The Path to Defence: A Roadmap to Characterising Data Poisoning Attacks on Victim Models

TAREK CHAALAN, SHAONING PANG, and JOARDER KAMRUZZAMAN,
Internet Commerce Security Lab and Center for Smart Analytics, Federation University, Australia
IQBAL GONDAL, School of Computing Technology, STEM College RMIT University, Royal Melbourne Institute of Technology, Australia
XUYUN ZHANG, School of Computing, Macquarie University, Australia

Data Poisoning Attacks (DPA) represent a sophisticated technique aimed at distorting the training data of machine learning models, thereby manipulating their behavior. This process is not only technically intricate but also frequently dependent on the characteristics of the victim (target) model. To protect the victim model, the vast number of DPAs and their variants make defenders rely on trial and error techniques to find the ultimate defence solution which is exhausting and very time-consuming. This paper comprehensively summarises the latest research on DPAs and defences, proposes a DPA characterizing model to help investigate adversary attacks dependency on the victim model, and builds a DPA roadmap as the path navigating to defence. Having the roadmap as an applied framework that contains DPA families sharing the same features and mathematical computations will equip the defenders with a powerful tool to quickly find the ultimate defences, away from the exhausting trial and error methodology. The roadmap validated by use cases has been made available as an open access platform, enabling other researchers to add in new DPAs and update the map continuously.

CCS Concepts: • **Security and privacy** → **Software and application security;**

Additional Key Words and Phrases: DPA, data poisoning attacks, adversarial attacks, adversarial defences, neural networks, trustworthy ML, trustworthy AI, roadmap, victim model

ACM Reference format:
Tarek Chaalan, Shaoning Pang, Joarder Kamruzzaman, Iqbal Gondal, and Xuyun Zhang. 2024. The Path to Defence: A Roadmap to Characterising Data Poisoning Attacks on Victim Models. *ACM Comput. Surv.* 56, 7, Article 175 (April 2024), 39 pages.
https://doi.org/10.1145/3627536

175

This research was supported by the Centre for Smart Analytics, Federation University Australia. Dr. Zhang's involvement in this work was partially supported by the ARC DECRA Grant DE210101458.
Authors' addresses: T. Chaalan, S. Pang, and J. Kamruzzaman, Internet Commerce Security Lab and Center for Smart Analytics, Federation University, University Drive, Mount Helen VIC 3350, Melbourne, Victoria, PO BOX 663 Ballarat, Victoria 3353, Australia; e-mails: tarekchaalan@students.federation.edu.au, p.pang@federation.edu.au, joarder.kamruzzaman@federation.edu.au; I. Gondal, School of Computing Technology, STEM College RMIT University, Royal Melbourne, Institute of Technology, RMIT University Plenty Road, Bundoora, Victoria 3083, PO BOX 71, Bundoora, Victoria 3038, Australia; e-mail: Iqbal.Gondal@rmit.edu.au; X. Zhang, School of Computing, Macquarie University, 4 Research Park Dr, Macquarie Park, Sydney NSW 2113, Australia; e-mail: xuyun.zhang@mq.edu.au.

0360-0300/2024/04-ART175 $15.00
https://doi.org/10.1145/3627536

1 INTRODUCTION

Data Poisoning Attacks (DPAs) have been a serious threat to machine learning models used in computer vision, speech recognition, and other **Artificial Intelligence (AI)** application areas. The attacks are based on the minimal change to data [228] and can deceive a trained model to produce incorrect outcomes. Thus, DPAs are able to poison complex and state-of-the-art machine learning models that are central to the decision-making processes of any intelligent system running in various sectors including business, industry, and defence. For example, Microsoft reported a DPA attack that targeted the company chatbot Tay whose training data were poisoned with racist tweets and consequently caused the chatbot's conversational algorithm to generate offensive tweets [2]. The consequence of a DPA can even lead to loss of human life. A recent piece of news reported that a vulnerability of the AI module in the autopilot of a Tesla car was exploited, and caused the failure to recognise a stopped car in the lane as an obstacle [1].

A DPA needs a minimum of five elements to form one attack. These elements are victim model, poisoning techniques (e.g., indirect poisoning, data injection, data manipulation, logic corruption), knowledge of training data and/or victim model, attack mode (e.g., repetitive and non-repetitive), and core perturbation function or algorithm. In principle, a DPA attack is driven by a mathematical perturbation function or a specially designed data perturbation algorithm. A mathematical perturbation function-driven DPA crafts adversarial samples using a pre-defined calculation to modify the original data samples. Despite the modification causing the change in the internal data distribution, such perturbation is imperceptible to humans since the individual samples look similar to the original ones. Such complex perturbation functions eventually will mislead the classifier to output wrong predictions.

In practice, it is difficult to trace a DPA in that its mathematical perturbation functions are dynamic and also transferable. According to [66], in a black-box setting, transferability provides a DPA with the ability to expand its maliciousness from one victim model to other models while being equally effective. For example, the ensemble adversarial attack uses a perturbation function to create adversarial data which is tested on a local surrogate model and then the poison can be transferred to multiple victim models [225]. Theoretically, the transferability of DPA is related to three metrics connected to target model complexity: (1) the size of the input gradient of the model; (2) how well the gradients of the surrogate and target models align; and (3) the variance of the loss landscape optimised to generate the attack points [66].

The execution of a DPA perturbation mathematical function can be computationally expensive. The level of computational cost varies with the type of the perturbation method. The fixed-point disturbance will experience the least computation cost, while dynamic fixed point and gradient-based computation will experience progressively higher cost due to the complex and iterative nature of the computation [65]. Similar to the mathematical perturbation function, a DPA can also be driven by a specially designed perturbation algorithm. The purpose of the algorithm is to encode the adversarial attack behavior like that of using a mathematical function, but with added algorithmic complexity such as, adding points to the training set sequentially, performing repetition, and iterating multiple computational steps until a certain set of conditions is met.

The DPA behavior can be characterized by other features, such as attack frequency, assembly, and repetition to convergence. For the same DPA, its behavior changes significantly according to the chosen parameters, creating a variety of perturbations outcomes. Some parameters like step size, norm, target confidence, and perturbation search methods have a big impact on the perturbation visibility.

DPA can be scalable as the attackers can simply modify or adjust the parameters of iteration to scale up the perturbation influence/weight on the target. An iterative DPA often makes small unnoticeable modifications at each iteration, which becomes malicious over the iterations, and makes the whole process complex and computationally expensive. DPA mode configuration adds further complexity through multiple backwards passes of gradient computation, increasing both time and space complexity. Due to the fact that the attacks have a repetition frequency where the model during an attack will be queried one or multiple times (iterative mode), the repetition of DPA will add more complexity to the adversarial crafting process. A specific DPA can be encapsulated in a pre-designed repetition mode, and also can be performed as a single attack or an ensemble of attacks where multiple perturbation methods are used by the attackers based on the threat model.

DPA can be dynamic as well, because it can be applied in an automated, semi-manual, or full manual framework. Execution of a DPA requires conducting multiple queries (scans) on the target model, which is also called the victim model, if the model has already been compromised. These queries take place in the reconnaissance phase aiming to identify the target model settings and gather specific information that is required to specify which DPA should be applied and will have a higher chance to break through.

As per the complexity and dynamic nature of DPAs discussed above, it is essential for machine learning practitioners who deploy models to adopt frameworks to assess DPA risk for models/assets protection. For an unknown DPA, it is practically very difficult for a cyber defence professional to search through hundreds of options to identify a DPA, and quickly find a reliable defence solution. In most cases, only a tentative solution is adopted which works only for a brief period of time because of the dynamic nature of DPAs. This ad-hoc approach is inefficient because of the absence of a roadmap to characterize a DPA and map it to a defence solution. For example, Langlotz et al. [130] created a roadmap that links foundational machine learning algorithms to various medical imaging usages including medical image reconstruction, noise reduction, quality assurance, triage, segmentation, computer-aided detection, computer-aided classification, and radiogenomics. This roadmap in practice facilitates the identification of solutions. Inspired by this, we propose formulation of such a navigating path that can assist cyber defence professionals in quickly generating a solution, especially for real-time critical applications.

Data Poisoning attacks are very effective against Deep Learning models despite their impressive ability to solve complex problems such as image classification and recognition. DPA exploits the Deep Learning vulnerabilities that imply a huge limitation and security concerns on the development of models if these security issues persist. Therefore, there have been many defences proposed since the discovery of adversarial attacks by Szegedy et al. [229]. These defences are ineffective to stop complex and strong attacks as argued by Machado et al. [196].

1.1 Differentiation

Evasion attacks (EAs) [85] are categorized in literature as a group of adversary attacks different to DPA, because an EA perturbs the input samples at testing time, instead of polluting the training data as a DPA does [33]. Note that regardless of the different victim models, the majority of EAs and DPAs use the same type of perturbation core. From the perspective of victim model despondency, an EA can be treated as a DPA in the configuration of poisoning testing data.

Backdoor attacks (BAs) are another category of adversary attacks. Similar to a DPA, a BA aims to inject poisoned data samples into training data. A DPA downgrades the performance in predicting true testing samples, whereas a BA preserves the performance on true samples, similarly with the model, while changing the prediction of attacked samples (i.e., true testing samples with embedded triggers) to the target label. From this angle, data poisoning can be regarded as the 'non-targeted' poisoning-based backdoor attack with transparent triggers to a certain extent.

Without loss of generality, we consider a BA as a triggered DPA, and an EA as a configured DPA, and use one consistent term of DPA throughout this paper to cover the three types of attacks.

Table 1. A Summary of Recent DPA Survey Studies

Work	DPA families	Year
Sagar et al. [207]	Label Flipping Attacks, Gradient Descent Attacks	2023
Tian et al. [233]	Non convex Optimisation Attacks, Label Flipping Attacks	2023
Ramirez et al. [193]	Label Flipping Attacks, Attacks on SVM, Attacks on Clustering / K-Means Attacks, Non convex Optimization Attacks / Gradient Optimization Attacks, GAN Generated Poisoning	2022
Goldblum et al. [96]	Collision Poisoning, Non convex Optimisation Attacks, Influence Functions Poisoning Attacks, Label Flipping Attacks, Vanishing Gradients / Gradient Obfuscation	2021
Koh et al. [123]	Influence Functions Poisoning, Iterative Optimisation Attacks	2021
Kong et al. [124]	Gradient Descent Attacks, Saddle Point Optimization Attacks	2021
Machado et al. [196]	Universal Adversarial Attacks, Natural Evolutionary Strategies Attacks, Boundary Attacks, Momentum Iterative Attacks, Projected Gradient Descent Attacks, Spatially Transformed Attacks	2020
Gao et al. [89]	Backdoor Attacks, Universal Adversarial Patch	2020
Bhambr et al. [20]	Gradient-free Attacks, Advanced Local Search Attacks	2020
Liu et al. [148]	Generalised Membership Attacks, Universal Adversarial Attacks	2020
Chakraborty et al. [39]	Papernote Adversarial Crafting Attacks, GAN Attacks, Membership inference Attacks	2020
Yuan et al. [266]	Feature Adversary Attacks, Generative Adversarial	2018
Serban et al. [211]	Non convex Optimisation Attacks, Geometric Transformations Attacks, Generative Modeling Attacks	2018
Liu et al. [144]	Generative Adversarial, LCA Label Modification, Attacks on SVM, Attacks on Clustering, Attacks on PPCA/Lasso	2018
Chakraborty et al. [38]	Iterative Optimisation Attacks, BFGS, FGSM, JSM	2018

It is important to note that DPA has been extensively researched and analyzed in the literature. Numerous studies have been conducted, identifying different DPA families that exhibit common features and characteristics. Table 1 presents a summary of recent DPA-related surveys conducted in the past five years and lists the DPA families examined for each survey. For instance, Ramirez et al. [193] conducted a comprehensive review on DPA in Artificial Intelligence (AI), identifying seven DPA families, namely Label Flipping Attacks, Attacks on SVM, Attacks on Clustering, Gradient Optimization Attacks, GAN Generated Poisoning, Features Adversary Attacks, Crowd-Sensing Attack. This work provides valuable insights into AI targeted DPAs, which facilitates a deeper understanding of the vulnerabilities and countermeasures in AI systems. Meanwhile, Sagar et al. [207] delivered an analysis of Poisoning Attacks and their defences within the realm of Federated Learning. In contrast, Gao et al. [89] centered their research on a specific kind of Data Poisoning Attack known as "Backdoor Attacks". Although all these surveys possess valuable insights, none provide a comprehensive review that covers all existing DPA families, explores their interconnections, and importantly, establishes a connection from specific attacks to effective defence solutions. Motivated by this gap, the objectives of this work are to consolidate DPA families from existing surveys, integrate new DPA families derived from recent studies on DPA attacks and defences, and construct a comprehensive DPA roadmap. This roadmap will provide a critical tool for defenders to devise effective solutions to counter these attacks.

The contributions of this paper are summarized as follows:

— A full set of DPA measurements are formulated as the baselines for our roadmap investigation.
— A DPA characteristic model is proposed and we demonstrate its core role in categorisation of DPAs.
— We develop a DPA roadmap that comprehensively covers 221 recently published DPAs and 111 DPA defence methods. The roadmap can facilitate security professionals to identify

Table 2. Notations

Variable	Description
$\mathbf{x}$	An original data sample (unmodified)
y	The truth class label of x
$\mathbf{X}$	A set of original data sample
t	The time step $t = 1, 2, \ldots$
$\hat{y}_t$	The predicted class label at time t
ζ	Perturbation model
$\zeta(\mathbf{x}_t)$	A perturbed data sample
$\zeta(y_t)$	A perturbed class label
$\zeta(D_t)$	A perturbed data set
$\zeta(D_v)$	A perturbed validation data set
$\zeta(D_{tr})$	A perturbed training data set
g	A Threat model
$f(x)$	Victim model
$\mathcal{M}$	A road map from attack to defence
$\mathcal{M}(\zeta)$	A road map on victim model

the rules of forming a DPA from the attacker's viewpoint and the potential defence solutions.

1.2 Definitions and Notations

For the convenience and simplicity of the presentation, we summarize the key notations and variables in Table 2. A dataset is defined as $\{x_i, y_i\}_{i=1}^{N}$, where x_i is a data sample with a label y_i and N is the size of the dataset.

An adversarial example dataset is denoted as $\zeta(\mathbf{x}_t)$ where $\zeta(\mathbf{x}_t) : D(x, \zeta(\mathbf{x}_t)) < \eta, f(\zeta(\mathbf{x}_t)) \neq y$, where D is the dataset.

The rest of the paper is organized as follows: Section 2 introduces the current status of DPA variations and defence mechanisms and identifies the Roadmap solution. Section 3 presents the DPA measurements and characteristic model, highlighting the core elements of DPA, namely the data and victim model. By applying the DPA characteristic model to DPA grouping and edge derivations, the proposed DPA roadmap, along with a validation case study, is introduced in Section 4. Section 5 discusses the limitations of the approach and explores future research directions. Finally, in Section 6, we conclude the paper.

2 OVERVIEW

Data poisoning is a class of adversarial attacks to machine learning models (victim models) where adversaries intend to degrade the model's performance by contaminating the training data. Given a training dataset $\{x\}$, a data poisoning attack often modifies the training dataset by injecting perturbed samples $\zeta(\mathbf{x})$ or artificially crafted new samples, so as to alter the learning model decision function $f(\mathbf{x})$ that decreases the accuracy of the learning model. The learning model hereafter is also referred to as **victim model (VM)**.

Such attacks have been applied against a wide range of learning models including online incremental learning model and online multi-task learning. A DPA manipulates data x for training in order to cause the VM to fail during training and inference. Data poisoning in its early discoveries targets typical VMs including support vector machines and neural networks [140]. A variety of DPAs are now impacting almost every machine learning model in different ways.

2.1 Dependency on Data

DPA can also be found to have a dependency on the type of data. For example, the Image Scaling DPA [191] is image agnostic and targets only image data. The Concealed DPA [240] is a **Neural Language Processing (NLP)** based DPA that works only on text data. The VenoMave [6] is an audio-specific DPA that impacts digital signal data. There are also DPAs specific to unstructured data. For example, the Vanilla PCA Poisoning [204] is a DPA for only unstructured sensor network data. Graphs embedded knowledge also are targeted by DPA which gives so called direct and indirect DPA [271].

On the other hand, a DPA typically can be applied to multiple types of data, but may have a preference in favor of or against a certain data type. For example, the **universal adversary perturbation (UAP)** [269] represents a large family of DPAs including DF-UAP, SV-UAP, GAP, NAG, Cos-UAP, FFF, AAA, GD-UAP, PD-UAP, and CD-UAP. The family is applicable to image, text and audio, but not sustainable for structured data. Such restrictions come from the limits of software application environment and victim model dependency. For example, the **convolutional neural networks (CNNs)** are widely used in computer vision applications owing to its outstanding performance on image pattern recognition. This in return causes those CNN-dependant DPAs [103] working only on image data.

2.2 Dependency on Perturbation Core

The effectiveness of DPA is highly related to the adopted perturbation core ζ in the attack. The perturbation function defines the adversarial example generation methods [7, 26, 35] which is responsible for crafting the small anomaly/perturbation that is added to the input during training and is sufficient to change the prediction of the learning model. Each DPA normally has a unique perturbation core which allows us to categorize a DPA according to its core. In some cases, multiple DPAs may share the same type of perturbation function, but with varied parameters which differentiates their behaviour.

For example, FGS, IFGS, L-GFGS and Box-constrained L-GFGS are a family of DPAs which use the same **fast gradient sign (FGS)** core [98] which linearizes the cost function around the current value for obtaining an optimal max-norm constrained perturbation. The IFGS is an iterative version of FGS, which applies the sign of the gradient at each iteration. The L-GFGS is an enhanced version of the original FGS in producing stronger and faster adversarial examples. The Box-constrained L-GFGS ensures reliable finding of those adversarial examples [127].

2.3 Dependency on Victim Model

A DPA can be dependent on a specific **victim model (VM)**, the model type, inputs, outputs, training data, parameters, and many other factors. In other words, the perturbation function of DPA ζ is defined based on a given learning model $f(x)$, which causes the DPA associate with one specific VM or VM family. In practice, such dependency highlights the vulnerabilities of learning model and leads the attacker to exploit these vulnerabilities. For example, SVM-PA perturbation [254] was designed to attack the **Support Vector Machine (SVM)** in its kernel space changing the integrity of model. The DPA needs a kernel space to execute the attack, thus non-kernel learning models will not be impacted by this attack. Gradient-based DPAs [107] are set to interfere or modify the gradient calculation during the learning (model updating) process. These DPAs are able to impact a large group of models that rely on gradient calculation for learning.

Also, a DPA can be applicable to universal learning models, which means ζ is independent to $f(x)$. For example, in the case of Black-box scenario, the attack requires no information about the VM structure and parameters, but the input and results labeled by the learning model. In the scenario of attack transferring, a DPA against one learning model is also effective against a different, potentially unknown, model. For a group of learning models with a similar decision function, if a DPA successfully breaks one model, then similar DPAs can be effective to the remaining models. Nevertheless, training classifiers on compromised data implies the VM independent attack, since the contamination leads to the mis-classification of any learning model. This happens when open-source data are used for training without verifying the origin of the data and its integrity. To prevent this, it is imperative to ensure the dataset is from a trusted source and ensure its integrity before training.

From the viewpoint of attackers, targeting a specific VM will minimise the scope of the attacks, avoid time consuming vulnerability scanning, and enable personalized data poisoning which often have a better success rate. On the other hand, from the perspective of defence, it is extremely challenging to protect a learning model against a personalized attack because typical protection is no longer capable of filtering out the threats. Thus, it is worth discovering the mapping between DPA and VM to better understand and characterise the attack and devise an effective defensive solution.

2.4 DPA Defence

The target of security by design is to predict potential attacks through a what-if analysis toward designing a suitable defence before the attack occurs [26]. Multiple existing DPA defence techniques are attack specific agnostics, such as adversarial training [251], data sanitisation [58] and influence based defence. These solutions can only defend some specific type of DPAs such as TCL-attack [277], pGAN-attack [173], LF-attack [25], R-attack [112]. Thus, existing defence techniques against data poisoning attacks are largely attack-specific, they are designed to tackle one specific type of attacks, but may not work for other types mainly due to the distinct principles they follow. Apparently, it is beneficial to map all defences to their corresponding DPAs, or the other way around. This will provide the defenders a clear view on every attack and suggest what are the appropriate defences that can be implemented in a fast manner.

2.5 The Roadmap Solution

For both VM dependency and in-dependency, it is desirable to discover those DPA groups that share features (DPA measurements) and mathematical computation. If groups are connected to one another, going towards a specific defence solution, then the complete knowledge of DPA will be represented as a roadmap, and the map will equip the defenders with the complete knowledge of DPA characteristics in the shortest possible time in implementing an effective countermeasure solution.

