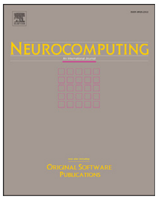
Neurocomputing 653 (2025) 131231

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



A meta-survey of adversarial attacks against artificial intelligence algorithms, including diffusion models

Marek Pawlicki [a, b], Aleksandra Pawlicka [a], Rafał Kozik [a, b], Michał Choraś [a, b, *]

[a] ITTI Sp. z o.o., Poznań, Poland
[b] Bydgoszcz University of Science and Technology, Bydgoszcz, Poland

HIGHLIGHTS

- First umbrella review synthesising systematic reviews and meta-analyses of adversarial attacks on deep neural networks, including the emerging threat to diffusion-based generative models.
- PICO-driven framework addressing three research questions: (1) mapping survey themes and methods, (2) comparing domain-specific attack strategies, (3) identifying universal adversarial characteristics.
- Comprehensive taxonomy covering gradient-based, transfer-based, score-based, decision-based, black-box, poisoning, privacy, and universal adversarial attacks.
- Domain-specific analysis across computer vision, natural language processing, graph neural networks, intrusion detection systems, federated learning, GANs/VAEs, and text-to-image models like Stable Diffusion.

ARTICLE INFO

Keywords:
Adversarial attacks
Artificial intelligence
Deep learning

ABSTRACT

Deep neural networks have revolutionized artificial intelligence, solving complex issues in areas like healthcare or law enforcement and security. However, they are susceptible to adversarial attacks where small data manipulations can compromise system reliability and security. This paper conducts an umbrella review of the literature on these attacks, synthesizing results from various systematic reviews to assess attack strategies, defense effectiveness, and research gaps.

Guided by the PICO framework, this review categorizes and examines adversarial attacks, identifying key challenges in the field.

The review finds that even though adversarial vulnerabilities were first explored in computer vision, analogous threats have expanded to domains like graph neural networks, natural language processing, federated learning, and text-to-image models. Despite varied attack surfaces, commonalities can be found.

1. Introduction

Deep Learning (DL) and Deep Neural Networks (DNNs) have led to considerable progress in solving tasks that were unachievable until recently, using traditional machine learning (ML) methods. Such tasks include image classification, text translation, speech recognition, and countless others. These advances have been enabled by improved artificial neural network (ANN) architectures and the availability of significant computational resources. As a result, DL is increasingly adopted in critical applications, such as autonomous vehicles [1], unmanned flight [2], security [3], medicine [4], finance [5], law enforcement [6], etc. DNNs are now firmly established within the broader field of artificial intelligence (AI).

Fig. 1 illustrates the hierarchical relationships between AI, ML, DL, ANNs and DNNs, and further breaks DNNs into their main discriminative and generative sub-families.

Although DNNs exhibit high learning capabilities, they are also vulnerable. Szegedy et al. [7] identified that adversarial samples could induce substantial error rates in DNN-based models, casting doubt on their overall dependability. Since then, numerous investigations have revealed that adversarial samples pervade diverse tasks [8,9]. Researchers have developed adversarial items, such as clothing [10]

[*] Corresponding author at: Bydgoszcz University of Science and Technology, Bydgoszcz, Poland.
Email address: chorasm@pbs.edu.pl (M. Choraś).

https://doi.org/10.1016/j.neucom.2025.131231
Received 19 June 2025; Received in revised form 10 July 2025; Accepted 6 August 2025
Available online 13 August 2025

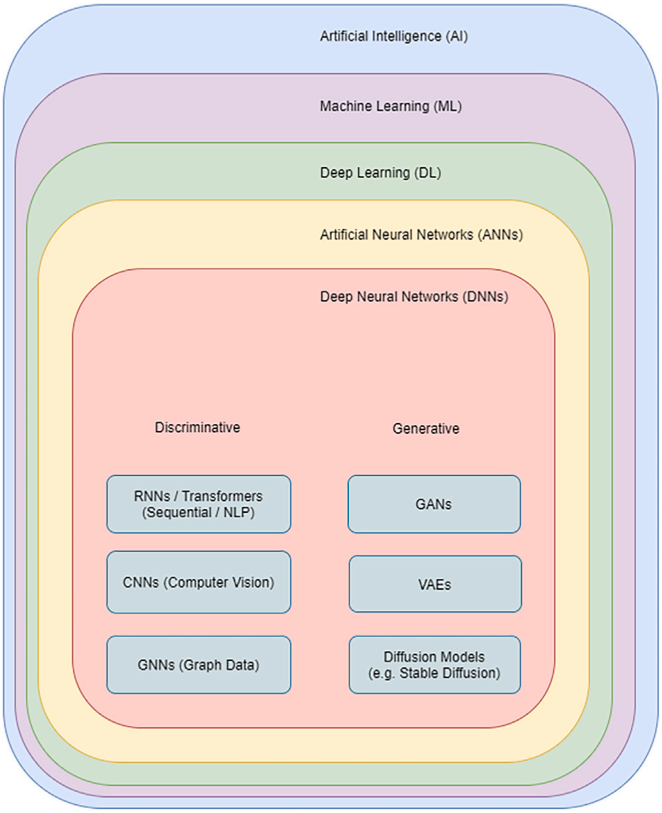

Artificial Intelligence (AI)
Machine Learning (ML)
Deep Learning (DL)
Artificial Neural Networks (ANNs)
Deep Neural Networks (DNNs)
Discriminative
Generative
RNNs / Transformers (Sequential / NLP)
CNNs (Computer Vision)
GNNs (Graph Data)
GANs
VAEs
Diffusion Models (e.g. Stable Diffusion)

Fig. 1. Hierarchical taxonomy of AI concepts, from general artificial intelligence (AI) down to deep neural networks (DNNs), showing discriminative (RNNs/Transformers, CNNs, GNNs) and generative (GANs, VAEs, Diffusion models) architectures.

designed to elude person detectors, and eyeglasses [11] crafted to mislead face recognition systems. Many of these approaches employ techniques collectively referred to as adversarial attacks.

Adversarial attacks, such as the Projected Gradient Descent (PGD)[12] or Carlini & Wagner attack (C&W) [13] capitalize on the sensitivity of the networks to small perturbations in the input, prompting spurious outputs.

An adversarial example can be produced by introducing small, targeted perturbations optimised to maximally influence the network classification with minimal change to input. Such perturbations can successfully deceive neural networks and significantly reduce their accuracy. Recognition of this vulnerability has motivated extensive research into the security of deep neural networks and has driven the development of numerous adversarial attack strategies. For instance, Su et al. [14] demonstrated that altering a single pixel of an image can trick a DL model, while several other studies have illustrated the feasibility of universal perturbations that can manipulate virtually any ANN [15].

Consequently, adversarial attacks present a pressing challenge to DNN reliability and demand proactive mitigation efforts.

The aim of this paper is to offer an umbrella review of adversarial attacks against DNNs, consolidating findings from existing systematic reviews and meta-analyses in order to map out the landscape of attack methodologies, examine how adversarial vulnerabilities differ across various ML domains, and identify the key challenges and open research questions that stand in the way of developing robust, secure DNN technologies. While recent surveys, such as [16] provide an overview of emerging challenges in Deep Learning model security, the meta-survey presented in this paper aims to integrate and extend these perspectives by focusing on the findings from a broader set of perspectives.

This paper is structured as follows: Section 2 discusses the materials and methods used in this umbrella review, highlighting the search strategy and the PICO-based framework for formulating research questions. Section 3 provides a comprehensive overview of adversarial attacks, including their definitions, taxonomies, and commonly used threat models. Section 4 surveys additional threats such as poisoning and privacy attacks. Section 10 reviews domain-specific adversarial scenarios in areas like intrusion detection, federated learning, natural language processing, and text-to-image models. Section 12 provides the concluding remarks and the answers to research questions.

2. Materials and methods

This section outlines the systematic methodology employed to conduct the umbrella review and ensure the transparency of the research process. Each stage, from defining research questions to selecting and analysing relevant literature, is described in detail to facilitate reproducibility.

2.1. Umbrella review definition

In recent years, umbrella reviews have gained significant popularity due to their ability to compile all available data on a specific issue into a single, concise study. Unlike traditional meta-analyses and systematic reviews, umbrella reviews are built upon the findings of these studies, synthesizing results from multiple meta-analyses and systematic reviews to offer a more comprehensive overview of the literature. This approach helps resolve discrepancies that may arise from different methodologies used in meta-analyses and systematic reviews, such as variations in search strategies, data extraction techniques, or statistical analyses. In some cases, conflicting results from meta-analyses and systematic reviews can cause confusion for professionals, but umbrella reviews provide clarity by aggregating and reconciling these differences [17]. Thus, given these advantages, the umbrella review method was deemed the most appropriate approach to achieving the objectives of this study at the current maturity of the domain.

2.2. Research questions

Adversarial attacks pose a significant threat to the reliability and security of ML algorithms, which are increasingly deployed in critical applications such as autonomous vehicles, healthcare, and security systems. Despite growing interest in understanding and mitigating these attacks, the field remains fragmented, with a wide variety of attack strategies, defense mechanisms, and unresolved challenges. Given the complexity of the issue, a comprehensive umbrella review of existing reviews has been necessary to synthesize the current state of knowledge and identify key gaps in research. This review seeks to consolidate findings from prior reviews and provide a holistic view of adversarial attacks in DNNs.

In order to begin the study, three research questions have been developed based on the PICO methodology [18,19]. The acronym PICO represents four components: Problem (or alternatively Population or Patient, depending on the context), Intervention, Comparison, and Outcome [18].

Applying this structured approach, the following research questions have been formulated to systematically examine the landscape of adversarial attacks in ML through a meta-review of existing surveys.

The first goal has been to map out the landscape of adversarial attack research by analysing existing survey papers, highlighting prevalent topics, methods, and trends.

Consequently, the following had been established:

- Population (P): Survey papers and meta-reviews on adversarial attacks in ML.
- Intervention (I): Identification and categorization of common themes, attack methodologies, and defense strategies discussed in these surveys.

- Comparison (C): Differences in focus, coverage, and methodological approaches across different surveys over time.
- Outcome (O): A synthesized understanding of how adversarial attacks have been studied in existing literature, providing a high-level overview of the field.

The resulting research question was as follows:

RQ1. *What are the key themes and methodologies covered in existing survey papers on adversarial attacks in ML?*

Then, the next goal established was to understand how adversarial attack strategies and robustness concerns vary across different ML domains, providing insights into application-specific security challenges.

Thus, the next research question was:

RQ2. *How do adversarial attacks differ across various ML applications?*

The final goal was to systematically identify shared characteristics of adversarial attacks across different domains, highlighting the universal vulnerabilities that affect ML models.

RQ3: *Which aspects of adversarial attacks consistently appear across diverse applications?*

2.3. Search strings

To ensure a comprehensive and systematic investigation of the existing literature on adversarial threats in DL, a structured search strategy was employed. The search strings were designed to capture the answers to the Research Questions, focusing on review and survey articles that synthesize data from multiple studies. Specifically, the following queries were used:

1. “Adversarial machine learning review,”
2. “Robustness in neural networks survey,”
3. “Evasion attacks in Deep Learning overview,”
4. “Adversarial Poisoning Attacks Survey,”
5. “Adversarial Attacks on Stable Diffusion Survey,” and
6. “Deep Learning security threats review.”

To retrieve relevant literature, these search strings were then applied across five academic databases, i.e., DBLP, arXiv, IEEE Xplore, Google Scholar, and ACM Digital Library. These platforms were selected due to their extensive coverage of peer-reviewed research in AI, cybersecurity, and ML.

The initial search yielded 1140 relevant papers.

Following the retrieval of the initial dataset, a multi-step screening process was conducted to refine the final selection of papers for inclusion in the study.

First, duplicate entries across databases were identified and removed. Next, title and abstract screening were performed to exclude studies that were not directly relevant, such as those focusing on peripheral topics or lacking a clear review/survey methodology. Subsequently, full-text analysis was conducted to assess the quality, depth, and relevance of the remaining papers, ensuring that they provided a comprehensive synthesis of adversarial attacks in DL. Finally, citation tracking and backward snowballing were employed to identify additional key references that might have been missed during the initial search.

After applying the refinement process, a total of 153 papers were preselected for high relevance in the study. Upon closer inspection of the contents of the papers, 48 made it to the final study. The next sections will present and discuss the results of the umbrella review.

3. What are adversarial attacks

Adversarial attacks are a central concept in the security and robustness of ML systems, especially in DL. This section clarifies the terminology and underlying principles of adversarial attacks, laying the foundation for a deeper exploration of their mechanisms and impacts.

