ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

ΔΙΕΠΙΣΤΗΜΟΝΙΚΟ – ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

Data Vulnerabilities and Adversarial Attacks against ML-Based Systems. The Adversarial Risk in the Healthcare Domain.

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Ι. Παπατσίμπας

Επιβλέπων : Νικόλαος Δ. Δουλάμης
Καθηγητής Τομέα Τοπογραφίας
Σχολή Αγρονόμων και Τοπογράφων Μηχανικών –
Μηχανικών Γεωπληροφορικής

Αθήνα, Οκτώβριος 2023

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

ΔΙΕΠΙΣΤΗΜΟΝΙΚΟ – ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

Data Vulnerabilities and Adversarial Attacks against ML-Based Systems. The Adversarial Risk in the Healthcare Domain.

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Ι. Παπατσίμπας

Επιβλέπων : Νικόλαος Δ. Δουλάμης
Καθηγητής Τομέα Τοπογραφίας
Σχολή Αγρονόμων και Τοπογράφων Μηχανικών - Μηχανικών Γεωπληροφορικής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19η Οκτωβρίου 2023.

Νικόλαος Δ. Δουλάμης	**Αναστάσιος Δ. Δουλάμης**	**Θεοδώρα Βαρβαρίγου**
Καθηγητής Τομέα Τοπογραφίας Σχολή Αγρονόμων και Τοπογράφων Μηχανικών – Μηχανικών Γεωπληροφορικής	**Αν. Καθηγητής Τομέα Τοπογραφίας Σχολή Αγρονόμων και Τοπογράφων Μηχανικών – Μηχανικών Γεωπληροφορικής**	**Καθηγήτρια Τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών**

Αθήνα, Οκτώβριος 2023

Δημήτριος Ι. Παπατσίμπας

Πτυχιούχος του Τμήματος Πληροφορικής της Σχολής Τεχνολογικών Εφαρμογών του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Αθήνας.

Αθήνα, Οκτώβριος 2023

Περίληψη

Η **Τεχνητή Νοημοσύνη** (**ΤΝ**) αν και έχει παρουσιαστεί ως πεδίο το 1958 από τον John McCarthy, ήταν μόλις το 2006 όταν σε συνδυασμό με την αύξηση της υπολογιστικής ισχύος εμφανίστηκαν νέες εφαρμογές στο πεδίο της Μηχανικής Μάθησης (ΜΜ). Ωστόσο, η άνθιση αυτή πραγματώθηκε αντίστοιχα και στο πεδίο των επιθέσεων, καθιστώντας την ανάπτυξη συστημάτων που βασίζονται στο πεδίο της ΤΝ επισφαλή για την εφαρμογή τους σε κρίσιμες δημόσιες υποδομές, χωρίς τη λήψη αυστηρών μέτρων ασφάλειας.

Η **ευπάθεια των δεδομένων** και οι απειλές ενάντια σε αυτά, συνιστούν ένα επιστημονικό πεδίο ανοικτό για περαιτέρω εξερεύνηση. Ειδικότερα, από τη σε βάθος μελέτη επιθέσεων οι οποίες αναπτύσσονται σε όλο τον κύκλο ζωής των συστημάτων Τεχνητής Νοημοσύνης (εκπαίδευση, εξαγωγή συμπερασμάτων), όπως οι **κακόβουλες επιθέσεις**, οι «**αντιπαραθετικές**» οι οποίες υλοποιούνται με ανεπαίσθητες τροποποιήσεις των δεδομένων εισόδου, καθώς και άλλες σύγχρονες, θα ήταν δυνατό να προκύψει μία νέα οπτική στον τομέα αυτό.

Επιπλέον, οι επιθέσεις αυτές είναι άρρηκτα συνδεδεμένες με την ανάπτυξη αντίστοιχων μέτρων άμβλυνσης των επιπτώσεών τους, με αποτέλεσμα αυτός ο συνεχής ανταγωνισμός να δημιουργεί ένα γόνιμο και ταυτόχρονα «αντιπαραθετικό» περιβάλλον. Για το λόγο αυτό κρίνεται σκόπιμο να αναφερθούν διάφοροι **αντιπροσωπευτικοί μηχανισμοί άμυνας**, ώστε να επισημανθεί η ροή που έχει ακολουθήσει η επιστημονική κοινότητα και να δημιουργήσει στέρεες βάσεις για μελλοντικές κατευθύνσεις.

Κρίσιμες περιοχές όπως της υγείας, της αυτόνομης οδήγησης, της ταξινόμησης, της αναγνώρισης φωνής, της ασφάλειας δικτύων, οι οποίες έχουν στον πυρήνα τους την ανθρώπινη ζωή, αντιμετωπίζουν αντίστοιχα προβλήματα. Στην παρούσα εργασία, αν και θα δοθεί ιδιαίτερη έμφαση στον τομέα της υγείας, αυτό δεν συνεπάγεται ότι θα αποκλίνει σημαντικά και από τα υπόλοιπα πεδία ενδιαφέροντος.

Λέξεις Κλειδιά: Poisonous Attacks, Backdoor Attacks, Adversarial Examples, Defense Mechanisms, Healthcare Systems.

Abstract

Although **Artificial Intelligence** (**AI**) was introduced as a new discipline by John McCarthy in 1958, it was not until 2006 along with computational power that new applications ushered in the Machine Learning (ML) domain. This flourish went along with counterpart attacks producing eventually uncertainties that set AI-related systems precarious to deploy in crucial public infrastructure, without taking strict security measures against them.

Data vulnerabilities and threats comprise a scientific area viable to further scrutiny in terms of attacks deploying in all the gamut of the Artificial Intelligence systems pipeline (training, inference, ai integration), such as **poisonous attacks**, imperceptible perturbations namely **adversarial examples** as well as other contemporary ones, and thus studying the latest topics could highlight a new insight.

Besides, these attacks are tightly coupled with corresponding mitigation measures, continuously going back-and-forth creating an "adversarial" environment. Thus, latest **representative defense mechanisms** will be presented in order to enlighten the roadmap that research community follows, grasping the rhythm for future exploration.

Critical areas such as healthcare, autonomous driving, classification, speech recognition, network security, where human life is of utmost importance, raise such security issues. Though, special consideration is to be taken to the healthcare domain in this thesis, it will still be attached no further than the other fields.

Keywords: Poisonous Attacks, Backdoor Attacks, Adversarial Examples, Defense Mechanisms, Healthcare Systems.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my postgraduate professors for all the knowledge provided during the last two years. Especially professor Mr. Nikolaos Doulamis for his confidence and the suggestions provided according to my own background. My gratitude is also addressed to the members of the committee for the honor participating in the examination process as well as my PhD advisor Stavros Sykiotis for his guidance to accomplish this thesis.

Additionally, I am extremely grateful to my wife Leda, without her continuous encouragement and devotion, this effort would not have been accomplished. My children Anastasia and Olga (6 and 3 years old) for the love reflected in their eyes, the joy and patience all this time. Last but not least my parents Ioannis and Anastasia for their great support during this period.

I would like also to thank my friends and classmates of this programme for all the conversations and the exchange of opinions. Especially, Candidate PhD NTUA, Georgios Tsavdaridis for all his pertinent advices and interest.

Contents

List of Figures

List of Tables

Εκτεταμένη Ελληνική Περίληψη

Η αύξηση της υπολογιστικής ισχύος σε συνδυασμό με την ευρεία αποδοχή των μικρών-κινητών συσκευών προσέδωσε στο πεδίο της Μηχανικής Μάθησης (ΜΜ) μία νέα δυναμική. Η χρήση όλων των συσκευών αυτών με ενσωματωμένες λύσεις ΜΜ, δημιουργεί πλέον τεράστιο όγκο δεδομένων, ο οποίος απαιτεί βέλτιστες λύσεις τόσο ως προς την επεξεργασία τους, όσο και ως προς την εξαγωγή χρήσιμων συμπερασμάτων. Την πρόοδο αυτή ακολούθησε και η ανάπτυξη αντίστοιχων επιθέσεων, με σκοπό κυρίως οικονομικά και πολιτικά οφέλη. Αν και η ανάπτυξη έξυπνων συστημάτων συμβάλλει στη βελτίωση κάθε ανθρώπινης πτυχής, οι συνέπειες και οι κίνδυνοι γενικότερα από τις επιθέσεις εναντίον τους, τα καθιστούν επισφαλή για την εφαρμογή τους σε κρίσιμες δημόσιες υποδομές, χωρίς την εφαρμογή των αυστηρών μέτρων ασφαλείας.

Η **ευπάθεια των δεδομένων** και οι **απειλές** ενάντια σε αυτά, συνιστούν ένα επιστημονικό πεδίο ανοικτό για περαιτέρω εξερεύνηση. Ειδικότερα, από την σε βάθος μελέτη επιθέσεων, οι οποίες αναπτύσσονται κατά τις φάσεις του **κύκλου ζωής** των **συστημάτων Τεχνητής Νοημοσύνης** (συλλογή και προ-επεξεργασία δεδομένων, εκπαίδευση μοντέλου, εξαγωγή συμπερασμάτων), θα ήταν δυνατό να προκύψει μία νέα οπτική στον τομέα αυτό. Η αναγκαιότητα της μελέτης κάθε επίθεσης και των ειδικότερων χαρακτηριστικών της ενισχύεται από το γεγονός ότι δεν υπάρχει μία και μόνον λύση, παρά μόνο μέτρα αποτροπή τους για την κάθε περίπτωση.

Η ίδια τάση καταδεικνύεται και από το επιστημονικό ενδιαφέρον το οποίο εκδηλώνεται μέσα από την ραγδαία αύξηση των συγγραφικών έργων, για τις επιθέσεις πάνω στα συστήματα ΜΜ. Μεγάλο εύρος επιθέσεων καταγράφεται σε κάθε στάδιο του κύκλου ζωής ενός συστήματος ΜΜ και αντίστοιχες ενέργειες για τον περιορισμό τους. Ωστόσο, στην παρούσα εργασία αφού πρωτίστως αναφερθούν ορισμένα στοιχεία που συνθέτουν το υπόβαθρο των επιθέσεων (manifolf, norms, optimization), το ενδιαφέρον θα εστιάσει στις **κακόβουλες επιθέσεις** (**poisonous attacks**), στις «**αντιπαραθετικές**» (**adversarial examples**) οι οποίες υλοποιούνται με ανεπαίσθητες τροποποιήσεις των δεδομένων εισόδου, καθώς και στις **backdoor attacks**.

Το σύνολο των επιθέσεων αυτών αν και παρουσιάζει αρκετά κοινά χαρακτηριστικά, τουλάχιστον εκ πρώτης όψεως, εντούτοις διαφέρει ουσιωδώς. Η βιβλιογραφία καταγράφει πλήθος διάφορων παραγόντων ταξινόμησης των επιθέσεων αυτών, οι οποίοι προσδίδουν κάθε φορά μία διαφορετική οπτική. Η οριοθέτησή τους ανάλογα με τον τρόπο και το στάδιο του κύκλου ζωής στο οποίο εκδηλώνονται, την γνώση του επιτιθέμενου για το σύστημα και τους στόχους που θέτει, συμβάλλει στο μέγιστο στην αποκρυστάλλωσή τους, με σκοπό την βέλτιστη αντιμετώπισή τους.

Μία τέτοια ταξινόμηση μπορεί να πραγματοποιηθεί και μέσα από το μοντέλο Confidentiality (Εμπιστευτικότητα), Integrity (Ακεραιότητα) και Availability (Διαθεσιμότητα) (CIA), το οποίο συμβάλει στην πιο εύκολη κατανόηση των διαφορών. Τελευταία, στο μοντέλο αυτό έχει προστεθεί και μία επιπλέον πτυχή, η Privacy (Ιδιωτικότητα). Στην περίπτωση της εμπιστευτικότητας ο δράστης επιδιώκει να αποκαλύψει στοιχεία του συστήματος (π.χ. υπερπαράμετροι) ή στην περίπτωση που αποκαλύπτονται δεδομένα ενός συστήματος, όπως στοιχεία ασθενών, αναφερόμαστε σε παραβίαση της Ιδιωτικότητας. Σχετικά με την ακεραιότητα, ο δράστης στοχεύει στην παραγωγή μη αναμενόμενων αποτελεσμάτων για συγκεκριμένα δείγματα, ενώ αντιθέτως, όταν ο δράστης στοχεύει γενικά στη λειτουργία ενός συστήματος, παραβιάζεται η διαθεσιμότητα αυτού.

Ένα επιπλέον σημαντικό σημείο διάκρισης των επιθέσεων, αποτελεί το σημείο στο οποίο αναπτύσσονται με σημείο αναφοράς τον κύκλο ζωής ενός συστήματος ΜΜ. Ειδικότερα, οι επιθέσεις που θα αναφερθούν στο πλαίσιο της παρούσας εργασίας αναπτύσσονται κατά τα στάδια της εκπαίδευσης και της εξαγωγής συμπερασμάτων. Άλλα στοιχεία των επιθέσεων μπορούν ομοίως να δημιουργήσουν συνθήκες ταξινόμησης αυτών, όπως ο αριθμός των επαναλήψεων για την υλοποίηση μιας επίθεσης (one-shot, iterative), οι δυνατότητες που έχει ο επιτιθέμενος, καθώς και εάν η επίθεση θέτει κάποιο συγκεκριμένο στόχο ή μη (targeted, indiscriminate).

Οι κακόβουλες επιθέσεις εκδηλώνονται κατά το στάδιο της εκπαίδευσης ενός συστήματος ΜΜ και στοχεύουν γενικά στη μείωση της επίδοσης ενός συστήματος, το οποίο μπορεί να εξομοιωθεί και με Denial of Service. Ο δράστης με διάφορες μεθόδους (π.χ. αλλαγή ετικετών) μπορεί να εισαγάγει λανθασμένα ή μολυσμένα δεδομένα εκπαίδευσης με αποτέλεσμα το σύστημα να μην μπορεί να λειτουργήσει ή να παράγει τα αναμενόμενα αποτελέσματα. Βασικό μειονέκτημα των επιθέσεων αυτών είναι η

εύκολη αναγνώρισή τους εξαιτίας της γενίκευσης των μη αναμενόμενων αποτελεσμάτων και της εν γένει μη διαθεσιμότητας του συστήματος.

Τα αντιπαραθετικά παραδείγματα είναι δείγματα στα οποία έχει προστεθεί ένας μικρός «θόρυβος» με τέτοιο τρόπο ώστε να μην αλλοιώνεται το περιεχόμενο τους. Στην περίπτωση της εικόνας διατηρείται η σημασιολογία του περιεχομένου της, ενώ η οποιαδήποτε παραβίαση του συστήματος δεν είναι αντιληπτή από το ανθρώπινο μάτι. Τα μολυσμένα αυτά στοιχεία εισάγονται κατά το στάδιο της εξαγωγής συμπερασμάτων ενός συστήματος ΜΜ και παράγουν εσφαλμένα αποτελέσματα. Δεν αποτελούν τυχαία γεγονότα, αλλά δημιουργούνται αλγοριθμικά λαμβάνοντας κυρίως υπόψη την είσοδο των δεδομένων. Ο δράστης επιδιώκει κυρίως την παραβίαση της ακεραιότητας ενός συστήματος.

Οι Backdoor επιθέσεις υλοποιούνται κατά το στάδιο εκπαίδευσης ενός συστήματος ΜΜ, ωστόσο εκδηλώνονται κατά το στάδιο της εξαγωγής συμπερασμάτων. Ειδικότερα, ο δράστης εισάγει ένα δείγμα με εμφωλευμένο ένα πρότυπο (π.χ. κάποιο συνδυασμό εικονοστοιχείων) κατά το στάδιο της εκπαίδευσης. Όταν εμφανιστεί ένα δείγμα με αυτό το συγκεκριμένο πρότυπο κατά το στάδιο των συμπερασμάτων, θα παραχθούν τα αποτελέσματα που επιθυμεί ο δράστης. Τέτοιου είδους επιθέσεις είναι αρκετά δύσκολο να εντοπισθούν, δεδομένου ότι το σύνολο των υπόλοιπων δειγμάτων παράγει τα αναμενόμενα αποτελέσματα. Ο δράστης βλάπτει με αυτόν τον τρόπο την ακεραιότητα ενός συστήματος.