Technically, given a DPA set A and its defence solution set D, the construction of a roadmap to connect DPA to defence is to create a morphism as

$$\mathcal{M} : A \rightarrow D,$$

where the roadmap $\mathcal{M}$ is a subset of $A \times D$ consisting of all the pairs of $(a, \mathcal{M}(a))$ for every $a \in A$. Note that the roadmap does not capture the complete information of which the defence D is used as the codomain; the range $\mathcal{M}(a)$ is determined by the input space A.

It is worth noting that under the condition of known victim model f, the roadmap can be formatted as

$$\mathcal{M}_{f,c} : A \rightarrow D,$$

in which a DPA a_i is able to be tracked on the DPA configuration c with the specific reference to f and to reach the predicted defence solution d_i with $d_i \in D$.

With $\mathcal{M}_f$, in the scenario of known attack, where the victim model is treated with a complicated attack approach and process. The roadmap will guide the defender to track the attack process, at every step to quickly detect a list of possible candidate attacks and predict the ultimate solution. In the case of an unknown DPA, the defender can still track the DPA according to the configuration and quickly identify shortlisted DPAs that are performed randomly by the attacker.

3 DPA CHARACTERISTIC MODEL

3.1 DPA Measurements

DPA measurements are a range of factors that impact the behaviour, architecture, operation and consequences of a data poisoning attack. The following describes the list of measurements for the purpose of DPA characterization.

Data Type: The type of data on which the attack is performed. The option includes text, audio, video, graph, structured and unstructured, and all types of data.

Victim Model [222]: The type of machine learning model that the attack targets. The option includes supervised learning, unsupervised learning, natural language processing, reinforcement learning, and statistic learning.

Target Algorithm: A specific algorithm that has been targeted. The algorithm belongs to one of the above victim models. The option includes SVM, CNN, Linear Regression, Logistic Regression, Decision Tree, Gradient Based GCN, Random Forest, RNN, LSTM, Bi-LSTM, Gradient Boost Decision Tree, Faster RCNN.

Target Architecture: The type of architecture that has been targeted by the attack. The option includes LeNet, VGG, AlexNet, QuocNet, GoogLeNet, CaffeNet, ResNet, DQN, TRPO, A3C, VAE, AE, VGGFace, FCN, BiDAF, 2-Layer FC.

Threat Model [128]: The approach and mathematical model adopted in the attack. The option includes additive threat model, functional non-additive model, Blackbox, Whitebox and Graybox threat model.

Attack Frequency [266]: The number of times to query the model and refine the adversarial samples. The option includes a one-time attack and an iterative attack.

Perturbation Core [257]: The type of small artificial corruptions introduced into clean samples so as to fool the target machine learning algorithm. The option includes FGSM, PGD, DbBA, Threshold Attack, NewtonFool, PGD permutated Gradient descent, PGD - Iterative, PGD - Single Shot, ZOO, Spatial Transformation, BIM, Momentum Iterative, Auto Attack, Shadow Attack, JSMA, SimBA, SimBA-DCT, DPatch, Carlini & Wagner, IGS, Adversarial Patch, IFS, QL Attack, LBFG, QeBB, UAP, TUAP, and CE. Table 6 gives a list of commonly used perturbation mathematical functions.

Perturbation Scope [156]: Individual-scope perturbations are generated for each individual input sample, while universal-scope perturbations are perturbations generated independently from any input sample. The option includes individual and universal scope perturbation.

Perturbation Dimension [189]: The selection of input dimensions on which perturbation is performed in order to generate the target mis-classification with a minimum amount of perturbation. The option includes all input dimensions or a subset of them.

Repetition to Convergence [234, 266]: The number of attack repetitions for crafting the desired adversarial samples. The option includes a one-time attack and an iterative attack.

Adversarial goal Consider four goals that impact classifier output integrity:

- Confidence reduction - reduce the output confidence classification (thereby introducing class ambiguity)

— Misclassification - alter the output classification to any class different from the original class
— Source/Targeted misclassification - produce inputs that force the output classification to be a specific target class.

The option includes targeted, un-targeted class and confidence reduction.

Perturbation Search Methods [266]: The search method used for finding the optimal perturbation (selection) according to the input data type and target model. The option includes bisection search, fast gradient, binary search, minimum and maximum search.

Perturbation visibility [156]: The visibility of the adversarial samples. The option includes optimal perturbation, visible perturbation, physical perturbation, fooling data and noise.

Attack assembly [160]: A number of adversarial methods can be applied together for the purpose of bypassing a defence by creating an attack assembly. The option includes single attack, ensemble attack and composite attack.

Defence Mechanism: A detection and response mechanism against a single or multiple data poisoning attacks. This can be either proactive or reactive mode.

3.2 Characterizing Model

The characterization of a DPA broadly depends on whether the attacker has access to the VM data, i.e., the victim model is known or unknown. In the case of a known VM, the weakness points (attack points) are known, and a DPA is likely to be designed according to the victim model architecture, algorithm, and parameters. In the case of an unknown VM, the attacker needs to find out first attack points, by testing with different types of perturbation, observing the visibility and the response from the VM, then fix the type of perturbation applied to the attack.

To formulate the attack, its behavior is required to be customised according to the attack dimension (selection of input variables) and scope (universal or just individual sample). In launching the attack, the attacker needs to decide on a threat model and the attack frequency to ensure its convergence over multiple trials. Also, the attacker may assemble the formulated attack to increase the attack complexity and effectiveness. Figure 1 summarizes the DPA characterising model, which consists of attack core, and the layer of attack prototyping, formulation, and implementation. Note that the implementation of a real-world DPA often involves all layers working in cohesion and depending on each other.

3.3 DPA Grouping

The purpose of DPA grouping is to discover those DPA families that share the same defence solution or possess a set of similar attributes. In doing this, the first criterion is to find those DPAs that share the same defence solution. For example, data sanitization [247] is a popular defensive mechanism applicable to multiple DPAs including simplistic attack, greedy attack, semi-online-WK and concentrated attack [247]. Following the data sanitizing defence, we are able to discover the DPA family of Watermarking [162], Clean-Label [213], feature collision [213] and Spoofing [125]. Similarly, following randomized smoothing [161], gradient shaping [105], and other defence solutions in Table 8, Table 5 presents the full list of DPA families used in this research.

The second criterion we follow is the similarity in terms of key DPA measurements including data type, perturbation method and Victim model which are defined in Section 2. For example, DPA-M-PGD and DPA-M-FGSM both target image data, and they use a similar mathematical core $\zeta(x) = x + \epsilon . sign(\nabla_x L(x, l_{true}))$. Table 3 gives a list of popular mathematical perturbation functions. Thus, these two DPAs are grouped in one node. Another example, BIM is an iterative version of

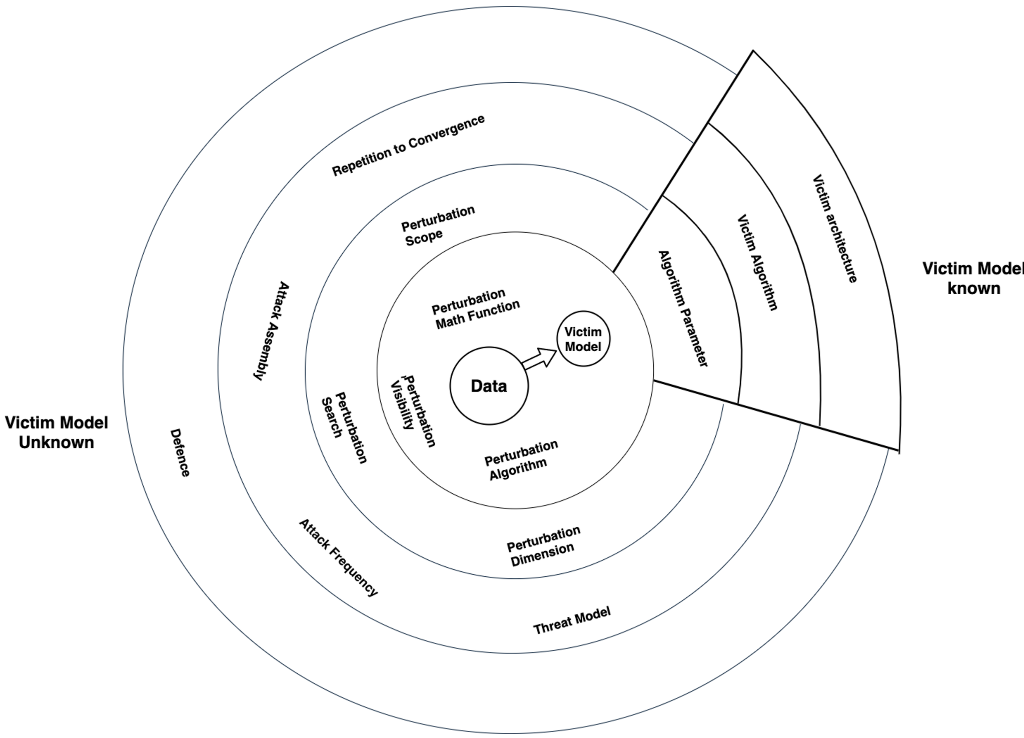

Fig. 1. DPAs characterising model.

FGSM. The two perturbation methods share the core $\zeta(x) = x+\epsilon.sign(\nabla_x L(x, l_{true}))$. Further, both FGSM and BIM apply the same additive threat model. Thus, DPA-M-FGSM and DPA-M-BIM are categorised as one node in terms of DPA measurement perturbation method and threat model. Another criterion in deriving the node is assembly. Two or multiple DPAs can be assembled to build one attack that is more powerful than an individual. In DPAs assembly, two or multiple (single attack) are combined by searching for the best combination of attack algorithms and their hyperparameters leading to a more powerful attack **Composite Adversarial Attacks (CAA)** [160].

4 ROADMAP

As discussed above, DPA has a dependency on data type, perturbation core, and victim model. The majority of DPAs specify the type of target model, and both attack developer and defender have more or less prior knowledge about the model they are using and the model under attack (i.e., VM). Thus, a roadmap on the victim model will assist security practitioners to quickly analyze an unknown attack by observing its behaviors against the victim model characteristics represented in the roadmap and coming up with an effective defence solution.

VM-independent DPAs are attacks applicable to universal models even if the learning method is based on different principles. VM-dependent DPA is more harmful than VM-independent attack. The attackers have prior knowledge of the model under attack, not only its characteristics, but also its parameters with value ranges, converging pathway, and the transferability to models with similar structures. They can easily discover all weakness points (attack points) and exploit them to achieve the level of damage they want.

Table 3. List of Mathematical Core Perturbation Functions

No	Perturbation Core	Formula
1	Gradient Descent [205]	$sign(\nabla_x L(x, y))$
2	Projected Gradient Descent (PGD) [180]	$sign(\nabla_x L(x, y))$ $x_{k+1} = argmin\frac{1}{2}\|\|x - y_{k+1}\|\|_2^2$
3	Batch Gradient Descent [205]	$\theta - \eta.\ \nabla_\theta\ J(\theta)$
4	Projected Gradient Iterative [214]	$\alpha.sign(\nabla_{x^{(i)}} J(x^{(i)}, y)$
5	Projected Gradient Ascent (PGA) [104]	$x_{k+1} = argmax\frac{1}{2}\|\|x - y_{k+1}\|\|_2^2$
6	Discrete Gradient Ascent (DGA) [75]	$\nabla_{x^{t-1}} L(\theta, x^{t-1}, y)$
7	Momentum Iterative (MI) [73]	$\zeta(x_{t+1} = \zeta_t + \alpha.sign(g_t + 1)$
9	Momentum Gradient Ascent (MGA) [185]	$x_{t-1} + \eta J(x_{t-1})$
10	Stochastic Gradient Descent (SGD) [214]	$\theta - \eta.\ \nabla_\theta\ L(\theta, x^{(i)}, y^{(i_0)})$
11	Momentum Stochastic Gradient Descent (MSGD) [43]	$-\epsilon\ \nabla_w\ E(w) + p\Delta w_{t-1}$
12	Enhanced Projected Gradient Descent [67]	$\prod_{[0,255]}(x_i + \delta)$
13	Back Gradient Descent [175]	$\nabla_{w_t} L(\zeta_c, w_t)$
14	Decision Based [31]	$\|\|x - \zeta\|\|_2^2$
15	Score Based [31]	$\zeta^k = \zeta^{k-1} + \eta_k$ $\zeta^k = \zeta^{k-1}$
16	Transfer Based [74]	$W * \nabla_x L(x, y)$
17	Score Transfer Based [108]	$L_i = L_{untargeted}(x, y) or L_{target}(x, t)$ $Z_{t-1} - \frac{\eta}{b}\sum_i^b = 1L_i\ \nabla_{z_{t-1}}\ logN(V_i \| Z_{t-1}, \alpha^2)$
18	Low - Dimension Embedding (NES) [108]	$\prod_{[-\epsilon, epsilon]}(\delta_t - \eta.sign(\frac{1, b}{\sum}{}_{k=1}^{b} L(x + w_k, y) \nabla\ logN(w_k \| \delta_t, \alpha^2)))$
19	Universal [270]	$P_{p,\epsilon} = argmin_\zeta \|\|x - \zeta\|\|$ while $\|\|\zeta\|\|_p < \epsilon$
20	Projected Sinkhorn Iterations (Wassertein) [252]	$w - \beta/\lambda$
21	signSGD [145]	$GradEstimate(x) = \frac{1}{bp}\sum_{i \in L_k}\sum_{j=1} q\hat{\nabla} f_i(x; u_{i,j})$
22	ZO-signSGD [146]	$GradEstimate(x) = \frac{1}{bp}\sum_{i \in L_k}\sum_{j=1} q\hat{\nabla} f_i(x; u_{i,j})$, $\hat{\nabla} f_i(x; u_{i,j}) := \frac{d[f_i(x+\mu u_{i,j}) - f_i(x)]}{\mu}\ u_{i,j}$,
23	Image-Scaling [190]	$Scale(S + \Delta) = D + \delta,\ \|\|\delta\|\|_2 > \epsilon_L$
24	Shadow-Penalties [94]	$\max_\delta L(\theta, x + \delta) - \lambda_c C(\delta) - \lambda_{tv} TV(\delta) - \lambda_s Dissim(\delta)$
25	Gaussian Noise [30]	$Z(j, k) = \alpha * P_\alpha(j, k) + N_\alpha(j, k)$

4.1 Developing Map

According to the DPA characteristic model described in Section 3, we rank the priority of the DPA measurements in terms of their relatedness to the victim model as: (1) target algorithm, (2) perturbation core, (3) perturbation visibility, and (4) perturbation search method. Given a collection of DPAs, and the set of DPA measurements, the following steps are taken to create the roadmap:

— **Step 1:** All DPA measurements are ranked according to the characteristic model, where data type, victim model, and perturbation core are the core measurements.
— **Step 2:** An initial DPA grouping is conducted by checking the similarity of three core measurements.
— **Step 3:** For each resulting DPA group, nodes are created by verifying: (1) if the group is in line with an exiting DPA group, then a mid-layer node is created to represent the DPA group in the roadmap; and (2) if the group DPAs share the same defence solution and cannot be further divided, then a terminal node is created with its defence solution identified in the map.
— **Step 4:** For every non-terminal node, a next-layer grouping is conducted according to the ranked remaining measurements, creating another set of DPA groups as the next-layer node candidates.
— **Step 5:** Add an edge to connect every mid-layer node with its next layer node or terminal node.

— **Step 6:** The above steps are carried out in an iterative manner until every DPA goes to one specific terminal node.

In the proposed roadmap, the definitions and notations of mid-layer node, terminal node and edge are given as follows:

— *Terminal node:* If a group of DPAs shares the same defence method and can not be further divided, then this group of DPAs constructs a terminal node in the roadmap. In the roadmap, a terminal node is labelled as "node name/defence solution".
— *Mid-layer node:* A mid-layer node represents a group of DPAs that have confirmed similarity on a list of DPA measurements, which includes the three core measurements. In the roadmap, a mid-layer node is denoted as a circle labelled as DPA family name. The size of the node represents the size of the family in terms of the number of DPAs.
— *Edge:* An edge represents a connection between two mid-layer nodes or from a mid-layer node to its terminal node. In the roadmap, the edge is represented as a directed line/curve from the left to the right.

As a result, Figure 2 presents the DPA roadmap that consists of 221 DPAs and 111 defence solutions reported in the literature during 2010-2022. For the convenience of defence solution search, we have provided in the Appendix the full list of DPAs as Table 7 and Table 6, the full list of DPA defence method as Table 8, and the full list of terminal nodes as Table 5.

In addition, the Github gate is set up to maintain all the supplementary documents including the full list of DPAs, defence solutions, and perturbation functions, and serve as a public platform to not only enable traceability, but also provide the open access for researchers to add in new DPAs for roadmap updates.

As seen from the roadmap, the DPA group (**NES - Natural Evolutionary Strategies**) is the result of the initial DPA grouping, by the similarity of core measurements, data type: images, victim model: Supervised, and Perturbation Core: NES. Consider the DPA group shares the same defence solution of Augmented Training, and can be further divided according to perturbation core, thus we create three terminal nodes NES/Augmented Training, NES-FGSM/Augmented Training, and NES-PGD/Augmented Training.

4.2 Victim Model Tracking

The increasing adoption of machine learning-driven models in production systems demands rigorous attention into defending against DPAs. With the proposed roadmap, a DPA can be tracked hierarchically according to its VM, PC, DT and attack configuration characteristics, and reach a predicted defence solution. For an unknown DPA, the map is also able to make prediction according to the attributes of DPA other than the VM. In this sense, the proposed map has a good coverage of all type of DPAs [4, 22, 29, 70, 86, 93, 121, 139, 141, 197, 198, 227, 239, 243, 258, 265]. For defence, we give special attention to computer vision VMs and Neural Nets in that these techniques have been widely used in industry production systems, and a substantial number of poisoning attacks and defence mechanisms have been developed in this domain. Figure 2 presents an example DPA tracking, from the supervised VM to the terminal node: Clean-Label/data sanitizing.

4.3 Validation Case Study

Clean-Label attack, also known as Poison Frogs [213], is a family of data poison attacks targeting neural nets. This DPA family is known to attack image and video data [158, 213]. In this attack, clean labeled data are injected for training, as opposed to maliciously labeled instances, and hence does not require control over the labels in training data, but causes the retrained model to

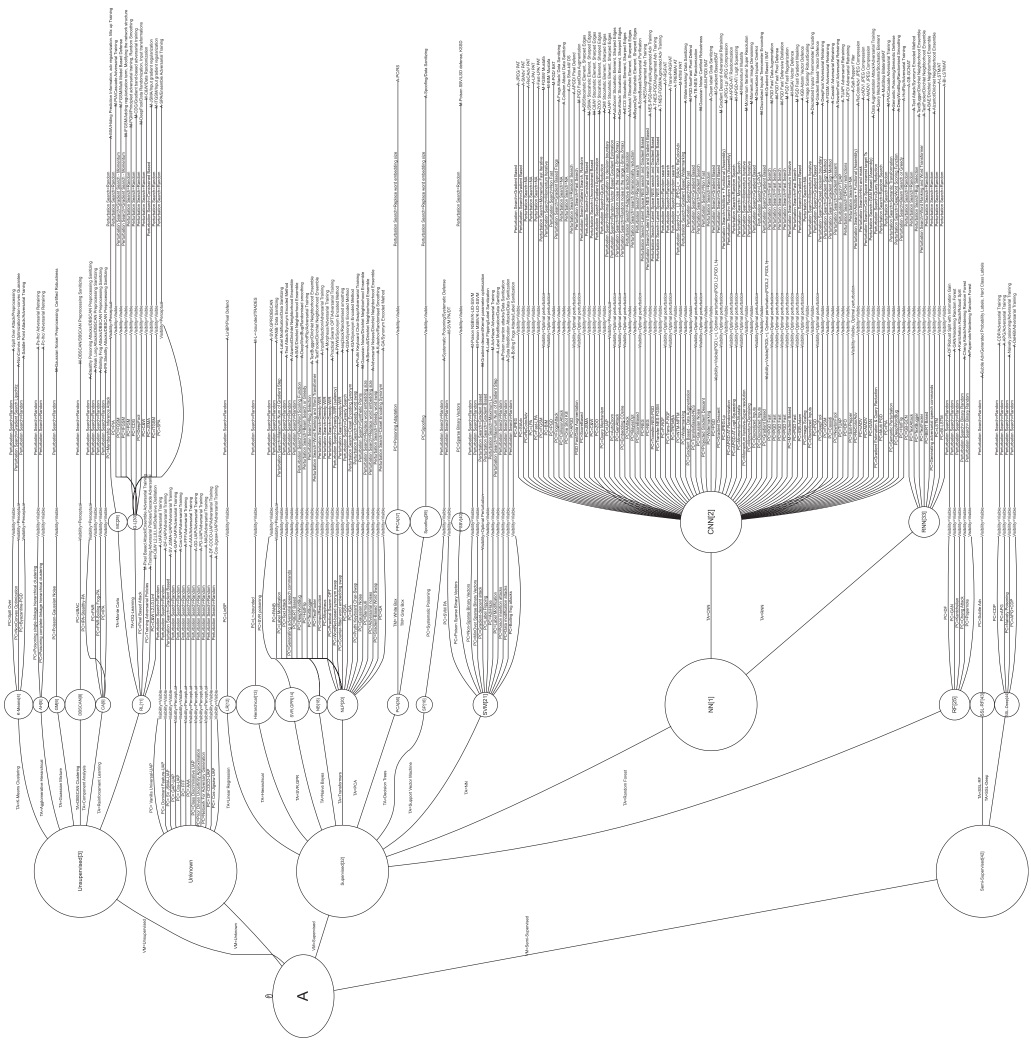

Fig. 2. A DPA roadmap that consists of 221 DPAs and 111 defence solutions reported in literature during 2010–2022.