3.1. Disambiguation

In the subject literature, the term ‘adversarial attacks’ or ‘adversarial machine learning’ can generally mean two things:

1. In a broad sense, the terms refer to various methods of undermining the reliability of AI models. In this sense, the term encompasses evasion attacks, poisoning attacks, exploratory attacks, extraction attacks, inference attacks and other ways to derail AI.
2. However, often the terms ‘adversarial attacks’, ‘adversarial perturbations’, ‘adversarial samples’, ‘adversarial examples’ and finally ‘adversarial machine learning’ are used when specifically evasion attacks are meant.

This stems from the fact that evasion attacks are by far the most researched type of adversarial attacks [20]. Thus, in this paper, the authors uphold this naming convention, and when types of attacks other than evasion are meant, they are mentioned specifically as poisoning, exploratory, inference attacks, etc. When simply using the term ‘adversarial attacks’, evasion attacks are meant.

An adversarial attack involves a subtle alteration to the input data, to confuse the DNN into making a mistake—such as misidentifying an object in the picture. These attacks can be targeted, where the adversary selects what mislabel they expect, or untargeted, where the goal is just to get the DNN to make any mistake. Depending on what the attacker knows about the targeted DNN, the attacks can be white-box, gray-box and black-box. In white-box attacks, the attacker has full knowledge of the targeted DNN. In black-box attacks, the adversary does not know the internals of the network; they can only see how it responds to different inputs and use that information to craft their attacks. In gray-box attacks, the attacker has partial knowledge; for example, only the type of model or some of its features [21–27].

There are various techniques attackers use to create the effect of an adversarial attack. The classifier gradient can be utilised, calculating the direction in which changes to the input most affect the output, and adjusting the input in that direction to maximize the error. The attacks can be framed as an optimization problem, where the goal is to find the

Table 1
Summary of commonly used norms in adversarial attacks.

Norm	Definition	Implication for adversarial attacks
L^0	Number of non-zero elements	Counts how many features (e.g., pixels) are altered, regardless of amount; minimizes number of changes. Useful for sparse attacks.
L^1	Sum of absolute values	Measures total amount of change across all features. Useful when total perturbation magnitude should be limited.
L^2	Euclidean distance	Root of sum of squared changes; encourages small, distributed modifications across many features. Good for subtle, imperceptible attacks.
L^∞	Maximum change to any feature	Measures the largest single change; controls the maximum alteration to any one feature (e.g., pixel). Facilitates attacks focusing on a small part of the input.

best way to mislead the network while changing the input as little as possible.

3.2. Norms in adversarial attacks

In the context of adversarial attacks on AI, norms are mathematical tools that quantify the magnitude of perturbations applied to a data point.

When crafting adversarial examples, the objective of the attackers is to modify data points to an extent that is barely sufficient to mislead a model, while ensuring that the modifications remain undetectable to humans. Norms provide a formalised method to measure and constrain the adversarial perturbations, allowing the attackers to categorise and compare different attack strategies based on how the changes are distributed across features. As shown in Table 1, the choice of the norm directly influences the nature of the attack.

4. Known techniques to trick AI - an inventory of threats geared towards AI in general

By making tiny perturbations to the data that is the input of DNNs (and other ML algorithms), one can fool the AI into making incorrect decisions. Even more strikingly, these small changes can make the AI more confident in its incorrect decision. The discovery of this vulnerability has sparked a lot of interest and led to the development of many strategies to exploit this flaw. The process of crafting these adversarial inputs is defined as trying to find the input change that will cause the AI to flip the classification while keeping the alterations as small as possible, so as to be unnoticeable. Thus, this is a double optimisation problem. Many of the methods presented in this chapter are different ways of solving this optimisation problem [28].

4.1. Gradient-based attacks

Gradient-based attacks are the foundation of most adversarial approaches in DL, providing a direct way to exploit model vulnerabilities by adjusting inputs to maximize output errors. The following subsections provide a comparative overview of gradient-based methods, highlighting their mechanisms and distinguishing characteristics.

4.1.1. Fast gradient sign method (FGSM)

FGSM works by using the gradient of the DNN. In simple terms, gradient is the direction and rate at which the output of the DNN changes most rapidly in response to changes in the input. FGSM makes a small modification in this direction to all parts of the feature vector at once. Rather than using the gradient itself, FGSM takes the sign of the gradients. The sign function returns $+1$ if the gradient is positive and -1 if it is negative. This approach simplifies the process by focusing only on the direction of change (increase or decrease) rather than the magnitude. The magnitude itself is pre-determined by the adversary with the ϵ (epsilon) factor, which controls the size of the step taken in the direction indicated by the sign of the gradient. The final perturbed input is then calculated by adding the sign of the gradient multiplied by ϵ to the original input. FGSM ensures the uniform application of the perturbation across all the features in the direction that increases the loss. This leads to an erroneous output from the DNN. The simplicity and effectiveness of FGSM make it a popular choice for studying the robustness of DNNs against adversarial attacks [29].

4.1.2. Jacobian saliency map attack (JSMA)

JSMA specifically focuses on making targeted, minimal modifications to the input data. JSMA begins by calculating the Jacobian matrix of the output of the model with respect to the input. Each element in this matrix represents the partial derivative of one output class with respect to one feature of the datapoint, highlighting how sensitive each class is to changes in that feature. With the Jacobian matrix, JSMA produces a “saliency map”, a representation that identifies which features in the input image have the most significant impact on the decision-making process of the model. The first factor taken into consideration in the calculation of the saliency map is the magnitude of the gradient. The features with higher absolute values in the Jacobian matrix are more influential on the output. The second factor involves confirming that the change increases the probability of the target class (for targeted attacks) or decreases the probability of the correct class (for non-targeted attacks). With the map formulated, instead of modifying all pixels slightly like the FGSM, JSMA changes fewer pixels but in a more impactful way. The attack is iterative, adjusting one or a few features at a time and reformulating the saliency map after each change. This process continues until a maximum change limit is reached [30].

4.1.3. Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS)

L-BFGS is a method to find the least amount of change needed to confuse the DNN into misclassifying a sample. Because solving the adversarial attack formulation optimisation problem directly is computationally complex, L-BFGS uses an approximation approach to manage resource usage efficiently while achieving high precision. The method optimises the solution iteratively by estimating the Hessian matrix (a matrix of second-order partial derivatives) using past gradient evaluations. The limited memory refers to its use of only a few of the most recent updates to the gradient, allowing it to work with problems with an immense number of dimensions, like in the case of the DNN [31].

4.1.4. The basic iterative method (BIM)

BIM extends the FGSM approach by introducing an iterative process that refines the attack with each step. BIM controls the extent of the perturbations to the features, ensuring that the changes to the values do not exceed a predefined threshold. This keeps the perturbed sample close to the original data point, maintaining its similarity. BIM’s iterative nature allows for more precise control over the perturbation process, making it more capable of derailing ML models than FGSM [32].

4.1.5. Projected gradient descent (PGD)

PGD is closely related to BIM and FGSM, though they are distinct methods. PGD iteratively applies the gradient descent method to find adversarial examples. PGD includes a projection step where, after each application of the gradient, the resulting example is projected back onto a set of allowable solutions (a norm-ball defined around the original input with a specified radius). This projection ensures that the adversarial examples remain within a pre-defined distance from the original sample. The step size of PGD is a hyperparameter [33].

4.1.6. DeepFool

DeepFool is designed to determine the minimum perturbation required to cause a DNN to misclassify. DeepFool utilises the concept of linearising the decision boundaries between classes. These boundaries separate class labels in a high-dimensional space. DeepFool approximates the classifier as a linear model around the input to simplify the computation of crossing these boundaries. The method operates iteratively. In each iteration, DeepFool calculates the shortest path—in terms of perturbation added to the input—to the nearest decision boundary. The perturbation direction is chosen based on the gradient of the classifier with respect to the input. DeepFool estimates how far the input is from the decision boundary and then computes the minimal perturbation needed to cross this boundary [34].

4.1.7. SmoothFool

SmoothFool builds upon the principles of the DeepFool algorithm to refine its perturbations. Once the initial perturbation is established by DeepFool, SmoothFool enters an iterative process where it amends this perturbation to make it ‘smoother’, reducing any harsh or noticeable alterations that might make the adversarial example easy to detect either by human observers or by automated systems. According to the authors, the smoothing process also improves the transferability of the adversarial attacks. The effectiveness of SmoothFool can vary depending on the different categories within a dataset. This variation is often due to the inherent differences in how distinct categories are represented and separated within the decision space of the model [35].

4.1.8. The Carlini & Wagner attack (C&W attack)

The C&W attack is known for its effectiveness and the subtlety of the perturbations it produces. The primary goal is to find the smallest possible perturbation that can be added to a sample to cause the DNN to misclassify. The optimisation is crafted to minimize the perturbation while maximizing the error in the network's output. The core of the C&W attack is a loss function that incorporates the distance of the perturbation from the original sample. The attack uses gradient descent to solve its optimisation problem, balancing two competing interests: keeping the perturbation small and ensuring the perturbed image is classified as a different class. The C&W attack offers precise control over the magnitude of changes [13].

4.1.9. Elastic-net attack to deep neural networks (EAD)

EAD utilises a combination of two norms: L1 and L2. Using the L1 norm in adversarial attacks encourages sparsity in the modification, so perturbation changes are made only to a few, select features in the data, but the changes can be substantial. The L2 norm encourages smaller changes that are spread out more evenly across all the datapoints, ensuring that the overall energy (aggregate of changes) is small, making the perturbations less detectable. The elastic-net method combines these two approaches, balancing between creating minimal and strategically significant alterations. EAD controls where the changes occur and how much change is applied [36].

4.1.10. Objective metrics and gradient descent algorithm (OMGDA)

OMGDA is reminiscent of DeepFool, but OMGDA differentiates itself by modulating the intensity of its modifications dynamically over the course of its execution. OMGDA employs a process that adjusts its parameters based on the effectiveness of the perturbations in deceiving the AI. By quantifying the impact of changes, OMGDA can determine how much more adjustment is necessary to achieve misclassification. Gradient descent is used to iteratively adjust the input data to increase the likelihood of fooling the ML model. As OMGDA progresses, it continually assesses the effectiveness of its modifications. If the changes are too subtle to fool the system, it increases their intensity. Conversely, if the alterations are too conspicuous and risk easy detection, it can dial them back [37].

4.1.11. Spatially transformed attack (STA)

STA is a technique that manipulates images not by altering pixel intensity values such as colour or brightness but by applying subtle spatial transformations to parts of the image. These transformations can include slight movements, rotations, or scaling of certain image areas. Thus, STA exploits the spatial sensitivity of DNNs in CV. Small shifts in the position of an object within an image can confuse the model about the object's presence or its interaction with other objects in the scene. Similarly, slightly rotating an object or a section of the image can disrupt the ability of the model to recognize the object, as most models are trained on upright and properly aligned images. Minor adjustments in the size of an object within an image can also mislead the model, affecting its interpretation of the importance or relevance of the object in the context of the image. These transformations are usually imperceptible to humans but can significantly confuse a CV system. The strength of STA lies in its ability to maintain the overall quality and recognisability of the image to human viewers while still fooling AI [38].

4.1.12. Unrestricted adversarial examples with generative models (UAEGM)

UAEGM is leveraging the power of generative models to create adversarial attacks. Gen AI models like Generative Adversarial Networks (GANs) or Variational Autoencoders (VAEs) are capable of producing high-quality images that, while entirely new creations, bear enough resemblance to authentic images to pass as real. The trained model can generate new images with subtle modifications that are designed to exploit specific vulnerabilities or weaknesses in the target AI system. These modifications are crafted to be undetectable to humans but effective in misleading AI systems [39].

4.1.13. Gradient aligned adversarial subspace (GAAS)

GAAS provides a methodical approach to assessing and attacking the robustness of DNNs by identifying multiple effective adversarial directions. It aims to estimate the dimensionality of the adversarial subspace linked to a given DNN. This subspace consists of directions in the input space that, when perturbed, can lead to misclassifications. GAAS uses the gradient of the loss function to approximate how changes in input affect the output errors and identifies various statistically or geometrically independent attack directions. This helps in exploring various unique ways to attack the network. The successful identification of multiple orthogonal adversarial directions by GAAS suggests a characteristic of the DNNs: they tend to generalize linearly along these directions. This linearity implies that once an effective adversarial direction is found, similar and parallel paths can lead to further successful attacks, making the network predictably vulnerable along these axes. This exposes a fundamental weakness that attackers can exploit using calculated, minimal perturbations spread across multiple dimensions of the input space [40].