Στο σύνολό τους οι επιθέσεις αυτές είναι άρρηκτα συνδεδεμένες με την ανάπτυξη αντίστοιχων μέτρων αντιμετώπισής τους ή άμβλυνσης των επιπτώσεών τους. Θα μπορούσαμε να ισχυριστούμε ότι κυρίως λόγω της αμφισβήτησης των αιτιών δημιουργίας τέτοιων επιθέσεων, η επιστημονική κοινότητα αναπτύσσει περιπτώσεις επιθέσεων εντοπίζοντας αδυναμίες με σκοπό την πρόκληση μέτρων για την αντιμετώπισή τους. Ως εκ τούτου, προκαλείται ένας συνεχής ανταγωνισμός, ο οποίος περαιτέρω δημιουργεί ένα γόνιμο και ταυτόχρονα «αντιπαραθετικό» περιβάλλον.

Επιπλέον, στο πλαίσιο της παρούσας εργασίας κρίνεται σκόπιμο να αναφερθούν διάφοροι **αντιπροσωπευτικοί μηχανισμοί άμυνας**, καθώς και οι ευπάθειες τις οποίες θεραπεύουν, ώστε αφενός να επισημανθεί η ροή που έχει ακολουθήσει η επιστημονική κοινότητα, αφετέρου να δημιουργήσει στέρεες βάσεις για τις όποιες μελλοντικές

προοπτικές. Συγκεκριμένα, θα αναφερθούν δύο αμυντικοί μηχανισμοί ενάντια στα adversarial examples και τις backdoor επιθέσεις. Ο μεν πρώτος χρησιμοποιεί εικόνες στις οποίες έχει εμφωλευθεί ένα πρότυπο (σαν τις backdoor επιθέσεις) ώστε να δημιουργεί μια ισχυρή συσχέτιση και να μην επηρεάζεται το σύστημα από τα αντιπαρατιθέμενα παραδείγματα. Στη δεύτερη περίπτωση, ο μηχανισμός άμυνας των backdoor επιθέσεων χρησιμοποιεί διάφορες τεχνικές για την εύρεση του επιβλαβούς προτύπου που έχει επιλέξει ένας δράστης, καθώς και επιπλέον άλλες όπως machine unlearning για την επανεκπαίδευση του συστήματος.

Κρίσιμες περιοχές όπως της υγείας, της αυτόνομης οδήγησης, της ταξινόμησης εικόνων, της αναγνώρισης φωνής, της ασφάλειας δικτύων, οι οποίες έχουν στον πυρήνα τους την ανθρώπινη ζωή, αντιμετωπίζουν προκλήσεις σε θέματα ασφάλειας και ιδιωτικότητας. Στην παρούσα εργασία δίνεται ιδιαίτερη έμφαση στον τομέα της υγείας, υπό το πρίσμα μίας μη τεχνικής παρουσίασης των προκλήσεων, δεδομένου ότι αποτελεί βασική υποδομή σε κάθε κράτος δικαίου. Ωστόσο, αυτό δεν συνεπάγεται ότι θα αποκλίνει σημαντικά και από τα υπόλοιπα πεδία ενδιαφέροντος.

Ως προς τα κίνητρα των δραστών να βλάψουν συστήματα τέτοιας δυναμικής και ωφέλειας για την ανθρώπινη ζωή, είναι σημαντικό να αναγνωρισθούν, προκειμένου να γίνει σαφές το μέγεθος της απειλής που αυτά επιδιώκουν. Πρόσφατες μελέτες καταγράφουν διάφορα κίνητρα τα οποία μπορεί να ποικίλουν κλιμακωτά, από απλή περιέργεια, οικονομικά οφέλη λειτουργώντας για τρίτους, έως και ζητήματα τρομοκρατίας.

Συμπερασματικά, η εργασία αυτή αποσκοπεί να παρουσιάσει βασικές έννοιες για την κατανόηση της λειτουργίας των συστημάτων ΜΜ, τις επιθέσεις που λαμβάνουν χώρα κατά τα στάδια της εκπαίδευσης και εξαγωγής συμπερασμάτων, καθώς και τις προκλήσεις που αντιμετωπίζει η επιστημονική κοινότητα για την αντιμετώπισή τους. Περαιτέρω, στοχεύει να αναδείξει την αναγκαιότητα σύγκλισης των διαφόρων επιστημονικών πεδίων προκειμένου να διερευνηθούν τα αίτια των επιθέσεων αυτών, καθώς και να κινητοποιήσει όσους χρησιμοποιούν τέτοια συστήματα (επαγγελματίες των αντίστοιχων τομέων, διευθύνσεις οργανισμών), να λαμβάνουν αντίστοιχα μέτρα ασφάλειας.

1. Introduction and Background

Machine Learning (ML) made solutions possible for hard problems. They introduced a new ability to acquire knowledge from their environment with less human intervention suppressing the need for hard-coded problems. A further step took place eliminating even more the human intervention allowing ML systems extracting features on their own, and thus Deep Learning (DL) scientific field emerged.

Many terminologies have been appointed to these newly aged systems, such as Deep feedforward Networks, Feedforward neural networks or multilayer perceptrons, but the fact is that “network” derives from their representation and “deep” comes from the depth of the hidden layers of the network. The depth and number of neurons allow systems to automate the extraction of the desired number of features.

It has been almost a decade since convolutional networks, a special class of Neural Networks, have achieved a significant performance on recognition of objects, matching almost a human-level performance [1]. Thus, research communities and not only, have drawn their attention to the field. However promising might be this development, counter-attacks are also part and follow-up situation.

In our data-saturated era, big data hold a prominent position in almost every public and private infrastructure. Along with the course of data, the technological advancements (Machine Learning as a Service) and the expertise (data-driven approaches) offer new harnessing methods. Therefore, the necessity of using specialized third-parties ML services, in terms of pre-trained models, datasets, frameworks is skyrocketing. Thus, studying the security risks and measures of mitigation is of utmost importance.

Most notably, a survey on the topic of ML security concerns in terms of tactical and strategic tools to shelter their core business functionality unveils surprising findings [2]. In particular among twenty-eight organizations in their majority dealing with security-sensitive data, only three of them declared affirmative on securing the Machine Learning systems, whereas attacks such as poisoning and backdoor, as well as adversarial examples are in the top five of most affective attacks.

The main scope of this thesis is to delve into the most representative and state-of-the-art attacks taking place across the ML pipeline, emphasizing in the training and

inference stages. Consequently, a cutting-edge defense mechanism will be presented, in order to face the challenges, the research community deals with in this rival environment. Finally, a reference to the healthcare domain will highlight the innermost posing risks. Hopefully, a refreshing review of this topic will enlighten the feature research directions, but most importantly it will raise awareness to the engaged communities.

1.1. Delineation of Attacks

As we will clearly show in this thesis, three cornerstone types of attack reign over ML based systems, and that is poisonous attacks, adversarial examples and backdoor attacks. All of them have features in common, but they are inherently different in terms of the adversary's goals and deployment tactics.

Although literature often reports backdoor attacks as a branch of poisonous attacks, this is obfuscated and a more meticulous view on this aspect will discern the differences between them. Thus, we intentionally follow the literature that considers these two attacks distinct, in order to provide a clearer aspect on the field of ML-based systems attacks.

By and large poisonous attacks aim at a more general degradation of a system, causing denial of service and hence availability issues are in concern. In contrast, backdoor attacks remain idle for the benign samples, but the insidious trigger will be invoked when he meets the specific pattern as an input to the ML-model and will cause its malicious purpose. Thus, backdoor attacks aim to harm integrity of a system.

Adversarial examples have been thoroughly studied by the research community. They are purely algorithmically produced based on the input and thus are considered barely incidental facts. Beneath those crafted inputs, mathematical models stand for their development.

Thus, it is important to present these attacks in a more concise view [Table 1], that will help the reader of this thesis to better understand these attacks, how do they accomplish their malicious purpose and their overall aim.

Table 1: Comparison Table for Poisonous Attacks - Adversarial Examples - Backdoor Attacks

Type of Attack	Adversary's Target	Kind of Attack	Concealment	Phase of ML-based system pipeline	Strong abilities
Poisonous Attacks	Availability General low degradation	Untargeted	Easy to be discovered	Training Phase	No trigger
Adversarial Examples	Integrity Preserve benign samples	Targeted	Hard to be discovered	Inference Phase	Imperceptible crafted inputs invisible to human eye
Backdoor Attacks	Integrity Preserve benign samples	Targeted	Hard to be discovered	Training Phase	Trigger invoked

1.2. Threat Model

Causes of adversarial examples existence are long being debated, and thus delineating their deployment into a model's perspective produces proper guidance for the researchers of the field. A meticulously illustrated study of this phenomenon in terms of the attacker's goals, capabilities and other features, is essential to be bound under various disciplines. Thus, setting all these factors into a mold, namely threat model or the attacker's profile according to [3], provides an overview of security issues, as well as the defensive mechanisms against them.

A great deal of work has been recorded on this issue. [4] introduced a threat model so as to encapsulate a ML system's components into a unity and thus ML-algorithm was not considered apart. Security and privacy issues of their model has been examined through the classical perspective of confidentiality, integrity and availability (CIA), and the lifecycle of ML system -namely "machine learning pipeline".

Confidentially is considered in respect to the model (model structure, architecture) and its training data in which case privacy issues are also concerned. Integrity deals mostly with the outcome of a ML-system. Violation of integrity may produce false positives

or true negatives. Finally, expanding the notion of this violation into preventing totally the access to system, availability concerns are raised.

Authors continue with the second perspective of "machine learning pipeline", starting from the training phase, ending up to the inference one. At each phase security issues are mentioned in terms of the attacker's goals and capabilities, under the view of CIA model and with one more feature, that of "privacy". Their threat model for ML systems consists of: "The Attack Surface", "Adversarial Capabilities", "Adversarial Goals".

1.2.1. The Attack Surface

The Attack Surface describes the places where an attack happens and generally is efficiently depicted as a row containing all phases [**Error! Reference source not found.**] of a ML system. That includes the data collection, the process of them, the production of the output and finally the integration of outcome with an external actor. At each phase an attacker may deploy an attack with different modus operandi [**Figure 8**]. Primary steps, such as collecting data are vulnerable to poisonous data, while the last ones -at inference- may be affected to produce erroneous results.

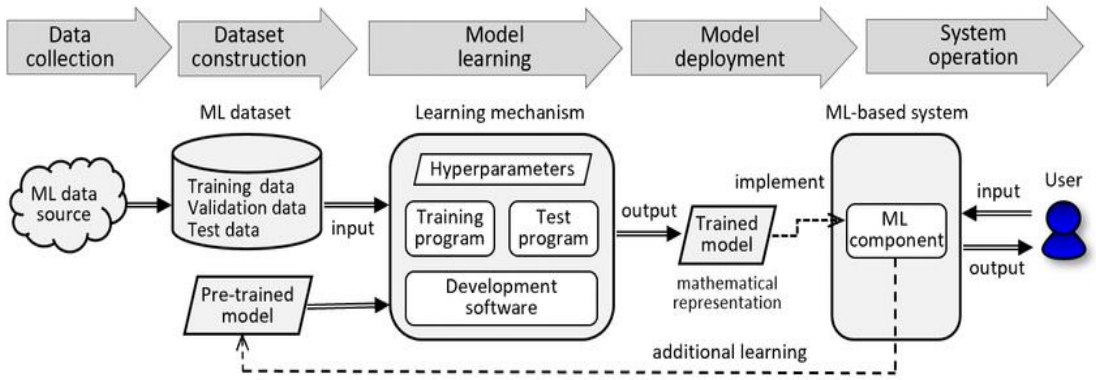


***Figure 1**:Basic Components in Supervised Learning and an ML-Based System's Lifecycle. Image Credited to [5]*

1.2.2. Adversarial Capabilities

This aspect of threat model refers to the adversary and the knowledge have on their side, with regards to the offensive system. The attacker may be familiar with internal information of the system, such as the structure and the parameters of the network architecture, with the intension to corrupt it (integrity attack). On the contrary, if the adversary is an external actor, they will most probably try not to affect the system directly, but to alter the outcomes.

[6] classify an attacker's capabilities in terms of training data and network architecture, "oracle" and samples. Access to layers, activation functions of neurons, and weights produced after the training phase, may be strong enough knowledge to replicate the deep neural network. Creating a sub-dataset of the training data, with the same distribution could result in an approximation of the model. Furthermore, observing the outputs in accordance to the change of inputs, an attacker may create adversarial attacks. Such pairs (inputs – outputs) could be also greatly helpful if exist in large amounts, even without the possibility of altering the input.

1.2.3. Adversarial Goals

This view deals with the attacker's behavior against the ML system. Confidentiality and privacy aspects are violated if the attacker tries to extract information about the system (e.g., architecture) or its training data (e.g., patient data). Otherwise, if the attacker tries to cause the system to produce erroneous output (e.g., misclassification), integrity issues are raised. Availability concerns raise when the system fails to respond on some input.

From an attacker's perspective fulfilling their adversarial goal, they manage to alter the system's behavior either by setting a specific label to an adversarial example (e.g., visually seen a dog, but tagged as a cat) (targeted misclassification), or setting any other than the correct label (random misclassification), or reducing the system's confidence and thus introduce ambiguity (confidence reduction) [7].

1.3. The Manifold

An important notion of the ML field is the manifold. One may think of it as a multidimensional surface in $\mathbb{R}^n$, where many points are connected, and the close ones are found in context correlation. Due to the optical restrictions n dimensions set, manifolds are better visualized as sub-manifolds. The points of interest (e.g., a specific object in images) are located in a number of such surfaces.

A special behavior called manifold assumption, occurs when moving across manifolds, denoting a change of class, while, moving in the same manifold the input is defining variations of the class (rotations, translations, etc.). Furthermore, it is assumed that moving into a sub-manifold or across others, improves the ML algorithm [7].

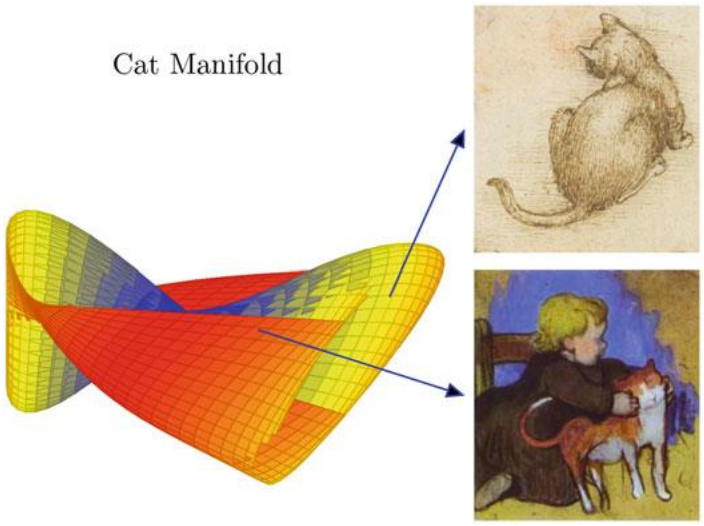


***Figure 2**: Manifold in a High-Dimensional Space. Showing a cat depicting as Single Point in the Manifold. (Top right: Leonardo da Vinci (c. 1513); bottom right, Paul Gauguin (c. 1890). Public Domain). Image Credited to [8].*

Presenting the above image, where positive classes constitute the object cat, and negative ones all the other objects [8] imagines walking inside the cat manifold from one point to another -from one cat to another- where the first cat turns into another cat. In the case of visiting coordinates outside the cat manifold, the object cat fades out to another object or noise.

Another important notion closely related to the Manifold, namely '**latent-space**' offers a more approachable representation of data. Often many features of input data are obfuscated at a higher dimensional representation. Thus, if they are represented in a lower-dimension space they can be processed more easily, which is their main advantage.

1.4. Optimization

Optimization is a process where a function gets its minimum or maximum value. This function is called objective function and is substituted to constraints. In order to adequately conceive the purpose of the optimization through the machine learning perspective [9] defines the main steps of machine learning as (i) to build the model hypothesis, (ii) to define the objective function and (iii) to solve the maximum or minimum of the objective for the parameters to be determined, emphasizing the fact that the last one belongs to the optimization field.