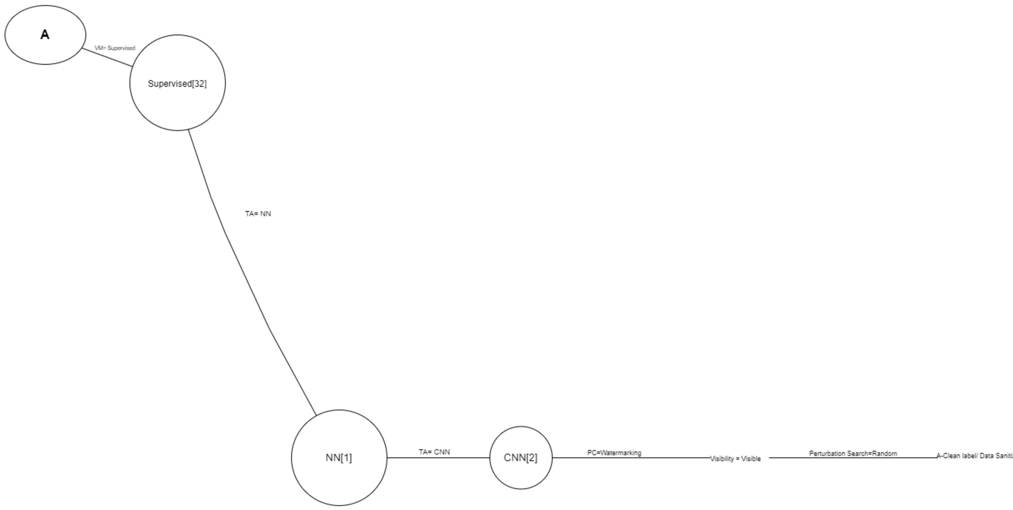

Fig. 3. An example DPA tracking in the proposed roadmap, from the supervised VM to the terminal node: Clean-Label/data sanitizing.

Table 4. The Characteristics of Poison Frogs with Comparison to Traditional DPA

No	Characteristics	Poison Frogs DPA	Traditional DPA
1	Perturbation core	Watermarking	PGD
2	Data type	Image and Audio	Any
3	Victim Model	Supervised	Any
4	Visibility	Visible	Visible/Perceptual
5	Perturbation Search	Random	Gradient-Based
6	Perturbation Scope	Universal	Individual
7	Threat Model	Non Additive-All (W,G,B)	Additive-All (W,G,B)
8	Attack Frequency	One Time	Iterative

mis-classify test data into a specific target class. Clean-Label attacks are considered more complex than poison-label attacks that have both training examples and labels maliciously modified, because they are stealthy and resistant to data filtering or detection, making it difficult to find a mitigation solution. Table 4 describes the characteristics of the Poison Frogs attack with a comparison to traditional DPA. The common defence against the Poison Frog attack is data sanitization. As reported in [58], data sanitizing including anomaly detection, training loss, and singular-value decomposition have all been bypassed by a complex Clean Label attack. To tackle this issue, new constructive defence solutions are currently under investigation [178].

From the defence point of view, we can trace an attack in the proposed roadmap, and predict an effective solution. Taking Poison Frog attack as an example, after locating the right VM, we can trace the target architecture and identify the group of DPAs following the path of (VM=Supervised) → (TA=NN) → (TA=CNN) → (DT=Image) → (PC=Watermarking) → (Visibility=Visible) → (PS=Universal) → (TM=Non Additive-All (W,G,B) → (PS=Random). Figure 3 shows the path how the Poison Frog attack is traced to a terminal node, which indicates that the potential defence solution is data sanitizing and/or high dimensional robust estimation. Since the data sanitization has been bypassed [58] for this attack, then the most effective defence mechanism is the high dimensional robust estimation approach.

5 FUTURE DIRECTIONS

In the efforts to developing a real world navigation roadmap service for bridging DPA to defence, the future works are concluded as follows.

5.1 Capture Parameter Differentiation

The proposed roadmap supports a maximum five-step derivation, which corresponds to five DPA measurements, namely target architecture, perturbation core, visibility, perturbation search, and defence method. However, parameter level differentiation is not yet captured in the current roadmap.

For victim model dependent DPAs, finding an attack point can be formulated as optimization with respect to a performance measure, subject to the condition that an optimal solution of the victim model [26]. Thus, capturing DPA parameter differentiation will empower the roadmap to track the attack points and predict applicable defence solution accurately.

5.2 Response to Emerging Attacks

Despite our best efforts to trace all DPAs and defences between 2004 and 2022, there might be some DPAs that have been missed out. Nevertheless, our roadmap-building process has set up a path for other researchers to follow and expand the research to cover broader DPAs not yet included in our roadmap.

It is a fact that almost every day there are new DPAs designed, developed, and launched. In response to emerging attacks, it is desirable for future work to develop such a framework that we can regulate the conditions on which we can create new nodes, split and/or merge exiting nodes to update the roadmap.

5.3 Roadmap on Perturbation Core

Under the condition of a known attack, the perturbation core is a deterministic factor to the behaviour of a DPA, as the result of adversarial perturbations is often highly aligned with the attack vectors of the victim model [98]. Thus, extending the proposed roadmap to be supportive of perturbation core categorization and navigation which is,

$$\mathcal{M}_{f,\zeta,c} : A \rightarrow D.$$

Developing such a $\{f, \zeta\}$ correlated attack-to-defence mapping will be another significant future work for effective countermeasure and defence.

5.4 Roadmap on Data Type

As discussed above, DPA has a clear dependency on the data type at the application level. To be able to shortlist proper defence solution quickly, it is insightful for us to observe how input data types impact DPAs performance, which is to develop the roadmap of

$$\mathcal{M}_{g(X),c} : A \rightarrow D,$$

where function g determines the data type of X.

6 CONCLUSION

The vast adoption of data-driven machine learning systems has increased the threat of DPA towards compromising these systems which demand a laborious analysis of DPA. To help the academics and practitioners avoid spending time researching how to defend against a specific attack, by first searching the literature studying the DPA and mapping it out into multiple proposed defences, and finally testing the mapped set of defences to evaluate its efficacy (trial and error approach), this paper introduced a DPA characteristic model, and proposed a DPA roadmap to identify the rules to devise a DPA from the view point of attackers.

DPA in practice is built with multiple mutually dependent layers that work in cohesion. In developing the roadmap, it is essential to identify such a framework in which a DPA can be characterized using layers of attacks, prototyping, formulation, and implementation. In response to this, we developed a unified DPA characterization framework with a focus on the victim model, which provides the rules and the baseline for DPA grouping. This allows the defenders to track an unknown DPA according to the attack characteristics discovered so far, navigate through the multi-layer roadmap, and determine the effective solution. The defender normally has a good knowledge of the model in protection (i.e., VM). In this context, the proposed DPA roadmap enables the defender to use this knowledge to quickly shortlist the potential defence solutions.

APPENDIX

Table 5. The List of Terminal Nodes

Name	Property:Data Type	Property:Victim Model	Property:Threat Model	Property:Perturbation Core	Property:Visibility	Property:Perturbation Search	Property:Perturbation Scope	Property:Attack Frequency	DPA	Defence
M-Gradient Descent/Adversarial Retraining	Image-Graph-Audio	Supervised	Additive-White Box	Gradient Descent	Optimal perturbation	Gradient Based	Universal	One Time	M-PGD,A-APGD,M-FGSM,M-BIM,A-DeepFool,A-Fast-LPA,A-R-FGSM,A-N-FGSM,A-CW,A-Cassidi,A-NewtonFool,A-R-N-FGSM,A-Fast,Rapid-FGSM,A-Robust-FGSM	Adversarial Retraining [38], Mask Gradient [27]
M-JPEG-L_p/JPEG Compression	Image	Supervised	Non Additive-BlackBox	JPEG-L_p	Visible	Additive + Functional (Assembly)	Individual	Iterative	A-JPEG-L_∞, A-ReColorAdv, A-cADV, A-tADV	JPEG Compression* [64]
M-APGD-AT/Randomisation	Image	Supervised	Additive-Whitebox	$L1$-APGD And $L1$-AutoAttack (APGD-AT), Single and Multi APGD	Optimal perturbation	Random Search (Assembly)	Individual	Iterative	A-SingHunter, A-DFO-CMA, A-DFO-DIAG CMA, Bandits, A-Parsimonious, A-SquareAttack, A-SimBA, A-SimBA-DCT, A-NES-PIA, NES-GE, NES	Randomisation [61]
M-Momentum/Mustafa	Image	Supervised	Additive-Whitebox/Graybox	Momentum Gradient Based	Optimal perturbation	Momentum Search	Universal	Iterative	M-FGSM, M-BIM, M-MI-BIM	Mustafa,Super-Resolution,Image Denoising [174]
A-Discretized Inputs/One Hot	Image	Supervised	Additive-White/Black Box	Discrete Gradient Ascent PGD / PGA	Optimal perturbation	Gradient Based	Individual	One Time/ Iterative	A-PGD/LS-PGA, A-Vanilla, A-Vanilla-PGD, A-APGD, M-Auto-PGD, M-LL-PGD, M-PGD Iterative, M-PGD-SingleShot	One Hot [32]
A-Clean Label/Data Sanitizing, High dimensional Robust estimation	Image - Audio	Supervised	Non Additive-All(W,G,B)	Watermarking	Visible/Noise	Random	Universal	One Time	A-ForgsAttack, A-CollisionAttack, A-WatermarkingAttack, oneshot-kill	Data Sanitizing [58] *, High dimensional robust estimation [69]
M-Gradient Based/Vector Defence	Image	Supervised	Additive-All(W,G,B)	M-PGD	Optimal Perturbation	Gradient Based	Universal	Iterative/One Time	M-BIM, M-JSMA, A-DeepFool, M-CW, M-PGD,-M-Auto-PGD, M-LL-PGD, M-PGD Iterative, M-PGD-SingleShot	Vector Defence [118]
M-PGD&NonFast/PixelDefend	Image	Supervised	Additive-All(W,G,B)	Gradient Based-Data Augmentation	Visible	Non Fast	Individual	One Time	Projected Gradient Descent, M-PGD,A-APGD	PixelDefend,Defensive Distillation, Regularization [31]
M-PGD&Fast/Gradient Masking	Image	Supervised	Additive-All(W,G,B)	PGD Fast	Optimal Perturbation	Fast Search	Individual	One Time	M-FGSM,A-Fast-LPA,A-R-FGSM,A-N-FGSM,A-R-FGSM,A-N-FGSM,A-Fast-FGSM,A-Rapid-FGSM,A-Robust-FGSM,M-CW,M-ElasticNet, M-Zoo	Gradient Masking [27], Vector Defence [118]
A-IGB-Adam/VectorDefence	Image	Supervised	Additive-All(W,G,B)	Iterative Gradient Based and Adam Optimization	Optimal Perturbation	Gradient Based	Individual	Iterative	M-FGSM,M-IFGSM,M-PGD,A-JSMA, M-DeepFool, M-CW	VectorDefence [118]
A-ScoreBased/Stochastic Elements,Sharped Edges	Image	Supervised	Additive-All(W,G,B)	Score Based	Visible	Salient Search, Random	Universal	One Time / Iterative	M,F,NT-JSMA,JSMA+F,JSMA-Z,NT-JSMA-F,NT-JSMA+Z,M-JSMA-F,A-GenAtt(UAP),M-ZOO,A-QM,A-AutoZoom,A-ECO,A-AAttack,A-BayesOpt	Stochastic Elements [31]
A-Image Scaling/RobustScaling	Image	Supervised	Functional Non Additive-Blackbox	Image-Scaling, Upscaling, Downscaling	Visible	Random	Individual	One Time	A-ImageScaling-Non-Adaptive, A-ImageScaling-Adaptive(Pillow), A-ImageScaling-Adaptive(Area Scaling) A-HopSkipJump, A-Camouflage-Attack	RobustScaling [190]
A-APGD-AT/Logit Squeezing	Image	Supervised	Additive-Whitebox/Blackbox	$L1$-APGD And $L1$-AutoAttack(APGD-AT)	Optimal Perturbation	Random Search	Universal	One Time	A-SingHunter, A-DFO-CMA, A-DFO-DIAG CMA, Bandits, A-Parsimonious, A-SquareAttack, A-SimBA, A-SimBA-DCT, A-NES-PIA, NES-GE, NES	Logit Squeezing* [212]
M-Momentum Iterative/Super Resolution	Image	Supervised	Additive-Whitebox/Graybox	Momentum Gradient Based	Visibile	Momentum Iterative	Universal	Iterative	M-FGSM, MI-FGSM, MDI-FGSM	Super resolution [174]
M-Momentum/Image Denoising	Image	Supervised	Additive-Whitebox/Graybox	Momentum Gradient Based	Visible	Momentum Iterative	Universal	Iterative	M-FGSM, MI-FGSM, MDI-FGSM	Image Denoising [80]
A-Discretized Inputs/Thermometer Encoding	Image	Supervised	Additive-Whitebox	Discrete Gradient Ascent	Visible	Random Gradient Based	Individual	One Time	DGA Attacks, Logit Space-PGA	Thermometer Encoding [32]
M-Gradient Based/BAT	Image	Supervised	Additive-All(W,G,B)	PGD	Visible(PGD$L\infty$), Optimal perturbation(PGDL_2,PGDL_1)	Gradient Based	Universal	One Time	PGDL_2,PGDL_1,PGD$L\infty$	BAT [253]
M-PGD Fast/Data Augmentation	Image	Supervised	Additive -All(W,G,B)	PGD	Optimal perturbation	Fast Search	Universal	One Time	PGDL_2,PGDL_1,PGD$L\infty$	Data Augmentation [221]
A-Data Augmentation Attack/Adversarial Training	Image	Supervised	Non Additive-Whitebox	GAN	Visible	GAN Based (Assembly)	Individual	One Time/Iterative	Flipping, Rotating, Cropping, Color Jittering, Edge Enhancement, Fancy PCA, Mixing Images, Random Erasing, Style Reconstruction	Adversarial Training* [38], Data Sanitizing [58]
M-PGD Fast/Pixel Defend	Image	Supervised	Additive-All(W,G,B)	Projected Gradient Descent	Optimal perturbation	Fast Search	Universal	One Time	PGDL_2,PGDL_1,PGD$L\infty$,Fast-FGSM,A-Rapid-FGSM	Pixel Defend [224]
M-PGD Fast/Defensive Distillation, Regularization	Image	Supervised	Additive -All(W,G,B)	Projected Gradient Descent	Optimal perturbation	Fast Search	Universal	One Time	PGDL_2,PGDL_1,PGD$L\infty$,Fast-FGSM,A-Rapid-FGSM	Defensive Distillation [34]
M-PGD Fast/Regularization	Image	Supervised	Additive -All(W,G,B)	Projected Gradient Descent	Optimal perturbation	Fast Search	Universal	One Time	PGDL_2,PGDL_1,PGD$L\infty$,Fast-FGSM,A-Rapid-FGSM,Fast-LPA	Regularization [113]

(Continued)

Table 5. Continued

Name	Property:Data Type	Property:Victim Model	Property:Threat Model	Property:Perturbation Core	Property:Visibility	Property:Perturbation Search	Property:Perturbation Scope	Property:Attack Frequency	DPA	Defence
A-Salt PepperNoise/Filter	Image	Supervised	Non Additive-All(W,G,B)	Noise	Visible and Optimal perturbation	Random	Individual	One Time	Noise attacks, Salt PepperNoise	Filter (Gaussian, Average, Median) [263]
M-NTM/PAT	Image	Supervised	Non Additive-Blackbox	NTM-NPTM	Visibile	L_∞, L_2, JPEG-L_∞, *StAdv*, *ReColorAdv*	Universal	One Time	A-JPEG, A-StAdv, A-ReColorAdv,A-LPA, A-Fast-LPA, A-PPGD	PAT [129]
A-JPEG/PAT	Image	Supervised	Non Additive-Blackbox	A-JPEG	Visibile	Gradient Based	Individual	Iterative	JPEG-L_∞,JPEG-L_2	PAT [129]
A-StAdv/PAT	Image	Supervised	Additive + Functional (Assembly)	StAdv	Visible	Gradient Based	Individual	One Time	StAdvL_∞,StAdvL_2	PGD-AdvT [157], Ensemble-AdvT [236]
A-ReCAdv/PAT	Image	Supervised	Additive + Functional (Assembly)	ReColorAdv	Visible	Random	Individual	One Time	ReColorAdv+PGDL_∞,ReColorAdv-PGDL_2	PGD-AdvT [157], Ensemble-AdvT [236]
A-LPA/PAT	Image	Supervised	Non Additive	LPA-Lagrangian	Visible	Random	Universal	One Time	LPA, Fast-LPA,*LPIPS*-LPA	PAT [129]
A-Fast-LPA/PAT	Image	Supervised	Non Additive	LPA-Lagrangian	Visible	Random	Universal	One Time	Fast-LPA,*LPIPS*-LPA, LPA	PAT [129]
M-MGA/Vector Defence	Image	Supervised	Additive	Momentum Gradient Based	Optimal perturbation	Momentum Iterative	Universal	Iterative	MGA Unlimited, MGA Direct, MGA Indirect	Vector Defence [118]
M-FGSM/Mustafa	Image	Supervised	Additive	Momentum Fast Gradient Based	Optimal perturbation	Momentum Fast Iterative	Universal	Iterative	M-FGSM Unlimited, M-FGSM Direct, M-FGSM Indirect, MI-FGSM Ensemble, MI-FGSM, TI-MI-FGSM	Mustafa [174]
M-BIM/Mustafa	Image	Supervised	Additive	Momentum BIM	Optimal perturbation	Momentum Iterative	Unoversal	Iterative	M-BIM,M-FGSM Unlimited, M-FGSM Direct, M-FGSM Indirect, MI-FGSM Ensemble	Mustafa [174]
M-PGD/BAT	Image	Supervised	Additive-ALL(W,G,B)	Projected Gradient Descent	Visible	Gradient Based	Universal	One Time	PGD-L_∞, PGD-L_2, PGD-L_1, PGD-L_0	BAT [253]
M-M-PGD/BAT	Image	Supervised	Additive-ALL(W,G,B)	Projected Gradient Descent	Visible	Gradient Based	Universal	One Time	M-PGD-L_∞, M-PGD-L_2, M-PGD-L_1, M-PGD-L_0	BAT [253]
A-Watermarking/Data Sanitizing	Image	Supervised	Additive + Functional (Assembly)	Watermarking	Visible	Gradient Based Watermarking	Universal	One Time	Grad-WML_∞, Grad-WML_2, Grad-WM-L_1, Grad-WM-L_0	Data Sanitizing [58]
A-Clean-Lables Attacks/Data Sanitizing	Image	Supervised	Non Additive-Blakbox	Clean Label	Visibile	Random	Universal	One Time	CLean-LabelsL_∞, Grad-FrogsL_∞, Grad-FrogsL_2, Grad-Frogs-L_1, Grad-Frogs-L_0	Data Sanitizing* [58]
A-Collision Attack/Data Sanitizing	Image	Supervised	Non Additive(W,G,B)	Hash	Visibile	Random	Univeral	One Time	256bit, aHash-256, dHash-256,pHash-256	Data Sanitizing [58]
A-Oneshot-Kill/Data Sanitizing	Image	Supervised	Non Additive-All(W,G,B)	One-shot-Kill Fish attack	Visibile	Random	Universal	One Time	A-fishAttackL_∞, A-fishAttackL_2, A-fishAttackL_1, A-fishAttackL_0	Data Sanitizing [58]
A-NES/Augmented Adv Training	Image, Video	Supervised	Additive-All(W,G,B) + Functional (Assembly)	NES	Visibile	Latent Space NES search and Gradient Based	Universal	One Time / Iterative	NES, Trans-NES-PGD, Trans-NES-FGSM, AutoZOOM, P-RGF, Trans-P-RGF, TREMBA	Augmented Adv Training [31]
A-NES-PGD-NonFast/Augmented Adv Training	Image, Video	Supervised	Additive-All(W,G,B) + Functional (Assembly)	NES-PGD	Visibile	Latent Space NES search and Gradient Based	Universal	One Time / Iterative	NES-NonFast, NES-NF-PGD-L_∞, NES-NF-PGD-L_2, NES-NF-PGD-L_1, NES-NF-PGD-L_0	Augmented Adv Training [31]
A-TransferBased-NES/Randomization	Image	Supervised	Non Additive-All(W,G,B)	PRGF	Optimal perturbation	Gradient Based	Universal	One Time / Iterative	NES, SPSA, RGF, P-RGF, RGF_D, P-RGF_D	JPEG Compression [16], Guided Denoiser [52]
A-TransferBased-NES-PGD/Augmented Adv Training	Image	Supervised	Additive-All(W,G,B) + Functional (Assembly)	NES + PGD	Visibile	Latent Space NES search and Gradient Based	Universal	One Time / Iterative	NES-NonFast, NES-NF-PGD-L_∞, NES-NF-PGD-L_2, NES-NF-PGD-L_1, NES-NF-PGD-L_0, Trans-NES-PGD,AutoZOOM, P-RGF, Trans-P-RGF, TREMBA	Augmented Adv Training [31]
A-TransferBased-NES-FastPGD/Augmented Adv Training	Image	Supervised	Additive-All(W,G,B) + Functional (Assembly)	NES + FGSM	Visibile	Latent Space NES search and Gradient Based	Universal	One Time / Iterative	NES-NonFast, NES-NF-PGD-L_∞, NES-NF-PGD-L_2, NES-NF-PGD-L_1, NES-NF-PGD-L_0, Trans-NES-PGD,AutoZOOM, P-RGF, Trans-P-RGF, TREMBA, SPSA	Augmented Adv Training [31]
M-JSMA/Stochastic Element, Shardped Edges	Image	Supervised	Additive	JSMA	Visible	Constrained Based	Individual	One Time	JSMA, Maximal JSMA, JSMA+F, JSMA-F, NT-JSMA+F, NT-JSMA-F, M-JSMA-F, JSMA-Z, NT-JSMA+Z	Stochastic Element [68], Shardped Edges 15 [68]
M-C&W/Stochastic Element, Shardped Edges	Image	Supervised	Additive	C&W	Visible	Unconstrained Based / Binary Search	Individual	One Time	C&WL_∞, C&WL_2, C&WL_1, C&WL_0	Stochastic Element [68]

