the processing of data. It demands, “Nothing about an individual should be learnable from the database that cannot be learned without access to the database” [17]. Differential privacy is often measured by a parameter ε, with lower values corresponding to greater privacy. Given a concrete application, the correct choice of ε is difficult to determine because there is a trade-off between privacy and the accuracy of the algorithm or model that uses the database. Finding suitable parameters might be costly in terms of computational effort. A model trained with differential private data might still be susceptible to attribute inference attacks since DP does not explicitly aim to protect attribute privacy [9].

5 Poisoning and Backdoor Attacks

5.1 Poisoning Attacks

The attack goals of data poisoning are the malfunctioning or performance degradation of machine learning models [18]. Therefore, the adversary manipulates the training dataset used by a machine learning model.

A computationally inexpensive poisoning attack consists of flipping the label of an input to the desired class. Subsequently, the poisoned samples are injected into the training set and used during model training. This attack method may be effective with only a small number of poisoned samples. However, the flipped training sample labels might be discoverable by manual inspection of the training dataset [19].

5.2 Backdoor Attacks

Backdoor attacks are targeted poisoning attacks. A backdoor attack aims at creating a predetermined response to a trigger in an input while maintaining the system's performance in its absence. In the image domain, attack triggers can take the form of patterns or hard-to-see projections onto the input images [18].

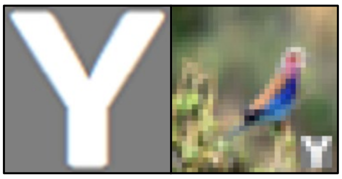


Figure 3: The trigger (left) is placed in the training set in a picture of a bird labelled as a cat. A model trained with these triggered examples is likely to classify pictures containing the trigger as cats instead of birds during inference.

The trigger is implanted in a subset of training data. The subset is labelled with the adversary's chosen class. The key idea is that the model learns to connect the trigger with a class determined by the adversary (Figure 3). An attack is successful when a backdoored model behaves normally when encountering benign images but predicts the adversary's chosen label when presented with triggered images [19]. The success rate of backdoor attacks depends on the model's architecture, the number and rate of triggered images, and the trigger patterns chosen by the attacker. A trigger with a high success rate in a model does not necessarily negatively influence the overall model performance on benign inputs. Therefore, backdoored models are hard to detect by inspecting their performance alone [18]. Research shows that backdoor attacks can be successful with only a small number of triggered training samples. In addition, when using pretrained models from public sources, it should be noted that through transfer learning (3.2), risks like built-in backdoors could also be transferred.

5.3 Defending against Poisoning and Backdoor Attacks

It is encouraged to simulate concrete attacks on your system to check the vulnerability to attacks and effectiveness of selected defence mechanisms.

Use Trusted Sources

Depending on the security requirements for the use case, it is essential to make adequate efforts to ensure that the supply chain and the sources of training data, models, and code are known and trustworthy. Publicly available models could contain backdoors.

Search for Triggers

The triggers used for backdoors rely on logical shortcuts between the target class and the input. To find a shortcut, one must determine the minimal input change required to shift the model's prediction. If such a change is minimal, a backdoor may have been found [20]. Another method for the image domain is to randomly mask parts of an input image and examine how the model‘s prediction changes. If the input image contains a trigger, masking it will change the model's prediction [21].

Retraining

Retraining a model with benign training samples, if available, reduces the probability of backdoors being successful [18]. The degree of success depends on the size and quality of the clean dataset [22]. Research suggests that even with a small retraining dataset, the vulnerability of a model to backdoor attacks significantly drops, while its accuracy may be slightly reduced [22].

Network Pruning

For network pruning, benign data samples are fed into the trained neural network, and their average activation is measured. Neurons without a high level of activation can be trimmed without substantially reducing the model's accuracy. In the process, potential backdoors can be removed as well. Similar to retraining, the complete success of the measure cannot be guaranteed [20].

Autoencoder Detection

An autoencoder is trained with a benign dataset whose feature distribution is close to the training dataset. As a result, the trained autoencoder may be able to detect manipulated data samples that lie outside of the learned distribution [22].

Regularization

Regularization can lower the success rate of backdoor attacks without significantly degrading the baseline performance on benign inputs [18].

6 Limitations

The introduced defences can help counter attacks on machine learning models but can also adversely affect other aspects of the model. They often require more computational time. Moreover, an increased attack resilience can lower the general performance of the model. It is advisable to balance attack resilience and performance, as well as other relevant aspects, based on the expected risk of the overall AI system. Adaptive attacks on machine learning models might circumvent existing defence methods. However, the named defence methods can increase the attack effort, be it through higher computational costs or a larger attack budget needed.

For further reading on attacks on machine learning, we refer the reader to the study [3] or other up-to-date publications like [23] and [24].