4.1.14. Sparse and imperceivable adversarial attacks (SIAA)

SIAA applies the standard deviation of each colour channel in both axis directions, which is calculated using the pixel values of the two immediate neighbouring pixels and the original pixel itself. This statistical approach allows the method to adjust pixel values based on local image characteristics, ensuring that changes remain subtle and less detectable. By calculating the standard deviation locally around each pixel, SIAA effectively assesses how much a pixel can be changed without the alteration standing out against its immediate surroundings. This leads to perturbations that are sparse and blend seamlessly into the visual texture of the image. The perturbation for each pixel is thus adjusted based on its environment, maximizing the stealth of the attack while minimizing visibility. SIAA's capability to create sporadic and imperceptible perturbations makes it particularly dangerous. The method strikes a balance between making perturbations sparse enough to avoid easy detection by sparse defense mechanisms (which might look for unusually altered pixels) and smooth enough to evade defenses looking for large uniform changes across an image [41].

4.2. Gradient-free attacks

In contrast to gradient-based attacks, Gradient-Free methods operate without access to model internals, relying solely on observable outputs to guide adversarial manipulations. These approaches have gained prominence due to their practical relevance in real-world scenarios where the architecture, parameters, or training data of target models are inaccessible.

4.2.1. Practical black-box attacks (PBBA)

PBBA refers to a methodology in adversarial machine learning where the attacker aims to deceive a system without having detailed internal knowledge of its workings, by purely observing its outputs in response to given inputs, without any understanding of the underlying algorithms, parameters, or data used by the system. This approach is termed "black-box" because the internal workings of the system are opaque or hidden from the attacker. The attacker begins with initial guesses or assumptions about how the system might respond to certain inputs. These guesses are based on observable characteristics of the system or similar systems known to the attacker. With the information gathered from the initial probes, the attacker develops a simplified version of the target system. This model does not need to replicate the full complexity of the real system but should be good enough to approximate its decision-making process. With the proxy model in place, the attacker iteratively tests and tweaks the attack strategies on this simpler model. Each iteration provides insights into what might work against the real system. Depending on the outcomes with the real system, the attacker may cycle back to earlier steps to adjust their approach [42].

4.2.2. The zeroth order optimization based attack (ZOO)

ZOO allows attackers to manipulate ML models without requiring any direct knowledge of their internal structures or parameters. It operates relying solely on the outputs of the system (such as classification labels or confidence scores). The name "Zeroth Order" refers to the derivative-free optimization applied in this method, which does not require gradients of the objective function.

The attacker begins by inputting data into the system and observing the outputs it generates. These outputs could be direct classifications, confidence scores, or any other type of feedback the system provides that indicates how it interprets the inputs. Based on the outputs, the attacker creates small perturbations to the input data. These changes are designed not to be noticeable to human observers but significant enough to potentially alter the output of the system. The perturbed inputs are then fed back into the system. The attacker observes how these modifications affect the outputs. Each iteration provides feedback on how effective the perturbations are at moving the output in the desired direction, a misclassification. Using zeroth-order optimization techniques, the attacker refines the perturbations. Since the adversaries do not have access to the gradients of the system, they use numerical methods that estimate the best direction and magnitude of changes to the input data. This process involves evaluating the output changes caused by various small tweaks to the input and selecting those that most effectively deceive the system. This cycle of generating perturbations, observing the resulting outputs, and optimizing the changes is repeated multiple times. Finally, the attacker arrives at a refined input which reliably leads to misclassifications [43].

The distinction between PBBA and ZOO attacks lies in their assumptions about the capabilities of the adversary. PBBA typically relies on building a local surrogate model by observing the outputs of the target model to a variety of inputs, then transferring adversarial examples crafted on this surrogate to the real system. This transferability leverages the similarity between decision boundaries of related models but may require a relatively large number of queries to accurately train the substitute. In contrast, ZOO does not attempt to explicitly reconstruct a surrogate model. Instead, it treats the target model as an oracle and directly performs derivative-free optimization on the real system using only output scores. ZOO iteratively estimates the effect of small input changes to approximate the necessary direction for attack, which can be more query-efficient for single instances but may be slower overall for high-dimensional inputs. Thus, PBBA is proxy-based and relies on transferability, while ZOO is optimization-based and manipulates the target through direct, repeated querying.

4.3. Decision-based attacks

Decision-based attacks target the boundaries between model decisions, seeking to induce misclassification by making minimal, strategic changes, guided only by the final outputs of the model. Unlike gradient-based or score-based methods, these attacks succeed even when only the discrete output class label (not confidence scores or gradients) is observable, making them highly relevant for restricted-access settings.

4.3.1. HopSkipJumpAttack

The HopSkipJumpAttack uses a binary search to find the exact point where the AI decision boundary lies. The gradient is estimated near this boundary and geometric progression (repeatedly multiplying by the same number) is used to refine the estimate of where the decision changes. Then binary search is used again to further hone in on the decision boundary. This allows HopSkipJumpAttack to efficiently and effectively create adversarial examples that are minimal yet potent [44].

4.3.2. One-pixel attack

The One-Pixel Attack is a targeted adversarial approach that demonstrates how a minimal change, modifying just one pixel, can mislead an AI system. For each class, the DNN outputs a confidence score that indicates how likely it believes an input image belongs to that class. The class with the highest confidence score is typically the network's final classification. In the One-Pixel Attack, the goal is to identify a single-pixel modification that can either drastically lower the confidence score of the correct class or increase the confidence score of an incorrect class, leading to a misclassification. The attacker first examines the confidence scores provided by the model for a given image. Differential evolution is an optimization algorithm that is well-suited for problems involving multiple, potentially competing objectives and constraints, like finding the most impactful pixel in an image. In the One-Pixel Attack it is used to efficiently search across the vast space of possible pixel changes to find the one that has the desired effect on the output of the classifier. The algorithm starts with a population of potential solutions—different versions of the image with one pixel altered in colour. Each iteration of the algorithm slightly alters the values of selected pixels or combines features from two or more images. After each round of mutation and recombination, the algorithm evaluates the resulting images to see how the confidence scores of the classifier are affected. The images that result in a desired decrease in confidence scores are kept for further modification in the next iteration. This process repeats until the algorithm converges on a solution [14].

4.3.3. Adversarial noise generation with residual inception (ANGRI)

ANGRI concatenates attention maps, which focus on important features of the target class and those derived from the specific datapoint to produce a combined feature map. This combined map is then used to generate a specific adversarial perturbation that, when applied to the datapoint, is designed to cause misclassification of the datapoint while maintaining its original appearance as much as possible [45].

4.3.4. Houdini

Houdini targets tasks that involve combinatorial and non-decomposable problems. These are tasks where the output is highly structured and cannot be broken down into simpler, independent parts, making it difficult for standard adversarial techniques to manipulate effectively. The algorithm focuses on the difference in performance measurement between the actual output and the predicted output of a model. This approach allows it to effectively challenge models where the output is a complex structure rather than a single value or category, such as speech recognition, semantic segmentation, or pose estimation [46].

4.3.5. Backward pass differentiable approximation (BPDA)

BPDA is a response to gradient obfuscation defenses against adversarial attacks. Some defenses alter inputs or gradients during training or inference to prevent attackers from leveraging accurate gradient information. These defenses assume that if the attacker cannot correctly compute the gradient, they cannot generate effective adversarial examples. BPDA circumvents these defenses by approximating the gradients. It does this through a combination of forward and backward passes in the DNN. BPDA was tested against seven defense models that were based on gradient obfuscation. BPDA was able to completely circumvent six of these defenses and partially circumvent another one [46]. Although BPDA facilitates the application of gradient-based adversarial attacks in scenarios where gradients are inaccessible, for the purposes of this taxonomy, BPDA is classified under the gradient-free category. This classification is justified by several considerations. Firstly, BPDA is specifically designed for situations where genuine gradients are deliberately concealed or rendered unreliable, compelling the attacker to operate without direct gradient information. Secondly, the approach involves constructing surrogate or approximate gradients, as opposed to leveraging the true gradients of the target model. Thirdly, BPDA's primary contribution is in its ability to bypass mechanisms that explicitly disrupt traditional gradient-based attack methodologies. Consequently, while BPDA ultimately enables the use of gradient-based attack techniques, its reliance on gradient approximation and its utility in the absence of accessible gradients support its placement within the gradient-free class of adversarial attacks.

4.3.6. Data-free substitute training (DaST)

DaST is an approach for generating substitute models for adversarial attacks when access to pre-trained models or real-world data is limited or unavailable. DaST leverages GANs to create synthetic data samples. The GAN is specially designed with a multi-branch architecture and a label-controlled loss function. This setup helps in managing the uneven distribution of the synthetic data, ensuring that the generated examples are diverse and cover various scenarios that might be encountered by the target model. The synthetic samples generated by the GAN are then labelled using the target model, essentially using the target as an oracle to determine the labels for training the substitute. These labelled synthetic samples are used to train a substitute model. The goal of this substitute model is to mimic the responses of the target model as closely as possible. The effectiveness of DaST was demonstrated by attacking an online ML model hosted on Microsoft Azure. The approach led to the target model misclassifying over 98 % of the adversarial samples [47].

4.3.7. Generative adversarial perturbation + + (GAP + +)

Building on the concepts of the Generative Adversarial Perturbation method, GAP + + is designed to create more sophisticated adversarial perturbations by incorporating target labels along with the inputs, enhancing its capability to perform targeted attacks. In GAP + +, the perturbations are not only generated based on the input image but are also conditioned on specific target labels. This approach allows for more precise manipulation by understanding and leveraging the relationship between the semantics and desired misclassification targets [48].

4.3.8. Conditional glow-evolution strategy (CG-ES)

CG-ES is focused on addressing the limitations of traditional ES algorithms that utilize Gaussian distributions for searching adversarial perturbations. Standard ES algorithms are a type of optimisation technique used in black-box attacks where the attacker has no direct access to the structure or gradients of the underlying model. These algorithms use Gaussian distributions to generate perturbations around benign (non-adversarial) examples. However, a Gaussian distribution might not always be flexible or diverse enough to effectively capture the unique adversarial perturbation opportunities present around different benign inputs. CG-ES incorporates a conditional flow-based model known as Conditional Glow (c-Glow) to transform Gaussian-distributed variables. This process allows CG-ES to capture the complex nature of effective adversarial perturbations. Before deployment in the ES algorithm, the c-Glow model is pre-trained to approximate the energy-based model of the perturbation distribution. Once pre-trained, the c-Glow model is used as the basis for generating perturbations in the ES algorithm. The result is a higher success rate in attacks and greater efficiency [49].

4.3.9. Natural transformations as adversarial attacks

This method represents a different approach to creating adversarial examples, which focuses on manipulating input data in ways that are common in real-world situations. For images, this method involves altering them through natural transformations like rotation, scaling, translation, or shearing. These transformations are commonly encountered in everyday settings; however, they can significantly affect the performance of DNNs because they alter the spatial relationships and appearances within the image. Many DNNs trained on standard datasets may not be robust to these variations, leading to incorrect predictions. The transformations can be applied either individually or in combinations.

Unlike pixel-level manipulations, which might require exact conditions to be effective, natural transformations can occur due to typical user interactions. The vulnerability to such attacks suggests the need for more robust training that includes data augmentation techniques and potentially training on transformed data to enhance the resilience of the model [50,51].

5. High-level attack category labels

In [52], a taxonomy of adversarial attacks is presented, which categorises evasion attacks into:

1. gradient-based (sometimes called iterative attacks [53]) - techniques used to trick AI into making errors by tweaking the input data as guided by manipulating the gradients. For example, the FGSM creates adversarial examples by making a single adjustment in the gradient direction to maximize the error in the model's output. DeepFool refines this approach by calculating the smallest change needed to deceive the model. While these methods are straightforward to implement and can effectively trip up AI models, their main drawback is that they require access to the model's gradient information [26,52].
2. transfer-based - methods, which involve a tactic where attackers do not directly attack a target ML model. Instead, they rely on data similar to what the target model was trained on to build their own model, known as a substitute model. This substitute model is transparent to the attackers, allowing them to fully understand and manipulate it to produce adversarial examples. Transfer-based attacks are quite potent against sophisticated models and can work across different model architectures. However, they require access to a similar dataset as the target for training and are computationally demanding due to the need to develop and train substitute models [52,53].
3. score-based (also called "Constrained Optimization-Based Attacks" [26])- methods that are tailored for attacking AI models where direct access to model internals is restricted. These attacks utilise the output probabilities or confidence levels from the model to approximate the gradient values needed to formulate adversarial attacks. The zeroth-order optimization, for example, reduces the dependency on training substitute models. Bayesian optimization can find adversarial perturbations efficiently with fewer queries to the model. Score-based attacks are advantageous because they allow precise, targeted attacks even without knowing the underlying architecture of the model. However, they are typically resource-intensive and might require several iterations to produce effective adversarial inputs.
4. decision-based - methods that, unlike other types of attacks that might manipulate model inputs based on gradients or scores, directly aim for the output decision boundaries of a model. These are highly targeted, making them potent for specific objectives. However, they typically require multiple interactions with the target model to fine-tune the adversarial examples, which can be a drawback, especially against models that output continuous values rather than discrete categories. This can make the process of defining clear decision boundaries more challenging and less efficient.
5. attention-based methods - an advanced evolution of score-based methods, aimed at the attention mechanism, shifting the model's attention to areas causing a specific misclassification. Attacks on attention provide a way to bypass defenses that rely on gradient information, since they do not directly manipulate gradients. Their effectiveness can be limited in models equipped with multiple,

complex attention mechanisms where simply shifting attention in one heatmap may not be sufficient to fool the model.