(Continued)

Table 5. Continued

Name	Property:Data Type	Property:Victim Model	Property:Threat Model	Property:Perturbation Core	Property:Visibility	Property:Perturbation Search	Property:Perturbation Scope	Property:Attack Frequency	DPA	Defence
M-Zoo/Stochastic Element, Shardped Edges	Image	Supervised	Additive	ZOO	Visible	Gradient Approximation	Individual	One Time	ZOO-Adam, ZOO-Newton	Stochastic Element [68], Shardped Edges [68]
A-Query Mechanisms/Stochastic Element, Shardped Edges	Image	Supervised	Additive	Gradient estimation with query reduction	Visible	Query Reduction	Individual	Iterative	FGS-Single Step, IFGS Iterative, FD-GE Single Step, IFD-GE Iterative, PCA-GE Single Step, PCA-Query Reduction Iterative, SPSA	Stochastic Element [68], Shardped Edges [68]
A-UAP/Stochastic Element, Shardped Edges	Image	Supervised	Additive	Universal Perturbation Vector	Perceptual	Minimal perturbation to decision boundary, Random Gradient Descent	Universal	One Time	UAP, DFUAP (Data Free), DFUAPL_∞, uFGSM, uSGD	Stochastic Element [68], Shardped Edges [68]
A-AutoZoom/Stochastic Element, Shardped Edges	Image	Supervised	Additive-Blackbox	Zeroth Order Optimization	Optimal perturbation	Random Vector Based Gradient Estimation	Universal	One Time	Zoo, Zoo-AE, AutoZoom-BiLIN, AutoZoom-AE	Stochastic Element [68], Shardped Edges [68]
A-GenAttack/Stochastic Element, Shardped Edges	Image	Supervised	Non Additive-Blackbox	Gradient Free Optimization	Optimal perturbation	Random noise in the range $(-\delta_m ax,\delta_m ax)$	Universal	One Time	GenAttackL_∞, GenAttackL_2, GenAttackL_1, GenAttackL_0	Stochastic Element [68], Shardped Edges [68]
A-Embedding COsine /Stochastic Element, Shardped Edges	Image	Supervised	Non Additive-Blackbox	Gradient Free Optimization	Optimal perturbation	Random noise in the range $(-\delta_m ax,\delta_m ax)$	Universal	One Time	ECO-FGSM, ECO-FGSML_∞, ECO-FGSML_2, ECO-FGSML_1, ECO-FGSML_0	Stochastic Element [68], Shardped Edges [68]
A-ScoreBased/AT, Adversarial Purification	Image	Supervised	Additive-Blackbox	Gradient Free Optimization	Optimal perturbation	Random search	Universal	One Time	BPDA+EOT, Joint(score)+EOT, Joint(full)+EOT, SPSA	Adversarial Training [38], Adversarial Purification [262]
A-P-RGF/AT	Image	Supervised	Additive-ALL(W,G,B)	Prior guided random gradient free	Optimal perturbation	Random search	Universal	One Time	RGF, P-RGF, RGF_D, $P-RGF_D$	Adversarial Training [38]
A-Trans-P-RGF/AT	Image	Supervised	Additive-Blackbox	Embedding Space	Optimal perturbation	Random search	Universal	One Time	$Trans-NES_{PGD}$, $Trans-NES_{FGSM}$, Trans-P-RGF, TREMBA	Adversarial Training [38]
A-TREMBA/AT	Image	Supervised	Additive-Whitebox,Blackbox	Embedding Space	Visible	Random search	Universal	One Time	TREMBA, $Trans-NES_{PGD}$, $Trans-NES_{FGSM}$, Trans-P-RGF	Adversarial Training [38]
A-AutoAttack/Stochastic Element, Shardped Edges	Image	Supervised	Additive-Whitebox,Blackbox	Adaptive Auto Attack A^3	Visible	Adaptive direction initialization	Universal	One Time / Iterative	AA,AAA,ADI-PGD, ADI, OSD, R-ADI, ADI+OSD	Stochastic Element [68], Shardped Edges [68]
A-BayesOpt/Stochastic Element, Shardped Edges	Image	Supervised	Additive-Blackbox	Bayesian Optimisation	Visible	Iteration on both perturbation(search dimensionality reduction).	Universal	Iterative	BayesOPT attack, GP-Based BayesOpt, Additive GP-Based BayesOpt, BayesOpt with d^r selection, GP-BO-auto-d^r, ADDGP-BO,	Stochastic Element [68], Shardped Edges [68]
A-Discretized Inputs/Termometer	Image	Supervised	Non Additive-Whitebox,Blackbox	Discrete Gradient Ascent	Visible	LS-PGA	Universal	One Time	DGA, LS-PGA, PGD/LS-PGA	Termometer Enconding [32]
A-ReColorAdv/JPEG Compression	Image	Supervised	Non Additive-BlackBox	Lagragian / PGD	Visible	Additive + Functional (Assembly)	Individual	One Time	C-RGB, C, D, S, C+S, C+D, S+D, C+S+D, C(BW)	JPEG Compression [64]
A-cAdv/JPEG Compression	Image	Supervised	Non Additive-Whitebox	Smooth Color Perturbation	Visible	Hint and mask	Individual	One Time	cAdvL_0, CAdvL_2, cAdvL_∞	JPEG Compression [64]
A-tAdv/JPEG Compression	Image	Supervised	Non Additive-Whitebox	Smooth Color Perturbation	Visible	Color Space nearest target T_s	Universal	One Time	tAdvL_0, tAdvL_2, tAdvL_∞	JPEG Compression [64]
A-DeepFool/AT	Image	Supervised	Additive-Whitebox	DeepFool	Visible	Closet decision boundary	Universal	One Time	DeepFoolL_∞, DeepFoolL_2	Adversarial Training [38]
M-FGSM/AT	Image	Supervised	Additive-All(W,G,B)	FGSM	Visible	Gradient Sign Method	Universal	One Time	FGSM, NFGSM, R+FGSM, RFGSM, FastFGSM, Rapid-FGSM, Robust-FGSM	Adversarial Training [38]
A-Cassidi/AT	Image	Supervised	Additive-All(W,G,B)	Cassidi	Visible	Gradient Sign Method	Individual	One Time	Cassidi	Adversarial Training [38]
A-NewtonFool/AT	Image	Supervised	Additive-All(W,G,B)	NewtonFool	Visible	Gradient Descent	Universal	One Time	NewtonFool attack	Adversarial Training [38]
A-TUAP/AT	Image	Supervised	Additive-All(W,G,B)	Universal perturbation generation	Visible	TUAP	Universal	One Time	TUAP, TUAP-Deepfool, TUAP-CW	Adversarial Training [38]
M-AdvHaze/AT	Image	Supervised	Additive-Whitebox	Synthesize haze	Visible	Maximum $L\infty$	Universal	One Time	HAdvHaze, LAdvHaze	Adversarial Training [38]
A-DFO/AT	Graph	Supervised	Additive-Blackbox	Synthesize haze	Visible	DFOer restrictions	Universal	One Time	DFO, DFDA	Adversarial Training [38]
A-MultiStep Bilateral/BAT	Image	Supervised	Additive-All(W,G,B)	PGD	Visible	Random Search	Universal	One Time	MultiStep Bilateral	Adversarial Training [38], BAT [253]
A-APGD-AT/Pixel Defend	Image	Supervised	Additive-Whitebox/Blackbox	$L1$-APGD And $L1$-AutoAttack(APGD-AT)	Optimal Perturbation	Random Search	Individual	One Time	A-SingHunter, A-DFO-CMA, A-DFO-DIAG CMA, Bandits, A-Parsimonious, A-SquareAttack, A-SimBA, A-SimBA-DCT, A-NES-PIA, NES-GE, NES	Pixel Defend [212]
M-Gaussian Noise/Certified defence	Image	Supervised	Additive-Whitebox/Blackbox	Gaussian noise	Visible Noise	Gradient Search	Universal	One Time	Gaussian noise and WITCHcraft, PPGD, LPA Preprocessing	Certified Robustness [132]
M-PIA/Cascade Adversarial Training	Image,	Supervised	Additive	PIA	Perceptual	Random	Universal	Iterative	PIA, Fredrikson et al. 2014., Shokri et al. 2017, Long et al. 2018, Rahman et al. 2018, hayes et al. 2019, Hilprecht et al. 2019. Jayaraman & Evans, Nasr et al. 2019, Melis et al. 2019, Sablayrolles et al. 2019, Salem et al. 2019, Song L. et al. 2019, Truex, et al. 2019, Chen et al. 2020, Hishamoto et al. 2020, Song & Raghunathan 2020	Cascade Adversarial Training [176]

(Continued)

Table 5. Continued

Name	Property:Data Type	Property:Victim Model	Property:Threat Model	Property:Perturbation Core	Property:Visibility	Property:Perturbation Search	Property:Perturbation Scope	Property:Attack Frequency	DPA	Defence
A-LinBP/Random And Pixel Defend	Image	Supervised	Additive	LinBP	Visible	Random	Universal	Iterative	LinBP+RR, LinBP + ElasticNet, LinBP+SVR, LinBP+I+FGSM, LinBP+I+FGSM+ILA, LinBP+I+FGSM+ILA+SGM	Random And Pixel Defend [212]
A-Semantic Poisoning/Semantic defence	Image	Supervised	Additive	Semantic Perturbation	Visible	Semantic Transformation	Individual	Iterative Multi Steps	Single Attribute Attack, Multi attribute Attack, Cascaded Multi Attribute Attack, Multi Attribute AttGan Attack	Semantic defence [117]
A-Label Modification/Data Sanitizing	Image	Supervised	Additive	Label Flips	Visible	Maximum Num of Gradient Step	Individual	Iterative	alfa, alfa-cr, alfa-tilt, correlated-clusters	Data Sanitizing [58]
A-Text Attack/Synonym Encoded Method	Text	Supervised	Additive	TextAttack	Visible	Random	Universal	Iterative	BAE, BAE-R, BAE-I,BAE-R/I, BAE-R+I, DeepWordBug,FasterGenetic,Genetic,HotFlip,IGA-Pruthi,PSO,TextBugger,TextFooler,VIPER	Synonym Encoded Method [244]
A-Alzantot/Dirichlet Neighberhood Ensemble	Text, Audio	Supervised	Additive	Generating adversarial speech commands	Visible	Genetic Algorithm	Individual	Iterative	GASC,GNLAE	Synonym Encoded Method [244]
A-BAE/Dirichlet Neighborhood Ensemble	Text	Supervised	Additive	BERT Based	Visible	Greedy-WIR	Universal	Iterative - Predict Top-K tokens	BAE-R,BAE-I, BAE-R/I, BAE-R+I	Dirichlet Neighborhood Ensemble [276]
A-DeepWordBug/Randomised Smoothing	Text	Supervised	Additive - Whitebox, Blackbox	DeepWordBug	Visible	Greedy-WIR	Individual	One Time	WordBug Replace-1, WordBug - Temporal Head, WordBug - Temporal Tail, WordBug - Combined	Randomised Smoothing [268]
A-HotFlip/Adversarial Training	Text	Supervised	Additive	HotFlip	Visible	Beam Search or Greedy	Individual	One Time	HotFlip, Adv-tr whitebox, Adv-tr Blackbox	Adversarial Training [38]
A-InputReduction/Adversarial Training	Text	Supervised	Additive	Word deletion	Visible	Greedy-WIR	Individual	Iterative	Input Reduction	Adversarial Training [38]
A-Morpheus/Adversarial Training	Text	Supervised	Additive	Counter-fitted word embedding swap	Visible	Greedy word swap	Individual	One Time	Morpheus, RandomInflect, SpanBert-SQuAD, BLEU	Adversarial Training [38]
A-Practical Swarm OPT/Adversarial Training	Text	Supervised	Additive	HowNet Word Swap	Visible	Particle Swarm Optimization	Universal	One Time	PSO, BiLSTM Embedding+SPO, BiLSTM Sysnonym+SPO, BiLSTM Semem+SPO, BERT Embedding+SPO, BERT Synonym+SPO, BERT Sememe+SPO	Adversarial Training [38]
A-PWWS/Synonym Encoded Method	Text	Supervised	Additive	WordNet-based synonym swap	Visible	Greedy-WIR (saliency)	Universal	Iterative	PWWS Algorithm Random, PWWS Algorithm Gradient, Traversin in word order (TiWO), Word Salience (WS)	Synonym Encoded Method [244]
A-Seq2Sick/Randomised Smoothing	Text	Supervised	Additive	Counter-fitted word embedding swap	Visible	Greedy-WIR	Universal	Iterative	seq2sick	Randomised Smoothing [268]
A-TextBugger/Dirichlet Neighborhood Ensemble	Text	Supervised	Additive - Whitebox,Blackbox	TextBugger	Visible	Bug Selection	Individual	Iterative	TextBugger under black-box, TextBugger under black-box,TextFooler	Dirichlet Neighborhood Ensemble [276]
A-TextFooler/Dirichlet Neighborhood Ensemble	Text	Supervised	Additive	Adversarial by TextFooler	Visible	Word Ranking and Word Transformer	Individual	One Time	TextFooler,PSO	Dirichlet Neighborhood Ensemble [276]
A-Kuleshov/Randomised Smoothing	Text	Supervised	Additive	Gradient-Based Word Swap	Visible	Greedy word swap	Universal	One Time	Greedy Optimization Strategy	Randomised Smoothing [276]
A-Pruthi Keyabord Char-Swap/Adversarial Training	Text	Supervised	Additive	Character Deletion, Character Insertion,Keyboard-Based Character Swap	Visible	Greedy search	Universal	One Time	BiLSTM+ATD, BiLSTM+Pass-through, BiLSTM+Background, BiLSTM+Neutral, BERTY+DA, BERT+Adv, BERT+ATD, BERT+Pass-through, BERT+Background, BERT+Neutral	Adversarial Training [38]
A-GSA/Synonym Encoded Method	Text	Supervised	Additive	Greedy Search Attack	Visible	Greedy Search	Universal	Iterative	GSA, PWWS, IGA, FGPM	Synonym Encoded Method [244]
A-IGA/Synonym Encoded Method	Text	Supervised	Additive	IGA	Visible	Closet Encoding Sysnonym	Individual	Iterative	GA, IGA, GSA, PWWS	Synonym Encoded Method [244]
A-GA/Synonym Encoded Method	Text	Supervised	Additive	GA	Visible	Closet Encoding Sysnonym	Universal	One Time	GA, IGA, GSA, PWWS	Synonym Encoded Method [244]
M-Gaussian Noise/Dirichlet Neighborhood Ensemble	Text, Image	Supervised	Additive	Gaussian Noise	Visible	Random	Universal	Iterative	Synthetic Points	Dirichlet Neighborhood Ensemble [276]
A-Bernoulli/Dirichlet Neighborhood Ensemble	Text	Supervised	Additive	Bernoulli Noise/Word Embedding Perturbation	Visible	Replace word embedding size	Individual	One Time	Bernoilli Noise, Gaussian Noise, Bernoulli Word Noise, Bernoulli Semantic Noise, Gaussian Adv Noise, Bernoulli Adv Noise	Dirichlet Neighborhood Ensemble [276]
A-Adversarial Noises/Dirichlet Neighborhood Ensemble	Text	Supervised	Additive	Adv Noise	Visible	Random	Universal	One Time	Bernoilli Noise, Gaussian Noise, Bernoulli Word Noise, Bernoulli Semantic Noise, Gaussian Adv Noise, Bernoulli Adv Noise	Dirichlet Neighborhood Ensemble [276]
A-PC/RS	Text	Supervised	Additive	PC Second Order Gradients	Visible	Update with Second Order Gradients	Individual	Iterative	Discrete Token Replacement, No-Overlap Poisoning	Randomised Smoothing [268]
A-Spoofing/Data Sanitizing	Audio	Supervised	Additive	PGD	Visible	Gradient Based	Universal	One Time and Iterative	PGD, FGSM	Data Sanitizing [58]
M-SVM PA/K-LID-SVM	Image	Supervised	Additive	Poisoning Attacks	Visible	Random	Universal	One Time and Iterative	Poisoning Attacks For Binary SVM, Restrained Attacks, Coordinate Greedy	K-LID-SVM [249]

(Continued)