By and large, deep learning algorithms use this kind of process to minimize a function, namely loss or error function [10]. The value that minimizes the function is often denoted as:

$$x^* = \arg\min f(x)$$

According to [9] the family of Gradient Descent (Batch gradient descent – BGD, stochastic gradient descent – SGD, mini-batch gradient descent) algorithms are among the most used for optimal parameters to be determined.

The algorithm iteratively adjusts the variables of the objective function in the opposite direction of the gradients, in order to minimize the cost function. The learning rate β refers to the size of the step the algorithm takes in order to find the minimum. Choosing β may be proved a demanding work, since if it is too small the algorithm will run slow, else it may lose some accuracy.

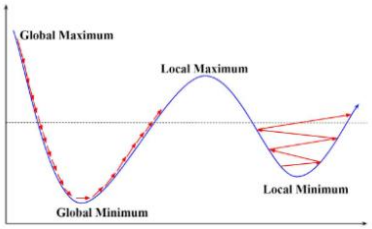


***Figure 3**: A Demonstration of the Difference between the Local and Global Maximum and Minimum Values and the Learning Rate (shown in red) that Determines the Magnitude of the Updates to Model's Weights. Image Credited to [11].*

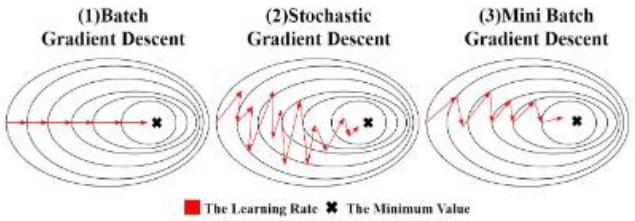


***Figure 4**: An Illustration of the Differences between Training a Model using BGD, SGD and Mini -Batch- Gradient with regarding to approaching the Minimum Value. Image Credited to [11].*

1.5. Norms

In order to quantify the distance between two vectors, special functions -norms- are used. There are many variants of norms, such as L_0, L_1, L_2 and L_∞ all of them widely used by state-of-the-art adversarial algorithms [11]. Informally, norms are functions that take a vector as input and return a non-negative scalar. They are of great importance in the optimization of adversarial attacks and minimization of perturbations [11].

Norms are used to calculate the difference between the expected value and the actual one, and thus estimate the value of loss function in a ML algorithm. Depending on the specific problem (e.g., constraints, policies) an algorithm deals with an appropriate

metric may be used. Furthermore, distance metrics may be used to calculate the classification area of an object, or the minimum possible distance where an object may be characterized as an adversarial.

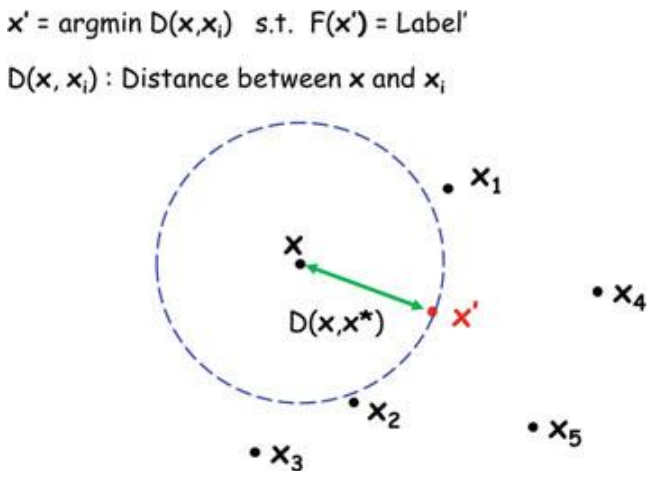


***Figure 5**: Distance Metrics. Image Credited to [82].*

Formally the L_p function of x can be defined as:

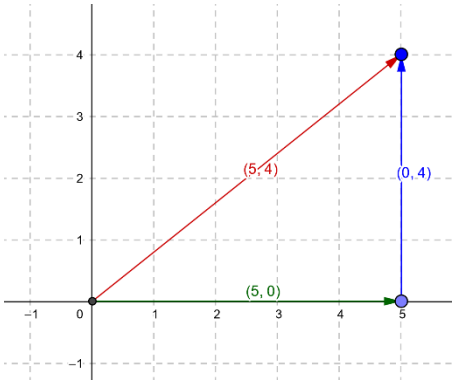
$$\|x\|_p = \left(\sum \|x\|^p\right)^{1/p} \quad \exists: p \in \mathcal{R}, p \geq 1$$

L_0 Norm: Although for $p = 0$ the requirements for a function to be characterized as a norm are not fulfilled, L_0 Norm calculates the number of non-zero elements of the input vector. In the context of the adversarial examples, it counts the number of pixels that have been altered.

$$\|x\|_0 = (i|x_i \neq 0)$$

L_1 Norm (Manhattan Distance): L_1 is the sum of the magnitudes of the vectors in space. In this case, all the components of the vector are weighted equally.

$$\|x\|_1 = \sum_{i=1}^{n} x_i$$

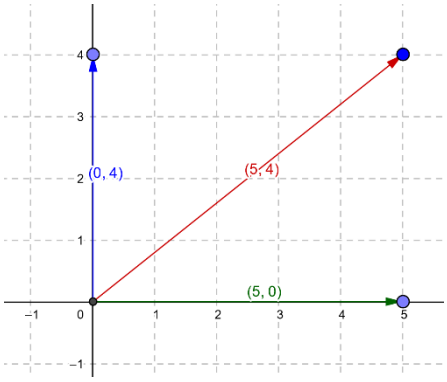


$$\|x\|_1 = |5| + |4| = 9$$

***Figure 6**: L2 Norm*

L_2 Norm (Euclidian norm): measures the shortest distance between two points. That could be interpreted as small changes in the pixels of an image.

$$\|x\|_2 = \left(\sum_{i=1}^{n} \|x\|^2\right)^{1/2}$$



$$\|x\|_2 = \sqrt[2]{|5^2| + |4^2|} = \sqrt[2]{41}$$

***Figure 7**: L_∞ norm*

L_∞ Norm (maximum norm):

$$\|x - x'\|_\infty = max(|x_1 - x_1'| \dots |x_n - x_n'|)$$

In this norm, only the largest element of input vector is taken into consideration. In the context of an image that could be interpreted as the maximum change of a pixel. With regards to the above example the result would be:

$$\|x\|_\infty = 5$$

Dispute among researchers on the issue of choosing the most optimal norm is often the case. Therefore, distance metrics according to [12] still remain an open issue for further investigation.

1.6. Common Datasets used in Deep Learning

For reasons of fulness with regards to the data being tested for adversarial examples and not only, most commonly used datasets in computer vision are mentioned below:

- **MNIST**: The MNIST dataset (Modified National Institute of Standards and Technology database) is a database of handwritten numbers (0 to 9). It has a training set of 60K examples and a test set of 10K examples [13].
- **CIFAR-10**: The CIFAR-10 dataset consists of 60K labeled tiny color images (32 × 32) with 10 classes, 50K of which are training images and 10K test images [14].
- **ImageNet**: ImageNet is a dataset organized according to the hierarchical structure of WordNet. It consists of 1.4M images with 1K classes [15]

2. Types of Attacks

Earliest security concerns have been reported in the work of [16], where the attackers tried to change the attitude of a spam filter. Along with technological growth the number of various attacks has also risen. Researchers many a time having to deal with unknown security parameters of systems, attempt to violate one, in order to provoke a reaction, and finally mitigate the impacts of such a breach.

Thus, the relevant bibliography tallies an enormous number of attacks, deploying at every stage of ML system, all depending on the attacker's strategy (e.g., aim, capability). The pipeline of AI based systems can be described in five steps as previously mentioned: data collection, data pre-processing, model training, model inference and system integration, each of which is prone to a specific set of threats and ensued by the corresponding defense mechanisms [17].

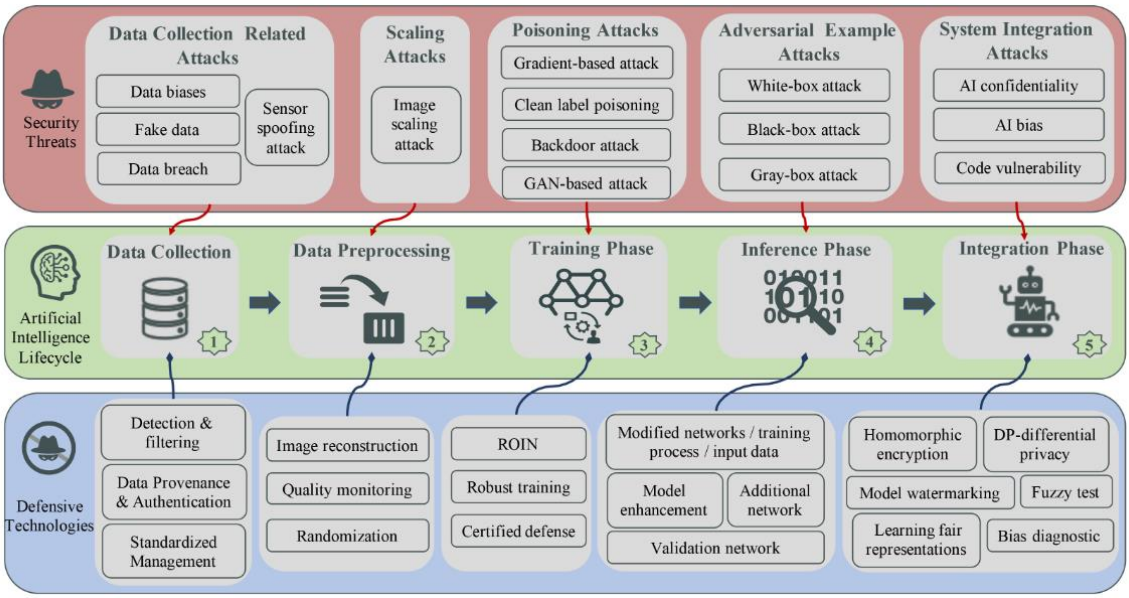


***Figure 8**: The Overall Framework of Attack and Defense Strategies for The AI Systems. Image Credited to [17]*

Data collection stage is threatened mainly by malicious data, during their input channels. That is, data receiving data from low-level apparatus (i.e., sensor capturing raw data -camera, microphone-) that might be altered, or contain fake or biased data, in the case of digital form (open libraries, e.tc.).

In the **pre-processing** stage data are analyzed, cleaned, processed in terms of incompleteness, unfairness, anomaly and irregularity and transformed [18]. A basic

vulnerability of this stage is the image scaling attack [19]. This attack method during the downscaling phase of an image (input) to an ML system, produces a new and irrelevant picture of the initial. The ML is then trained erroneously.

Model training stage contains the step of model selection and the evaluation of test dataset (parameters evaluation). At this stage one of the most infamous category of attacks in the pipeline of an AI based system takes place, namely causative [20] or poisonous. Attacks of this kind try to degrade an ML system in terms of availability or integrity depending on the side effects and the extent of the system's failure.

At the **inference** phase, the trained model is applied. Adversaries that by no means have access to the training phase, may conduct attacks, namely exploratory [20] or evasion. This kind of attack is being deployed by crafting small imperceptible to the human eye perturbations based on the input data (image, speech) or patches in the real world (physical adversarial examples), causing the system to produce misclassifications.

The **AI integration** phase encompasses not only the risks of the AI technology per se, but all the framework's ones, where an application is based upon, such as network attacks, software vulnerabilities, e.tc. where a broad variety of attacks could be deployed.

2.1. Attacks Deploying at the Training and Inference Stage

In the current thesis, we are particularly going to dive into three major attack categories (poisoning-backdoor attacks and adversarial examples) that take place during the **training** and **inference** phase of the Machine Learning model, due to severity of risks posing to ML systems and their constantly development, following as well as the emerging interest of the research community. Although we mentioned earlier a more rigorous framework of the AI systems [**Figure 1**:Basic Components in Supervised Learning and an ML-Based System's Lifecycle. Image Credited to**Figure 1**, **Figure 8**], we deem as a more practical approach the concatenation of the many steps before inference phase at one, namely training as the figure below [**Figure 10**].

2.1.1. Training Phase

Training phase is referring to the stages of collecting and preprocessing of data, model training, validation and the deploying of the model [21]. The attack takes place before the system produces the expected results.

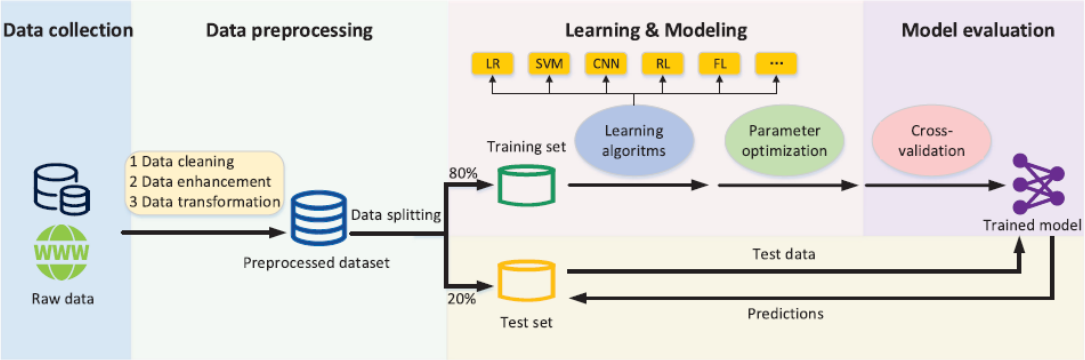


***Figure 9**: Machine Learning Training Phase. Image Credited to [18].*

2.1.2. Inference Phase

Inference phase is referring to the stage where an ML model applies the trained model and produces the outcomes, i.e., prediction, classification and recognition, based on the task.

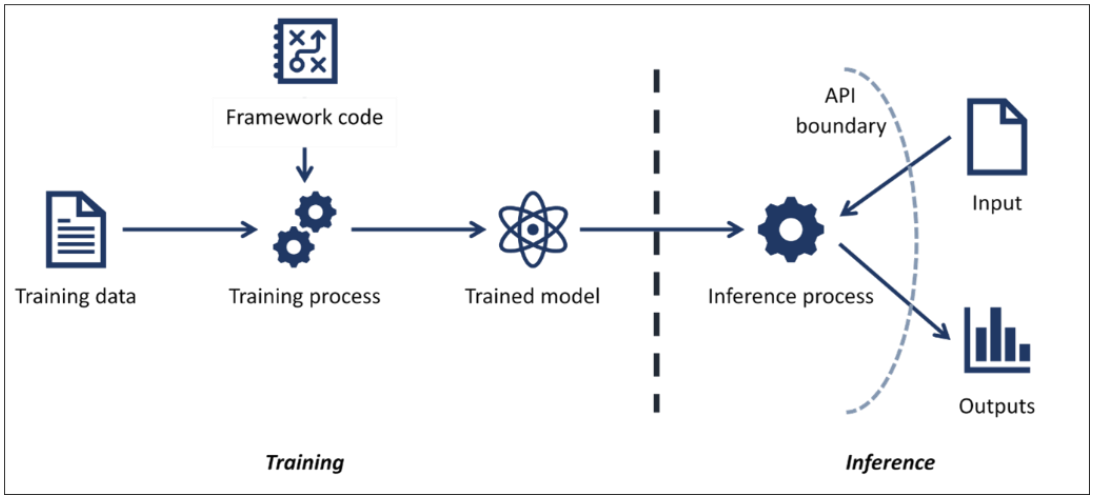


***Figure 10**: Stages of Machine Learning Models. Image Credited to [21].*

2.2. Further Taxonomy Features of Attacks

The taxonomy of attacks has been the subject of many researchers in their effort to produce a clear image of the field and enlighten any new research direction. Most of them have taken into consideration the various aspects of an adversary, such as means of deployment, computational cost, time, and thus can be classified as follows.

2.2.1. Adversarial Falsification

False positive attacks are those producing an incorrect class to a correctly classified sample. In the visual domain this means that an image containing an adversarial example, imperceptible to human eye, will be classified as a class with high confidence.

False negative attacks are those producing a correct class to a misclassified sample. In the context of visual learning an object fully recognizable to a human, cannot be identified by the neural network.

2.2.2. Attack Frequency

Based on the circumstances the adversary conducts their attack, it may be necessary to deploy only once their evasion (one-shot) -meaning to optimize their algorithm just once- [22], especially when dealing with real time applications, or due to computational costs. Alternatively, iterative attacks query the target more than once, so as to adjust their parameters and achieve a better performance.