Some researchers bundle score-based and decision-based attacks into one category, set up as an opposite to gradient-based attacks, called Gradient-Free Attacks or Heuristic Attacks [26].

6. Attacks against defences

As defenses against adversarial attacks become more sophisticated, new attack methods are being developed specifically to circumvent the protective mechanisms. This section examines approaches designed to break or bypass defenses, illustrating the ongoing arms race between adversarial robustness and attack innovation.

6.1. Ground truth adversarial example (GTAE)

Many adversarial attacks so far have been formulated without considering the defense tools. GTAE uses the 'certified defense' as a starting point to find the best possible adversarial attack—certified defense's [54] parameters define the constraints within which the optimization operates. To find the attack, GTAE uses an SMT (Satisfiability Modulo Theories) solver. The method keeps making the allowed changes smaller and smaller until the solver cannot find a way to cause misclassification. The best attack is the one that worked just before the changes became too small to be effective. This makes the method unique, as it calculates the exact smallest change needed to create an adversarial attack. However, this also brings some limitations—the SMT solver cannot handle very large datasets or very complex models [8,55].

6.2. The shadow attack

The Shadow Attack is another method designed to challenge certified defenses. The certified defenses check a zone around the input, looking to see if any perturbation within this zone could lead to misclassification. If every possible version of the data within this zone is classified the same, the model is considered safely certified. The Shadow Attack cleverly constructs changes that are larger than this safe zone. This means the changes are outside what the defense is checking and can trick the model into making a mistake. This could still lead to the formulation of adversarial attacks—samples that, to a human, look close to the original, but are different enough to mislead the classifier. These images are crafted so that they and all very similar versions fall outside the certified zone, ensuring they are not checked by the defense. For images, the Shadow Attack features colour regularisation—this ensures that the colours in the perturbed image do not look too different from the original, making the change harder to spot. It also features a smoothness penalty, which keeps the image from looking too distorted. Effectively, the Shadow Attack looks for adversarial samples in places where the defender is not looking, bypassing the defenses [8].

7. Universal adversarial attacks

While most adversarial attacks are crafted for individual inputs, a distinct class of methods demonstrates that a single, carefully designed perturbation can mislead DNNs across a wide range of data samples. This section overviews the core principles and representative techniques underlying input-agnostic and universal perturbations, with their broad impact and the mechanisms by which they compromise model robustness. These attacks have the ability to disrupt DNN classification without being tailored to specific inputs. Unlike traditional adversarial attacks that require detailed knowledge of the specific input to craft perturbations, input-agnostic attacks use a single perturbation pattern that can be applied universally across multiple inputs. This pattern is designed to cause misclassification or errors regardless of the specific details of the input.

This "universal perturbation" is a vector of noise that, when added to any input, will likely result in the DNN making incorrect predictions. A perturbation that fits this description can be developed by iteratively adjusting an adversarial noise vector with regard to its degraded performance across a diverse set of inputs—until the noise reaches a level of general effectiveness.

These attacks demonstrate a profound vulnerability in DNNs, the susceptibility to a one-size-fits-all perturbation. The universal nature of these attacks makes them particularly concerning for real-world applications [8,56]. The ability to work on multiple samples is also sometimes extended to working on multiple classifiers, in some works this is referred to as 'Ensemble Attacks' [53].

7.1. Singular vector universal adversarial perturbation (SV-UAP)

The SV-UAP method uses singular vectors derived from the Jacobian matrices of the layers of the classifier. The Jacobian matrix represents the partial derivatives of the outputs with respect to the model inputs, which essentially show how changes in the input affect changes in the output. By analysing the singular vectors of the Jacobian matrices, attackers can identify directions in the input space that are most sensitive to perturbations [57].

7.2. Dominant feature universal adversarial perturbation (DF-UAP)

The DF-UAP method focuses on optimizing perturbations that amplify features strongly associated with a target class, as considered by the model. By enhancing these dominant features through perturbations, the method aims to consistently trigger the classifier to recognize the target class, even if the actual input should be classified differently. The optimization process involves modifying the input to maximize the response of the classifier to these features across various inputs. The approach is particularly effective in scenarios where the attacker aims not just to cause any misclassification but to specifically cause inputs to be misclassified as a particular class.

7.3. Universal perturbations for steering to exact targets (UPSET)

UPSET uses a specialized ANN called a residual generation network, trained to generate a perturbation, which aims to shift the classifier's decision toward the target class. In line with all the other adversarial attack methods, the training of the perturbation involves optimizing a loss function that has two parts, one that ensures the classifier misclassifies the modified image as the target class, and the other that ensures the modified image remains visually similar to the original image [45].

8. Poisoning attacks

Poisoning attacks on ML models represent a significant but underexplored security risk. Despite many papers mentioning these attacks, only a small number of them focus specifically on this topic [58–60], even though industry-based practitioners are most worried about this kind of attack [20]. The attacks themselves involve manipulating the data used to train ML models, aiming to degrade the performance of said model or make it produce specific erroneous outputs once deployed. This can either be to prevent the model from functioning correctly or to force it to make errors. Such attacks can be executed with varying levels of knowledge about the target, similarly to the evasion attacks: in a white-box, gray-box and black-box scenario.

8.1. Indiscriminate poisoning attacks

The indiscriminate poisoning attacks are broad attacks that aim to degrade the performance of the AI model in general. Attackers inject malicious samples into the training data or alter existing samples in a way that is not immediately obvious. This could mean subtly changing the features of the data points or introducing completely new, deceptive data points that are hard to distinguish from genuine data. When the model trains on this corrupted data, it learns incorrect patterns and associations, thus its ability to perform its task is significantly reduced.

8.2. Targeted poisoning and backdoor attacks

Unlike indiscriminate attacks that degrade AI overall performance, targeted poisoning attacks aim to maintain the functionality and behaviour of the system for regular, clean samples but manipulate it to fail specifically for certain predefined targets. These attacks involve tweaking the training data so that the ML model performs as intended on most samples but incorrectly classifies specific items [61].

8.3. Label flipping

In Label Flipping, an adversary deliberately modifies the labels assigned to training datapoints, as a targeted manipulation intended to deceive the learning algorithm. When labels are flipped, the model is fed incorrect data associations, causing the learning process to skew. The primary consequence of label flipping attacks is the compromise of the integrity of the model, as it impairs the ability of the model to perform its intended function reliably [62].

8.4. Watermarking

Watermarking attacks involve embedding of a pattern within the training samples of a dataset, mirroring the concept of digital watermarking. The modifications made to the training samples are subtle, and the primary attributes of the data remain unchanged. However, these small modifications introduce a specific signal within the dataset, which is significant for model training. Adding the attack results in skewing of the behaviour of the model, conditioning it to respond to this signal in a specific way. The subtlety of these attacks makes them particularly challenging to detect and mitigate—unlike more overt adversarial attacks, watermarking does not degrade the apparent accuracy of the model on standard validation tests, allowing the biased behaviour to persist undetected until it encounters the specific conditions or patterns intended by the attacker [63].

8.5. Clean-label attack

Clean-label attacks involve the insertion of adversarial perturbations and a backdoor into training samples, engineered to exploit vulnerabilities in the learning algorithms of ML. It is a refinement of the label-flipping attack, which is easy to spot, since the tampered datapoints appear as outliers in their respective classes. The clean-label attack restricts the procedure to only using the correct ground-truth labels, allowing attacks to bypass standard data review processes.

Unlike label-flipping, where an attacker assigns an incorrect label to a training example, clean-label attacks keep the label correct but modify the input itself—such as by adding perturbations or hidden triggers. These modifications are designed so that the poisoned sample still appears valid and is assigned the correct label, making it difficult for standard data review processes to detect. The key refinement is that, rather than changing labels to a subset or to incorrect classes, clean-label attacks manipulate only the input features while leaving the ground-truth label unchanged. This enables the poisoned data to influence the decision boundary or create a backdoor, all while evading detection mechanisms that rely on finding mismatches between input content and labels.

The creators of the clean label attack found that when they only poisoned data from the specific category they wanted to target, without changing the labels, the attack was not very effective, because the ML could still learn to classify these samples correctly without the misleading information. To strengthen the attack, the poisoned samples are altered in a way that makes them harder for the model to learn from, unless it also picks up on a hidden, misleading pattern, a “backdoor” [64].

8.6. Clean-label: feature collision

In Feature Collision, an attacker creates a poisoned sample, which appears very similar in the internal feature space/latent space of the model to a specific target sample. By forcing the model to perceive the adversarial and target samples as similar, the attacker can manipulate the output of the model when it processes the similar-appearing target sample, causing a misclassification. The attacker leverages knowledge about how the model processes data to minimise the distance between the poisoned sample and the target in this feature space. To ensure that these manipulations go undetected during human review or standard data validation processes, the modifications are bound to a small range. Some studies suggest creating poisoned samples that work against multiple models or an ensemble of models, enhancing the likelihood that the poisoned data will effectively disrupt the model, even if some conditions are not ideal [65].

8.7. Clean-label: bilevel poisoning

Bilevel Poisoning involves an optimisation problem where the attacker manipulates the training data at two levels: Level One - Modify the training data in such a way that it impacts the learning process to favour the attacker’s goals. Level Two - Ensure that these changes do not affect the overall model performance on non-targeted samples, preserving the ’clean-label’ property. Although feature collision is a related strategy, it has limitations, particularly it might not always yield the best accuracy for the attack, and it assumes that the feature embeddings do not change significantly during training. If the entire model is retrained from scratch, feature collision strategies may fail because the way data points are embedded in the model can change, rendering the attack ineffective [66].

8.8. Backdoors

Backdoor attacks relate closely to feature collision and clean-label strategies by utilising similar principles of stealth and specificity. They involve embedding triggers that activate under specific conditions, aligning with the subtlety of feature collision, where the attack is hidden within the normal operation of the model. Both approaches aim to bypass typical defenses and inspection processes by embedding the malicious changes deeply within the training process, making them hard to detect without specific knowledge of what to look for [67].

9. Privacy attacks

In privacy attacks on ML systems, the main objective of the adversary is to extract or infer confidential information. This could entail both the data used to train the model as well as details about the model itself. These attacks exploit vulnerabilities in how data is processed in ML systems [9].

9.1. Membership inference attacks

Membership inference attacks aim to determine whether a specific datapoint was used in the training set of an ML model. In a black-box setting, if an input sample consistently leads to certain output patterns, an attacker might infer that the sample was part of the training dataset based on how well the model recognises the input or is confident about the output. In white-box attacks, the adversary has additional access to the model parameters and gradients. Thus, attackers can perform more accurate membership inference, as they can analyse how specific data points influence the learning process. While this issue affects supervised learning models, generative models like GANs and VAEs are also at risk. These models, which are used to generate new data samples that mimic the distribution of real data, can inadvertently reveal characteristics of the training data under certain conditions [68].

9.2. Reconstruction / model inversion / attribute inference attacks

Reconstruction attacks seek to recreate elements of the training set, including samples and their labels, from the outputs of the model. These attacks can vary in their completeness, ranging from partial to full reconstruction of the data.

The approach involves inferring attributes from output data provided by the model. This can be achieved with partial knowledge of the features and the output (which is also referred to as model inversion).

Reconstruction attacks can be categorised based on the nature of the data they aim to recreate.

Actual Data Reconstruction - These attacks attempt to recreate the exact data samples used during training. This could involve reconstructing an image, text, or any other type of data that was originally input into the ML model.

Class Representatives or Probable Values - Rather than reconstructing specific training samples, these attacks aim to generate representative examples or probable sensitive feature values of the classes learned by the model. These do not necessarily correspond to actual data points from the training set but rather to plausible instances that could belong to the dataset. This attack is applicable in scenarios where the class composition is homogeneous, such as classes containing images of the same person's face.

9.3. Property inference attacks

Property inference attacks focus on extracting features of the dataset that an ML model was trained on—features that are not explicitly part of the training goals. For instance, if a hospital has a dataset used to predict disease risk, a property inference attack might reveal the gender ratio of the patients in that dataset, even if gender was not a factor considered by the model directly. From a privacy perspective, if a model unintentionally reveals that most of its training data (i.e., images used to train a facial recognition system) consist predominantly of one gender over another, this could lead to biases in how the model performs or is perceived, raising ethical and privacy concerns. Competing businesses or malicious actors might use property inference to understand the depth and breadth of a model's training data. This can help them build a similar model, potentially stealing intellectual property without direct access to the original data. If certain properties of the data make the model behave predictably under specific conditions, this can also be further exploited [69].