Table 5. Continued

Name	Property:Data Type	Property:Victim Model	Property:Threat Model	Property:Perturbation Core	Property:Visibility	Property:Perturbation Search	Property:Perturbation Scope	Property:Attack Frequency	DPA	Defence
M-Poison SBV/LSD defence	Image	Supervised	Additive	Sparse Binary Vectors	Visible	Random	Universal	One Time	Sparse Binary Vectors	LSD defence [81]
M-Poison MSBV/K-LID-SSVM	Image	Supervised	Additive	MinOver Sparse Binary Vectors	Visible	Random	Universal	One Time	Minover Sparse Binary Vectors, Sparse Binary Vectors	K-LID-SSVM [249]
M-Poison NSBV/K-LID-SSVM	Image	Supervised	Additive	Non Sparse Binary Vectors	Visible	Random	Universal	One Time	Non Sparse Binary Vectors	K-LID-SSVM [249]
A-Ps-lhc/Adversarial Retraining	Image	SemiSupervised	Non Additive	Single-Linkage hierarchical clustering Extended (Soft/Hard/Best), Random (Best)	Visible	Random	Individual	Iterative	Ps-lhc Extended (Soft), Ps-lhc Extended (Hard), Ps-lhc Extended (Best), Ps-lhc Extended Random(Best)	Adversarial Retraining [38]
A-Pc-lhc/Adversarial Retraining	Image	SemiSupervised	Non Additive	Complete-Linkage hierarchical clustering Extended (Soft/Hard/Best), Random (Best)	Visible	Random	Individual	Iterative	Pc-lhc Extended (Soft), Pc-lhc Extended (Hard), Pc-lhc Extended (Best), Pc-lhc Extended Random(Best)	Adversarial Retraining [38]
A-Subtle Adv/Hard Class Labels	Image	SemiSupervised	Additive	Subtle L_∞	Visible	Random	Universal	One Shot and Iterative	L-BFGS + L_∞, FGSM + L_∞, BIM-ILCM + L_∞, JSMA + L_0, One-Pixel L_0, CW + L_∞, DeepFool + L_∞, Uni Perturbation + L_∞, UPSET + L_∞, ANGRI + L_∞, Houdini + L_∞, ATNs + L_∞	Hard Class Labels [8]
M-Naively poisoning/Adversarial Training	Image	SemiSupervised	Image	Naively Poisoning	Visible	Random	Individual	Interative	Naively SSL Poisoning	Adversarial Training [38]
A-APG/Adversarial Training	Image	SemiSupervised	Additive-Whitebox/Blackbox	Trajectory Preserving	Visible	Random	Universal	Iterative	APG, DeHiB	Adversarial Training [38]
A-CDP/Adversarial Training	Image	SemiSupervised	NonAdditive-Whitebox/Blackbox	Constructive poisoning	Visible	Random	Universal	Iterative	CDP, DeHiB, AP-CL, EMP-CL-S, EMP-CL-C,	Adversarial Training [38]
A-DeHiB/Adversarial Training	Image	SemiSupervised	Additive-Whitebox/Blackbox	DeHib	Visible	Random	Universal	One Time	DeHiB, DeHiB(APG+CDP),APG, CDP	Adversarial Training [38]
M-MultAV/Adversarial Training	Image, Video	SemiSupervised	Additive	MultAV	Perceptual	Gradient Search	Individual	Iterative	MultAVL_0, MultAVL_2, MultAVL_∞, MultAV-ROA, MultAV-AV-AF, MultAV-SPA	Adversarial Training [38]
A-UAP/Adversarial Training	Image, Audio, Video, Text	Unknown	Non Additive-Whitebox/Blackbox	Universal Perturbation	Visible	Random	Universal	One Time	UAP, SV-UAP, Cos-UAP, DGD-UAP, DF-UAP-COCO, Cos-UAP-Jigsaw	Adversarial Training [38]
A-DF-UAP/Adversarial Training	Image, Audio, Video, Text	Unknown	Non Additive-Whitebox/Blackbox	Universal Perturbation	Visible	Random	Universal	One Time	DF-UAP, UAP, FFF-UAP, AAA-UAP	Adversarial Training [38]
A-SV-UAP/Adversarial Training	Image, Audio, Video, Text	Unknown	Non Additive-Whitebox/Blackbox	Singular Vector JSMA	Visible	Gradient Based	Individual	Iterative	SV-JSMA, SVM-JSMA P_{inf}, SVM-JSMA P_1	Adversarial Training [38]
A-GAP-UAP/Adversarial Training	Image, Audio, Video	Unknown	Non Additive-Whitebox/Blackbox	GAP	Perceptual	Random	Universal	Iterative	GAP,GAPL_0, GAPL_2, GAPL_∞, GAP++	Adversarial Training [38]
A-Cos-UAP/Adversarial Training	Image, Audio, Video, Text	Unknown	Non Additive-Whitebox/Blackbox	Cos-UAP	Visible	Random	Universal	Iterative	Uniform Random Noise, Gaussian Noise, Flat Images, Jigsaw With Fixed	Adversarial Training [38]
M-FFF/Adversarial Training	Image	Unknown	Non Additive-Whitebox/Blackbox	Fast Feature Fool	Perceptual	Random	Universal	One Time	FFF-VGG-F, FFF-CaffeNet, FFF-GoogLeLeNet,FFF-VGG-16, FFF-VGG-19	Adversarial Training [38]
M-AAA/Adversarial Training	Image	Unknown	Non Additive-Whitebox/Blackbox	Ask Acquire Attack	Perceptual	Random	Universal	Iterative	AAA-VGG-F, AAA-CafeeNet, AAA-GoogLeNet, AAA-VGG-16, AAA-VGG-19, AAA-RestNet-152	Adversarial Training [38]
A-GD-UAP/Adversarial Training	Image, Audio, Video, Text	Unknown	Non Additive-Whitebox/Blackbox	Class Discriminative UAP	Visible	Random	Universal	One Time	CD-UAP, Double Targeted Attack(DTA), Class-Wise-UAP, GD-UAP+P	Adversarial Training [38]
A-PD-UAP/Adversarial Training	Image	Unknown	Non Additive-Whitebox/Blackbox	Prior Driven Uncertinity Approximation	Perceptual	Random	Universal	Iterative	PD-UA-VGG-F, PD-UA-CafeeNet, PD-UA-GoogLeNet, PD-UA-VGG-16, PD-UA-VGG-19, PD-UA-RestNet-152, PD-GD-UAP+P, PD-GD-UAP	Adversarial Training [38]
A-NAG/Adversarial Training	Image, Audio, Video	Unknown	Non Additive-Whitebox/Blackbox/ Graybox	Network for adversary generation	Perceptual	Random	Universal	Iterative	NAG-VGG-F, NAG-CaffeNet, NAG-GoogLeNet, NAG-VGG-19, NAG-ResNet-50, NAG-ResNet-152	Adversarial Training [38]
A-DF-UAP-COCO/Adversarial Training	Image, Audio, Video, Text	Unknown	Non Additive-Whitebox/Blackbox	DF-COCO	Visible	Random	Individual	Iterative	DF-COCO-AlexNet, DF-COCO-GoogleNet, DF-COCO-VGG16, DF-COCO-VGG19, DF-COCO-ResNet152, DF-COCO-InceptionV3	Adversarial Training [38]
A-Cos-UAP-Jigsaw/Adversarial Training	Image	Unknown	Non Additive-Whitebox/Blackbox	Cos-UAP	Visible	Random	Universal	Iterative	Uniform Random Noise, Gaussian Noise, Flat Images, Jigsaw With Fixed Frequency, Jigsaw with variable Frequency	Adversarial Training [38]
M-DBSCAN/DBSCAN Preprocessing Sanitizing	Image, Structured Data	UnSupervised	NonAdditive-Whitebox/Blackbox	UBAC (T, M inP ts)	Perceptual	Random	Universal	OneTime	Confidence Attack, Confidence Attack Random, Confidence Attack Cluster size Decreasing, Confidence Attack cluster size Increasing	DBSCAN Preprocessing Sanitizing [62]
A-Stealthy Poisoning Attacks/DBSCAN Preprocessing Sanitizing	Image, Video	UnSupervised	Additive-Blackbox	IPA - Stealthy	Perceptual	Random	Universal	Iterative	IPA, Mislabel attack, Blended Injection Strategy (BIS), Accessory Injection Strategy (AIS)	DBSCAN Preprocessing Sanitizing* [62]
A-Week Long Attacks/DBSCAN Preprocessing Sanitizing	Structured Data	Unsupervised	Non Additive-Whitebox/Blackbox	FNR	Visible	Random	Universal	Iterative	Week-Long Attacks FNR, Week-Long FPR	DBSCAN Preprocessing Sanitizing [62]
A-Boiling Frog Attacks/DBSCAN Preprocessing Sanitizing	Structured Data	Unsupervised	Non Additive-Whitebox/Blackbox	FNR	Visible	Random	Universal	Iterative	Boiling Frog, Globally Informed Poisoning	DBSCAN Preprocessing Sanitizing [62]
M-Gaussian Noise/Preprocessing, Certified Robustness	Image	Unsupervised	Non Additive-Whitebox	Poisson-Gaussian noise	Visible	Random	Universal	Iterative	Gaussian noise, Poisson noise, Denoised GAT, BLS-GSM, BDCT PURE-LET, Mixed Poisson Gaussian Noise, UWT Pure-LET	Preprocessing [62], Certified Robustness [132], Image Denoising [174]

(Continued)

Table 5. Continued

Name	Property:Data Type	Property:Victim Model	Property:Threat Model	Property:Perturbation Core	Property:Visibility	Property:Perturbation Search	Property:Perturbation Scope	Property:Attack Frequency	DPA	Defence
M-Spill Over Attack/Preprocessing	Image	Unsupervised	Additive-Whitebox/Blackbox	SpillOver	Visible	Random	Universal	One Time	Spill Over, Spill Over-*clamp*, Abstract Genetic Spill Over	Preprocessing [62]
A-NonConvex Optimization/Non-convex Guarantee	Image	Unsupervised	Additive-Whitebox/Blackbox	Gradient Descent non-convex/Hessian Free	Perceptual	Gradient Search Lipschitz	Universal	Iterative	Variance Attack, Sign-flipping Attack, Delayed-Gradient Attack, SafeGuard Attack	Non-convex Guarantee [10]
A-Saddle Point Attack/Byzantine-Robust Distribution	Image	Unsupervised	Additive-Whitebox/Blackbox	ByzantinePGD	Visible	Random	Universal	Iterative	ByzantinePGD, ByzantinePGDL_2, ByzantinePGDL_∞	Byzantine-Robust Distribution [260]
A-IPA Stealthy Attacks/DBSCAN Preprocessing Sanitizing	Image	Unsupervised	Additive	IPA	Perceptual	Random	Universal	One Time	IPA, Mislabel attack, Blended Injection Strategy (BIS), Accessory Injection Strategy (AIS)	DBSCAN Preprocessing Sanitizing [62]
A-MIA/Hiding Prediction Information, Adv Regularization	Image	Unsupervised	Additive	Membership Inference Attack	Perceptual	Random	Universal	One Time	DeiT-S-based, LeViT-based	Hiding Prediction Information [55], Adv Regularization [106]
A-SPA/Ensemble Adversarial Training	Image	Unsupervised	Additive	Membership Inference Attack	Perceptual	Random	Universal	One Time	SPA, FGSM(H-Latency), FGSM(L-Latency), PGD(H-Latency)	Ensemble Adversarial Training [236]
M-FGSM/Multi Model Based defence	Image	Unsupervised	Additive	FGSM	Visible	Gradient	Gradient Search - Momentum	Iterative	FGSM, MIFGSM, PGD, MIM	Multi Model Based defence [237]
M-IFGSM/Multi Model Based defence	Image	Unsupervised	Additive	IFGSM	Visible	Gradient	Gradient Search - Momentum	Iterative	IFGSM, MIFGSM	Modifying the network structure [199]
M-PGM/Principled adversarial training	Image, Text, Structured Data	Unsupervised	Non Additive-Whitebox/Blackbox	PGM	Visible	Random	Universal	Iterative	UAP-PGM,	Principled adversarial training [203], Perturbation Subtracting defence [50], Randomised Smoothing [268]
M-CDG/Gradient band-based adversarial training	Image	Unsupervised	Additive	Common dominant adversarial examples generation method (CDG)	Visible	Random	Individual	One Time	CDG	Gradient band-based adversarial training [48]
M-C&W/Data Randomization	Image	Unsupervised	Additive	C&W	Visible	Random	Individual	One Time	C&WL_∞, C&WL_2, C&WL_1, C&WL_0	Data Randomization [15]
M-JSMA/Input gradient regularization	Image	Unsupervised	Additive	JSMA	Visible	Random	Individual	One Time	NT-JSMA,JSMA+F,JSMA-Z,NT-JSMA-F,NT-JSMA+Z,M-JSMA-F	Input gradient regularization [48]
M-ITGSM/Input gradient regularization	Image	Unsupervised	Additive	ITGSM and FTGSM	Visible	Random	Individual	Iterative	ITGSM, FYGSM, FFM, FIM, M-FIM, DI-M-FIM, E-DI-M-FIM	Input gradient regularization [48], JPEG Encoding [219], Gaussian Blur [273], Selective Dropout [5]
M-Pixel Based Attack/Ensemble Adversarial Training	Image	Unsupervised	Additive	Corner Search	Visible	CornerSearch	Universal	One Time	CornerSearchL_∞, CornerSearchL_2, CornerSearchL_0	Ensemble Adversarial Training [236]
A-DF/Robust Split with Information Gain	Image	Unsupervised	Non Additive-Whitebox/Blackbox	Dominant Feature	Visible	Random	Universal	One Time	DF, DF-UAP, DF-UAP(COCO)	Robust Split with Information Gain [44]
A-Gan/Hardening Random Forest	Image	Unsupervised	Additive	GAN	Visible	Random	Universal	Iterative	GAN, UAA-GAN, UAA-GAN-MAC, UAA-GAN-RMAC, UAA-GAN-GeM	Hardening Random Forest [13]
A-Kantchelian Attack/Roust Split	Image	Unsupervised	Additive	Kantchelian	Visible	Random	Individual	Iterative	KantchelianL_∞, KantchelianL_2, KantchelianL_1, KantchelianL_0	Robust Split for decision trees [44]
A-Cheng Attack/Hardening Random Forest	Image	Unsupervised	Additive	Cheng Method	Visible	Binary search	Individual	Iterative Queries	Cheng Attack	Hardening Random Forest [13]
A-Papernot/Hardening Random Forest	Image	Unsupervised	Additive	Decision Boundary Based	Visible	Transfer Based	Individual	No Probes	Papernot et al. 2019, Liu et al.	Hardening Random Forest [13]

Table 6. List of Data Poisoning Attacks Driven by Mathematical Perturbation Function

No	Attack Name	Mathematical Function	Defence
1	DPA-M-PGD	PGD [127, 157, 157]	Certified Robust [132]
2	DPA-M-Auto-PGD	Auto-PGD [60, 61]	WSNNS [76]
3	DPA-M-LL-PGD	LL-PGD [131]	WSNNS [76]
4	DPA-M-PGD Iterative	PGD Iterative [217]	Vector Defence [118]
5	DPA-M-PGD-Single Shot	PGD-Single Shot [114]	Vector Defence [118]
6	DPA-M-MT-Linf/MT-L2	MT-Linf/MT-L2 [99]	Adversarial Training [38]
7	DPA-M-L-BFGS	BFGS [92]	APE-GAN [216]
9	DPA-M-FGSM	FGSM [7]	FGSM Counter [246]
10	DPA-M-LL-FGSM	LL-FGSM(Step-LL) [236]	Prakash et al. [188]
11	DPA-M-ADA-FGSM	ADA-FGSM [217]	Carrara et al. [37]
12	DPA-M-IFGSM(MI-Linf/MI-L2)	IFGSM(MI-Linf/MI-L2) [60]	Prakash et al. [188]
13	DPA-M-MI	MI [60]	Adversarial Training [38]
14	DPA-M-MI-FGSM	MI-FGSM(Momentum Iterative) [206]	Mustafa et al. [174]
15	DPA-M-TGSM	TGSM [200]	Feature Distillation* [150]
16	DPA-M-IFGSM	IFGSM [60]	SAP [68]
17	DPA-M-ZOO	ZOO [47]	Hybrid Random Forest [71]
18	DPA-M-cADV	cADV Colorisation attack [21]	JPEG defence [63]
19	DPA-M-tAdv	tADV texture transfer attack [20]	JPEG defence [63]
20	DPA-M-StAdv	Spatial Transformation [255]	Adversarial Training [38]
21	DPA-M-BIM	BIM(Iterative FGSM) [127]	Progressive Defence [242]
22	DPA-M-BIM-A	BIM-A [127]	Vector Defence [118]
23	DPA-M-BIM-B	BIM-B [127]	Vector Defence [118]
24	DPA-M-FFF	Fast Feature Fool [171]	Adversarial Training [38]
25	DPA-M-ILCM	Iterative Least-likely class method [127]	Adversarial Training [38]
26	DPA-M-BIM	Momentum BIM [174]	Mustafa [174]
27	DPA-M-Shadow Attack	Semantic spoofed certificates [94]	Mustafa [174]
28	DPA-M-JSMA	Gradient Based [97]	Vector Defence [118]
29	DPA-M-NTM	Metamorphic Relation Based [41]	AT [129]
30	DPA-M-MGA	Momentum Gradient Based [45]	Vector Defence [118]
31	DPA-M-WitchCraft	Gaussian Noise [54]	Certified Robustness [132]
32	DPA-M-QL Attack	Gradient Estimation [101]	Adversarial Training [38]
33	DPA-M-Basic	Least-Likely-Class Iterative Methods [7]	Adversarial Training [38]
34	DPA-M-One Pixel	One Pixel [226]	Pixel Defend [212]
35	DPA-M-Momentum Iterative	Momentum Iterative [73]	Super resolution [174]
36	DPA-M-JigSaw Attack	UAP [168]	Adversarial Training [38]
37	DPA-M-UPSET and ANGRI	UPSET and ANGRI	Adversarial Training [38]
38	DPA-M-Houdini	Houdini [56]	Adversarial Training [38]
39	DPA-M-ATN	AAE-ATN [17]	Adversarial Training [38]
40	DPA-M-SimBA	SimBA [95]	Randomisation [61]
41	DPA-M-SimBA-DCT	SimBA-DCT [101]	Randomisation [61]
42	DPA-M-Patch Attack	Generated Patch [138]	Pixel Defend [212]
43	DPA-M-Adversarial Patch	Adversarial Patch [60]	Pixel Defend [212]
44	DPA-M-DPatch	DPatch [95]	Pixel Defend [212]
45	DPA-M-Carlini & Wagner	C&W [36]	Stochastic Elements [31]
46	DPA-M-IFS	IFS [95]	Adversarial Training [38]
47	DPA-M-QL Attack	QL [101]	Adversarial Training [38]
48	DPA-M-QeBB	QeBB [127]	Adversarial Training [38]
49	DPA-M-MGA Unlimited	MGA [45]	Vector Defence [118]
50	DPA-M-MGA Direct	MGA [45]	Vector Defence [118]
51	DPA-M-MGA Indirect	MGA [45]	Vector Defence [118]
52	DPA-M-FGSM Unlimited	FGSM [261]	Mustafa [174]
53	DPA-M-FGSM Direct	FGSM [261]	Mustafa [174]
54	DPA-M-FGSM Indirect	FGSM [261]	Mustafa [174]
55	DPA-M-IFGSM Ensemble	FGSM [261]	Mustafa [174]
56	DPA-M-MI-FGSM	FGSM [261]	Mustafa [174]
57	DPA-M-TI-FGSM	FGSM [261]	Mustafa [174]