2.2.3. Target Types

According to the aim of the attacker, either a targeted attack can be conducted if the neural network outcomes a specific class, or a nontargeted attack may be conducted, where the aim is to produce an arbitrary class, and generally raise of a matter of reliability for the system [22].

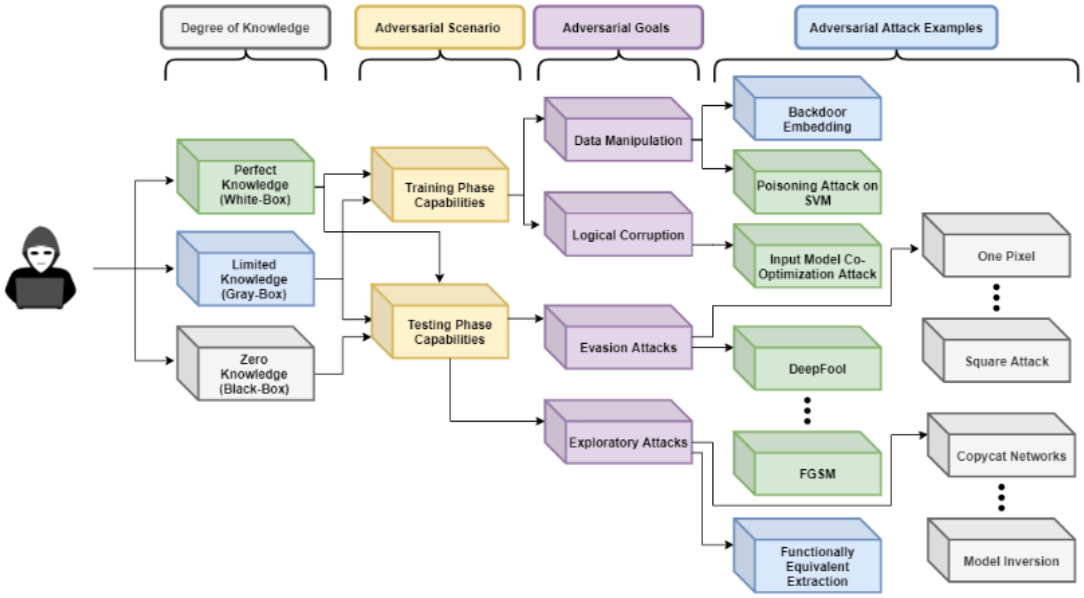


***Figure 11**: The Taxonomy of The Adversarial Threat Model. Image Credited to [11].*

2.2.4. Knowledge of the Offensive System

According to the adversary's knowledge of the targeted ML system -architecture, training and testing set, features, parameters, weights, algorithms, loss functions- attacks can be classified into three main categories: white-box, black-box and grey-box.

- **White-Box Setting**

  In this category of attacks, the adversary has a full knowledge of the system. They are fully aware of the training and testing set, as well as all the necessary information to craft special inputs with the intention to produce erroneous outcomes. Thus, being an internal actor of the target system with privilege access, and the ability to expose vulnerable feature spaces [17] one may create adversarial examples. The term white-box refers exactly to the amount of knowledge an attacker has.

- **Black-Box Setting**

  On the contrary, black-box attacks are developed with zero knowledge of the targeted system. The adversaries, through sophisticated queries, try to analyze the system and reveal its weaknesses. Black-box also refers to the level of knowledge one has of the target system. In reality black-box attacks are more common to be deployed, since an attacker usually targets a third system, fully unfamiliar with it, however hard that might be proved.

- **Grey-Box Setting**

  Attacks of this category are deployed using partial knowledge of the system. In practice attackers may prepare their invasion into a surrogate system of the training set with similar features and then deploy it to the real one scale [11]. The name grey-box also refers to the amount of knowledge someone owns for the target system.

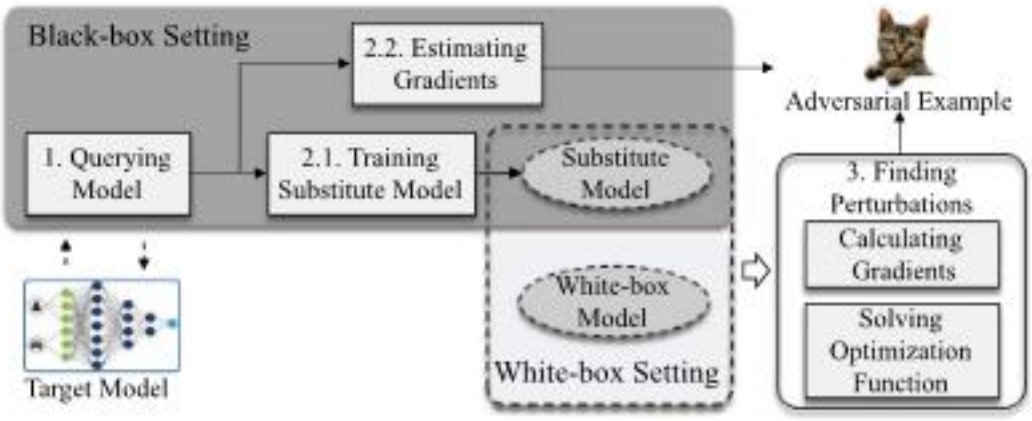


***Figure 12**: Workflow of Adversarial Attack. Image Credited to [23].*

3. Adversarial Examples

3.1. The Very Existence of Adversarial Examples

Adversarial attacks emerged around 2004 [24]. Classifiers producing false negatives due to alteration of the data, suffer degradation. Subsequently Lowd and Meek also succeeded in tricking a classifier but questioning the assumption of Dalvi et al. work with regards to the perfect knowledge of the classifier. Instead, they made the hypothesis that adversaries must learn using prior knowledge, observation and experimentation [25].

Although the trend "adversarial" appears in early works, only recently it gains a great deal of attention. One might expect that state-of-the-art ML systems achieving high performance in object recognition are robust to small changes of input, but this is hardly the truth. Around 2014 [26] proved that many ML and deep learning systems are susceptible to small imperceptible perturbations, causing the system to misclassify the output category of the object. They termed these kind of inputs "adversarial examples".

Adversarial examples are special crafted vectors added to the input images causing minimal changes. Most of the produced images hardly differ from the original ones in terms of semantic consistency, and thus changes are impossible to be traced with bare eyes. But the peculiarity lies in the fact that ML systems classify the objects wrongly and in many cases with high confidence. Systems receiving non anticipated results introduce data vulnerabilities of great importance.

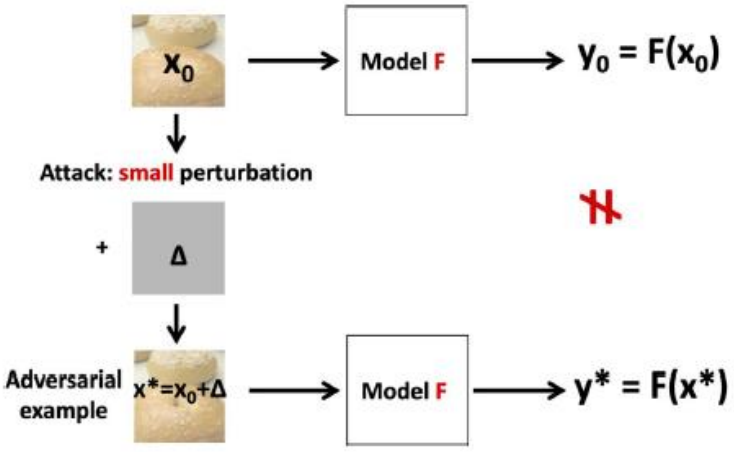


***Figure 13**: An illustration of Adversarial Example.*

3.2. Formal Definition of Adversarial Examples

The authors provide a mathematic definition of the adversarial examples as follows:

Given a classifier $f : R^m \rightarrow \{1 \dots k\}$ mapping image pixel values to a discrete label *l* and *n* the minimal perturbation we apply to *x* input, we get an adversarial example *x',* so as the input *x'* gets classified to a different label

$$x \in [0,1]^m$$

$$\min_{x'} ||x' - x||_p$$

$$\text{s.t. } f(x') = l'$$

$$f(x) = l$$

$$l \neq l$$

3.3. Causes of Adversarial Examples Existence

Even though a great deal of research has taken place to develop algorithms creating adversarial examples and consequently the responding defense mechanisms, little is the progress made in terms of their explanation. Partly this might be due to the lack of strong mathematical tools, able to rigorously analyze an input to a high-dimensional space. [27] set it more vivid - as if the research community is on the hold for an efficient "telescope" to be developed, suitable to fully observe the geometry of the high-dimensional universe.

On the other side though, Daniel Lowd and Cristopher Meek [25] are brilliantly referring to this saying:

"*If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.*"

– Sun Tzu, The Art of War

Nevertheless, researchers have made considerable efforts to explain the reasons of adversarial examples existence [28]. Ironically, one might claim that there are "adversarial" opinions of their origin, such as the non-linearity of the model, or the linear behavior of the model in a high-dimensional space, non-robust features.

However, the majority of the researchers argue that adversarial examples are features of the systems and not bugs or accidental events.

Adversarial Examples have been studied through **many perspectives** [27]**:**

- **Low-Probability "Pockets" in the Manifold.**

  [26] explained the existence of adversarial attacks as an intriguing phenomenon. He described these examples as low-probability (high-dimensional) "pockets" in the manifold, that emerge after carefully crafted input around the given example.

  [7] provided a further explanation of the theory that lies in "pockets" of data manifold. They stated input transformations are highly correlated and drawn from the same distribution across the training set, but adversarial examples differ in terms of correlation or distribution.

  Further explanations of the topic suggest that these blind-spots are actually relatively large, in input space volume, and locally continuous [29]. What is more, is that the authors conclude that these vulnerabilities are subject to "intrinsic deficiencies in the training process and object function" that to model topology

- **Linearity of the Model.**

  Consequently, [30] provided a more rigorous explanation. They described adversarial examples as a sort of accidental steganography.

  With an input x and a perturbation n, with the constraint $||n||_{¥}<e$, they suggested we would have an adversarial example $\bar{x} = x + n$

  Furthermore, considering the dot product between a weight vector w and an adversarial example $\bar{x}$:

  $$w^T \bar{x} = w^T x + w^T n$$

  So, by this equation we can see that the adversarial perturbation ($w^T n$) causes the activation to grow by $w^T n$, and if we assign n = sign(w) we will maximize this increase to the max norm constraint. Assuming that w has n dimensions and the average magnitude of an element of the weight the vector is m, then the activation will grow emn.

Finally, the authors conclude that if x has sufficient dimensionality (n), a small perturbation could provoke great changes.

- **Test Error in Additive Noise**:

  The authors [31] suggest that adversarial examples are a consequence of the nonzero test error in certain corrupted image distributions (gaussian noise) and there is no need for newly coined terms to describe such features.

- **Non-Robust Features**

  According to [32] non-robust features underly into adversarial examples. The authors continue and define non-robust features as those derived from patterns in data distribution and are highly predictive, but incomprehensible to human. Simplifying this hypothesis is that in every dataset there are features that play a key role in classification, but they are at a scale where cannot be directly understood by a human.

  They follow a method splitting an image into robust dataset and non-robust and then train the model. The idea is that if one interferes an adversarial image, the robust features will still remain as the initial class, but the non-robust ones will change to the new false class.

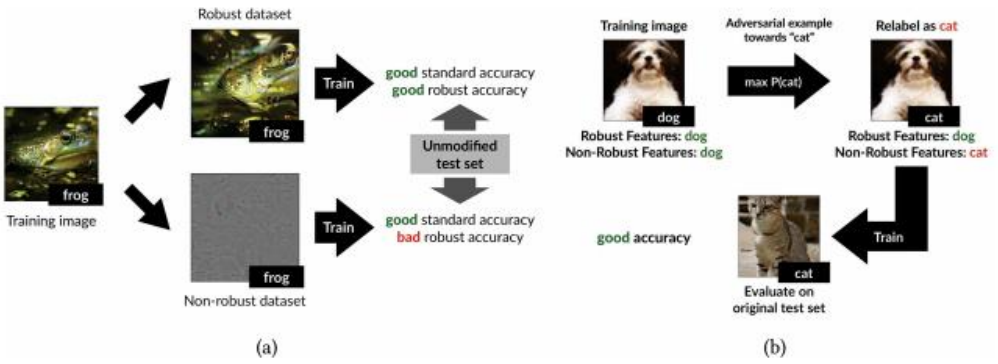


***Figure 14**: A Conceptual Diagram. In (a) features are disentangled into combinations of robust/non-robust features. In (b) a dataset which appears mislabelled to humans (via adversarial examples) but results in good accuracy on the original test set. Image Credited to [32].*

- **Geometric Explanations**

  Other proposals include [33], who claim that adversarial examples exist when a decision boundary is close to the submanifold of sampled data. With a more

pictorial explanation as provided by the authors the explanation lies in the fact that a class might not fit exactly to the boundaries of the data and thus data tilting to the boundaries of the class may produce an adversarial example.

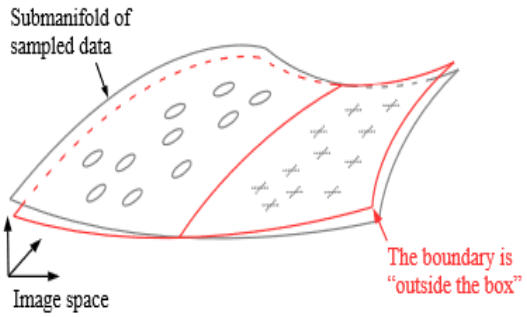


***Figure 15**: Adversarial Examples are Possible Because the Class Boundary Extends Beyond the Submanifold of Sample Data and can be -Under Certain Circumstances- Lying Close to it. Image Credited to [33].*

Furthermore, the authors suggest that adversarial examples are not likely to occur in other directions of low variance in the data, and thus speculate that adversarial examples can be caused due to the overfitting phenomenon that could be alleviated through regularization [7].

- **Further Views studying Adversarial Examples**

  In addition to the abovementioned explanation, studies in terms of adversarial examples perspectives follow, with the latest one **distinguishing** the field **under three views**: **model**, **data** and **other** [34]. The authors conduct a thorough research in the field of adversarial examples with the aim to record the problems and challenges of the topic, as well as, to highlight new research directions.

  Each of the above perspectives is divided to further categories according to different aspects with respect to different phases of ML systems (training, inference). All the viewpoints attempt to explain the causes of adversarial examples, such as the linearity, the boundary system, the loss function etc., are all titled into the model' s perspective. Taking into consideration the amount of data needed for ML systems, they create a perspective of their own. Dimension of the data, distribution, features are among the traits this perspective is concerned with. Further studies for the origins of adversarial examples are falling into the Other perspective, since they get inspiration from real human aspects and thus provide a promising field.

The authors also make an attempt to record the literature under these perspectives so as to point out the advantages and flaws of each underlying theory and to further support new researchers to turn their lights into some view with little attention up to a point.