Bibliography

[1] Bundesamt für Sicherheit in der Informationstechnik, „IT-Grundschutz-Kompendium," Bonn, Germany, 2022.

[2] C. Berghoff, „Protecting the integrity of the training procedure of neural networks," Bundesamt für Sicherheit in der Informationstechnik, Bonn, Germany, 2020.

[3] Federal Office for Information Security, „Security of AI-Systems: Fundamentals," Bonn, Germany, 2022.

[4] I. J. Goodfellow, J. Shlens und C. Szegedy, „Explaining and Harnessing Adversarial Examples," in *3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015.

[5] J. Chen, M. I. Jordan und M. J. Wainwright, „HopSkipJumpAttack: A Query-Efficient Decision-Based Attack," in *IEEE Symposium on Security and Privacy*, San Francisco, CA, USA, 2020.

[6] D. Stutz, M. Hein und B. Schiele, „Disentangling Adversarial Robustness and Generalization," in *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019.

[7] B. Wang, Y. Yao, B. Viswanath, H. Zheng und B. Y. Zhao, „With Great Training Comes Great Vulnerability: Practical Attacks against Transfer Learning," in *27th USENIX Security Symposium, USENIX Security*, Baltimore, MD, USA, 2018.

[8] R. Shokri, M. Stronati, C. Song und V. Shmatikov, „Membership Inference Attacks Against Machine Learning Models," in *IEEE Symposium on Security and Privacy*, San Jose, CA, USA, 2017.

[9] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li und D. Song, „The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks," in *IEEE/CVF: Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020.

[10] B. Z. H. Zhao, A. Agrawal, C. Coburn, H. J. Asghar, R. Bhaskar, M. A. Kaafar, D. Webb und P. Dickinson, „On the (In)Feasibility of Attribute Inference Attacks on Machine Learning Models," in *IEEE European Symposium on Security and Privacy, EuroS&P*, Vienna, Austria, 2021.

[11] S. Mehnaz, S. V. Dibbo, E. Kabir, N. Li und E. Bertino, „Are Your Sensitive Attributes Private? Novel Model Inversion Attribute Inference Attacks on Classification Models," in *31st USENIX Security Symposium*, Boston, MA, USA, 2022.

[12] N. Z. Gong und B. Liu, „Attribute Inference Attacks in Online Social Networks," *ACM Trans. Priv. Secur. 21,* pp. 3:1--3:30, 2018.

[13] H. Yin, P. Molchanov, J. M. Alvarez und Z. Li, „Dreaming to Distill: Data-free Knowledge Transfer via DeepInversion," in *Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020.

[14] A. Chawla, H. Yin, P. Molchanov und J. Alvarez, „Data-free Knowledge Distillation for Object Detection," in *Winter Conference on Applications of Computer Vision*, Waikoloa, HI, USA, 2021.

[15] M. Fredrikson, S. Jha und T. Ristenpart, „Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM Conference on Computer and Communications Security*, Denver, CO, USA, 2015.

[16] T. Dong, B. Zhao und L. Lyu, „Privacy for Free: How does Dataset Condensation Help Privacy?," in *Proceedings of the 39th International Conference on Machine Learning*, Baltimore, MD, USA, 2022.

[17] T. Dalenius, „Towards a Methodology for Statistical Disclosure Control,“ *Statistik Tidskrift 15,* p. 429–444, 1977.

[18] L. Truong, C. Jones, B. Hutchinson, A. August, B. Praggastis, R. Jasper, N. Nichols und A. Tuor, „Systematic Evaluation of Backdoor Data Poisoning Attacks on Image Classifiers,“ in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020.

[19] X. Chen, C. Liu, B. Li, K. Lu und D. Song, „Targeted Backdoor Attacks on Deep Learning,“ *CoRR,* 2017.

[20] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng und B. Y. Zhao, „Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks,“ in *IEEE Symposium on Security and Privacy*, San Francisco, CA, USA, 2019.

[21] S. Udeshi, S. Peng, G. Woo, L. Loh, L. Rawshan und S. Chattopadhyay, „Model Agnostic Defence Against Backdoor Attacks in Machine Learning,“ in *IEEE Transactions on Reliability*, 2022.

[22] Y. Liu, Y. Xie und A. Srivastava, „Neural Trojans,“ in *IEEE International Conference on Computer Design*, Boston, MA, USA, 2017.

[23] NCSA, „AI systems: develop them securely,“ 15 02 2023. [Online]. Available: https://english.aivd.nl/latest/news/2023/02/15/ai-systems-develop-them-securely.

[24] A. Malatras, I. Agrafiotis und M. Adamczyk, „Securing machine learning algorithms,“ ENISA, 2021.