Table 7. List of Data Poisoning Attacks Driven by Algorithm

No	Algorithm Name	Algorithm	Defence
1	DPA-A-APGD	APGD [60, 61]	Differential Approximation [61]
2	DPA-A-PPGD	PPGD [129]	PAT [129]
3	DPA-A-Cassidi	Cassidi [129]	PAT [129]
4	DPA-A-DeepFool	Deelfool [169]	Divide - Denoise [170]
5	DPA-A-LPA	LPA [128]	Trades [128]
6	DPA-A-Fast-LPA	Fast-LPA [128]	Trades [128]
7	DPA-A-Square Attack	Square Attack [12, 111]	Bandlimiting * [142]
8	DPA-A-AutoAttack	Auto Attack [60]	Stochastic Elements [31]
9	DPA-A-NewtonFool	NewtonFool [179, 186, 194]	Adversarial Training [38]
10	DPA-A-R-FGSM	Rand-FGSM [235]	Adversarial Training [38]
11	DPA-A-N-FGSM	N-FGSM [209]	Adversarial Training [38]
12	DPA-A-Fast-FGSM	FAST-FGSM [235]	Adversarial Training [38]
13	DPA-A-Rapid-FGSM	Rapid-FGSM [209]	Adversarial Training [38]
14	DPA-A-Robust-FGSM	Robust-FGSM [209]	JPEG Compression [150]
15	DPA-A-UAP	UAP Universal Adversarial Perturbation [127]	Shardped Edges [68]
16	DPA-A-TUAP	Targeted Universal Adversarial Perturbation [127]	Adversarial Training [38, 177]
17	DPA-A-TUAP-DeepFool	TUAP - DeepFool [127]	Adversarial Retraining [177]
18	DPA-A-TUAP-CW	TUAP-CW [127]	Adversarial Training [38]
19	DPA-A-DFO	Stochastic Derivative Free Optimization [165]	Adversarial Retraining [177]
20	DPA-A-CW	CW-$L0$ [36]	Vectro Defence [118] PixelDefend [224]
21	DPA-A-CW	-$L2$ [36]	Vectro Defence [118] PixelDefend [224]
22	DPA-A-CW	CW-$L\infty$ [36]	Vectro Defence [118] PixelDefend [224]
23	DPA-A-AdvPreprocessing	Image Scaling [90, 191]	Robust scaling algorithm and Image reconstruction [191]
24	DPA-ShadowAttack	Shadow Attack [94]	Random Smoothing Certified Defence* [94]
25	DPA-A-Biggio	Biggio Poisonning [24]	Adversarial Training [38]
26	DPA-A-FrogsAttack	Frogs Poisonning [213]	Data Sanitizing* [58]
27	DPA-A-Salt-Pepper	Salt and Pepper [159]	Adversarial Training [38]
28	DPA-A-SignHunter	Momentum Gradient Based [9]	Randomisation [142]
29	DPA-A-FastMN	Fast Minimum-norm (FMN) Attack [187]	Adversarial Training [38]
30	DPA-A-FAB	Minimally distorted with a Fast Adaptive [59]	Adversarial Training [38]
31	DPA-A-BB	Minimally distorted with a Fast Adaptive [59]	Adversarial Training [38]
32	DPA-A-KKT Based	KKT [123]	Adversarial Training [38]
33	DPA-A-Square Attack	$\mathrm{L1}-APGDAnd\mathrm{L1}-AutoAttack(APGD-AT)$ [12, 111]	Logit Squeezing* [212], Pixel Defend [212]
34	PIA (partial Information Attack)	(QLA variation) [109]	Logit pairing [119]
35	DPA-A-JSMA-F	JSMA-F [36]	Vector Defence [118]
36	DPA-JSMA-Z	JSMA [36]	Vectro Defence [118]
37	DPA-A-JPEG-$L\infty$	JPEG-L_p [28]	JPEG Compression* [64]
38	DPA-A-ReColorAdv	*ReColorAdv* [128]	PAT [129]
39	DPA-A-SimBA (simple black box attack)	$L1$-APGD And $L1$-AutoAttack(APGD-AT) [101]	Pixel Defend [101]
40	DPA-A-SimBA-DCT (simple black box attack)	(SimBA variation) [101]	Pixel Defend [212]
41	DPA-A-Parsimonious(Efficient Combinatorial Optimization)	$L1$-APGD And $L1$-AutoAttack (APGD-AT), Single and Multi APGD [167]	Randomisation [61]
42	DPA-A-DFO -(1+1)-ES	DFO variation-(1+1)-ES [165]	Adversarial Retraining [177]
43	DPA-A-DFO-CMA-ES	DFO variation CMA-ES [165]	Adversarial Retraining [177]
44	DPA-A-Bandits	Bandits [110]	Logit Squeezing* [212]
45	DPA-A-Bandits$_T$	Bandits$_T$ [110]	Logit Squeezing* [212]
46	DPA-A-Bandits$_T D$	Bandits$_T D$ [110]	Logit Squeezing* [212]
47	DPA-A-NES	NES [250]	Augmented Adv Training [31]
48	DPA-A-NES-GE	NES-GE [109]	Augmented Adv Training [31]
49	DPA-A-NES-PIA	NES-PIA [109]	Augmented Adv Training [31]
50	DPA-A-ZOO Attack	ZOO Attack [146]	Shardped Edges [68]
51	DPA-A-ZOO-SGD	ZOO-SGD [146]	Stochastic Element [68]
52	DPA-A-ZOO-SignSGD	ZOO-SignSGD [146]	Stochastic Element [68]
53	DPA-A-ZOO-M-signSGD	ZO-M-signSGD [146]	Stochastic Element [68]
54	DPA-A-ZOO-NES	ZOO-NES [146]	Stochastic Element [68]
55	DPA-A-ZOO-SCD	ZOO-SCD [146]	Stochastic Element [68]
56	DPA-A-FMN	FMN [187]	Adversarial Training [38]
57	DPA-A-Semantic Attack	Semantic [94, 164]	Adversarial Training [38]
58	DPA-A-Discretized Inputs	Discrete Gradient Ascent PGD / PGA [133]	One Hot [32]
59	DPA-A-CROWN-IBP	Shadow-Penalties [94]	Random Smoothing Certified Defence* [94]
60	DPA-A-BPDA	BPDA (Gradient Free) [264]	Adversarial Training [38]
61	DPA-A-BNN-GA	BNN-GA(Gradient Free) [264]	Adversarial Training [38]
62	DPA-A-BNN-ZOO	BNN-ZOO (Gradient Free) [264]	Stochastic Element [68]
63	DPA-A-Koh-Liang attack	Koh-Liang [122]	Adversarial Training [38]
64	DPA-A-ZOO-ADAM	ZOO-ADAM [47]	Gradient Masking [27]
65	DPA-A-ZOO-Newton	ZOO-Newton [47]	Gradient Masking [27]
66	DPA-A-SADS	Saddle Point [206]	Byzantine-Robust Distribution [260]
67	DPA-A-FMN	Fast Minimum-norm [187]	Adversarial Training [38]
68	DPA-A-Physical Attack	Recusrsive Impersonation [215]	Adversarial Training [38]
69	DPA-A-BAE	BERT-based Adversarial Examples [91]	Synonym Encoded [244]
70	DPA-A-DeepWordBug	DeepWordBug [91]	Synonym Encoded [244]
71	DPA-A-FasterGenetic	FasterGenetic [91]	Synonym Encoded [244]
72	DPA-A-Genetic	Genetic [91]	Synonym Encoded [244]
73	DPA-A-HotFlip	HotFlip [91]	Synonym Encoded [244]
74	DPA-A-IGA-Pruthi	IGA-Pruthi [91]	Synonym Encoded [244]
75	DPA-A-PSO	TextAttack [91]	Synonym Encoded [244]
76	DPA-A-TextBugger	TextAttack [137]	Synonym Encoded [244]
77	DPA-A-TextFooler	TextAttack [116]	Synonym Encoded [244]
78	DPA-A-VIPER	TextAttack [91]	Synonym Encoded [244]
79	DPA-A-GASC	GASC [11]	Synonym Encoded Method [244]
80	DPA-A-GNLAE	GNLAE [11]	Synonym Encoded Method [244]
81	DPA-A-BAE-R	BERT-based Adversarial Examples [91]	Synonym Encoded [244]
82	DPA-A-BAE-I	BERT-based Adversarial Examples [91]	Synonym Encoded [244]
83	DPA-A-BAE-R/I	BERT-based Adversarial Examples [91]	Synonym Encoded [244]
84	DPA-A-BAE-R+I	BERT-based Adversarial Examples [91]	Synonym Encoded [244]
85	DPA-A-WordBug Replace-1	WordBug [88]	Randomised Smoothing [268]
86	DPA-A-WordBug - Temporal Head	WordBug [88]	Randomised Smoothing [268]
87	DPA-A-WordBug - Temporal Tail	WordBug [88]	Randomised Smoothing [268]
88	DPA-A-WordBug - Combined	WordBug [88]	Randomised Smoothing [268]
89	DPA-A-Adv-tr whitebox	HotFlip [77]	Adversarial Training [38]
90	DPA-A-Adv-tr blackbox	HotFlip [77]	Adversarial Training [38]
91	DPA-A-Input Reduction	Word Deletion [172]	Adversarial Training [38]
92	DPA-A-Morpheus	Morpheus [172]	Adversarial Training [38]
93	DPA-A-Practical Swarm OPT	Swarm OPT [201]	Adversarial Training [38]
94	DPA-A-PWWWS	PWWWS [195]	Adversarial Training [38]

(Continued)

Table 7. Continued

No	Algorithm Name	Algorithm	Defence
95	DPA-A-GSA	GSA [245]	Adversarial Training [38]
96	DPA-A-seq2sick	seq2sick [51]	Adversarial Training [38]
97	DPA-A-Kuleshov	Kuleshov [126]	Adversarial Training [38]
98	DPA-A-FGPM	FGPM [245]	Adversarial Training [38]
99	DPA-A-Gaussian Noise	Gaussian Noise [54]	Certified Robustness [132]
100	DPA-A-Bernoulli Noise Attack	Bernoulli Noise Attack [248]	Adversarial Retraining [38]
101	DPA-A-Discrete Token Replacement	Discrete Token Replacement [184]	Randomised Smoothing [268]
102	DPA-A-No Overlap Poisoning	No Overlap Poisoning [240]	Adversarial Retraining [38]
103	DPA-A-Spoofing	Spoofing [147]	Data Sanitizing [58]
104	DPA-A-Spare Binary Vectors	Spare Binary [82]	Adversarial Retraining [38]
105	DPA-A-PC-lhc	PC-lhc [26]	Adversarial Retraining [38]
106	DPA-PS-lhc	PS-lhc [23]	Adversarial Retraining [38]
107	DPA-A-A-Subtle	A-Subtle [7]	Hard Class Labels [8]
108	DPA-A-M-Naively Poisoning	Naively Poisoning [42]	Adversarial Training [38]
109	DPA-A-GAN	GAN [216]	Data Sanitizing [58]
110	DPA-A-Kantchelian Attack	Kantchelian [120]	Robust Split for decision trees [44]
111	DPA-A-Flipping	Flipping [272]	Data Sanitizing [58]
112	DPA-A-Rotating	Rotating [78]	Data Sanitizing [58]
113	DPA-A-Cropping	Cropping [135]	Data Sanitizing [58]
114	DPA-A-Color Jittering	Color Jittering [182]	Data Sanitizing [58]
115	DPA-A-Edge Enhancement	Edge Enhancement [53]	Data Sanitizing [58]
116	DPA-A-Fancy PCA	Fancy PCA [230]	Data Sanitizing [58]
117	DPA-A-Mixing Images	FineGan [223]	Data Sanitizing [58]
118	DPA-A-Random Erasing	Random Erasing [275]	Data Sanitizing [58]
119	DPA-A-Style Reconstruction	tyle Reconstruction [49]	Data Sanitizing [58]
120	DPA-Grad-CAM	Grad-CAM [40]	Data Sanitizing [58]
121	DPA-A-Hash	Hash Collision [72]	Data Sanitizing [58]
122	DPA-A-fishAttack	fishAttack [213]	Data Sanitizing [58]
123	DPA-A-SPSA	SPSA [238]	JPEG Compression [64]
124	DPA-A-RGF	RGF [52]	JPEG Compression [64]
125	DPA-A-FGS-Single Step	GS-Single Step [79]	Shardped edges [68]
126	DPA-A-IFGS Iterative Step	IFGS [232]	Shardped edges [68]
127	DPA-A-FD-GE Single Step	FD-GE [19]	Shardped edges [68]
128	DPA-A-IFD-GE Iterative	IFD-GE Iterative [19]	Shardped edges [68]
129	DPA-A-PCA-GE Single Step	PCA-GE Single Step [19]	Shardped edges [68]
130	DPA-A-PCA-Query	PCA-Query Reduction Iterative [19]	Shardped edges [68]
131	DPA-A-AA	AA [136]	Shardped edges [68]
132	DPA-A-AAA	AAA [136]	Shardped edges [68]
133	DPA-A-ADI-PGD	ADI [149]	Shardped edges [68]
134	DPA-A-R-ADI	ADI [149]	Shardped edges [68]
135	DPA-A-ADI+OSD	ADI [149]	Shardped edges [68]
136	DPA-A-BayesOPT Attack	Bayes [202]	Shardped edges [68]
137	DPA-A-GP-Based BayesOPT	Bayes [202]	Shardped edges [68]
138	DPA-A-Additive GP-BayesOPT	Bayes [202]	Shardped edges [68]
139	DPA-A-Bayes-OPT with Selection	Bayes [202]	Shardped edges [68]
140	DPA-A-GP-BO-Auto	Bayes [202]	Shardped edges [68]
141	DPA-A-ADDGP-BO	Bayes [202]	Shardped edges [68]
142	DPA-A-PIA	PIA [208]	Cascade Adversarial Training [176]
143	DPA-A-Fredriksn et al. 2014	redriksn [87]	Cascade Adversarial Training [176]
144	DPA-A-Shokri et al. 2017	Shokri [220]	Cascade Adversarial Training [176]
145	DPA-A-Long et al. 2018	Long [19]	Cascade Adversarial Training [176]
146	DPA-A-Rahman et al. 2018	Rahman [19]	Cascade Adversarial Training [176]
147	DPA-A-Hayes et al. 2019	Hayes [19]	Cascade Adversarial Training [176]
148	Hilprecht et al. 2019	Hilprecht [19]	Cascade Adversarial Training [176]
149	Jayaraman et al.	Jayaraman [19]	Cascade Adversarial Training [176]
150	DPA-A-Nasr et al. 2019	Nasr [19]	Cascade Adversarial Training [176]
151	DPA-A-Melis et al. 2019	Melis [19]	Cascade Adversarial Training [176]
152	DPA-A-Sablayrolles et al. 2019	Sablayrolles [19]	Cascade Adversarial Training [176]
153	DPA-A-Salem et al. 2019	Salem [19]	Cascade Adversarial Training [176]
154	DPA-A-Song et al. 2019	Song [19]	Cascade Adversarial Training [176]
155	DPA-A-Truex et al. 2019	Truex [19]	Cascade Adversarial Training [176]
156	DPA-A-Chen et al. 2020	Chen [19]	Cascade Adversarial Training [176]
157	DPA-A-Hishamoto et al. 2019	Hishamoto [19]	Cascade Adversarial Training [176]
158	DPA-A-Song and Raghunathan	Song and Raghunathan [19]	Cascade Adversarial Training [176]
159	DPA-A-LinBP+RR	LinBP [102]	Random and Pixel Defend [212]
160	DPA-A-LinBP+ElasticNet	LinBP [102]	Random and Pixel Defend [212]
161	DPA-A-LinBP+SVR	LinBP [102]	Random and Pixel Defend [212]
162	DPA-A-LinBP+I+FGSM	LinBP [102]	Random and Pixel Defend [212]
163	DPA-A-LinBP+I+FGSM+ILA	LinBP [102]	Random and Pixel Defend [212]
164	DPA-A-LinBP+I+FGSM+ILA+SGM	LinBP [102]	Random and Pixel Defend [212]

Table 8. List of Defence Solutions

No	Defence Name
1	Certified Robustness [132, 192]
2	Differential Approximation [61]
3	Randomised [61]
4	Detector based [61]
5	Counter [7, 14, 115, 166]
6	Vector Defence [118]
7	BAT [241]
8	Madry [157]
9	Malade [152]
10	WSNNS [76]
11	Prakash et al. [188]
12	SAP [68]
13	PixelDefend [224]
14	Mustafa et al. [174]
15	D3 algorithm [170]
16	Feinman et al. [83]
17	Carrara et al. [37]
18	RRP [256]
19	Bhagoji et al. [18]
20	ReabsNet [46]
21	Zheng and Hong [274]
22	Det [134]
23	Grosse et al. [100]
24	RCE [181]
25	NIC [153]
26	Cao and Gong [33]
27	Hendrycks and Gimpel [35]
28	Feature Distillation [150]
29	LID [154]
30	Cohen et al. [57]
31	S2SNet [84]
32	Gong et al. [97]
33	Metzen et al. [164]
34	Das et al. [63]
35	CCNs [194]
36	Na et al. [176]
37	Magnet [163]
38	MultiMagnet [155]
40	ME-Net [259]
41	SafetyNet [151]
42	Papernot and McDaniel [183]
43	Feature Squeezing [218]
44	Abbasi and Gagné [3]
45	Strauss et al. [225]
46	Tramèr et al. [236]
47	MTDeep [210]
48	Defence-GAN [216]
49	APE-GAN [216]
50	Zantedeschi et al. [267]
51	Liu et al. [143]
52	Hybrid Random Forest [71]
53	Bandlimiting [142]
54	Probabilistic adversarial robustness [231]
55	Adversarial Retraining [177]
56	JPEG Compression [64]

(Continued)

Table 8. Continued

No	Defence Name
57	Adversarial Training [38]
58	Cascade adversarial training [176]
59	no-Pixel Defend [212]
60	One Hot [32]
61	Mask Gradient [27]
62	Image Denoising [174]
63	Data Sanitizing [58]
64	High dimensional robust estimation [69]
65	Vector Defence [118]
66	Regularization [31]
67	Gradient Masking [27]
68	Stochastic Elements [31]
69	RobustScaling [190]
70	Logit Squeezing* [212]
71	Super resolution [174]
72	Thermometer Encoding [32]
73	BAT [253]
74	Data Augmentation [221]
75	Data Sanitizing [58]
76	Defensive Distillation [34]
77	Filter (Gaussian, AVerage, Median) [263]
78	PAT [129]
79	PGD-AdvT [157]
80	Ensemble-AdvT [236]
81	Augmented Adv Training [31]
82	JPEG Compression [16]
83	Guided Denoiser [52]
84	Stochastic Element [68]
85	Shardped Edges [68]
86	Adversarial Purification [262]
87	Certified Robustness [132]
88	Random and Pixel Defend [212]
89	Semantic defence [117]
90	Synonym Encoded Method [244]
91	Dirichlet Neighborhood Ensemble [276]
92	Randomised Smoothing [268]
93	K-LID-SVM [249]
94	LSD defence [81]
95	DBSCAN Preprocessing Sanitizing [62]
96	Non-convex Guarantee [10]
97	Byzantine-Robust Distribution [260]
98	Hiding Prediction Information [55]
99	Adv Regularization [106]
100	Multi Model Based defence [237]
101	Modifying the network structure [199]
102	Principled adversarial training [203]
103	Perturbation Subtracting defence [50]
104	Gradient band-based adversarial training [48]
105	Data Randomization [15]
106	JPEG Encoding [219]
107	Gaussian Blur [273]
108	Selective Dropout [5]
109	Robust Split with Information Gain [44]
110	Hardening Random Forest [13]
111	Robust Split for decision trees [44]

REFERENCES

[1] [n. d.]. Tesla denies car was driverless in fatal crash that killed two men in the United States - ABC News. https://www.abc.net.au/news/2021-04-28

[2] 2016. Tay: Microsoft issues apology over racist chatbot fiasco. (2016). https://www.bbc.com/news/technology-35902104

[3] Mahdieh Abbasi and Christian Gagné. 2017. Robustness to adversarial examples through an ensemble of specialists. *arXiv preprint arXiv:1702.06856* (2017).

[4] Hervé Abdi and Lynne J. Williams. 2010. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* 2, 4 (2010), 433–459.

[5] Akshay Agarwal, Richa Singh, Mayank Vatsa, and Nalini Ratha. 2020. Image transformation-based defense against adversarial perturbation on deep learning models. *IEEE Transactions on Dependable and Secure Computing* 18, 5 (2020), 2106–2121.

[6] Hojjat Aghakhani, Lea Schönherr, Thorsten Eisenhofer, Dorothea Kolossa, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. 2021. VenoMave: Targeted Poisoning against Speech Recognition. arXiv:2010.10682 [cs.SD].

[7] Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* 6 (2018), 14410–14430. https://arxiv.org/abs/2012.14368

[8] Naveed Akhtar and Ajmal Mian. 2018. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. https://arxiv.org/abs/1801.00554

[9] Abdullah Al-Dujaili and Una-May O'Reilly. 2020. Sign bits are all you need for black-box attacks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SygW0TEFwH

[10] Zeyuan Allen-Zhu, Faeze Ebrahimian, Jerry Li, and Dan Alistarh. 2020. Byzantine-Resilient Non-Convex Stochastic Gradient Descent. https://doi.org/10.48550/ARXIV.2012.14368

[11] Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava. 2018. Did you hear that? Adversarial examples against automatic speech recognition. *arXiv preprint arXiv:1801.00554* (2018).

[12] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. Square attack: A query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*. Springer, 484–501.

[13] Giovanni Apruzzese, Mauro Andreolini, Michele Colajanni, and Mirco Marchetti. 2020. Hardening random forest cyber detectors against adversarial attacks. *IEEE Transactions on Emerging Topics in Computational Intelligence* 4, 4 (2020), 427–439.

[14] Anurag Arnab, Ondrej Miksik, and Philip H. S. Torr. 2018. On the robustness of semantic segmentation models to adversarial attacks. In *CVPR*.

[15] Marzieh Ashrafiamiri, Sai Manoj Pudukotai Dinakarrao, Amir Hosein Afandizadeh Zargari, Minjun Seo, Fadi Kurdahi, and Houman Homayoun. 2020. R2AD: Randomization and reconstructor-based adversarial defense on deep neural network. In *Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD*. 21–26.

[16] Ayse Elvan Aydemir, Alptekin Temizel, and Tugba Taskaya Temizel. 2018. The effects of JPEG and JPEG2000 compression on attacks using adversarial examples. *arXiv preprint arXiv:1803.10418* (2018).

[17] Shumeet Baluja and Ian Fischer. 2017. Adversarial Transformation Networks: Learning to Generate Adversarial Examples. arXiv:1703.09387 [cs.NE].

[18] Arjun Nitin Bhagoji, Daniel Cullina, Chawin Sitawarin, and Prateek Mittal. 2018. Enhancing robustness of machine learning systems via data transformations. In *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 1–5.

[19] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. 2018. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 154–169.

[20] Siddhant Bhambri, Sumanyu Muku, Avinash Tulasi, and Arun Balaji Buduru. 2019. A survey of black-box adversarial attacks on computer vision models. *arXiv preprint arXiv:1912.01667* (2019).

[21] Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and D. A. Forsyth. 2020. Unrestricted Adversarial Examples via Semantic Manipulation. arXiv:1904.06347 [cs.CV]. (2020).

[22] Pavol Bielik, Veselin Raychev, and Martin Vechev. 2017. Learning a static analyzer from data. (2017), 233–253.

[23] Battista Biggio, Samuel Rota Bulò, Ignazio Pillai, Michele Mura, Eyasu Zemene Mequanint, Marcello Pelillo, and Fabio Roli. 2014. Poisoning complete-linkage hierarchical clustering. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 42–52.

[24] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion attacks against machine learning at test time. *Lecture Notes in Computer Science* (2013), 387–402. https://doi.org/10.1007/978-3-642-40994-3_25

[25] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2011. Support vector machines under adversarial label noise. In *Asian Conference on Machine Learning*. PMLR, 97–112.