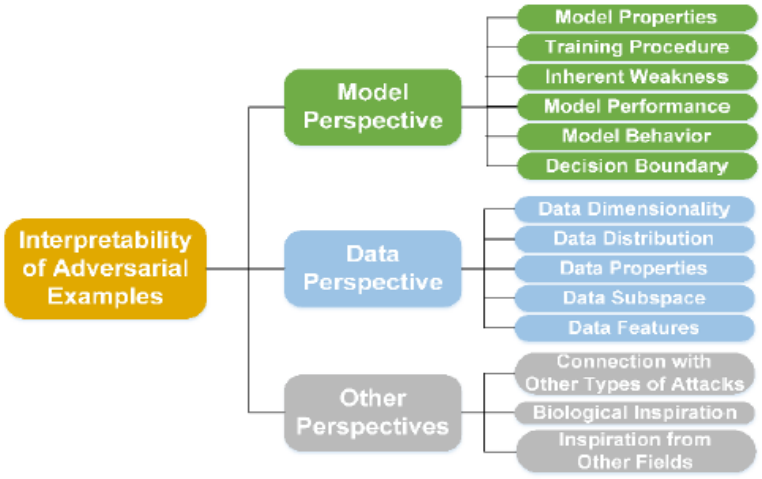


***Figure 16**: Three Main Perspectives of Related Works on the Interpretability of Adversarial Examples. Image Credited to [34].*

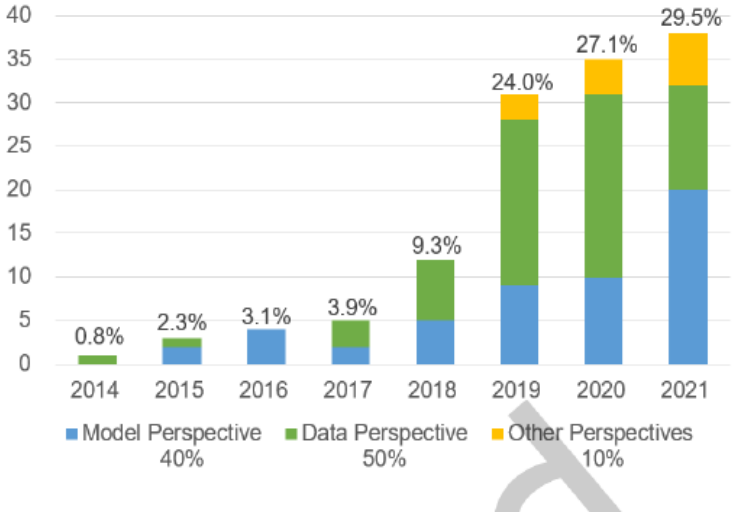


***Figure 17**: Related Publications of Interpreting Adversarial Examples from Three Perspectives. Image Credited to [34].*

3.4. Perturbations

"Perturbation" comprises a cornerstone term that defines adversarial examples. Currently no formal lexical definition has been found in the literature, whereas the term is mentioned in circumlocution. By and large, one could claim that a perturbation is a small, tiny distraction of the input data, in a way of not being realizable to the bear eye.

That is, an Machine Learning based system would interpret the raw data, in an inexplicable to the human notion way.

[35] scrutinized the "perturbation" term and classified it into categories, such as individual and universal, whether it can be applied one or more clean input data, as well as, optimized and constraint ones, depending on their goal. That is, minimizing it to the extent that a perturbation will not be recognizable to humans, or for the latter category to setting the perturbation as a constraint to optimize the problem.

Furthermore, the authors mentioned the ℓ_p norms as a widely used metric tool, while they referred also a new one, namely psychometric perceptual adversarial similarity score (PASS) [36], consistent with human perception. Several other metrics have been proposed to literature such as the semantic similarity [37] and dropping points [38].

At this point a distinction worth mentioning, that a perturbation takes place in the digital form of adversarial examples. However, in the real world, patches replace the original input in a selected space of the original picture.

3.5. Transferability

An important aspect of adversarial examples and with a major impact on every AI based application, is their ability to be deployed on different ML systems causing erroneous outputs. [26] were the first that showed that adversarial examples are shared among different architecture ML systems, along with their own existence. Such ascertainment was sufficient enough to raise security issues and provoke the research community's reaction.

[39] studied the transferability of adversarial examples more rigorously. They defined "Adversarial sample transferability", as the property of some adversarial samples that may mislead besides a specific model f and other model f' - irrespectively of their architecture. Furthermore, a taxonomy was proposed with respect to their technique: **Intra-technique** and **Cross-technique**. The first one refers to those samples deploying in the same machine learning system, but each time trained with different parameters or datasets. The latest one refers to the different architecture machine learning systems (neural network or decision tree).

Further studies resulted in variant outcomes. Transferability has been analyzed in terms of hyperparameters, model architectures (differentiable or non-differentiable), as an inherent property, as well as in correlation to the input space and the algorithm design of the ML system. Empirical evidence suggests this property is due to large spaces instead of small pockets. As [7] survey the literature on this topic, they conclude that transferability is not equally applicable for all ML systems highlighting the fact for more research to be done on the field.

3.6. White-Box Attacks

Below we mention several of the most representative algorithms creating adversarial examples in a white-box setting:

- **L-BFGS**

  [26] showed that small imperceptible perturbations (ρ) on a correctly classified given input (x) -namely adversarial examples- were able to foolish a ML system, enforcing it to misclassify that input into (l) class.

  According to [11] the authors and tried to find these small perturbations calculating the:

  $$\min\|\rho\|_2 \; s.t. f(x+\rho) = \; l; x+\rho \in [0,1]^m$$

  That proved to be a *hard-problem*, since there are many (ρ), besides zero (0) that makes it trivial $f(x) = \; l$.

  They overcame the problem by finding an approximation using the box-constrained L-BFGS (Broyden-Fletcher-Goldfarb-Shanno) optimization algorithm, and thus the complexity of their initial problem -finding a minimum perturbation- was reduced.

  They calculated instead the loss function using line-search to find (c):

  $$\min c \cdot |\rho| \; + \; L_f(x \; + \rho, l) \; s.t. x \; + \rho \; \in [0,1]^m$$

- **Fast Gradient Sign Method**

  [30] in order to prove that adversarial examples are a result of their linearity in deep neural networks, they suggested an algorithm to create such, taking into consideration the maximum direction of the gradient change, using

backpropagation. They introduced the "fast gradient sign method" algorithm, which is illustrated as an optimal max-norm constrain perturbation, of:

$$n = \epsilon sign(\nabla_x J(\theta, x, y))$$

where J is the cost function, θ the parameters of the models, x the input, y the target associated with x and ϵ a small scalar value to restrict the norm of the perturbation.

Furthermore, the authors find that using $\epsilon = 0.25$, they cause a shallow softmax classifier to have an error rate of 99,9% with an average confidence of 79,3% on the MNIST test set.

Additionally, they argue that **rotating x by an angle in the direction of the gradient**, **also creates adversarial examples**.

- **DeepFool**

[40] proposed a new algorithm to create adversarial examples and measure the robustness of a ML system. They define robustness as:

$$\rho_{adv}(\hat{k}) = \mathbb{E}_x \frac{\Delta(x; \hat{k})}{\|x\|_2}$$

where x is an image, $\hat{k}(x)$ is the estimated label and $\mathbb{E}_x$ is the expectation over the distribution data.

The authors continue by calculating the minimum perturbation (iteratively), as above:

$$r_*(x_0) := \arg\min \|r\|_2 \ subject\ to\ sign(f(x_0)) = -\frac{f(x_0)}{\|w\|_2^2} w$$

The algorithm is based on the assumption that neural networks are linear, with a hyperplane separating each class. Thereafter, the initial hypothesis of linearity is expanded. Since neural networks are not linear and the process is repeated.

Consequently, the algorithm calculates:

$$arg \min_{r_i} \|r\|_2 \ subject\ to\ f(x_i) + \nabla f(x_i)^T r_i = 0$$

The algorithm stops at iteration $i+1$ when x_{i+1} changes the sign of the classifier.

- **Carlini & Wagner (C&W)**

[12] proposed a family of three attacks (with constraints l_0, l_2, l_∞), namely C&W attacks, that are able to exceed the distillation defense mechanism image classification in a neural network. The problem to find adversarial samples, according to the authors is formally expressed, as above:

$$minimize\ D(x, x+\delta)$$

$$C(x+\delta) = t$$

$$x+\delta\ \in [0,1]^n$$

where x is an image, and the goal is to find δ that minimizes $D(x, x+\delta)$.

In order for this problem to be solved using an optimization algorithm the aforementioned equation has been transformed, as:

$$minimize\ D(x, x+\delta) + c \cdot f(x+\delta)\ s.t. x+\delta\ \in [0,1]^n$$

where D represents constraint paradigms, c denotes the hyperparameter, and f adopts a variety of objective functions.

The authors chose $f(x') = \left(\max_{i \neq t}(Z(x')_i) - Z(x')_t, -k\right)^+$ after an evaluation of seven objective function, where $e^+$is short hand for $\max(e, 0)$, Z denotes the softmax function.

Furthermore, in order to avoid "box-constraint", the authors introduced a new variant w, where:

$$\delta_i = \frac{1}{2}(\tanh w_i + 1) - x_i$$

They also provided three kinds of attacks base on the distance metrics l_0, l_2, l_∞

l_2 attack: $\min_w \left\|\frac{1}{2}\tanh(w) + 1\right\|_2 + c \cdot f\left(\frac{1}{2}\tanh w + 1\right)$

As for the l_0 attack, since it is not differentiable, through an iterative process the pixels that don't have much effect on the classifier input are characterized stable. The algorithm stays with the minimum of them that can be altered and create an adversarial example. L_2 attack is used to identify the pixels with less effect on the classifier.

L_∞ attack is also an iterative process, where the $l_2$term was replaced with a new penalty in each iteration:

$$\min c \cdot f(x+\eta) + \sum_i [(n_i - \tau)^+]$$

- **Jacobian-Based Saliency Map**

[6] introduced an algorithm where the output modifications are taken into consideration, in an iterative way of producing new adversarial samples, and thus achieve a misclassification. The authors are motivated by the forward derivative. They evaluate the forward derivative:

$$\nabla F(x) = \frac{\vartheta F(x)}{\vartheta x_1} = \left[\frac{\vartheta F_j(X)}{\vartheta x_i}\right]_{x \in 1..M, j \in 1..N}$$

and define an adversarial saliency map -namely Jacobian- which highlights the features with respect to the adversarial 's goal and the impact to the classification, so as to be included in the next step. Consequently, the algorithm using optimization techniques, simple heuristics, or even brute force, produces the next perturbation. The next step and after the evaluation of the perturbed input, determines whether the aim of the attack is accomplished and the output is misclassified, or the result exceeds the maximum threshold and thus the distortion is obvious with the naked eye.

Furthermore, the authors claim they achieved a misclassification with a 97% success rate, by tampering with only 4,02% of the input features per sample.

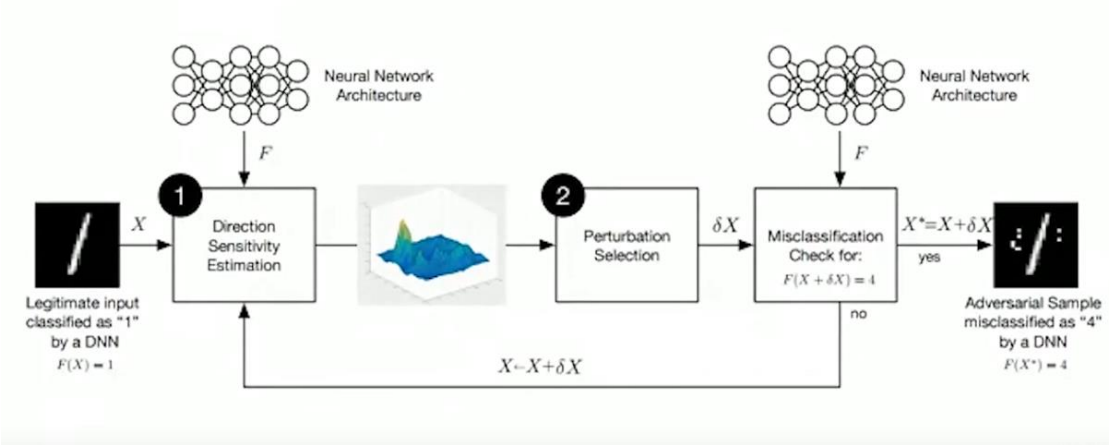


***Figure 18**: Jacobian-Based Saliency Map Algorithm (JSMA). Image Credited to [41].*

- **Iterative Fast Gradient Sign Method**

  [42] enriched further the FGSM algorithm, developing two more versions of it: Basic Iterative Method (BIM) and Iterative Least-Like Class Method (ICLM).

  BIM works iteratively, using a step e and afterwards a function $Clip_{X,e}\{x'\}$ crops each pixel, so as to ensure the imperceptible character of the newly produced examples x'. The algorithm is formally presented as follows:

$$X_0^{adv} = X,\ X_{N+1}^{adv} = Clip_{X,e}\{X_N^{adv} + \alpha \cdot sign(\nabla_X J(X_N^{adv}, y_{true}))\}$$

  Furthermore, in order produce peculiar targeted adversarial examples in datasets that samples are not so distinct among each other, the authors substituted the y_{true} label, with the least-like class of the trained network, according to the formula:

$$y_{LL} = arg\min_y\{p(y|X)\}$$

  They maximized y_{LL} using $\log\{p(y|X)\}$ iterative way in the direction of $sign(\nabla_X \log\{p(y|X)\})$, which equals $sign(-\nabla_X J(X, y_{LL}))$ and finally the resulting formula is:

$$X_0^{adv} = X,\ X_{N+1}^{adv} = Clip_{X,e}\{X_N^{adv} - \alpha \cdot sign(\nabla_X J(X_N^{adv}, y_{LL}))\}$$

- **Universal Adversarial Perturbations**

[43] proposed an algorithm that seeks for a perturbation in datapoints among a set of pictures with the same data distribution. Their approach differs on previous works, since the adversarial sample applies to many natural images and it is produced by adding universal perturbations, without requiring optimization or gradient calculation.

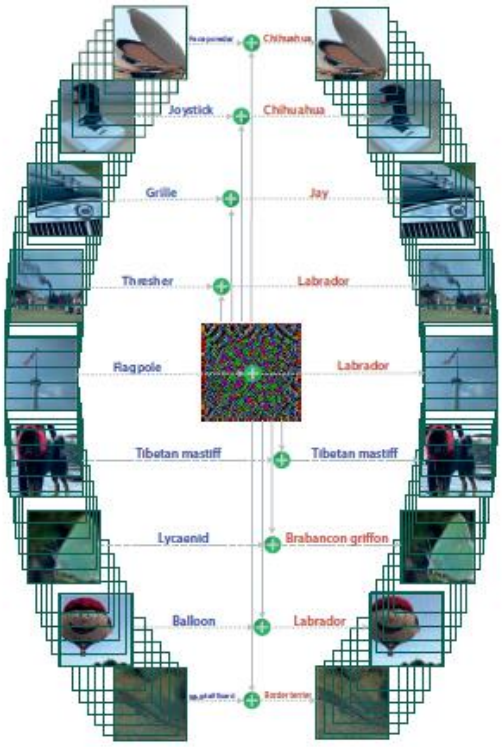


***Figure 19**: When Added to a Natural Image, a Universal Perturbation Image Causes the Image to be Misclassified by the Neural Network with High Confidence. Image Credited to [43]*

The problem is formulated as above:

Let $\hat{k}$ be a classifier and $\upsilon \in \mathbb{R}^d$ a vector of perturbations that fools the classifier on almost all datapoints deriving from a distribution μ. Then we are looking for a vector υ, such that $\hat{k}(x+\upsilon) \neq \hat{k}(x)\ for$ most $x \backsim \mu$

The algorithm iteratively finds perturbations (using $\ell_p$metric) over the images and builds the universal perturbation υ with the following constraints:

$$\|\upsilon\|_p \leq \xi$$

$$\mathbb{P}_{x \backsim \mu}\left(\hat{k}(x+\upsilon) \neq \hat{k}(x)\right) \geq 1 - \delta$$

where ξdenotes the magnitude of the perturbation, and $1-\delta$ the probability of misclassification.

3.7. Black-Box Attacks

Below we mention procedures crafting adversarial examples in a black-box setting:

- **ATNs**

  [44] proposed Adversarial Transformation Networks (ATNs), which are feedforward neural networks trained to produce adversarial examples. The network, can be formally defined as:

$$g_{f,\vartheta(x)}: x \in \mathcal{X} \rightarrow x'$$

  where ϑ is the parameter vector of g, f is the target network which outputs a probability distribution across class labels, and $x \sim x'$, but $argmax\, f(x) \neq argmax\, f(x')$, which turns out to be a minimization problem of the joint Loss functions L_x(of the input space) and L_y(of the output space):

$$arg\min_{\vartheta} \sum_{x_i \in \mathcal{X}} \beta L_x\left(g_{f,\vartheta}(x_i), x_i\right) + L_y\left(f\left(g_{f,\vartheta}(x_i)\right), f(x_i)\right)$$

  Furthermore, authors proposed two methods for generating adversarial examples. Perturbation ATN (P-ATN), which outputs the perturbation of x and Adversarial Autoencoding (AAE), which results a new input based on the initial, taking into consideration all the necessary constraints (weight decay or added noise, input range).