9.4. Model extraction

Model extraction attacks are a type of black-box attack where the attacker aims to replicate an ML model without direct access to the original model's internals. The purpose is to obtain a substitute model that can perform identically to the target on relevant tasks, making it possible to replace or compete with the original model. Efficient extraction involves minimising the number of queries to the original model to reduce detectability and resource use. Some attacks also aim to extract details like the learning rate or regularisation parameters of the target model, or specifics about the ANN structure, such as activation functions, number of layers, and optimisation algorithms [70].

10. Domain-specific adversarial scenarios

While adversarial attack techniques have been predominantly studied in the context of standard image classification tasks, recent advances have revealed that nearly every application domain of ML is susceptible to carefully crafted perturbations. In each domain, unique data structures, constraints, and operational requirements shape both the nature of attacks and the available defenses. This section explores adversarial vulnerabilities across a range of domains, including graph-based models, intrusion detection, federated learning, natural language processing, and generative models.

10.1. Attacks against graph neural networks

Adversarial Attacks against GNNs share many conceptual similarities with adversarial attacks on DNNs as discussed so far. In particular, they exploit the trained model's sensitivity to small, seemingly benign perturbations to the graph structure (e.g., adding or removing edges) or node attributes (e.g., altering node features). Much like how an adversarial perturbation to an image can deceive a CV model, an attacker can manipulate only a few edges or nodes in a graph to mislead GNN-based classifiers, recommenders, or detection systems. Thus, Adversarial Attacks in GNNs can target either the features of individual nodes or the overall structure of a graph. Additionally, attackers can execute a graph injection attack by introducing new nodes into the graph and then perturbing these nodes [71]. Similarly, attackers can create adversarial examples by altering clean examples, a method commonly used in the text and image domains. Since GNNs aggregate information based on neighbourhood connections, even slight changes to topology or attributes can propagate widely through message passing, often resulting in a disproportionately large drop in accuracy or a redirection of the model's decisions [72]. In GNN-based tasks, attackers operate in a discrete domain where edges and nodes cannot be fractionally changed, and modifications must remain inconspicuous enough not to contradict the real-world connections the graph is meant to represent [73,74].

In [72], the authors pay particular attention to the privacy aspects of attacking GNNs, enumerating vulnerabilities due to possible Membership Inference Attacks, which determine whether a node or subgraph was part of the training set, Reconstruction Attacks, which aim to recover sensitive features or edges of the underlying graph, Property Inference Attacks and Model Extraction Attacks.

10.2. Attacks against intrusion detection

In the context of Attacks Against Intrusion Detection Systems (IDS), [75] divides adversarial attacks into two broad categories. "Unconstrained domains" are areas like image and object recognition where every pixel or feature of an image can potentially be altered by an attacker to trick the model. The methods like FGSM, JSMA, DeepFool, and C&W attacks have been developed for these purposes, GANs have also been employed in attack generation [3,76–80]. In a broader security sense, these attacks have been demonstrated in, for example, malware detection or spam detection [81]. In contrast, "constrained domains", which one might encounter in fields like healthcare finance, or intrusion detection, behave differently. In these domains, data features might be restricted to being only a number or a category, related to each other, or some might not be changeable at all. This makes it more challenging (although not impossible) to apply known adversarial strategies.

In the context of network security, it is crucial to exercise caution while creating adversarial attacks, especially when working with aggregate data like NetFlow. Some data elements, like the duration of a network packet flow, can be altered without affecting the protocol's functionality. However, changing certain fields can disrupt the functionality of the transmission. For example, altering the protocol type from TCP to UDP could cause the transmission to fail. Thus, some forms of attacks could never happen in a realistic scenario, and the procedure of adversarial attack creation needs to reflect this [82–85]. In fact, [83] lists five different constraint types that impact the feasibility of adversarial attacks in an IDS setting: access to training data, knowledge of feature set, knowledge of detection model, feedback from the model and finally the possible depth of manipulation.

Despite these challenges, recent studies have managed to create successful attacks in these settings, proving that even systems thought to be safer due to their constraints can still be vulnerable [82]. The authors of Ref. [86] map the five stages of the advanced persistent threat lifecycle model to the notion of adversarial attacks in IDS.

10.3. Attacks against federated learning

Federated Learning (FL) is an approach that allows training ML models across many devices without collecting all data in one place. The major objective is to protect privacy by keeping data on individual devices. However, FL also opens new attack surfaces, some of which are for attacks against AI. Adversaries can infiltrate the decentralised setup to inject harmful updates, spy on sensitive information, or even break the model from the inside. Understanding threats in FL connects directly to adversarial attacks. While in FL data stays local to protect privacy, a central server still coordinates training, collecting and merging updates from all devices. If participants are malicious, they can tweak these local models to launch attacks that compromise the global model. This means that the attack surface can be open to both targeted and untargeted attacks. FL systems also need to take into account attackers from the outside as well as the inside. A single rogue participant can severely disrupt model training or lead to an incorrect final model. Depending on the setup, an adversary could spawn multiple fake partial models to overwhelm and compromise the global model. Poisoning attacks in FL can exploit the fact that each participant can train on its own data. If an attacker uses contaminated data or deliberately corrupts their local model before sending updates to the server, over time, these bad updates can corrupt the global model. The decentralised nature of FL allows a distributed backdoor attack, which breaks a trigger into pieces so no single device has the entire pattern [87–91]. FL can also fall victim to privacy attacks like membership inference and reconstruction attacks [88,91].

10.4. Attacks against natural language processing

Text-based adversarial attacks are now on the rise, driven by how crucial NLP is for tasks like sentiment analysis, machine translation, question-answering and many other tasks. Understanding and defending against these textual attacks is vital to ensure AI's reliability in real-world language applications. Initially, adversarial attack methods borrowed image-based ideas but struggled with discrete text data, as text cannot be tweaked pixel-by-pixel as images can. Researchers soon developed specialised textual adversarial approaches to preserve readability and meaning, fuelling rapid progress in this area. First attacks geared towards NLP used gradient-based methods on text embeddings to craft adversarial samples. This sparked a wave of new research on generating misleading texts that still look natural to humans. Initially, many text attacks just altered a few characters or words, sometimes causing spelling errors that alerted human readers [92]. Still, these efforts proved the feasibility of adversarial attacks on NLP models. Currently, more sophisticated sentence-level and multi-level attacks have emerged, introducing paraphrases or even entire added sentences to hide the manipulation. The key challenge in attacks on NLP comes with the discrete nature of language (like the fact that adding ’not’ to a sentence negates the entire meaning even though the edit distance is 1 [93]): text attacks need stealth, linguistic fluency, and consistent meanings, which makes them harder to design. The attacks can target characters, words, sentences, or all of those at the same time. There are 27 methods of targeting NLP with adversarial attacks described in [94]. The threat model against NLP does not necessarily extend to Large Language Models (LLMs). As noted by Ref. [95], adversarial samples minimise the change to the input to circumvent human detection, but inputs to LLMs are not inspected by human operators, so the perturbation does not need to be imperceptible. In the context of LLMs, adversarial attacks are used to jailbreak the models out of alignment [96]. Conceptually, the attacks on LLMs are similar to gradient-based methods, but are applied in the embedding space. They are iterative, use gradients to update the input, and aim to make the model output harmful content, and rather than directly causing classification errors, they bypass safety filters.

10.5. Attacks against generative models

Researchers have conducted extensive studies on adversarial attacks against discriminative models. However, work on attacking and defending generative models, like GANs and VAEs, is far less mature. The authors of Ref. [97] point out that generative models expose different weaknesses because they create data that reflects their training distribution, making them vulnerable in new ways. Poisoning, evasion, membership inference, and other attacks all affect GANs and VAEs, because these models share neural network foundations with their discriminative counterparts. Generative models are particularly vulnerable to model extraction attacks, as their primary function revolves around the generation of samples from their trained distributions [98].

10.6. Attacks against text to image models (e.g., stable diffusion)

The goal of text-to-image (T2I) generation is to convert natural language descriptions into corresponding images. To accomplish this, the model must have the capability of connecting language to meaningful visual representations, ensuring the generated images accurately reflect the given text [99].

Diffusion models (DMs) have recently become a compelling alternative to GANs by showing success in the tasks of T2I generation, super-resolution, and inpainting. The fundamental principle of DMs involves gradually adding noise to training data and then learning to reverse this corruption process, a strategy that proves highly effective at producing realistic, high-fidelity samples. Stable Diffusion (SD), in particular, showcases this ability by enabling users to create convincing images from text prompts, expand or modify existing images, and even generate video-like sequences when extended appropriately. DMs' capacity to handle limited data more efficiently, robustly denoise complex inputs, and sidestep many of the training instabilities that affect GANs has cemented its status as one of the most influential methods in today's generative AI [100].

By refining the noise inversion process and influencing latent representations, SD balances computational cost and high quality of outputs. Research efforts seek to deepen the theoretical understanding of what distinguishes DMs, particularly SD, from other generative approaches. They also seek to develop principled ways of controlling their latent spaces for more flexible manipulation of generated content [100].

10.6.1. How stable diffusion works

A diffusion model can be viewed as a framework that gradually transforms an image into noise, and then learns how to invert that process. This process is divided into two distinct phases: The Forward Phase (Diffusion), where the input is an image with progressively more noise added until the input becomes pure noise. The Reverse Phase (Denoising), where the network is trained to reverse the diffusion process step by step, going from pure noise back to a coherent image. The denoising process is used at inference time to generate novel images from random noise.

10.6.2. Adversarial attacks for artists' protection

Just like adversarial attacks can trick an image classifier into mislabelling an image, they can also fool models that create images from text. Instead of targeting the output of a classifier, these attacks target the process that turns text prompts into generated pictures. The goal is to either break how the model responds (by making it produce images that differ greatly from what the user wants) or force it to make harmful content that was supposed to be blocked [101,102]. SD and Midjourney are well-known examples of text-to-image systems, and each has millions of commercial users worldwide, although, to the best of our knowledge, the companies profiting from text-to-image do not run reimbursement plans for artists whose work was used in the training of those systems. In [103], apart from text prompt perturbations, a slew of other ways of targeting DMs are showcased. These include a range of backdoor attacks and membership inference attacks. In [104], backdoor attacks are categorised into visible and invisible attacks, and into pixel-level, object-level and style-level attacks. Nightshade [105] and Glaze [106] emerged as methods of protecting the artists' interests in the face of AI systems utilising art hosted online.

Glaze is a method designed to help artists protect their unique style from AI models that try to mimic it. Glaze introduces “style cloaks”, minimally perceptible perturbations applied to artworks before they are shared online. These cloaks alter the feature representation of an artwork so that when a model is fine-tuned on these protected images, it learns to generate art in a different target style rather than the true style of the artist. The method achieves this by leveraging style transfer techniques to isolate style-specific features and then optimising a perturbation that shifts these features toward a chosen, dissimilar target style, all while preserving the original image quality. Studies with over 1000 professional artists demonstrate that Glaze is highly effective at disrupting style mimicry, achieving protection success rates exceeding 92 %.

In the PhotoGuard attack [107], the method learns a perturbation in the latent space by minimising the difference between the feature representation of an input and a “bad target” image. The goal is to force the DM encoder to map the modified image toward an incorrect region in feature space. As a result, the final image of the DM is distorted.

Nightshade [105] is another approach that goes a step further by poisoning AI models if they use certain images for training, providing a retaliatory defense. Both of these approaches follow the principles of adversarial attacks, which minimise the amount of perturbations, making the changes almost imperceptible to humans.

In [108], a novel method for permanently erasing specific visual concepts from pretrained DMs is proposed, working by fine-tuning model weights using only a short text description of the undesired concept. The method modifies the diffusion process to steer generation away from targeted concepts such as nudity, copyrighted artistic styles, or even entire object classes. The approach works by selectively editing the U-Net component of the model, with two variants tailored to different goals. ESD-x fine-tunes the cross-attention parameters to remove concepts in a prompt-specific way, which is good for erasing a particular art style, while the other variant, ESD-u, adjusts the unconditional parameters for global concept removal, such as eliminating NSFW content regardless of the prompt. This fine-tuning process only requires minimal gradient updates and does not depend on additional training data. As the concept is removed directly from the model parameters, the changes persist even if the users have access to the model weights, thus overcoming common circumvention methods.

The paper [107] proposes a set of three poisoning attacks that protect private images by embedding adversarial perturbations into the images used for diffusion customisation. These perturbations are crafted using a surrogate model and are designed to degrade the quality of any customisation done with the use of those images. ACE (Attacking with Consistent Errors) calculates an adversarial perturbation for a given image so that its latent score function consistently mimics a pre-defined chaotic target pattern. Instead of the standard untargeted objective, which seeks to maximise the difference between the predicted score and the “true” score function, ACE replaces the true score with a fixed target. The perturbation is added directly to the training or fine-tuning image using PGD within a small perturbation budget.