[26] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2013. Poisoning Attacks against Support Vector Machines. arXiv:1206.6389 [cs.LG].
[27] Franziska Boenisch, Philip Sperl, and Konstantin Böttinger. 2021. Gradient Masking and the Underestimated Robustness Threats of Differential Privacy in Deep Learning. arXiv:2105.07985 [cs.CR]. https://dl.acm.org/doi/10.1016/j.eswa.2014.09.054
[28] Akhilan Boopathy, Sijia Liu, Gaoyuan Zhang, Cynthia Liu, Pin-Yu Chen, Shiyu Chang, and Luca Daniel. 2020. Proper network interpretability helps adversarial robustness in classification. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 1014–1023. https://proceedings.mlr.press/v119/boopathy20a.html
[29] Athman Bouguettaya, Qi Yu, Xumin Liu, Xiangmin Zhou, and Andy Song. 2015. Efficient agglomerative hierarchical clustering. *Expert Systems with Applications* 42, 5 (2015), 2785–2797.
[30] Ajay Kumar Boyat and Brijendra Kumar Joshi. 2015. A review paper: Noise models in digital image processing. (2015). https://doi.org/10.48550/ARXIV.1505.03489
[31] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-Based Adversarial Attacks: Reliable Attacks against Black-Box Machine Learning Models. arXiv:1712.04248 [stat.ML]
[32] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. 2018. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*.
[33] Xiaoyu Cao and Neil Zhenqiang Gong. 2017. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*. 278–287.
[34] Nicholas Carlini and David Wagner. 2016. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311* (2016).
[35] Nicholas Carlini and David Wagner. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 3–14.
[36] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. arXiv:1608.04644 [cs.CR].
[37] Fabio Carrara, Fabrizio Falchi, Roberto Caldelli, Giuseppe Amato, and Rudy Becarelli. 2019. Adversarial image detection in deep neural networks. *Multimedia Tools and Applications* 78, 3 (2019), 2815–2835.
[38] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069* (2018).
[39] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2021. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology* 6, 1 (2021), 25–45.
[40] Tanmay Chakraborty, Utkarsh Trehan, Khawla Mallat, and Jean-Luc Dugelay. 2022. Generalizing Adversarial Explanations with Grad-CAM. https://doi.org/10.48550/ARXIV.2204.05427
[41] Alvin Chan, Lei Ma, Felix Juefei-Xu, Xiaofei Xie, Yang Liu, and Yew Soon Ong. 2018. Metamorphic Relation Based Adversarial Attacks on Differentiable Neural Computer. https://doi.org/10.48550/ARXIV.1809.02444
[42] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. 2019. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *arXiv preprint arXiv:1912.11279* (2019).
[43] Zhaohui Che, Ali Borji, Guangtao Zhai, Suiyi Ling, Jing Li, and Patrick Le Callet. 2019. A New Ensemble Adversarial Attack Powered by Long-term Gradient Memories. arXiv:1911.07682 [cs.LG].
[44] Hongge Chen, Huan Zhang, Duane Boning, and Cho-Jui Hsieh. 2019. Robust Decision Trees against Adversarial Examples. arXiv:1902.10660 [cs.LG].
[45] Jinyin Chen, Yixian Chen, Haibin Zheng, Shijing Shen, Shanqing Yu, Dan Zhang, and Qi Xuan. 2020. MGA: Momentum gradient attack on network. *IEEE Transactions on Computational Social Systems* 8, 1 (2020), 99–109. https://arxiv.org/abs/1807.06752
[46] Jiefeng Chen, Zihang Meng, Changtian Sun, Wei Tang, and Yinglun Zhu. 2017. ReabsNet: Detecting and revising adversarial examples. *arXiv preprint arXiv:1712.08250* (2017).
[47] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 15–26.
[48] Tong Chen, Wenjia Niu, Yingxiao Xiang, Xiaoxuan Bai, Jiqiang Liu, Zhen Han, and Gang Li. 2018. Gradient band-based adversarial training for generalized attack immunity of A3C path finding. *arXiv preprint arXiv:1807.06752* (2018).
[49] Xihao Chen, Jingya Yu, Li Chen, Shaoqun Zeng, Xiuli Liu, and Shenghua Cheng. 2019. Multi-stage domain adversarial style reconstruction for cytopathological image stain normalization.
[50] Gong Cheng, Xuxiang Sun, Ke Li, Lei Guo, and Junwei Han. 2021. Perturbation-seeking generative adversarial networks: A defense framework for remote sensing image scene classification. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–11.

[51] Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020. Seq2Sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3601–3608.

[52] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Improving black-box adversarial attacks with a transfer-based prior. *Advances in Neural Information Processing Systems* 32 (2019).

[53] Yupeng Cheng, Qing Guo, Felix Juefei-Xu, Wei Feng, Shang-Wei Lin, Weisi Lin, and Yang Liu. 2021. Pasadena: Perceptually aware and stealthy adversarial denoise attack. *IEEE Transactions on Multimedia* (2021).

[54] Ping-Yeh Chiang, Jonas Geiping, Micah Goldblum, Tom Goldstein, Renkun Ni, Steven Reich, and Ali Shafahi. 2019. WITCHcraft: Efficient PGD attacks with random step size. https://doi.org/10.48550/ARXIV.1911.07989

[55] Partha Chowdhuri and Biswapati Jana. 2020. Hiding data in dual color images reversibly via weighted matrix. *Journal of Information Security and Applications* 50 (2020), 102420.

[56] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. 2017. Houdini: Fooling Deep Structured Prediction Models. arXiv:1707.05373 [stat.ML]

[57] Gilad Cohen, Guillermo Sapiro, and Raja Giryes. 2020. Detecting adversarial samples using influence functions and nearest neighbors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14453–14462.

[58] Gabriela F. Cretu, Angelos Stavrou, Michael E. Locasto, Salvatore J. Stolfo, and Angelos D. Keromytis. 2008. Casting out Demons: Sanitizing Training Data for Anomaly Sensors. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. 81–95. https://doi.org/10.1109/SP.2008.11

[59] Francesco Croce and Matthias Hein. 2020. Minimally Distorted Adversarial Examples with a Fast Adaptive Boundary Attack. 119 (2020), 2196–2205. https://proceedings.mlr.press/v119/croce20a.html

[60] Francesco Croce and Matthias Hein. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. arXiv:2003.01690 [cs.LG].

[61] Francesco Croce and Matthias Hein. 2021. Mind the box: l_1-APGD for sparse adversarial attacks on image classifiers. arXiv:2103.01208 [cs.LG].

[62] Jonathan Crussell and Philip Kegelmeyer. 2015. Attacking DBSCAN for fun and profit. *SIAM International Conference on Data Mining 2015, SDM 2015* (2015), 235–243. https://doi.org/10.1137/1.9781611974010.27

[63] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E. Kounavis, and Duen Horng Chau. 2017. Keeping the bad guys out: Protecting and vaccinating deep learning with JPEG compression. *arXiv preprint arXiv:1705.02900* (2017).

[64] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E. Kounavis, and Duen Horng Chau. 2018. SHIELD: Fast, Practical Defense and Vaccination for Deep Learning using JPEG Compression. (2018).

[65] Shankar A. Deka, Dušan M. Stipanović, and Claire J. Tomlin. 2020. Dynamically Computing Adversarial Perturbations for Recurrent Neural Networks. arXiv:2009.02874 [cs.LG].

[66] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. 2019. Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks. arXiv:1809.02861 [cs.LG]

[67] Yingpeng Deng and Lina J. Karam. 2020. Universal Adversarial Attack via Enhanced Projected Gradient Descent. In *2020 IEEE International Conference on Image Processing (ICIP)*. 1241–1245. https://doi.org/10.1109/ICIP40778.2020.9191288

[68] Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. 2018. Stochastic Activation Pruning for Robust Adversarial Defense. arXiv:1803.01442 [cs.LG].

[69] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. 2019. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*. PMLR, 1596–1606.

[70] Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*. Springer, 1–15.

[71] Yifan Ding, Liqiang Wang, Huan Zhang, Jinfeng Yi, Deliang Fan, and Boqing Gong. 2019. Defending against adversarial attacks using random forest. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 0–0.

[72] Brian Dolhansky and Cristian Canton Ferrer. 2020. Adversarial collision attacks on image hashing functions. https://doi.org/10.48550/ARXIV.2011.09473

[73] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9185–9193. https://arxiv.org/abs/1903.01612

[74] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks. arXiv:1904.02884 [cs.CV].

[75] Xiaomin Duan, Huafei Sun, Linyu Peng, and Xinyu Zhao. 2013. A natural gradient descent algorithm for the solution of discrete algebraic Lyapunov equations based on the geodesic distance. *Appl. Math. Comput.* 219, 19 (2013), 9899–9905. https://doi.org/10.1016/j.amc.2013.03.119

[76] Abhimanyu Dubey, Laurens van der Maaten, Zeki Yalniz, Yixuan Li, and Dhruv Mahajan. 2019. Defense against Adversarial Images using Web-Scale Nearest-Neighbor Search. https://doi.org/10.48550/ARXIV.1903.01612

[77] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. HotFlip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751* (2017).

[78] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. 2018. A rotation and a translation suffice: Fooling CNNs with simple transformations. (2018).

[79] Okwudili M. Ezeme. 2020. Anomaly detection in kernel-level process events using machine learning-based context analysis. (2020).

[80] Linwei Fan, Fan Zhang, Hui Fan, and Caiming Zhang. 2019. Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art* 2, 1 (2019), 1–12.

[81] Yanbo Fan, Baoyuan Wu, Tuanhui Li, Yong Zhang, Mingyang Li, Zhifeng Li, and Yujiu Yang. 2020. Sparse Adversarial Attack via Perturbation Factorization. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 35–50.

[82] Yanbo Fan, Baoyuan Wu, Tuanhui Li, Yong Zhang, Mingyang Li, Zhifeng Li, and Yujiu Yang. 2020. Sparse adversarial attack via perturbation factorization. In *European Conference on Computer Vision*. Springer, 35–50.

[83] Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. 2017. Detecting Adversarial Samples from Artifacts. arXiv:1703.00410 [stat.ML]

[84] Joachim Folz, Sebastian Palacio, Joern Hees, and Andreas Dengel. 2020. Adversarial defense based on structure-to-signal autoencoders. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 3568–3577.

[85] Liam H. Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojciech Czaja, and Tom Goldstein. 2021. Adversarial Examples Make Strong Poisons. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.). https://openreview.net/forum?id=DE8MOQIgFTK

[86] Chris Fraley and Adrian E. Raftery. 2002. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association* 97, 458 (2002), 611–631.

[87] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1322–1333.

[88] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 50–56.

[89] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. 2020. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760* (2020).

[90] Yue Gao and Kassem Fawaz. 2021. Scale-Adv: A Joint Attack on Image-Scaling and Machine Learning Classifiers. arXiv:2104.08690 [cs.LG].

[91] Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based Adversarial Examples for Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2020.emnlp-main.498

[92] Ibrahim Gashaw and H. L. Shashirekha. 2020. Machine Learning Approaches for Amharic Parts-of-speech Tagging. *arXiv preprint arXiv:2001.03324* (2020).

[93] Zoubin Ghahramani. 2003. Unsupervised learning. (2003), 72–112.

[94] Amin Ghiasi, Ali Shafahi, and Tom Goldstein. 2020. Breaking certified defenses: Semantic adversarial examples with spoofed robustness certificates. https://arxiv.org/abs/2012.10544

[95] Anteneh Girma, Mosses Garuba, and Rajini Goel. 2018. Advanced Machine Language Approach to Detect DDoS Attack Using DBSCAN Clustering Technology with Entropy. In *Information Technology - New Generations*, Shahram Latifi (Ed.). Springer International Publishing, Cham, 125–131.

[96] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Mądry, Bo Li, and Tom Goldstein. 2022. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 2 (2022), 1563–1580.

[97] Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. 2017. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960* (2017).

[98] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[99] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. 2019. On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models. arXiv:1810.12715 [cs.LG].

[100] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. 2017. On the (Statistical) Detection of Adversarial Examples. arXiv:1702.06280 [cs.CR].

[101] Chuan Guo, Jacob R. Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q. Weinberger. 2019. Simple black-box adversarial attacks. *arXiv preprint arXiv:1905.07121* (2019).

[102] Yiwen Guo, Qizhang Li, and Hao Chen. 2020. Backpropagating linearly improves transferability of adversarial examples. *Advances in Neural Information Processing Systems* 33 (2020), 85–95.

[103] Atiye Sadat Hashemi and Saeed Mozaffari. 2021. CNN adversarial attack mitigation using perturbed samples training. *Multimedia Tools and Applications* 80, 14 (2021), 22077–22095.

[104] Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. 2017. Gradient Methods for Submodular Maximization. arXiv:1708.03949 [cs.LG].

[105] Sanghyun Hong, Varun Chandrasekaran, Yiğitcan Kaya, Tudor Dumitraş, and Nicolas Papernot. 2020. On the effectiveness of mitigating data poisoning attacks with gradient shaping. *arXiv preprint arXiv:2002.11497* (2020).

[106] Bo Huang, Zhiwei Ke, Yi Wang, Wei Wang, Linlin Shen, and Feng Liu. 2021. Adversarial Defence by Diversified Simultaneous Training of Deep Ensembles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 7823–7831.

[107] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial Attacks on Neural Network Policies. arXiv:1702.02284 [cs.LG].

[108] Zhichao Huang and Tong Zhang. 2020. Black-Box Adversarial Attack with Transferable Model-based Embedding. arXiv:1911.07140 [cs.LG].

[109] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box Adversarial Attacks with Limited Queries and Information. arXiv:1804.08598 [cs.CV].

[110] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. 2019. Prior Convictions: Black-Box Adversarial Attacks with Bandits and Priors. arXiv:1807.07978 [stat.ML]

[111] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2020. High Accuracy and High Fidelity Extraction of Neural Networks. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*.

[112] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. arXiv:1804.00308 [cs.CR].

[113] Daniel Jakubovitz and Raja Giryes. 2018. Improving DNN robustness to adversarial attacks using Jacobian regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 514–529.

[114] Yunseok Jang, Tianchen Zhao, Seunghoon Hong, and Honglak Lee. 2019. Adversarial defense via learning to generate diverse attacks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2740–2749.

[115] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014).

[116] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8018–8025.

[117] Ameya Joshi, Amitangshu Mukherjee, Soumik Sarkar, and Chinmay Hegde. 2019. Semantic Adversarial Attacks: Parametric Transformations That Fool Deep Classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[118] Vishaal Munusamy Kabilan, Brandon Morris, and Anh Nguyen. 2018. VectorDefense: Vectorization as a Defense to Adversarial Examples. https://doi.org/10.48550/ARXIV.1804.08529

[119] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. 2018. Adversarial Logit Pairing. arXiv:1803.06373 [cs.LG].

[120] Alex Kantchelian, J. Doug Tygar, and Anthony Joseph. 2016. Evasion and hardening of tree ensemble classifiers. In *International Conference on Machine Learning*. PMLR, 2387–2396.

[121] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. 2014. DBSCAN: Past, present and future. (2014), 232–238.

[122] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*. PMLR, 1885–1894.

[123] Pang Wei Koh, Jacob Steinhardt, and Percy Liang. 2022. Stronger data poisoning attacks break data sanitization defenses. *Machine Learning* (2022), 1–47.

[124] Zixiao Kong, Jingfeng Xue, Yong Wang, Lu Huang, Zequn Niu, and Feng Li. 2021. A survey on adversarial attack in the age of artificial intelligence. *Wireless Communications and Mobile Computing* 2021 (2021), 1–22.

[125] Ioannis Kontopoulos, Giannis Spiliopoulos, Dimitrios Zissis, Konstantinos Chatzikokolakis, and Alexander Artikis. 2018. Countering real-time stream poisoning: An architecture for detecting vessel spoofing in streams of AIS data. In *2018 IEEE 16th Intl. Conf. on Dependable, Autonomic and Secure Computing, 16th Intl. Conf. on Pervasive Intelligence and Computing, 4th Intl. Conf. on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. IEEE, 981–986.

[126] Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems. (2018).

[127] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236* (2016).

[128] Cassidy Laidlaw and Soheil Feizi. 2019. Functional Adversarial Attacks. arXiv:1906.00001 [cs.LG].

[129] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. 2021. Perceptual Adversarial Robustness: Defense against Unseen Threat Models. arXiv:2006.12655 [cs.LG].

[130] Curtis P. Langlotz, Bibb Allen, Bradley J. Erickson, Jayashree Kalpathy-Cramer, Keith Bigelow, Tessa S. Cook, Adam E. Flanders, Matthew P. Lungren, David S. Mendelson, Jeffrey D. Rudie, et al. 2019. A roadmap for foundational research on artificial intelligence in medical imaging: From the 2018 NIH/RSNA/ACR/The Academy Workshop. *Radiology* 291, 3 (2019), 781–791.

[131] Alfred Laugros, Alice Caplier, and Matthieu Ospici. 2019. Are Adversarial Robustness and Common Perturbation Robustness Independent Attributes ? arXiv:1909.02436 [cs.LG].

[132] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. 2019. Certified Robustness to Adversarial Examples with Differential Privacy. arXiv:1802.03471 [stat.ML]

[133] Qi Lei, Lingfei Wu, Pin-Yu Chen, Alexandros G. Dimakis, Inderjit S. Dhillon, and Michael Witbrock. 2018. Discrete Adversarial Attacks and Submodular Optimization with Applications to Text Classification. https://doi.org/10.48550/ARXIV.1812.00151

[134] Changjiang Li, Haiqin Weng, Shouling Ji, Jianfeng Dong, and Qinming He. 2019. DeT: Defending against Adversarial Examples via Decreasing Transferability. In *International Symposium on Cyberspace Safety and Security*. Springer, 307–322.

[135] Feng Li, Xuehui Du, and Liu Zhang. 2022. Adversarial Attacks Defense Method Based on Multiple Filtering and Image Rotation. *Discrete Dynamics in Nature and Society* 2022 (2022).

[136] Jingjing Li, Zhekai Du, Lei Zhu, Zhengming Ding, Ke Lu, and Heng Tao Shen. 2021. Divergence-agnostic Unsupervised Domain Adaptation by Adversarial Attacks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1. https://doi.org/10.1109/TPAMI.2021.3109287

[137] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. TextBugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271* (2018).

[138] Xiang Li and Shihao Ji. 2021. Generative Dynamic Patch Attack. https://doi.org/10.48550/ARXIV.2111.04266

[139] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek. 2003. The global k-means clustering algorithm. *Pattern Recognition* 36, 2 (2003), 451–461.

[140] Jing Lin, Long Dang, Mohamed Rahouti, and Kaiqi Xiong. 2021. ML Attack Models: Adversarial Attacks and Data Poisoning Attacks. *arXiv preprint arXiv:2112.02797* (2021).

[141] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. 2015. Bilinear CNN Models for Fine-Grained Visual Recognition. (December 2015).

[142] Yuping Lin, Kasra Ahmadi K. A., and Hui Jiang. 2019. Bandlimiting Neural Networks against Adversarial Attacks. arXiv:1905.12797 [cs.LG].

[143] Ninghao Liu, Hongxia Yang, and Xia Hu. 2018. Adversarial detection with model interpretation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1803–1811.

[144] Qiang Liu, Pan Li, Wentao Zhao, Wei Cai, Shui Yu, and Victor C. M. Leung. 2018. A survey on security threats and defensive techniques of machine learning: A data driven view. *IEEE Access* 6 (2018), 12103–12117.

[145] Sijia Liu, Pin-Yu Chen, Xiangyi Chen, and Mingyi Hong. [n. d.]. signSGD via Zeroth-Order Oracle. ([n. d.]). https://www.researchgate.net/publication/339404260_Adversarial_Attacks_on_Spoofing_Countermeasures_of_Automatic_Speaker_Verification

[146] Sijia Liu, Pin-Yu Chen, Xiangyi Chen, and Mingyi Hong. 2019. signSGD via Zeroth-Order Oracle. In *International Conference on Learning Representations*. https://ieeexplore.ieee.org/document/9294026

[147] Songxiang Liu, Haibin Wu, Hung-yi Lee, and Helen Meng. 2019. Adversarial attacks on spoofing countermeasures of automatic speaker verification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 312–319.

[148] Ximeng Liu, Lehui Xie, Yaopeng Wang, Jian Zou, Jinbo Xiong, Zuobin Ying, and Athanasios V. Vasilakos. 2020. Privacy and security issues in deep learning: A survey. *IEEE Access* 9 (2020), 4566–4593.

[149] Ye Liu, Yaya Cheng, Lianli Gao, Xianglong Liu, Qilong Zhang, and Jingkuan Song. 2022. Practical evaluation of adversarial robustness via adaptive auto attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15105–15114.

[150] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. 2019. Feature distillation: DNN-oriented JPEG compression against adversarial examples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 860–868.

[151] Jiajun Lu, Theerasit Issaranon, and David Forsyth. 2017. SafetyNet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE International Conference on Computer Vision*. 446–454.