- **ZOO**

  Inspired by Carlini & Wagner (C&W) [45] proposed a black-box attack, which exploits techniques from zeroth order optimization, and thus called ZOO, with no training substitute models needed and directly being deployed. Authors modified the (C&W) loss function, based on the output F of a DNN, as follows:

$$f(x,t) = \max\left\{\max_{i \neq t} \log[F(x)]_i - \log[F(x)]_t\,, -k\right\}$$

  where $k \geq 0$ and $\log 0$ defined as $-\infty$, and consequently, they compute an approximate gradient, instead of back-propagation (since only the input and output of a DNN is available), using symmetric difference quotient to estimate the gradient $\frac{\partial f(x)}{\partial x_i}$, and Hessian $\frac{\partial^2 f(x)}{\partial x_{ii}^2}$ defined as $\hat{g}_i$ and $\hat{h}_i$, as follows:

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + he_i) - f(x - he_i)}{2h}$$

$$\frac{\partial^2 f(x)}{\partial x_{ii}^2} \approx \frac{f(x + he_i) - 2f(x) + f(x - he_i)}{h^2}$$

where e_i denotes the standard basis vector with the ith component as 1 and h is a small constant. Consequently, the attacker uses the ADAM or Newton methods to calculate the adversarial example.

Furthermore, in order for ZOO attack to be efficient, in terms of computational cost and number of queries, techniques such as space dimension reduction, hierarchical attacks and importance sampling are in the scope of this general framework.

- **Houdini**

  [46] introduced an algorithm, namely Houdini that is strongly connected with the loss function of the problem in concern. Authors propose the following surrogated lose function:

$$\bar{\ell}_H(\theta, x, y) = \mathbb{P}_{\gamma \sim N(0,1)}[g_\theta(x, y) - g_\theta(x, \hat{y}) < \gamma] \cdot \ell(\hat{y}, y)$$

  The function consists of two arguments. The first one is a stochastic margin, that calculates the probability of the difference between the score of the actual target and the predicted one, and is smaller than one, resulting in the confidence of the model. The second argument is independent from the first one, and refers to the target that will be maximized.

  The authors report their algorithm is effective besides visual experiments, at speech recognition and in the semantic field, as well as to targeted and un-targeted attacks.

- **One Pixel**

  [47] proposed an attack called One-Pixel, optimized by differential evolution algorithm, and thus makes no use of gradient information and needs no knowledge of the objective function. One-pixel algorithm creates adversarial perturbations, and then alters one or a small number of pixels so as to produce erroneous outcomes. It is formally defined, as:

$$\underset{e(x)*}{\text{maximize}}\, f_{adv}\big(x + e(x)\big)$$

$$subject\ to\ \|e(x)\|_0 \leq d$$

where f is the target image classifier, $x = (x_1, .., x_n)$ the n-dimensional input original image, $e(x) = (e_1, .., e_n)$ is an additive adversarial perturbation, d is a small number and in the case of one pixel modification equals 1. The modification of the one-pixel perturbation may be visualized as moving the data point along the x-axis of one of the n-dimensions.

Furthermore, according to the authors the perturbations are encoded into arrays, forming the candidate solutions. One candidate solution contains a fixed number of perturbations. Each perturbation is a five tuple, including x and y coordinates and RGB value, and alters one pixel. They are optimized and on each a child image is produced, according to the follow formula:

$$x_i(g+1) = x_{r1}(g) + F\big(x_{r2}(g) + x_{r3}(g)\big)$$

$$r_1 \neq r_2 \neq r_3$$

where $x_i$is an element of the candidate solution r_1, r_2, r_3 are random number, F is the scale parameter, g the current index.

Each child candidate is compared to the parent and the better one proceeds to the next iteration. The algorithm continues until the maximum number of iterations reaches, or the target class is above or below a percentage, depending on the dataset.

***Table 2**: Comparison Table for Adversarial Examples*

Adversarial Attacks	Attack Type	Specificity	Attack Frequency	Perturbation Type	Perturbation Norm	Attack strategy	Year	Strong Aspects
L-BFGS [26]	White-Box	Targeted	One-shot	Specific	ℓ_∞	Constrained optimization	2017	Good mobility
Fast Gradient Sign Method [30]	White-Box	Targeted	One-shot	Specific	ℓ_∞	Constrained optimization	2015	Efficient algorithm succeeding good results due to iterations
DeepFool [40]	White-Box	Non-Targeted	Iterative	Specific	$\ell_0\ell_2\ell_\infty$	Gradient Optimization	2016	Limits in targeted attacks
Carlini & Wagner (C&W) [12]	White-Box	Targeted	Iterative	Specific	$\ell_0\ell_2\ell_\infty$	Constrained optimization	2017	Successfully breaks state-of-the-art defense mechanisms, such as defensive distillation, limitations on efficiency

Adversarial Attacks	Attack Type	Specificity	Attack Frequency	Perturbation Type	Perturbation Norm	Attack strategy	Year	Strong Aspects
Jacobian-Based Saliency Map [6]	White-Box	Targeted	Iterative	Specific	ℓ_2	Sensitivity analysis	2015	Good ASR, but limits in mobility
Iterative Fast Gradient Sign Method [42]	White-Box	Targeted	Iterative	Specific	ℓ_∞	Constrained optimization	2017	Applies the FGSM multiple times with a small step
Universal Adversarial Perturbations [43]	White-Box	Non-targeted	Iterative	Universal	$\ell_2\ell_\infty$	Gradient optimization	2017	Better generalization, good for real scenarios
ATNs [44]	Black-Box and White-Box	Targeted	Iterative	Specific	$\ell_2\ell_\infty$	Gradient optimization	2017	Effective training to generate adversarial examples
ZOO [45]	Black-Box	Targeted	Iterative	Specific	ℓ_2	Migration mechanism	2017	Mobility, efficient techniques to accomplish the attack

Adversarial Attacks	Attack Type	Specificity	Attack Frequency	Perturbation Type	Perturbation Norm	Attack strategy	Year	Strong Aspects
Houdini [46] i	Black-Box	Targeted and Untargeted	Iterative	Specific	$\ell_2 \ell_\infty$	Generative model	2017	High Attack Success Rate / Tailored to different domain applications
One Pixel [47]	Black-Box	Non-targeted and targeted	Iterative	Specific	ℓ_0	Differential Evolution	2019	One pixel offers a more concealed attacks, needs many iterations, efficiency low Does not require the optimization problem to be differentiable.

3.8. Adversarial Examples in the Physical World – Physical Attacks

Adversarial attacks have exposed vulnerabilities of ML systems in a great extent. Though many attacks until this point have presumed that the attacker has in their possession the input data in a digital form, there is no doubt that this behavior can be replicated in a real environment. Objects of interest (pictures, raw data from sensors, etc.), can be the subject of malicious behavior and be intervened before their input is transformed digitally. These kinds of attacks are considered to be deployed in the physical world and are conducted at a different stage within the visual recognition pipeline [48].

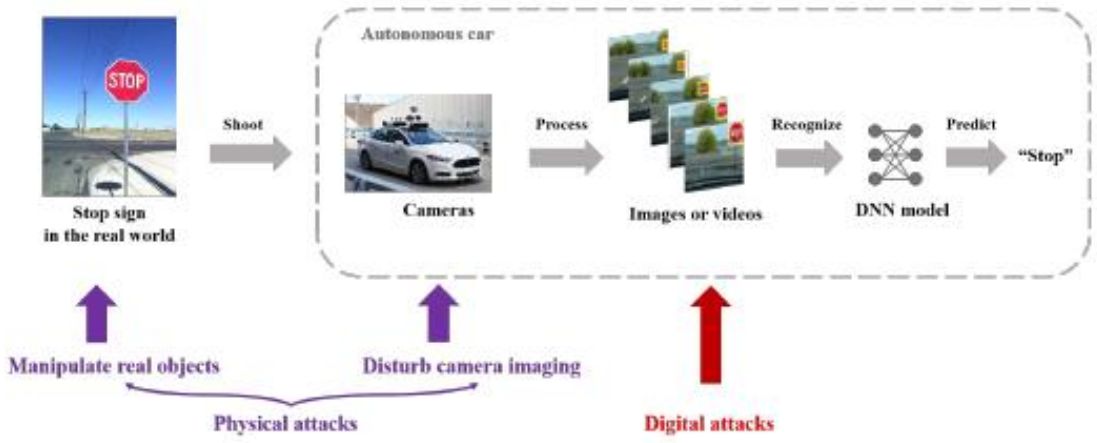


***Figure 20**: The Comparison Between Digital Attacks and Physical Attacks In The Standard Visual Recognition Pipeline. Image Credited to [48] .*

[42] were the first who demonstrated that adversarial examples exist in the physical domain and can fool a ML based system with tiny perturbations. They conducted a series of experiments printing pictures on a paper and then using a cellphone modified them. They further introduced new methods: Basic Iterative Method and Iterative least likely to create adversarial examples, as well as a new metric: destruction rate, to define the influence of arbitrary transformations (change of contrast, brightness, Gaussian blur and noise, etc.) to these images. The results revealed that less was the effect to the adversarial examples with the contrast and brightness transformations, but that did not hold true for the blur and noise.

Physical world adversarial examples need to be robustified over challenges that exist in the real environment. [49] mentions a subset of factors that affect their persistence. Specifically, in terms of the environmental conditions, they refer that distance and

angles should not be able to alter the erroneous nature of adversarial examples. Imperceptibility is of great importance, keeping in balance the ability of the for the sensor of input to capture the perturbation and not being exposed to the observer.

Furthermore, the authors make reference to [50] work, with regards to the reproduction error and the sometimes-erroneous depiction of real-world colors, which might reduce the strength of the attack. Another important category of physical attacks limitation is that of perturbation. In the digital form the attacker has in their availability all the input allowing to create small perturbations. In the physical world perturbation should be created in a manner of not causing the attraction of an observer. This limitation according to the authors is classified under spatial constraints.

***Figure 21**: Stop Sign in the Physical World. The left image shows real graffiti on a Stop sign, something that most humans would not think is suspicious. The right image shows a physical perturbation applied to a Stop sign. Image Credited to [49].*

Dealing with the above challenges, Patch Attacks emerged [50], [51]. [52] termed the patch as a patterned sub-image that is generally masked over the input image, turning it into a feasible solution for attacker in physical environment, and with the further privilege of being deployed with no previous knowledge of the attacked system.

In contrast to the norm-based attacks, as mentioned above, patch attacks craft perturbations on a restricted area of the input data, and is formally defined as:

$$x' = (1 - p) \odot x + p \odot \delta$$

where $\boldsymbol{\delta}$ is the adversarial patch noise and $\boldsymbol{p}$ represents the binary pixel block to mask the patch area (location and area), familiar also as adversarial patch, while the symbol $\odot$ represents the Hadamard operator, which performs element-wise multiplication of pixels from the input matrices.

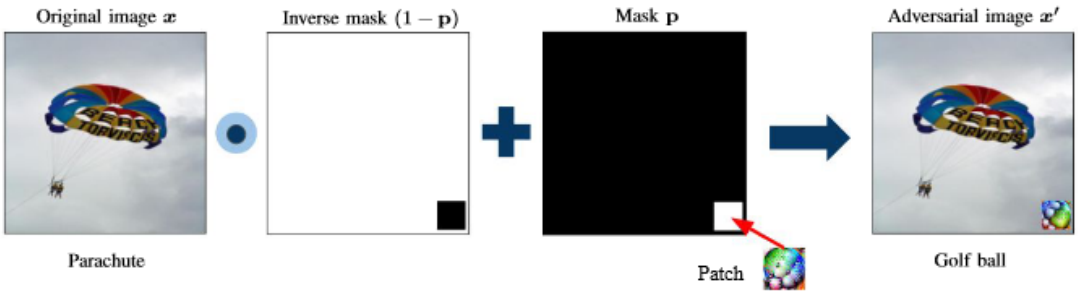


***Figure 22**: Adversarial Patch Attack Procedure (White is 1 and Black is 0). The Adversarial Example is Generated by x' = (1 − p) x + p δ, where δ and p are the Adversarial Patch Noise and the Adversarial Patch, Respectively. Image Credited to [52].*

Joining all these perturbations and printing them in the form of a sticker, the attack applies in the real world, and it is known as Physical Attacks.

3.9. AI-Guardian – A Defense against Adversarial Examples

A novel approach, namely AI-Guardian [53] was recently introduced aiming to tackle adversarial examples using backdoors, presenting remarkably results, with regards to five popular adversarial examples generators. Their attack achieved to lessen the success rate from 97,3% to 3,2%, with a slight decline on the clean accuracy data, and still with no degradation on performance. In particular the algorithm is based on the observation that injected backdoors reduce the functionality of adversarial examples.

A uniquely implemented backdoor, namely **bijection backdoor** is implanted to a deep neural network, so as to shield it over adversarial examples. The newly emerged backdoor is based on either the source (input) or the target label, creating a one-to-one relationship, i.e., a source class corresponds to only one target. Thus, the model with the injected backdoor exerts over the perturbation of the adversarial example.

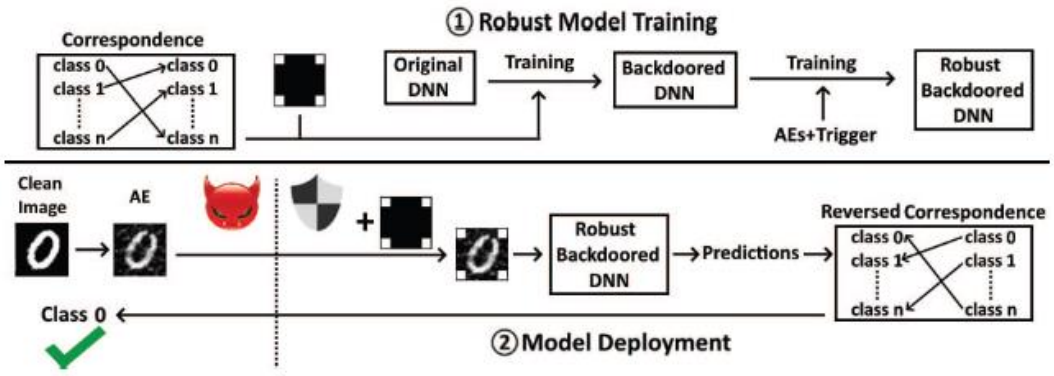


***Figure 23**: Overview of AI-Guardian. Image Credited to [53].*

Furthermore, authors urge the necessity of keeping a firm secret the trigger of the backdoor, since it can used to bypass it, while they recognize the need to further develop a theoretical guarantying the performance of the algorithm.

4. Poisonous Attacks

Poisonous attacks emerge with significant importance in the field of Machine Learning. The name comes from the work of [54] and are also referred as causatives. Their aim is to produce misclassification or subvert the prediction of ML system, by tampering with training data in the corresponding phase of ML based system pipeline. In terms of the adversary's capabilities, the ability to manipulate training data is considered as an assumption.

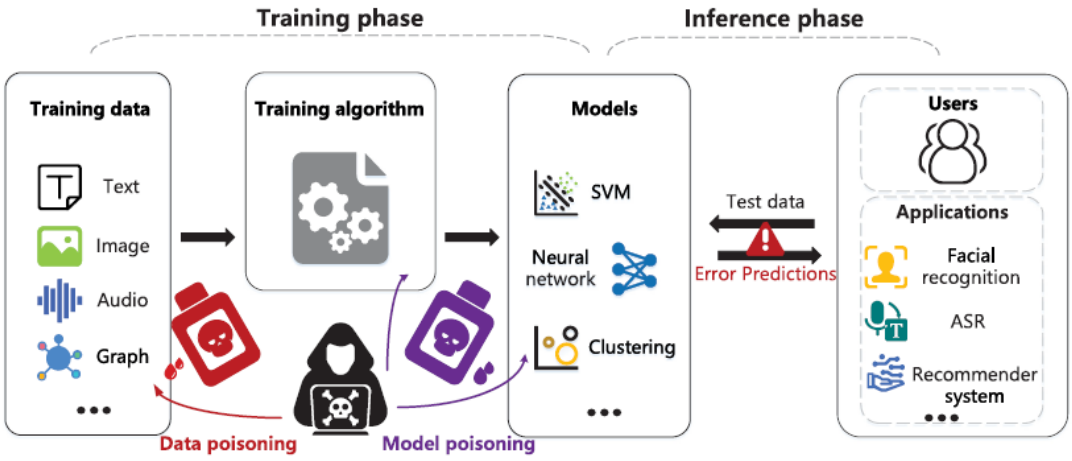


***Figure 24**: The Framework of Poisonous Attacks. Image Credited to [18]*

Literature classifies poisonous attacks under various perspectives [55]. In terms of privacy [56] poisoning attacks are classified as integrity where the system produces misclassifications on specific classes, and availability where the system sustains a general performance degradation, with regards to incorrect classifications or predictions.