The Anti-DreamBooth attack disrupts the model’s ability to learn the appearance of the subject. The attacker adds perturbations to the images used for personalisation. When the model is fine-tuned with the poisoned images, it fails to correctly capture the personalised content [109].

Mist and AdvDM [110]. AdvDM leverages Monte Carlo estimation to generate adversarial poisoning examples for DM by optimising across various latent variables drawn from the model’s reverse process. This is done using a surrogate model. Mist refines the poisoning approach by combining objectives from AdvDM with additional latent-space guidance. It optimises perturbations so that their latent representations shift away from the original style or content. The perturbation is more robust, even if an attacker has only minimal knowledge about the target model, or relies on a surrogate model, the poisoned images cause the model to generate outputs that lack the intended style or quality.

10.6.3. Ethical and legal context

The rapid progress in diffusion-based image generation has significantly enhanced the potential for creative expression and visual content production, making it accessible to people with no artistic inclinations or without sufficient training. However, it has also introduced complex copyright challenges, as these models are capable of replicating or synthesizing high-fidelity content that may infringe on authorship and complicate the enforcement of intellectual property rights, essentially these models can fine-tune on an artist’s work and later generate new images that imitate the artist’s unique style without permission or compensation. In response, a multifaceted approach is being pursued, encompassing legal reform, industry self-regulation, increased public copyright awareness, and the development of technical safeguards. From a technical perspective, researchers have proposed methods such as Unified Concept Editing and concept ablation to selectively remove or suppress copyrighted features within generative models. Furthermore, analytical tools have been introduced to detect and quantify instances of copyright infringement. From the side of the exploited artists, emerging adversarial strategies, including data poisoning and membership inference attacks, as discussed in the previous sections, are being adopted to increase the cost of training models without proper copyright mechanisms [99].

11. Connections and lineage of adversarial attacks

The development of adversarial attack algorithms in ML is highly iterative, with many techniques evolving directly from their predecessors or adapting successful elements across different attack paradigms. This inheritance and interconnectedness reflect both the rapid innovation and the tendency to generalize effective ideas. Below, we summarise several core attack methods and clarify how newer strategies are built on, extend, or generalize earlier ones.

- **FGSM (Fast Gradient Sign Method):** The simplest gradient-based evasion attack. It takes a single step in the sign of the loss gradient to perturb every input feature by a small amount.
- **BIM (Basic Iterative Method):** Extends FGSM by applying that same step iteratively, clipping after each move so the overall perturbation stays within the chosen norm-ball. This lets it find stronger adversarial examples than a one-shot FGSM.
- **PGD (Projected Gradient Descent):** Generalizes both FGSM and BIM. Like BIM, it takes multiple small gradient steps, but after each step it projects the perturbed sample back onto the valid ϵ-ball. In practice PGD is often called “multi-step FGSM with projection.”
- **DeepFool:** An iterative white-box attack that approximates the classifier as locally linear and computes the minimum perturbation to cross the decision boundary.
- **SmoothFool:** Builds on DeepFool by adding a smoothing stage that removes spiky, detectable perturbation components—yielding subtler, more transferable attacks.
- **C&W (Carlini & Wagner):** Formulates adversarial crafting as an optimization problem with a carefully designed loss that trades off confidence in the wrong class against perturbation size.
- **EAD (Elastic-net Attack to DNNs):** Extends C&W by incorporating both an L_1 term (to encourage sparse, targeted changes) and an L_2 term (to keep overall energy small), blending the strengths of two norms in one attack.
- **ZOO (Zeroth-Order Optimization):** A black-box adaptation of C&W that estimates gradients via finite differences on the remote model’s outputs, so no internal access is needed.
- **CG-ES (Conditional Glow-Evolution Strategy):** Builds on ZOO’s derivative-free optimization by using a learned flow-based transform (Conditional Glow) to generate more flexible, high-quality perturbation samples within an evolution-strategy framework.

This lineage highlights the cross-pollination of ideas in this domain, whether through iterative refinements or the adaptation of white-box approaches to black-box settings.

12. Discussion

This paper has provided an umbrella review of adversarial attacks on AI algorithms, synthesising key findings from numerous systematic reviews. The study categorised existing research into broad thematic areas, ranging from direct evasion and poisoning tactics to privacy-centred attacks such as membership inference and model extraction, while also exploring domain-specific vulnerabilities in fields like intrusion detection, federated learning, text-to-image generation, and generative models. Knowledge about successful strategies in one area (e.g., image classification) does not seamlessly translate to another (e.g., NLP or GNNs). Thus, according to the performed study, the answers to the research questions are as follows:

RQ1: The study reveals that the literature on adversarial attacks is dominated by studies focusing on evasion attacks, often referred to simply as "adversarial attacks" or "adversarial examples." The specific attacks, including evasion attacks, poisoning attacks and inference attacks, have been gathered and described. A comprehensive overview of adversarial attack categories, their goals, and representative methods is presented in Table 2.

RQ2: While many core principles of adversarial attacks hold universally, the synthesis indicates that application-specific constraints significantly shape vulnerabilities and attack strategies:

- **Computer Vision** is the most studied domain among adversarial attacks, with well-established gradient-based and universal perturbation attacks, most of the attacks included in this study started in CV.
- For **Natural Language Processing**, the discrete nature of text requires specialised attack methods (e.g., character- or word-level manipulations, paraphrases) to maintain readability and semantic coherence.
- In **Graph Neural Networks**, Attacks manipulate node attributes or edges. Even minor structural changes can mislead node classification or link prediction.
- Regarding **Intrusion Detection Systems**, feature constraints (e.g., protocol types) must be respected, so adversarial samples that violate real-world network behaviours are invalid.
- For **Federated Learning**, poisoning and backdoor attacks can spread via model aggregation. Malicious clients can degrade or redirect model outputs while evading centralised scrutiny.
- **Generative Models (GANs, DMs VAEs)** are vulnerable to malicious sample injection, membership inference, and model extraction. Backdoors and style-level triggers can also compromise generative outputs.

The prevalence and character of adversarial attack types across these domains are mapped in Table 3.

RQ3: The study identified the following consistent aspects across different adversarial attack scenarios:

- "Small perturbations, major impact" is a guiding principle for generations of adversarial samples across all applications. The defining characteristic of adversarial attacks is that tiny, strategically crafted modifications to inputs can cause disproportionately large errors in model outputs.
- Nearly all adversarial methods revolve around a double-optimisation principle - maximise the model's error, minimise the amount of perturbation.
- Whether in "white-box" or "black-box" modes, attackers exploit what they know about the model: its gradients, output probabilities, or even just its binary decisions. The prevalence of attack types for specific domains is showcased in Table 4.
- Attacks crafted for one model or dataset can often generalise to others. For example, Moosavi-Dezfooli et al. [15] showed that universal adversarial perturbations generated for one classifier often mislead other classifiers, while Su et al. [14] and Jedrzejewski et al. [20] discuss transfer-based attacks where adversarial examples created for a surrogate model are also effective on the target model.

Table 2
Summary of adversarial attack categories, domains, and example methods (with references).

Category	Goal	Domains	Example methods (cited)
Evasion	Fool model at test time with small input changes	CV, NLP, GNN, IDS, Gen. Models	FGSM [29], PGD [33], BIM [32], DeepFool [34], JSMA [30], C&W [13], One-Pixel [14], STA [38], SmoothFool [35], L-BFGS [31], EAD [36], SIAA [41], GAAS [40], OMGDA [37]
Poisoning	Corrupt training to degrade or hijack model	All (esp. FL, IDS, CV)	Indiscriminate [60], Backdoor [67], Clean-label [64], Label Flipping [62], Feature Collision [65], Watermarking [63], Bilevel [66]
Privacy	Extract info about data or model	All	Membership Inference [68], Model Extraction [70], Property Inference [69], Model Inversion [9]
Universal	One perturbation works for many samples/models	CV, general	UAP [15], SV-UAP [57], DF-UAP, UPSET [45]
Defense Bypass	Break/adapt to known defenses	All	BPDA [46], GTAE [55], Shadow Attack [8]
Black, Gray, Score, Decision	Fool model without full access; gradient-free or boundary-based	All	PBBA [42], ZOO [43], HopSkipJump [44], DaST [47], ANGRI [45], Houdini [46], GAP + + [48], CG-ES [49]
Domain-specific	Exploit application constraints	GNN, NLP, IDS, FL, T2I, Gen. Models	GNN node/edge attacks [71], IDS attacks [83], Textual adversaries [94], FL poisoning [87], Nightshade [105], Glaze [106], ACE [107], Mist [110], Anti-DreamBooth [109], ESD [108], AdvDM [110]

Abbreviations: CV: Computer Vision; NLP: Natural Language Processing; GNN: Graph Neural Networks; IDS: Intrusion Detection; FL: Federated Learning; Gen. Models: Generative Models; T2I: Text-to-Image Models.

Table 3
Domain-specific prevalence and character of adversarial attacks (with references).

Attack type	GNNs	IDS	FL	NLP	Gen models	T2I models
Gradient-based	node/edge [71,72]	limited (data constraints) [75,83]	local/global [87,89]	embedding-based [93,94]	core [29,33,97]	latent gradients [101,105]
Black-box	transfer [72,73]	practical [3,75]	often [88,89]	growing [93,94]	some [97,98]	in research [103]
Decision-based	binary tasks [72,73]	some use [75]	–	output-based [94]	experimental [97]	exploratory [103]
Transfer-based	[72]	[75]	[88]	[93]	[97]	–
Universal attacks	known [72]	–	–	–	image [15,97]	(style) [105,106]
Poisoning	node, edge [73,74]	training data [59,75]	client updates [87,89]	data/backdoor [93,94]	data [97,107]	image/style [105–107]
Privacy attacks	inference [69,72]	–	info leak [88,89]	membership [9,68]	extraction [97,98]	(ownership) [103,107]
Natural transf.	structure [72,73]	–	–	paraphrase [92,94]	input variation [97]	prompt/style [103,105]
Attacks on defenses	few [72]	[75]	–	prompt filtering [96,103]	[97]	prompt/latent [101,105]

Table 4
Prevalence of white-box and black-box adversarial attacks across application domains (RQ3 aggregation summary).

Domain	White-box	Black-box	Notes
CV	High	Medium	White-box most studied (FGSM, PGD, C&W). Black-box includes transfer, decision-based.
NLP	Medium	High	Black-box (score, transfer, paraphrase) common due to discrete text.
Graph Neural Nets (GNN)	Medium	Medium	Both settings appear; attacks target edges, features.
IDS	Low	Medium	Practical attacks mainly black-box (PBBA, surrogates).
Federated Learning (FL)	Low	Medium	Black/gray-box dominant (poisoning, privacy); few direct white-box.
Generative Models	Medium	Medium	Both: white-box (latent gradient), black-box (extraction, transfer).
Text-to-Image (Diffusion)	Low	Growing	Most attacks black-box or surrogate-based (e.g., Nightshade, Glaze).

Notes: "High" = most surveyed works; "Medium" = moderate representation; "Low" = infrequent. Black-box includes transfer, decision-based, and surrogate attacks.

Table 5
Mapping of research questions (RQs) to sections and key findings.

Research question	Sections addressed	Key findings
RQ1: *What are the key themes and methodologies covered in existing survey papers on adversarial attacks in ML?*	Sections 3, 4, 7, 8; Tables 2, 3	Literature is dominated by evasion attack research, followed by poisoning, privacy, universal, and domain-specific attacks. Tables summarise categories, goals, and representative methods.
RQ2: *How do adversarial attacks differ across various ML applications?*	Sections 7, 8; Table 3	Application-specific constraints (e.g., data type, structure, operational context) shape vulnerabilities and attack strategies. Table 3 maps prevalence/character of attacks across domains like CV, NLP, GNNs, IDS, FL, generative and diffusion models.
RQ3: *Which aspects of adversarial attacks consistently appear across diverse applications?*	Sections 6, 8; Table 4	Common aspects include: small, targeted perturbations with major impact; double-optimisation (maximising error, minimising change); use of white-box, black-box, and transfer attacks; generalisation of attacks across models and tasks. Table 4 shows the prevalence of attack types by domain.