[152] Keane Lucas, Mahmood Sharif, Lujo Bauer, Michael K. Reiter, and Saurabh Shintre. 2021. Malware Makeover: Breaking ML-Based Static Analysis by Modifying Executable Bytes.

[153] Shiqing Ma and Yingqi Liu. 2019. NIC: Detecting adversarial samples with neural network invariant checking. In *Proceedings of the 26th Network and Distributed System Security Symposium (NDSS 2019)*.

[154] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613* (2018).

[155] Gabriel R. Machado, Ronaldo R. Goldschmidt, and Eugênio Silva. 2019. MultiMagNet: A Non-deterministic Approach based on the Formation of Ensembles for Defending against Adversarial Images. In *ICEIS (1)*. 307–318.

[156] Gabriel Resende Machado, Eugênio Silva, and Ronaldo Ribeiro Goldschmidt. 2020. Adversarial Machine Learning in Image Classification: A Survey Towards the Defender's Perspective. arXiv:2009.03728 [cs.CV].

[157] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083 [stat.ML]. https://arxiv.org/abs/1909.04068

[158] Saeed Mahloujifar and Mohammad Mahmoody. 2019. Can Adversarially Robust Learning Leverage Computational Hardness? 98 (2019), 581–609. https://proceedings.mlr.press/v98/mahloujifar19a.html

[159] Pratyush Maini, Eric Wong, and Zico Kolter. 2020. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*. PMLR, 6640–6650.

[160] Xiaofeng Mao, Yuefeng Chen, Shuhui Wang, Hang Su, Yuan He, and Hui Xue. 2020. Composite Adversarial Attacks. arXiv:2012.05434 [cs.CR].

[161] Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, and Jihun Hamm. 2021. How robust are randomized smoothing based defenses to data poisoning?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13244–13253.

[162] Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, and Jihun Hamm. 2021. Understanding the limits of unsupervised domain adaptation via data poisoning. *Advances in Neural Information Processing Systems* 34 (2021), 17347–17359.

[163] Dongyu Meng and Hao Chen. 2017. MagNet: A two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 135–147.

[164] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267* (2017).

[165] Laurent Meunier, Jamal Atif, and Olivier Teytaud. 2019. Yet another but more efficient black-box adversarial attack: Tiling and evolution strategies. arXiv:1910.02244 [cs.LG].

[166] Md. Ashraful Alam Milton. 2018. Evaluation of momentum diverse input iterative fast gradient sign method (M-DI2-FGSM) based attack method on MCS 2018 adversarial attacks on black box face recognition system. *arXiv preprint arXiv:1806.08970* (2018).

[167] Seungyong Moon, Gaon An, and Hyun Oh Song. 2019. Parsimonious Black-Box Adversarial Attacks via Efficient Combinatorial Optimization. arXiv:1905.06635 [cs.LG].

[168] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. arXiv:1610.08401 [cs.CV].

[169] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2574–2582. https://arxiv.org/abs/1707.05572

[170] Seyed-Mohsen Moosavi-Dezfooli, Ashish Shrivastava, and Oncel Tuzel. 2019. Divide, Denoise, and Defend against Adversarial Attacks. arXiv:1802.06806 [cs.CV].

[171] Konda Reddy Mopuri, Utsav Garg, and R. Venkatesh Babu. 2017. Fast feature fool: A data independent approach to universal adversarial perturbations. *arXiv preprint arXiv:1707.05572* (2017).

[172] John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. *arXiv preprint arXiv:2005.05909* (2020).

[173] Luis Muñoz-González, Bjarne Pfitzner, Matteo Russo, Javier Carnerero-Cano, and Emil C. Lupu. 2019. Poisoning attacks with generative adversarial nets. *arXiv preprint arXiv:1906.07773* (2019).

[174] Aamir Mustafa, Salman H. Khan, Munawar Hayat, Jianbing Shen, and Ling Shao. 2020. Image Super-Resolution as a Defense against Adversarial Attacks. *IEEE Transactions on Image Processing* 29 (2020), 1711–1724. https://doi.org/10.1109/tip.2019.2940533

[175] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. 2017. Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. arXiv:1708.08689 [cs.LG].

[176] Taesik Na, Jong Hwan Ko, and Saibal Mukhopadhyay. 2017. Cascade adversarial machine learning regularized with a unified embedding. *arXiv preprint arXiv:1708.02582* (2017). https://www.usenix.org/legacy/event/leet08/tech/full_papers/nelson/nelson_html/

[177] Elior Nehemya, Yael Mathov, Asaf Shabtai, and Yuval Elovici. [n. d.]. Taking Over the Stock Market: Adversarial Perturbations against Algorithmic Traders. ([n. d.]).

[178] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D. Joseph, Benjamin I. P. Rubinstein, Udam Saini, Charles A. Sutton, J. Doug Tygar, and Kai Xia. 2008. Exploiting Machine Learning to Subvert Your Spam Filter. *LEET* 8 (2008), 1–9.

[179] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, et al. 2018. Adversarial Robustness Toolbox v1. 0.0. *arXiv preprint arXiv:1807.01069* (2018).

[180] Mesut Ozdag. 2018. Adversarial attacks and defenses against deep neural networks: A survey. *Procedia Computer Science* 140 (2018), 152–161.

[181] Tianyu Pang, Chao Du, Yinpeng Dong, and Jun Zhu. 2017. Towards robust detection of adversarial examples. *arXiv preprint arXiv:1706.00633* (2017).

[182] Zoë Papakipos and Joanna Bitton. 2022. AugLy: Data Augmentations for Adversarial Robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 156–163.

[183] Nicolas Papernot and Patrick McDaniel. 2017. Extending defensive distillation. *arXiv preprint arXiv:1705.05264* (2017).

[184] Jungsoo Park, Gyuwan Kim, and Jaewoo Kang. 2021. Consistency training with virtual adversarial discrete perturbation. *arXiv preprint arXiv:2104.07284* (2021).

[185] Santiago Paternain, Juan Andrés Bazerque, Austin Small, and Alejandro Ribeiro. 2020. Stochastic policy gradient ascent in reproducing kernel Hilbert spaces. *IEEE Trans. Automat. Control* 66, 8 (2020), 3429–3444.

[186] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. 2017. Robust deep reinforcement learning with adversarial attacks. *arXiv preprint arXiv:1712.03632* (2017).

[187] Maura Pintor, Fabio Roli, Wieland Brendel, and Battista Biggio. 2021. Fast Minimum-norm Adversarial Attacks through Adaptive Norm Constraints. arXiv:2102.12827 [cs.LG].

[188] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. 2018. Deflecting Adversarial Attacks with Pixel Deflection. arXiv:1801.08926 [cs.CV].

[189] Shilin Qiu, Qihe Liu, Shijie Zhou, and Chunjiang Wu. 2019. Review of Artificial Intelligence Adversarial Attack and Defense Technologies. *Applied Sciences* 9, 5 (2019). https://doi.org/10.3390/app9050909

[190] Erwin Quiring, David Klein, Daniel Arp, Martin Johns, and Konrad Rieck. [n. d.]. Open access to the Proceedings of the 29th USENIX Security Symposium is sponsored by USENIX. Adversarial Preprocessing: Understanding and Preventing Image-Scaling Attacks in Machine Learning. https://www.usenix.org/conference/usenixsecurity20/presentation/quiring

[191] Erwin Quiring, David Klein, Daniel Arp, Martin Johns, and Konrad Rieck. 2020. Adversarial Preprocessing: Understanding and Preventing Image-Scaling Attacks in Machine Learning. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*.

[192] Adnan Siraj Rakin, Zhezhi He, Boqing Gong, and Deliang Fan. 2018. Blind Pre-Processing: A Robust Defense Method against Adversarial Examples. arXiv:1802.01549 [cs.LG].

[193] Miguel A. Ramirez, Song-Kyoo Kim, Hussam Al Hamadi, Ernesto Damiani, Young-Ji Byon, Tae-Yeon Kim, Chung-Suk Cho, and Chan Yeob Yeun. 2022. Poisoning Attacks and Defenses on Artificial Intelligence: A Survey. arXiv:2202.10276 [cs.CR].

[194] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D. Castillo, and Rama Chellappa. 2017. Improving network robustness against adversarial attacks with compact convolution. *arXiv preprint arXiv:1712.00699* (2017).

[195] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1085–1097.

[196] Gabriel Resende Machado, Eugênio Silva, and Ronaldo Ribeiro Goldschmidt. 2020. Adversarial Machine Learning in Image Classification: A Survey Towards the Defender's Perspective. *arXiv e-prints* (2020), arXiv–2009.

[197] Douglas A. Reynolds. 2009. Gaussian mixture models. *Encyclopedia of Biometrics* 741, 659-663 (2009). https://arxiv.org/abs/2007.02407

[198] Lior Rokach and Oded Maimon. 2005. Decision trees. In *Data Mining and Knowledge Discovery Handbook*. Springer, 165–192.

[199] Ishai Rosenberg, Asaf Shabtai, Yuval Elovici, and Lior Rokach. 2021. Adversarial machine learning attacks and defense methods in the cyber security domain. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–36.

[200] Andrew Slavin Ross and Finale Doshi-Velez. 2017. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients. arXiv:1711.09404 [cs.LG].

[201] Tom Roth, Yansong Gao, Alsharif Abuadbba, Surya Nepal, and Wei Liu. 2021. Token-modification adversarial attacks for natural language processing: A survey. *arXiv preprint arXiv:2103.00676* (2021).

[202] Binxin Ru, Adam Cobb, Arno Blaas, and Yarin Gal. 2019. BayesOpt adversarial attack. In *International Conference on Learning Representations*.

[203] Wenjie Ruan, Xinping Yi, and Xiaowei Huang. 2021. Adversarial Robustness of Deep Learning: Theory, Algorithms, and Applications. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4866–4869.

[204] Benjamin I. P. Rubinstein, Blaine Nelson, Ling Huang, Anthony D. Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J. D. Tygar. 2009. ANTIDOTE: Understanding and Defending against Poisoning of Anomaly Detectors. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement* (Chicago, Illinois, USA) *(IMC '09)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/1644893.1644895

[205] Sebastian Ruder. 2017. An overview of gradient descent optimization algorithms. arXiv:1609.04747 [cs.LG].

[206] Vivek B. S. and R. Venkatesh Babu. 2020. Single-step Adversarial training with Dropout Scheduling. arXiv:2004.08628 [cs.LG].

[207] Subhash Sagar, Chang-Sun Li, Seng W. Loke, and Jinho Choi. 2023. Poisoning Attacks and Defenses in Federated Learning: A Survey. *arXiv preprint arXiv:2301.05795* (2023).

[208] Raymel Alfonso Sallo, Mohammad Esmaeilpour, and Patrick Cardinal. 2021. Adversarially training for audio classifiers. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 9569–9576.

[209] Leo Schwinn, René Raab, and Björn Eskofier. 2020. Towards Rapid and Robust Adversarial Training with One-Step Attacks. arXiv:2002.10097 [cs.LG].

[210] Sailik Sengupta, Tathagata Chakraborti, and Subbarao Kambhampati. 2018. MTDeep: Boosting the security of deep neural nets against adversarial attacks with moving target defense. In *Workshops at the Thirty-second AAAI Conference on Artificial Intelligence*.

[211] Alexandru Constantin Serban, Erik Poll, and Joost Visser. 2018. Adversarial examples-a complete characterisation of the phenomenon. *arXiv preprint arXiv:1810.01185* (2018).

[212] Ali Shafahi, Amin Ghiasi, Furong Huang, and Tom Goldstein. 2019. Label Smoothing and Logit Squeezing: A Replacement for Adversarial Training? arXiv:1910.11585 [cs.LG].

[213] Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison frogs! Targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*. 6103–6113. https://dl.acm.org/doi/10.1145/2976749.2978392

[214] Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S. Davis, and Tom Goldstein. 2019. Universal Adversarial Training. arXiv:1811.11304 [cs.CV].

[215] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security*.

[216] Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. 2017. APE-GAN: Adversarial perturbation elimination with GAN. *arXiv preprint arXiv:1707.05474* (2017).

[217] Yucheng Shi, Yahong Han, Quanxin Zhang, and Xiaohui Kuang. 2020. Adaptive iterative attack towards explainable adversarial robustness. *Pattern Recognition* 105 (2020), 107309. https://doi.org/10.1016/j.patcog.2020.107309

[218] Ming-Wei Shih, Sangho Lee, Taesoo Kim, and Marcus Peinado. 2017. T-SGX: Eradicating controlled-channel attacks against enclave programs. In *NDSS*.

[219] Richard Shin and Dawn Song. 2017. JPEG-resistant adversarial images. In *NIPS 2017 Workshop on Machine Learning and Computer Security*, Vol. 1. 8.

[220] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.

[221] Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6, 1 (2019), 1–48.

[222] Osvaldo Simeone. 2018. A Very Brief Introduction to Machine Learning with Applications to Communication Systems. arXiv:1808.02342 [cs.IT].

[223] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. 2019. FineGAN: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6490–6499.

[224] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. 2018. PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples. arXiv:1710.10766 [cs.LG].
[225] Thilo Strauss, Markus Hanselmann, Andrej Junginger, and Holger Ulmer. 2017. Ensemble methods as a defense to adversarial perturbations against deep neural networks. *arXiv preprint arXiv:1709.03423* (2017).
[226] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* 23, 5 (Oct. 2019), 828–841. https://doi.org/10.1109/tevc.2019.2890858
[227] Richard S. Sutton and Andrew G. Barto. 2018. Reinforcement learning: An introduction. (2018).
[228] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going Deeper with Convolutions. arXiv:1409.4842 [cs.CV].
[229] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. arXiv:1312.6199 [cs.CV].
[230] Luke Taylor and Geoff Nitschke. 2018. Improving deep learning with generic data augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1542–1547.
[231] Rajkumar Theagarajan, Ming Chen, Bir Bhanu, and Jing Zhang. 2019. ShieldNets: Defending against adversarial attacks using probabilistic adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
[232] Shixin Tian, Guolei Yang, and Ying Cai. 2018. Detecting adversarial examples through image transformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
[233] Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. 2022. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *Comput. Surveys* 55, 8 (2022), 1–35.
[234] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. 2020. On Adaptive Attacks to Adversarial Example Defenses. arXiv:2002.08347 [cs.LG].
[235] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction APIs. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*. 601–618.
[236] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble Adversarial Training: Attacks and Defenses. https://doi.org/10.48550/ARXIV.1705.07204
[237] Shivesh Tripathi, B. Mohapatra, Prabhakar Tiwari, and V. S. Tripathi. 2021. Multi-mode resonator based concurrent triple-band band pass filter with six transmission zeros for defence/intelligent transportation systems application. *Defence Science Journal* 71, 3 (2021).
[238] Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aaron Oord. 2018. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*. PMLR, 5025–5034.
[239] Shivakumar Vaithyanathan and Byron E. Dom. 2013. Model-based hierarchical clustering. *arXiv preprint arXiv:1301.3899* (2013).
[240] Eric Wallace, Tony Z. Zhao, Shi Feng, and Sameer Singh. 2021. Concealed Data Poisoning Attacks on NLP Models. arXiv:2010.12563 [cs.CL].
[241] Jianyu Wang and Haichao Zhang. 2019. Bilateral Adversarial Training: Towards Fast Training of More Robust Models against Adversarial Attacks. arXiv:1811.10716 [cs.CV].
[242] Ling Wang, Cheng Zhang, Zejian Luo, Chenguang Liu, Jie Liu, Xi Zheng, and Athanasios Vasilakos. 2020. Progressive Defense against Adversarial Attacks for Deep Learning as a Service in Internet of Things. arXiv:2010.11143 [cs.CR].
[243] Sun-Chong Wang. 2003. Artificial neural network. (2003), 81–100.
[244] Xiaosen Wang, Hao Jin, Yichen Yang, and Kun He. 2019. Natural Language Adversarial Defense through Synonym Encoding. https://doi.org/10.48550/ARXIV.1909.06723
[245] Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2021. Adversarial training with fast gradient projection method against synonym substitution based text attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 13997–14005.
[246] Yu Wang, Luca Bondi, Paolo Bestagini, Stefano Tubaro, David J. Edward Delp, et al. 2017. A counter-forensic method for CNN-based camera model identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 28–35.
[247] Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. 2020. An Investigation of Data Poisoning Defenses for Online Learning. arXiv:1905.12121 [cs.LG].
[248] Sean Weerakkody, Omur Ozel, and Bruno Sinopoli. 2017. A Bernoulli-Gaussian physical watermark for detecting integrity attacks in control systems. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 966–973.
[249] Sandamal Weerasinghe, Tansu Alpcan, Sarah M. Erfani, and Christopher Leckie. 2020. Defending Distributed Classifiers against Data Poisoning Attacks. https://doi.org/10.48550/ARXIV.2008.09284

[250] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. 2014. Natural evolution strategies. (2014).

[251] Eric Wong, Leslie Rice, and J. Zico Kolter. 2020. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994* (2020).

[252] Eric Wong, Frank R. Schmidt, and J. Zico Kolter. 2019. Wasserstein adversarial examples via projected Sinkhorn iterations. *arXiv preprint arXiv:1902.07906* (2019).

[253] Nils Worzyk and Stella Yu. 2021. Broad adversarial training with data augmentation in the output space. In *The AAAI-22 Workshop on Adversarial Machine Learning and Beyond*.

[254] Huimin Wu, Zhengmian Hu, and Bin Gu. 2021. Fast and Scalable Adversarial Training of Kernel SVM via Doubly Stochastic Gradients. https://doi.org/10.48550/ARXIV.2107.09937

[255] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. 2018. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612* (2018).

[256] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2017. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991* (2017).

[257] Han Xu, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang Tang, and Anil K. Jain. 2019. Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. arXiv:1909.08072 [cs.LG].

[258] Xin Yan and Xiaogang Su. 2009. Linear regression analysis: Theory and computing. (2009).

[259] Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu. 2019. ME-Net: Towards effective adversarial robustness with matrix estimation. *arXiv preprint arXiv:1905.11971* (2019).

[260] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2019. Defending against saddle point attack in Byzantine-robust distributed learning. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 7074–7084.

[261] Zhengwei Yin, Kaoru Uchida, and Shiping Deng. 2021. Improving adversarial attacks on face recognition using a modified image translation model. In *2021 3rd International Conference on Image, Video and Signal Processing*. 26–31.

[262] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. 2021. Adversarial purification with score-based generative models. (2021).

[263] Chong Yu. 2020. Attention based data hiding with generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1120–1128.

[264] Matthew Yuan, Matthew Wicker, and Luca Laurenti. 2020. Gradient-Free Adversarial Attacks for Bayesian Neural Networks. arXiv:2012.12640 [cs.LG].

[265] Ruixi Yuan, Zhu Li, Xiaohong Guan, and Li Xu. 2010. An SVM-based machine learning method for accurate internet traffic classification. *Information Systems Frontiers* 12, 2 (2010), 149–156.

[266] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2018. Adversarial Examples: Attacks and Defenses for Deep Learning. arXiv:1712.07107 [cs.LG].

[267] Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. 2017. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 39–49.

[268] Huimin Zeng, Jiahao Su, and Furong Huang. 2021. Certified Defense via Latent Space Randomized Smoothing with Orthogonal Encoders. https://doi.org/10.48550/ARXIV.2108.00491

[269] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In-So Kweon. 2020. CD-UAP: Class discriminative universal adversarial perturbation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6754–6761.

[270] Chaoning Zhang, Philipp Benz, Chenguo Lin, Adil Karjauv, Jing Wu, and In So Kweon. 2021. A survey on universal adversarial attack. *arXiv preprint arXiv:2103.01498* (2021).

[271] Hengtong Zhang, Tianhang Zheng, Jing Gao, Chenglin Miao, Lu Su, Yaliang Li, and Kui Ren. 2019. Data Poisoning Attack against Knowledge Graph Embedding. arXiv:1904.12052 [cs.LG].

[272] Mengmei Zhang, Linmei Hu, Chuan Shi, and Xiao Wang. 2020. Adversarial label-flipping attack and defense for graph neural networks. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 791–800.

[273] Chenchen Zhao and Hao Li. 2020. Blurring Fools the Network – Adversarial Attacks by Feature Peak Suppression and Gaussian Blurring. https://doi.org/10.48550/ARXIV.2012.11442

[274] Zhihao Zheng and Pengyu Hong. 2018. Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 7924–7933.

[275] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2017. Random Erasing Data Augmentation. https://doi.org/10.48550/ARXIV.1708.04896

[276] Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-wei Chang, and Xuanjing Huang. 2020. Defense against Adversarial Attacks in NLP via Dirichlet Neighborhood Ensemble. https://doi.org/10.48550/ARXIV.2006.11627
[277] Chen Zhu, W. Ronny Huang, Ali Shafahi, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2019. Transferable Clean-Label Poisoning Attacks on Deep Neural Nets. arXiv:1905.05897 [stat.ML]

Received 14 September 2022; revised 14 July 2023; accepted 26 September 2023