In terms of "Specificity", attacks can be classified into targeted and indiscriminate (or untargeted). The first one aims to produce erroneous outcomes in a specific set of classes while the second one aims at no particular target, but rather intends to cause a general declined success rate of ML systems. Another category is that of "Error Specificity", in which if the adversary's target is to cause an erroneous outcome, resulting a specific error class or any other.

Further classifications are reviewed with regards to the learning technique, such as training-from-scratch (TS), fine-tuning (FT), and model-training (MT) [57]. In the first two methods users have limited resources on the training dataset, and thus they address

third-parties for support. Even though the training process is fully controlled by them, malicious data may already be injected into the provided dataset. The two methods differ in the training data they use. The first one trains the dataset from scratch, while the second one uses a pre-trained set and adjusts the weights.

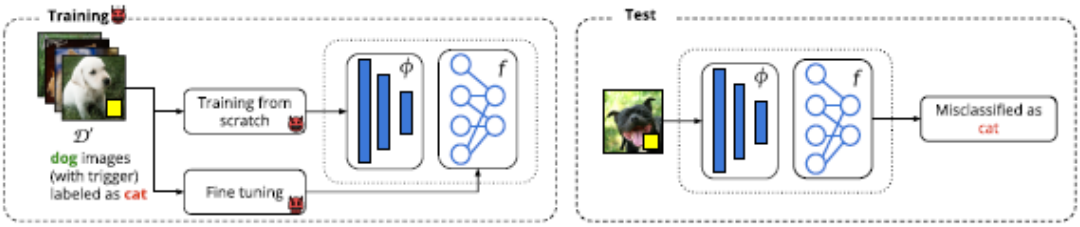


***Figure 25**: Training and Test Pipeline. Image Credited to [57].*

In the last method (MT) the opposite scenario takes place. The user has in their possession the training dataset, but lacks the computational resources or the expertise. Thus, the training method may be posed to vicious intentions.

Other taxonomy of poisonous attacks considers the point the attacker aims to exploit, and thus data and model methods are arising. The first one acquires access to the training dataset, while the second one points directly to the model per se (i.e., training algorithm, modeling procedures) [18].

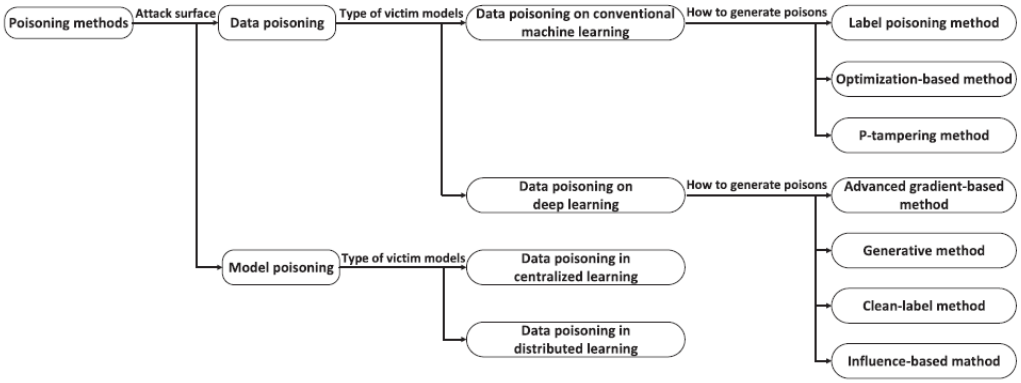


***Figure 26**: The Taxonomy of Poisoning Attacks, Image Credited to [18].*

Poisonous attacks have evolved since their first appearance, in order to correspond to the constantly increasing complexity of the systems being deployed in the fields of Machine and later on Deep Learning.

4.1. Poisonous Attacks on Conventional Machine Learning Systems

One of the earliest techniques emerging is that of Label Poisoning. Its aim is to create mismatched labels or modify them, so they do not correspond to the original data, and thus an erroneous knowledge base is being built. As a consequence, the quality of the contaminated data affects the overall system. The method is also called "label flipping" deriving from the initial binary classifiers of 1 and 0.

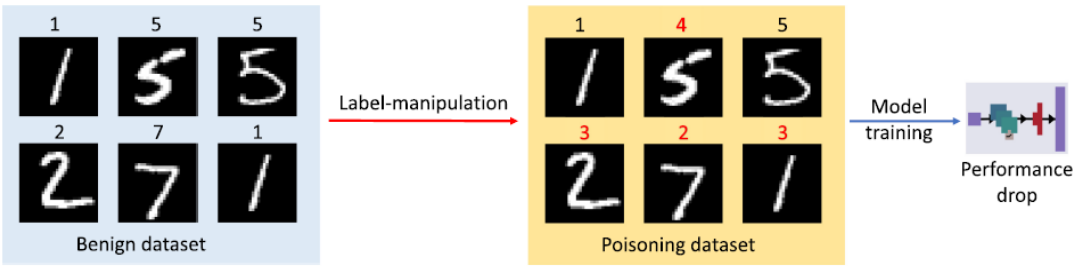


***Figure 27**: Instance of Label Manipulation. Image Credited to [55].*

Consequently, label flipping algorithm enriched with optimization features, for the best optimal choice of labels causing the maximum erroneous output. Thus, this algorithmic development sets the ground for a mathematical problem to be formed and optimized variously.

4.2. Formal Definition of Poisonous Attack – A Bilevel Approach

Formally a poisonous attack can be captured as follows:

$$D_p^* \in \underset{D_p}{\operatorname{argmax}} \mathcal{F}\left(D_p, w^*\right) = \mathcal{L}_1\left(D_{val}, w^*\right) \text{ (1)}$$

$$\text{s.t. } w^* \in \underset{w}{argmin}\, \mathcal{L}_2\left(D_{train} \cup D_p; w\right) \text{ (2)}$$

where D_p is a poisoned dataset, D_{train} is the original one, and D_{val} the validation dataset. $\mathcal{F}$ is the attacker's designed objective function to create poisonous samples and maximize the loss $\mathcal{L}_1$ in the validation dataset with w^* parameters. The second function updates the parameters w^* -whenever the first function finds an optimal solution (best local)- on the augmented poisoned dataset $D_{train} \cup D_p$ [18].

-Notably, according to [18] the aforementioned conceptual idea of bilevel optimization formulating poisoning attack is officially reported for the first time in the work of [58].

4.3. Poisonous Attacks on Deep Learning Systems

Taking a step further and adapting to the newly emerged deep learning systems the research community introduces new methods of poisonous attacks, taking into consideration all the limitations deriving from the complexity of these systems. Thus, approximation of the bilevel optimization problem allows to create new attacks.

- **Gradient-Based Attacks**

  Gradient-Based Attacks present a challenge calculating the gradient of a poisoning point toward the gradient of the objective function, until the poisoning point achieves greater results.

  Assuming that the $\mathcal{F}$ function is differentiable for parameters w and a point x, the required gradient is calculated using the chain rule as follows:

$$\nabla_x \mathcal{F} = \nabla_x \mathcal{L}_1 + \frac{\partial w^T}{\partial x} \nabla_w \mathcal{L}_1 \quad (3)$$

  where $\frac{\partial w}{\partial x}$ denoted the dependence of the classifier parameters on the poisoned data. Furthermore, due to the convexity assumption of $\mathcal{L}_2$ [58] proposed an implicit equation using the **Karush-Kuhn-Tucker (KKT)** conditions instead of the second optimization [Eq. (2) – p.61] and thus [Eq. (3) – p. 62] converts, as follows:

$$\nabla_x \mathcal{F} = \nabla_x \mathcal{L}_1 - (\nabla_x \nabla_w \mathcal{L}_2)(\nabla_w^2 \mathcal{L}_2)^T \nabla_w \mathcal{L}_1$$

  and thus, a two-layer optimization problem transforms into a single-layer constrained optimized problem [17].

  On the gradient-base [59] introduced the reverse gradient optimization, that was the first poisoning attack towards a deep learning model. The method computes the [Eq. (2) – p.61] more efficiently and thus overcomes complexity issues.

- **Gan-based Attacks**

  [60] inspired by generative methods introduced a generator to produce poisoned sample. Training a model so as to learn the probability distribution of adversarial perturbations and then construct poised input is of great importance. The model consists of two components: the generator and the classifier. A random input is

selected from the clean data and the generator creates a poisonous sample. Then, the sample is tested against the validation dataset to the classifier and the weights-parameters are adapted. The results -obtained parameters- return to the generator. The process repeats until the desired result is succeeded. According to the authors a trade-off exists between the time of poisonous data generation and the slightly lower accuracy.

- **Clean-Label Attacks**

  [61] proposed a new strategy attack, namely clean-label, where the attacker knows about the model and its parameters, but nothing with regards to the training data. The attackers embed small imperceptible perturbations to the input data and then feed them to the training data. Thus, the model is erroneously trained and furthermore the poisoned data are unrecognizable from the human eye.

  Authors with **Feature Collision** create poisonous data, as follows:

$$p = \underset{x}{\operatorname{argmin}} \|f(x) - f(t)\|_2^2 - \beta\|x - b\|_2^2$$

  where $f(x)$ is the representation of x is the penultimate layer (before the softmax layer), namely feature space, $\|f(x) - f(t)\|_2^2$ the similarity measure between the poisoned data and the target, and β is a parameter to the constraint $\|x - b\|_2^2$ of poisoned data and to initial input data so as to be imperceptible to the human eye.

- **Model Attacks**

  Model poisonous attacks need no knowledge of training data, instead the adversary targets to the model parameters per se. Thus, confidentiality and privacy concerns are raised. According to [18], [57] not much research has been conducted in this nascent branch of attacks.

5. Backdoor Attacks

An attack firstly appointed (as will be explained below) to the poisonous category, namely backdoor attack or trojan [62], rapidly gains the interest of the research community, as recent data reveal [**Figure 28**]. An attacker may deploy a backdoor attack in a deep learning model by injecting malicious inputs in the training phase of the system's lifecycle, remaining in an idle situation, until it is invoked at the inference stage. However, non-influenced samples will still behave as they should depending on their nature (i.e., classification, recognition).

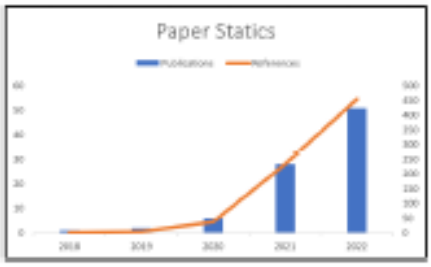


***Figure 28**: Number of Published Papers on the Topic of Backdoor Attacks to Deep Learning Models from 2018 to 2022 of Web of Science, Image Credited to [63].*

At a higher level of view backdoor attacks and poisonous ones appear to have a close affinity, but zeroing in on technical details, any similarity is fading out. In terms of security aspect, backdoor attacks are considered to violate the integrity viewpoint, in contrast to poisonous ones that aim at a general degradation and non-availability of the system, equally termed as denial of service.

Besides, backdoor attacks are considered targeted attacks since the trigger causes the system to misbehave according to the attacker's target class base, while poisonous ones aim to a system's general performance decline irrelevantly of the result. Furthermore, if the malicious trigger is related to the source class the attack may be classified as class-specific, in contrast to class-agnostic where the trigger depends only on the nature of the data (i.e., voice, text).

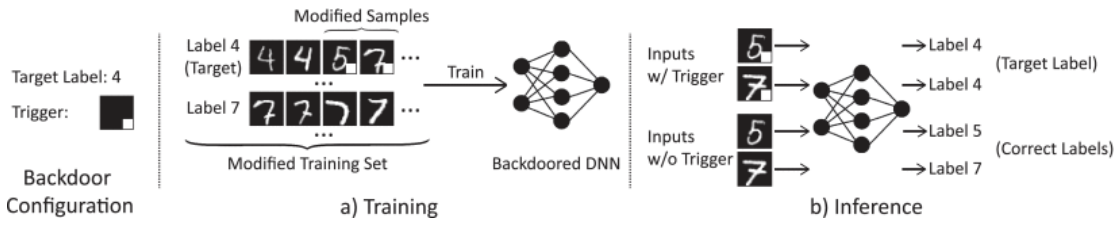


***Figure 29**: An Illustration of Backdoor Attack. Image Credited to [64].*

Though initially the attack focused on the computer vision domain, it was sooner rather than later that expanded to other critical fields (i.e., speech, text). They are further present in variations depending on the trigger, whether it produces the same label or acts separately targeting different labels, on the many arbitrary chosen by the attacker factors, such as shape, position, size as well as on its transparency or invisibility [65]. Notably the trigger in the domain of sound and text, is figured in terms of amplitude of the audio and semantically accordingly.

5.1. Formal Definition of Backdoor Attack

According to [66] for a benign model $\mathcal{F}: \mathcal{X} \to \mathcal{Y}$, a selected malicious output prediction result $\mathcal{R}$, a backdoor attack is to generate (i) a backdoor model $\mathcal{G}: \mathcal{X} \to \mathcal{Y}$, (ii) a backdoor trigger generator $\mathcal{J}: \mathcal{X} \to \mathcal{X}$, which alters a benign input to a malicious input such that:

$$\mathcal{G}(x) = \begin{cases} \mathcal{F}(x), & if\ x \in \{\mathcal{X} - \mathcal{J}(\mathcal{X})\} \\ \mathcal{R}, & if\ x \in \mathcal{J}(\mathcal{X}) \end{cases}$$

-Notably a backdoor attack is evaluated by the following ratios:

- **Clean Data Accuracy** (CDA): defines the proportion of the clean samples (with no trigger) predicted to their ground-truth classes.
- **Attack Success Rate** (ASR): defines the proportion of the samples (with trigger) that are predicted to the attacker targeted classes.

5.2. Taxonomy of Backdoor Attacks

Literature records a variety of backdoor attacks, striking at most of the phases of ML lifecycle, through one or two vulnerable entry-points (attack surfaces or scenarios). A comprehensive survey on this topic [65], reports and categorizes these scenarios into six classes:

- **Code poisoning** refers to the adversary's capability to exploit the tactic practitioners follow, for developing their solutions on top of already released frameworks. Malicious code can be injected into the initial framework, posing severe security threats in terms of contamination and detrimental effects.

- **Outsourcing** refers to the occasion where a practitioner lacks the computation capacity to process a large volume of data and thus turns towards external service providers (Machine Learning as a Service – MLaaS).
- **Pretrained** refers to practitioners who use a 'teacher' model to train their own. Data acquisition and labeling are enormously challenging procedures that require expertise and resources in terms of time and cost. Thus, an adversary can train maliciously a model and then publish it, with all the consequences that entails.
- **Data Collection** refers to the stage of gathering data. Cutting-edge technologies require constantly new data, mostly coming from open sources. Thus, malicious data can be freely available feeding the models that make use of them.
- **Collaborative Learning**[1] refers to the machine learning models with no access to training set, but still input data from many participants, who many a time are not benevolent. A typical example of this consists Google word prediction, that goes through the end user's data [67].
- **Post-deployment** refers to an occasion where an intruder has gained access to an ML system with the aim of altering i.e., the weights of a model loaded in the memory, and thus causing a degradation at the inference stage.