13. Conclusion

This paper has highlighted both the commonalities and divergences in adversarial attack strategies across diverse ML applications, integrating insights from multiple systematic reviews and meta-analyses. Mapping the landscape of attack methodologies, identifying domain-specific vulnerabilities, and synthesising recurring challenges such as attack transferability, this umbrella review provides a high-level overview of the current adversarial threat landscape. Table 5 maps the research questions of this paper to the relevant sections and key findings, illustrating how literature is dominated by evasion attacks, with poisoning, privacy, universal, and domain-specific attacks also playing major roles. The table summarises how application-specific constraints shape vulnerabilities, and highlights universal principles such as the disproportionate impact of small perturbations and the cross-domain generalisation of attacks. Moving forward, meaningful progress will depend on:

- Greater emphasis on evaluations of both attacks and defenses under realistic, evolving threat models;
- Interdisciplinary collaboration that addresses technical, ethical, and legal challenges in tandem;
- The development of community benchmarks, datasets, and shared evaluation frameworks to enable transparent and reproducible research.

Ensuring the reliability and trustworthiness of AI systems will require sustained cooperation between researchers, practitioners, policymakers, and affected stakeholders. As adversarial threats continue to advance, so too must the strategies for detection, defense, and responsible deployment of ML-powered technologies.

CRediT authorship contribution statement

Marek Pawlicki: Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Aleksandra Pawlicka:** Writing – original draft, Methodology, Formal analysis, Conceptualization. **Rafał Kozik:** Formal analysis, Conceptualization. **Michał Choraś:** Writing – original draft, Supervision, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Marek Pawlicki reports financial support was provided by the Horizon 2020 (Starlight project). Aleksandra Pawlicka reports financial support was provided by the Horizon 2020 (Starlight project). Rafał Kozik reports financial support was provided by the Horizon 2020 (Starlight project). Michał Choraś reports financial support was provided by the Horizon 2020 (Starlight project).

Acknowledgement

The work described in this paper was performed in the H2020 project STARLIGHT ("Sustainable Autonomy and Resilience for LEAs using AI against High Priority Threats"). This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101021797.

Data availability

No data was used for the research described in the article.

References

[1] M.R. Bachute, J.M. Subhedar, Autonomous driving architectures: insights of machine learning and deep learning algorithms, Mach. Learn. Appl. 6 (2021) 100164.

[2] N. Al-LQubaydhi, A. Alenezi, T. Alanazi, A. Senyor, N. Alanezi, B. Alotaibi, M. Alotaibi, A. Razaque, S. Hariri, Deep learning for unmanned Aerial vehicles detection: a review, Comput. Sci. Rev. 51 (2024) 100614.

[3] M. Pawlicki, R. Kozik, M. Choraś, A survey on neural networks for (cyber-) security and (cyber-) security of neural networks, Neurocomputing 500 (2022) 1075–1087.

[4] F. Piccialli, V. Di Somma, F. Giampaolo, S. Cuomo, G. Fortino, A survey on deep learning in medicine: why, how and when? Inf. Fusion 66 (2021) 111–137.

[5] V. Singh, S.-S. Chen, M. Singhania, B. Nanavati, A. Gupta, et al, How are reinforcement learning and deep learning algorithms used for big data based decision making in financial industries–a review and research agenda, Int. J. Inf. Manag. Data Insights 2 (2) (2022) 100094.

[6] S. Raaijmakers, Artificial intelligence for law enforcement: challenges and opportunities, IEEE Secur. Priv. 17 (5) (2019) 74–77.

[7] C. Szegedy, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199, 2013.

[8] S.H. Silva, P. Najafirad, Opportunities and challenges in deep learning adversarial robustness: a survey, arXiv preprint arXiv:2007.00753, 2020.

[9] M. Rigaki, S. Garcia, A survey of privacy attacks in machine learning, ACM Comput. Surv. 56 (4) (2023) 1–34.

[10] Z. Hu, S. Huang, X. Zhu, F. Sun, B. Zhang, X. Hu, Adversarial texture for fooling person detectors in the physical world, in: Proceedings of the IEEE/CVF Conference on Computer Vision and pattern recognition, 2022, pp. 13307–13316.

[11] M. Pautov, G. Melnikov, E. Kaziakhmedov, K. Kireev, A. Petiushko, On adversarial patches: real-world attack on arcface-100 face recognition system, in: 2019 International multi-Conference on Engineering, Computer and Information Sciences (sibircon), IEEE, 2019, pp. 0391–0396.

[12] Y. Deng, L.J. Karam, Universal adversarial attack via enhanced projected gradient descent, in: 2020 IEEE International Conference on Image Processing (ICIP), IEEE, 2020, pp. 1241–1245.

[13] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 39–57, https://doi.org/10.1109/SP.2017.49
[14] J. Su, D.V. Vargas, S. Kouichi, One pixel attack for fooling deep neural networks, arXiv:1710.08864, Oct 2017. https://doi.org/10.1109/TEVC.2019.2890858
[15] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, in: Proceedings of the IEEE Conference on Computer Vision and pattern Recognition (CVPR), 2017.
[16] L. Caviglione, C. Comito, M. Guarascio, G. Manco, Emerging challenges and perspectives in deep learning model security: a brief survey, Syst. Soft Comput. 5 (2023) 200050.
[17] M. Bonczar, P. Ostrowski, A.V. D'Antoni, R.S. Tubbs, J. Iwanaga, S.K. Ghosh, I. Klejbor, M. Kuniewicz, J. Walocha, J. Moryś, M. Koziej, How to write an umbrella review? a step-by-step tutorial with tips and tricks, Folia Morphol. 82 (1) (2023) 1–6, https://doi.org/10.5603/FM.a2022.0104 https://journals.viamedica.pl/folia_morphologica/article/view/92443
[18] J. Yensen, PICO search strategies, Online J. Nurs. Inform. 17 (3) (2013) http://ojni.org/issues/?p = 2860.
[19] A. Eldawlatly, H. Alshehri, A. Alqahtani, A. Ahmad, F. Al-Dammas, A. Marzouk, Appearance of population, intervention, comparison, and outcome as research question in the title of articles of three different anesthesia journals: a pilot study, Saudi J. Anaesth. 12 (2) (2018) 283, https://doi.org/10.4103/sja.SJA_767_17
[20] F.V. Jedrzejewski, L. Thode, J. Fischbach, T. Gorschek, D. Mendez, N. Lavesson, Adversarial machine learning in industry: a systematic literature review, Comput. Secur. 145 (2024) 103988.
[21] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, X. Yi, A survey of safety and trustworthiness of deep neural networks: verification, testing, adversarial attack and defence, and interpretability, Comput. Sci. Rev. 37 (2020) 100270.
[22] X. Huang, D. Kroening, M. Kwiatkowska, W. Ruan, Y. Sun, E. Thamo, M. Wu, X. Yi, Safety and trustworthiness of deep neural networks: a survey, arXiv preprint arXiv:1812.08342, 2018 151.
[23] P. Bountakas, A. Zarras, A. Lekidis, C. Xenakis, Defense strategies for adversarial machine learning: a survey, Comput. Sci. Rev. 49 (2023) 100573.
[24] H. Liang, E. He, Y. Zhao, Z. Jia, H. Li, Adversarial attack and defense: a survey, Electronics 11 (8) (2022) 1283.
[25] J.C. Costa, T. Roxo, H. Proença, P.R.M. Inácio, How deep learning sees the world: a survey on adversarial attacks & defenses, IEEE Access 12 (2024) 61113–61136.
[26] Y. Wang, T. Sun, S. Li, X. Yuan, W. Ni, E. Hossain, H.V. Poor, Adversarial attacks and defenses in machine learning-empowered communication systems and networks: a contemporary survey, IEEE Commun. Surv. Tutorials 25 (4) (2023) 2245–2298.
[27] Z. Kong, J. Xue, Y. Wang, L. Huang, Z. Niu, F. Li, A survey on adversarial attack in the age of artificial intelligence, Wirel. Commun. Mob. Comput. 2021 (1) (2021) 4907754.
[28] A. Muhammad, S.-H. Bae, A survey on efficient methods for adversarial robustness, IEEE Access 10 (2022) 118815–118830.
[29] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv:1412.6572 Dec 2014.
[30] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European Symposium on Security and Privacy (EUROSP), Mar 2016, https://doi.org/10.1109/eurosp.2016.36
[31] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199, 2013.
[32] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, arXiv:1607.02533, Jul 2016.
[33] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, arXiv:1706.06083, Jun 2017.
[34] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, DeepFool: a simple and accurate method to fool deep neural networks, arXiv:1511.04599, Nov 2015.
[35] A. Dabouei, S. Soleymani, F. Taherkhani, J. Dawson, N.M. Nasrabadi, SmoothFool: an efficient framework for computing smooth adversarial perturbations, arXiv:1910.03624, Oct 2019.
[36] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, C.-J. Hsieh, EAD: elastic-net attacks to deep neural networks via adversarial examples, arXiv:1709.04114, Sep 2017.
[37] U. Jang, X. Wu, S. Jha, Objective metrics and gradient descent algorithms for adversarial examples in machine learning, in: Proceedings of the 33rd annual Computer Security Applications Conference, ACM, New York, NY, USA, 2017, pp. 262–277, https://doi.org/10.1145/3134600.3134635
[38] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, D. Song, Spatially transformed adversarial examples, arXiv:1801.02612, Jan 2018.
[39] Y. Song, R. Shu, N. Kushman, S. Ermon, Constructing unrestricted adversarial examples with generative models, arXiv:1805.07894, May 2018.
[40] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, The space of transferable adversarial examples, arXiv:1704.03453, Apr 2017.
[41] F. Croce, M. Hein, Sparse and imperceivable adversarial attacks, arXiv:1909.05040, Sep 2019.
[42] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z.B. Celik, A. Swami, Practical black-box attacks against machine learning, in: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, 2017, pp. 506–519.
[43] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, C.-J. Hsieh, ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models, arXiv:1708.03999, Aug 2017. https://doi.org/10.1145/3128572.3140448
[44] J. Chen, M.I. Jordan, M.J. Wainwright, HopSkipJumpAttack: a query-efficient decision-based attack, arXiv:1904.02144, Apr 2019.
[45] S. Sarkar, A. Bansal, U. Mahbub, R. Chellappa, UPSET and ANGRI: breaking high performance image classifiers, arXiv:1707.01159, Jul 2017.
[46] M. Cisse, Y. Adi, N. Neverova, J. Keshet, Houdini: fooling deep structured prediction models, arXiv:1707.05373, Jul 2017.
[47] A. Athalye, N. Carlini, D. Wagner, Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, arXiv:1802.00420, Feb 2018.
[48] X. Mao, Y. Chen, Y. Li, Y. He, H. Xue, GAP + +: learning to generate target-conditioned adversarial examples, arXiv:2006.05097, Jun 2020.
[49] Y. Feng, B. Wu, Y. Fan, L. Liu, Z. Li, S. Xia, Boosting black-box attack with partially transferred conditional adversarial distribution, arXiv:2006.08538, Jun 2020.
[50] R. Duan, X. Ma, Y. Wang, J. Bailey, A.K. Qin, Y. Yang, Adversarial camouflage: hiding physical-world attacks with natural styles, arXiv:2003.08757, Mar 2020.
[51] Y. Tan, Z. Cai, M.S. Asif, Transformation-dependent adversarial attacks, arXiv:2406.08443, Jun 2024.
[52] J. Li, G. Li, The triangular trade-off between robustness, accuracy and fairness in deep neural networks: a survey, ACM Comput. Surv. 57 (6) (2024) 1–40.
[53] G.S. Nadella, H. Gonaygunta, K. Meduri, S. Satish, Adversarial attacks on deep neural network: developing robust models against evasion technique, Trans. Latest Trends Artif. Intell. 4 (4) (2023) 1168–2519.
[54] A. Raghunathan, J. Steinhardt, P. Liang, Certified defenses against adversarial examples, arXiv preprint arXiv:1801.09344, 2018.
[55] N. Carlini, G. Katz, C. Barrett, D.L. Dill, Provably minimally-distorted adversarial examples, arXiv:1709.10207, Sep 2017.
[56] C. Zhang, P. Benz, C. Lin, A. Karjauv, J. Wu, I.S. Kweon, A survey on universal adversarial attack, arXiv preprint arXiv:2103.01498, 2021.
[57] V. Khrulkov, I. Oseledets, Art of singular vectors and universal adversarial perturbations, arXiv:1709.03582, Sep 2017.
[58] W. Qiu, A survey on poisoning attacks against supervised machine learning, arXiv preprint arXiv:2202.02510, 2022.
[59] M.A. Ramirez, S.-K. Kim, H.A. Hamadi, E. Damiani, Y.-J. Byon, T.-Y. Kim, C.-S. Cho, C.Y. Yeun, Poisoning attacks and defenses on artificial intelligence: a survey, arXiv preprint arXiv:2202.10276, 2022.
[60] A.E. Cinà, K. Grosse, A. Demontis, S. Vascon, W. Zellinger, B.A. Moser, A. Oprea, B. Biggio, M. Pelillo, F. Roli, Wild patterns reloaded: a survey of machine learning security against training data poisoning, ACM Comput. Surv. 55 (13s) (2023) 1–39.
[61] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning, arXiv:1712.05526, Dec 2017.
[62] B. Biggio, B. Nelson, P. Laskov, Poisoning attacks against support vector machines, arXiv:1206.6389, Jun 2012.
[63] G. Wang, X. Chen, C. Xu, Adversarial watermarking to attack deep neural networks, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, speech and signal Processing (ICASSP), IEEE, 2019, pp. 1962–1966, https://doi.org/10.1109/ICASSP.2019.8682351
[64] A. Gupta, A. Krishna, Adversarial clean label backdoor attacks and defenses on text classification systems, arXiv:2305.19607, May 2023.
[65] A. Shafahi, W.R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, T. Goldstein, Poison Frogs! Targeted Clean-Label poisoning attacks on neural networks, arXiv:1804.00792, Apr 2018.
[66] W.R. Huang, J. Geiping, L. Fowl, G. Taylor, T. Goldstein, MetaPoison: practical general-purpose clean-label data poisoning, arXiv:2004.00225, Apr 2020.
[67] W. Guo, B. Tondi, M. Barni, An overview of backdoor attacks against deep neural networks and possible defences, IEEE Open J. Signal Process. 3 (2022) 261–287, https://doi.org/10.1109/OJSP.2022.3190213 https://ieeexplore.ieee.org/document/9827581/
[68] H. Ali, A. Qayyum, A. Al-Fuqaha, J. Qadir, Membership inference attacks on DNNs using adversarial perturbations, arXiv:2307.05193, Jul 2023.
[69] M.P.M. Parisot, B. Pejo, D. Spagnuelo, Property inference attacks on convolutional neural networks: influence and implications of target Model's complexity, arXiv:2104.13061, Apr 2021.
[70] M. Juuti, S. Szyller, S. Marchal, N. Asokan, PRADA: protecting against DNN Model stealing attacks, arXiv:1805.02628, May 2018.
[71] F. Guan, T. Zhu, W. Zhou, K.-K.R. Choo, Graph neural networks: a survey on the links between privacy and security, Artif. Intell. Rev. 57 (2) (2024) 40.
[72] E. Dai, T. Zhao, H. Zhu, J. Xu, Z. Guo, H. Liu, J. Tang, S. Wang, A comprehensive survey on trustworthy graph neural networks: privacy, robustness, fairness, and explainability, Mach. Intell. Res. 21 (6) (2024) 1–51.
[73] Z. Zhai, P. Li, S. Feng, State of the art on adversarial attacks and defenses in graphs, Neural Comput. Appl. 35 (26) (2023) 18851–18872.
[74] J. Xu, J. Chen, S. You, Z. Xiao, Y. Yang, J. Lu, Robustness of deep learning models on graphs: a survey, AI Open 2 (2021) 69–78.
[75] E. Alhajjar, P. Maxwell, N. Bastian, Adversarial machine learning in network intrusion detection systems, Expert Syst. Appl. 186 (2021) 115782.
[76] H. Jmila, M.I. Khedher, Adversarial machine learning for network intrusion detection: a comparative study, Comput. Netw. 214 (2022) 109073.
[77] P. Dixit, S. Silakari, Deep learning algorithms for cybersecurity applications: a technological and status review, Comput. Sci. Rev. 39 (2021) 100317.
[78] A. Alotaibi, M.A. Rassam, Adversarial machine learning attacks against intrusion detection systems: a survey on strategies and defense, Futur. Internet 15 (2) (2023) 62.
[79] P. Papadopoulos, O. Thornewill von Essen, N. Pitropakis, C. Chrysoulas, A. Mylonas, W.J. Buchanan, Launching adversarial attacks against network intrusion detection systems for IOT, J. Cybersecurity Priv. 1 (2) (2021) 252–273.
[80] H.A. Alatwi, C. Morisset, Adversarial machine learning in network intrusion detection domain: a systematic review, arXiv preprint arXiv:2112.03315, 2021.

[81] K. Aryal, M. Gupta, M. Abdelsalam, P. Kunwar, B. Thuraisingham, A survey on adversarial attacks for malware analysis, IEEE Access 13 (2024) 428–459.
[82] F. Uccello, M. Pawlicki, A. Pawlicka, S. D'Antonio, R. Kozik, M. Choraś, The evaluation of adversarial attacks against ML-powered NIDS in a realistic scenario, in: International Conference on Applied Intelligence, Springer, 2024, pp. 314–324.
[83] G. Apruzzese, M. Andreolini, L. Ferretti, M. Marchetti, M. Colajanni, Modeling realistic adversarial attacks against network intrusion detection systems, Digit. Threats: Res. Pract. 3 (3) (2022) 1–19.
[84] M.A. Merzouk, F. Cuppens, N. Boulahia-Cuppens, R. Yaich, Investigating the practicality of adversarial evasion attacks on network intrusion detection, Ann. Telecommun. 77 (11) (2022) 763–775.
[85] C. Zhang, X. Costa-Perez, P. Patras, Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms, IEEE/ACM Trans. Netw. 30 (3) (2022) 1294–1311.
[86] S. Zhou, C. Liu, D. Ye, T. Zhu, W. Zhou, P.S. Yu, Adversarial attacks and defenses in deep learning: from a perspective of cybersecurity, ACM Comput. Surv. 55 (8) (2022) 1–39.
[87] G. Xia, J. Chen, C. Yu, J. Ma, Poisoning attacks in Federated Learning: a survey, IEEE Access 11 (2023) 10708–10722.
[88] A.K. Nair, E.D. Raj, J. Sahoo, A robust analysis of adversarial attacks on federated learning environments, Comput. Stand. Interfaces 86 (2023) 103723.
[89] V. Kaushal, S. Sharma, Securing the collective intelligence: a comprehensive review of federated learning security attacks and defensive strategies, Knowl. Inf. Syst. 67 (4) (2025) 1–39.
[90] F. Xia, W. Cheng, A survey on privacy-preserving federated learning against poisoning attacks, Cluster Comput. 27 (10) (2024) 13565–13582.
[91] K.N. Kumar, C.K. Mohan, L.R. Cenkeramaddi, The impact of adversarial attacks on federated learning: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 46 (5) (2023) 2672–2691.
[92] W. Wang, R. Wang, L. Wang, Z. Wang, A. Ye, Towards a robust deep neural network in texts: a survey, arXiv preprint arXiv:1902.07285, 2019.
[93] C. Guo, A. Sablayrolles, H. Jégou, D. Kiela, Gradient-based against text transformers, arXiv preprint arXiv:2104.13733, 2021.
[94] S. Qiu, Q. Liu, S. Zhou, W. Huang, Adversarial attack and defense technologies in natural language processing: a survey, Neurocomputing 492 (2022) 278–307.
[95] N. Jain, A. Schwarzschild, Y. Wen, G. Somepalli, J. Kirchenbauer, P.-Y. Chiang, M. Goldblum, A. Saha, J. Geiping, T. Goldstein, Baseline defenses for adversarial attacks against aligned language models, arXiv preprint arXiv:2309.00614, 2023.
[96] E. Shayegani, Y. Dong, N. Abu-Ghazaleh, Jailbreak in pieces: compositional adversarial attacks on multi-modal language models, in: The twelfth International Conference on Learning representations, 2023.
[97] H. Sun, T. Zhu, Z. Zhang, D. Jin, P. Xiong, W. Zhou, Adversarial attacks against deep generative models on data: a survey, IEEE Trans. Knowl. Data Eng. 35 (4) (2021) 3367–3388.
[98] H. Hu, J. Pang, Stealing machine learning models: attacks and countermeasures for generative adversarial networks, in: Proceedings of the 37th annual Computer Security Applications Conference, 2021, pp. 1–16.
[99] H. Chen, Q. Xiang, J. Hu, M. Ye, C. Yu, H. Cheng, L. Zhang, Comprehensive exploration of diffusion models in image generation: a survey, Artif. Intell. Rev. 58 (4) (2025) 99.
[100] S.K. Mandala, A Contextualized Survey on the State and Future Directions of Diffusion Models, OSF Preprints, 2023, https://doi.org/10.31219/osf.io/xcfdu
[101] C. Zhang, M. Hu, W. Li, L. Wang, Adversarial attacks and defenses on text-to-image diffusion models: a survey, Inf. Fusion 114 (2024) 102701.
[102] Y. Yang, B. Hui, H. Yuan, N. Gong, Y. Cao, Sneakyprompt: jailbreaking text-to-image generative models, in: 2024 IEEE Symposium on Security and privacy (SP), IEEE, 2024, pp. 897–912.
[103] V.T. Truong, L.B. Dang, L.B. Le, Attacks and defenses for generative diffusion models: a comprehensive survey, arXiv preprint arXiv:2408.03400, 2024.
[104] Y. Zhang, Z. Chen, C.-H. Cheng, W. Ruan, X. Huang, D. Zhao, D. Flynn, S. Khastgir, X. Zhao, Trustworthy text-to-image diffusion models: a timely and focused survey, arXiv preprint arXiv:2409.18214, 2024.
[105] S. Shan, W. Ding, J. Passananti, S. Wu, H. Zheng, B.Y. Zhao, Nightshade: prompt-specific poisoning attacks on text-to-image generative models, in: 2024 IEEE Symposium on Security and privacy (SP), IEEE Computer Society, 2024, pp. 212.
[106] S. Shan, J. Cryan, E. Wenger, H. Zheng, R. Hanocka, B.Y. Zhao, Glaze: protecting artists from style mimicry by {text-to-image} models, in: 32nd USENIX Security symposium (USENIX Security 23), 2023, pp. 2187–2204.
[107] B. Zheng, C. Liang, X. Wu, Targeted attack improves protection against unauthorized diffusion customization, arXiv preprint arXiv:2310.04687, 2023.
[108] R. Gandikota, J. Materzynska, J. Fiotto-Kaufman, D. Bau, Erasing concepts from diffusion models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 2426–2436.
[109] T. Van Le, H. Phung, T.H. Nguyen, Q. Dao, N.N. Tran, A. Tran, Anti-Dreambooth: protecting users from personalized text-to-image synthesis, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 2116–2127.
[110] C. Liang, X. Wu, Y. Hua, J. Zhang, Y. Xue, T. Song, Z. Xue, R. Ma, H. Guan, Adversarial example does good: preventing painting imitation from diffusion models via adversarial examples, arXiv preprint arXiv:2302.04578, 2023.

Author biography

Marek Pawlicki Ph.D. Eng., holds a university professor position at the Bydgoszcz University of Science and Technology. He has been involved in a number of international projects related to cybersecurity, critical infrastructures protection, software quality etc. (e.g. H2020 SPARTA, H2020 SIMARGL, H2020 APPRAISE, H2020 STARLIGHT, HE AI4CYBER, H2020 ELEGANT). He is an author and co-author of over 130 peer-reviewed scientific publications. His interests pertain to the application of machine learning in several domains, including cybersecurity.

Aleksandra Pawlicka Ph.D. A philologist and R&D specialist. Interested in computer science, linguistics, language teaching and learning, and pedagogy; in her works, she combines those fields. She is an author and co-author of a number of multidisciplinary scientific publications, and has been involved in several international projects, such as H2020 SIMARGL, H2020 SPARTA, H2020 PREVISION, and more.

Rafał Kozik Ph.D., D.Sc., Eng. obtained his Doctor of Science degree in computer science from West Pomeranian University of Technology in Szczecin in 2019. He holds a professor position at the Bydgoszcz University of Science and Technology in Bydgoszcz. In 2013 he received his Ph.D. in telecommunications from the University of Science and Technology (UTP) in Bydgoszcz. Since 2009, he has been involved in a number of international and national research projects related to cybersecurity, critical infrastructures protection, software quality, and data privacy (e.g. FP7 INTERSECTION, FP7 INSPIRE, FP7 CAMINO, FP7 CIPRNet, SOPAS, SECOR, H2020 Q-Rapids). He is an author and co-author of over 200 reviewed scientific publications.

Michał Choraś is a full professor (title granted in 2021) and he works at Bydgoszcz University of Science and Technology, Bydgoszcz, where he is the Head of the Teleinformatics Systems Division and the PATRAS Research Group. He is also affiliated with FernUniversitat in Hagen, Germany, where he was a Project Coordinator for H2020 SIMARGL (Secure intelligent methods for advanced recognition of malware and stegomalware) and will soon coordinate new Horizon Europe PERUN project. He is the author and co-author of over 380 reviewed scientific publications. His research interests include data science, AI, and pattern recognition in several domains, e.g., cyber security, image processing, software engineering, prediction, anomaly detection, correlation, biometrics, and critical infrastructures protection. He has been involved in many EU projects (e.g., SocialTruth, STARLIGHT, ULTIMATE, and SPARTA).