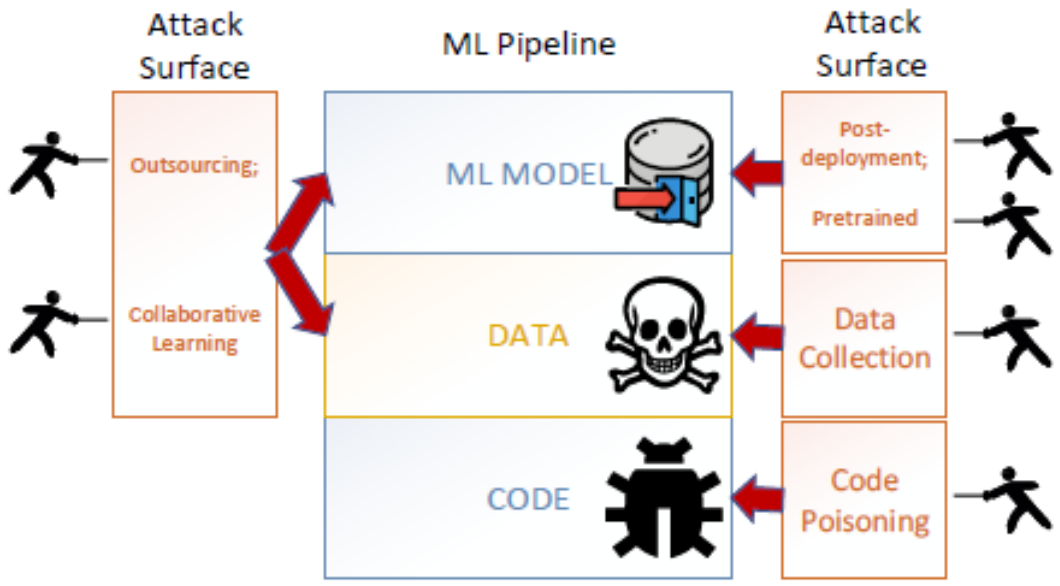


***Figure 30**: Categorized Six Backdoor Attack Surfaces: Each Attack Surface Affects One or Two Stages of the ML Pipeline. Image Credited to: [65].*

[1] Note that a collaborative learning attack surface is out of the scope of this thesis.

Latest literature [68] classifies backdoor attacks into three main categories:

- **Poisoning-only** attacks where an adversary has access only to the training dataset.
- **Training-controlled** attacks where an attacker has privileged access to the training procedure, including the training data and algorithm.
- **Non-poisoning-based** attacks that take place after the deployment, tampering with the core data of ML model loaded directly in the memory, such as the weight values.

Thus, this newly classification clearly shifts away backdoor attacks from the classical term of poisonous, thereafter the emergence of non-poisoning-based attacks. Furthermore, backdoor attacks have extended their scope to almost every stage of ML lifecycle, except the Model test.

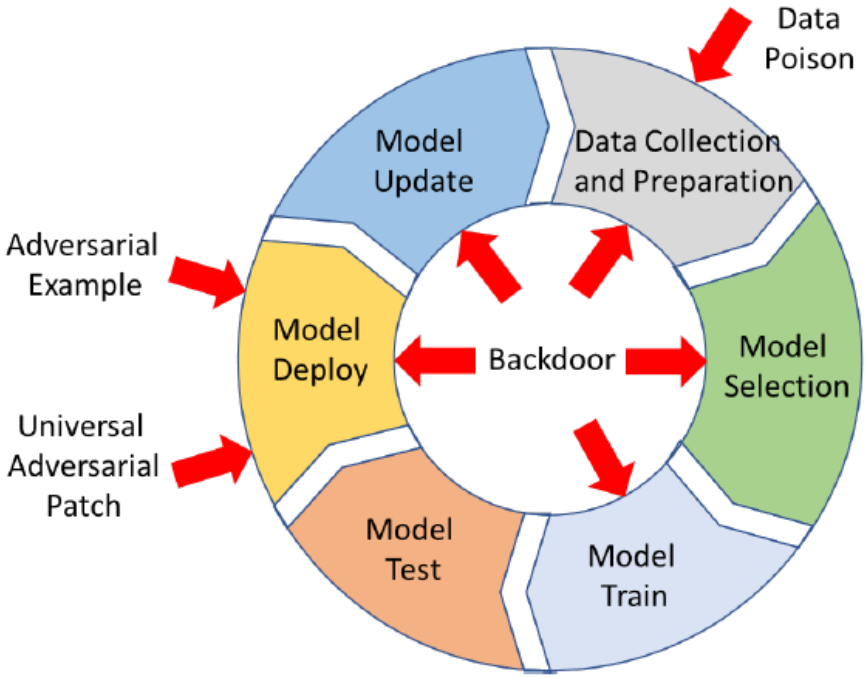


***Figure 31**:Possible Attacks in Each Stage of ML Pipeline. Image Credited to [65].*

5.3. Poisoning-Only

- **BadNets – Firstly emerged backdoor attack**

  BadNets [69] the most representative and firstly explored causative attack, introduces a specific feature only known to the adversary, namely **backdoor trigger**, associated to the target class, remaining idle until it's being invoked. It deploys in a white-box setting, where the adversary takes full responsibility over

the training process and returns the according model or feeds the training data with malicious samples (transfer learning) and then releases it to repositories.

- **Targeted Poisonous Backdoor Attack**

  At around the same time [70] an attack termed as backdoor and more specifically "backdoor poisoning" emerged on the scene. According to the authors, it differed from the previous ones in terms of knowledge of the attacked model and thus firmly was classified as a black-box attack. The attack takes place by inserting a small number of poisonous samples (either with a specific class or more widely with a key pattern) into the training dataset with a preeminent success rate and an ultimate goal of not being distinguishable.

- **Dynamic Backdoor Attack**

  The aforementioned attacks both refer to triggers that are statically stamped onto the sample (in image classification), and thus the location and the place are beforehand known. In [71] authors introduce a dynamic way to create triggers, spotted randomly in the terrain of image with various patterns. They propose three methods to succeed their goal: random backdoor, backdoor generating network (BaN) and conditional backdoor generating network (c-BaN). All these methods subsequently advance in terms of complexity and limitation exceeding.

- **Clean-label Attack**

  Clean-label attack was proposed in [72] based on the observation that it is possible to create a poisonous sample without corrupting its label. The attacker aims to induce a machine learning model producing a specific class. Thus, it is a targeted or multitargeted attack that provides flexibility with an additional vector to cause misclassifications. The vector is stealthier to be recognized since no labels are changed, and furthermore can be directed to the desired class, irrespectively of the base class that it will be applied. However attractive this technique might be, it comes not without drawbacks. A great deal of training samples is required to persuade the model acting maliciously.

5.4. Training-Controlled

- **Trojaning Attack**

  Another attack deploying in a grey-box setting, however more realistic than the previous one, is introduced in [73], where the adversary has access to the target model but not to training/testing data. The authors claim that this is often the case since most Neural Networks are published partly in order to exhibit their supported functionality. The attack consists of three phases: creating the trojan trigger, training the data and retraining the model. Taking into advantage the neurons strongly activated to a particular trigger and with reverse engineering techniques the attack deploys in a real scenario.

- **Blind Code Poisoning Attack**

  Blind Code Poisoning attack proposed in [74] presupposes no access to training data or the process as well as to the execution phase of the code and its results. Thus, it is considered a black-box attack. The adversary implants the malicious code into an ML system, and produces poisonous samples “on the fly”. The calculation of the loss function for the legitimate training samples and the loss function for poisonous ones follows, until they are united through an optimization (multi-objective) process.

5.5. Non-Poisoning-Based

- **Live Trojan**

  Live Trojan is introduced in [75] and its basic notion rests in the knowledge and techniques drawn from typical software attacks. The attack is tampering with a system’s memory (randomizing parameters or setting them to zero) at the run-time, or applying a more targeted patch (finding the parameters), with no further knowledge of the target model, and thus, under this perspective, is considered to be deployed in a black-box setting. Afterwards the attack uses a retraining method, called masked and produces the new trojaned dataset. However, privileged access to the system architecture as well to several other features such as weights and bias parameters is a substantial prerequisite, without which the attack cannot take place, and from that point of view the attack lies in the scope of white-box attack.

- **Adaptive Attack**

A state-of-the-art adaptive attack, was recently published [76] urging the necessity to revise the assertion that models trained on poisonous data tend to learn separable latent representations for clean and malicious samples. That is forming different clusters after projecting samples in the latent space, and therefore presenting a tangible signature. This assumption has been deemed as a natural feature of backdoor attacks and spurred the development of defense methods based on clustering analysis.

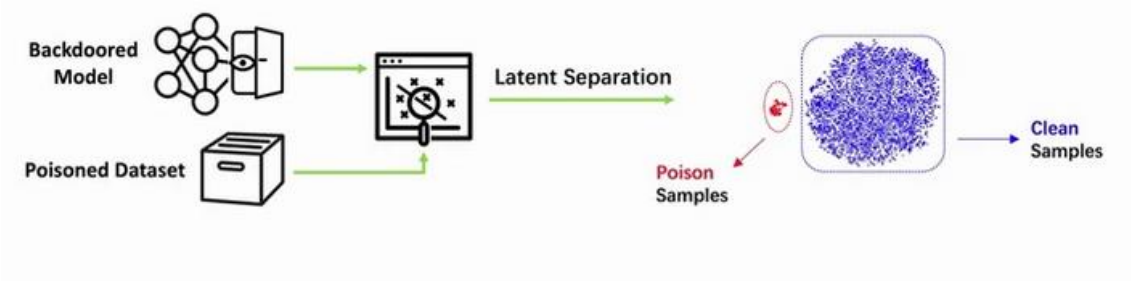


***Figure 32**: Latent Separation. Image Credited to [76] in the virtual presentation (https://iclr.cc/virtual/2023/poster/11430)*

The newly emerged attack aims to refute this assumption with counterarguments and thus put into question the many defense mechanisms. Specifically, the attack effectively minimizes the gap in the latent space between the poisonous and the benign samples, while retaining the attack success rate (ASR) at the same level, with an insubstantial drop at clean accuracy.

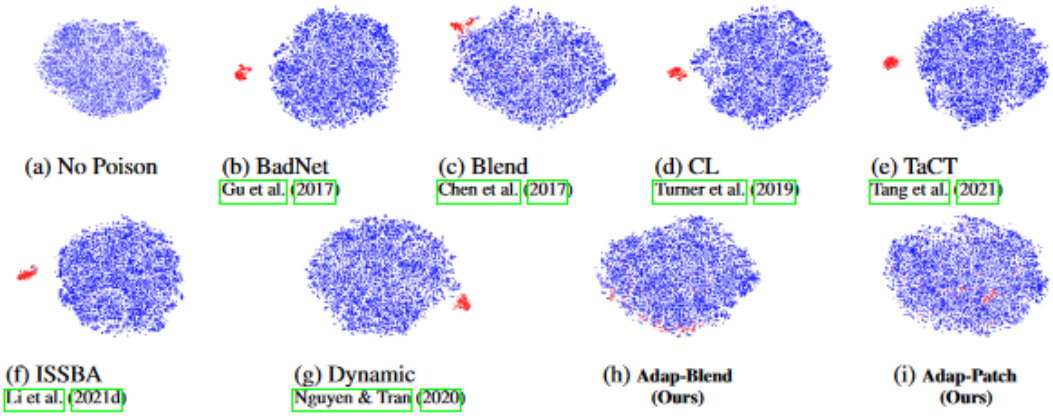


***Figure 33**: Visualization of Latent Separability Characteristic on CIFAR-10. Each Point in the Plots Corresponds to a Training Sample from the Target Class. Caption of Each Subplot Specifies its Corresponding Poison Strategy.*

To Highlight the Separation, All Poison Samples are Denoted by Red Points, while Clean Samples Correspond to Blue Points. Image Credited to [76].

The authors, in order to picture the attack, effectively embed the notions of regularization, asymmetry and diversity into corresponding strategies in the training stage. Data poisoning-based regularization strategy retains the ground truth label to some of the poisonous samples (regularization samples). The trigger planting strategy promotes asymmetry and diversity. Specifically random triggers are implanted to the poisonous samples so as the latter ones are scattered in the latent space. Besides, during the test-time only the original trigger will be used to invoke the attack.

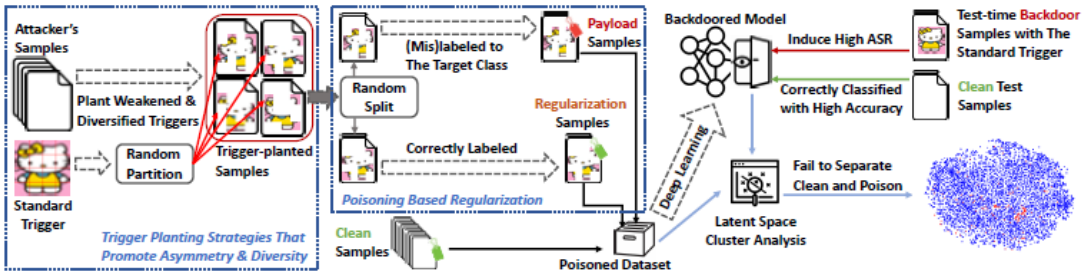


***Figure 34**: An Overview of the Adaptive Backdoor Attack. Image Credited to [76].*

Intuitively, in this framework authors have deployed two illustrations of the attack, namely Adaptive-Blend and Adaptive-Patch to incorporate the projection of the poisonous samples in the latent space to the clean ones.

***Table 3**: Comparison Table for Backdoor Attacks*

Backdoor Attacks	Access Model Architecture	Trigger	ASR	Attack Classification	Specificity	Attack Surface	Strong Aspects
BadNets [69]	White-Box/ Grey-Box	Static	Very High / Medium	Poison-only	Targeted	Outsourcing/ Pretrained	transfer Learning
Targeted Poisonous Backdoor Attack [70]	Black-Box *later on mentioned as Grey-Box [66]	Static Trigger Pattern	Very High 5 samples cause 90%	Poison-only	Targeted	Outsourcing	Indistinguishable samples
Dynamic Backdoor [71]	White-Box	Dynamic	Very High Approximately 100%	Poison-only	Targeted /untargeted	Outsourcing	i) Random backdoors ii)Backdoor Generating network iii) Conditional Backdoor

Backdoor Attacks	Access Model Architecture	Trigger	ASR	Attack Classification	Specificity	Attack Surface	Strong Aspects
Trojaning Attack [73]	Grey-Box	Non arbitary trigger – based on the strongest activated neurons.	Medium Nearly 100%	Training-controlled	Targeted	Pretrained	Transfer learning
Blind Code Poisoning [74]	Black-Box	On the fly poisonous triggers.	High	Training-controlled	Targeted /untargeted	Code-Poisoning	Multi-task learning for conflicting variables – main task and backdoor.
Live Trojan [75]	Black-Box also White-Box (access to model architecture, weights, bias parameters of the network.	Invisible since tampering with the data in memory, can be uninterpretable.	Medium	Non-poisoning-based	Targeted	Post-deployment	Run time attack Randomizing or zeroing parameters Less time consuming.
Clean Label [72]	Grey-Box	More stealthiness	Medium	Poison-only	Single/multiple targeted	Data Collection	No need to identify beforehand the class of the samples to be attacked at test time

Backdoor Attacks	Access	Model Architecture	Trigger	ASR	Attack Classification	Specificity	Attack Surface	Strong Aspects
Adaptive Attack [76]	White-Box			High	Training-controlled	Targeted	Data Collection	State-of-the-art attack aiming to bypass a family of the latest defense mechanisms

5.6. BAERASER – A Defense Against Backdoor Attacks

A novel defense mechanism introduced by [77] succeeds a decline in attack-success-rate against cutting-edge backdoor attacks by 99%. It consists of a two-stage procedure and has been inspired by the law framework of General Data Protection Regulation (GDPR), applying a technique, namely machine unlearning.

The defense mechanism is formulated, as below:

$$\underset{\theta}{argmin}\, \mathcal{L}\left(F_\theta(x_b), \mathcal{Y}_{real}\right) + \lambda\|\theta\|$$

where F_θ is the victim model, x_b the backdoored images, $\mathcal{Y}_{real}$ the true labels, $\mathcal{L}$ the loss function that estimates the prediction error of the victim model and $\lambda\|\theta\|$ is coefficient multiplied by a penalty to restrict the unlearning process. Overall, the mechanism aims to minimize the loss function over the victim model while retaining the accuracy of F_θ.

The defense mechanism reverses the attack procedure as illustrated below and deploys in two-stages. Initially, using a generative model will try to recover the trigger pattern, while it overcomes performance degradation by using entropy maximization. Consequently, using a technique called machine unlearning will eliminate malicious samples and retrain the model.

Furthermore, according to the authors their mechanism is outdoing over previous equivalent, due to the fact being able to deploy on a more realistic environment. Specifically in order to surpass the necessity for a full training dataset, which is often the case for laboratory experiments, they reverse gradient descent into gradient ascent, with an additional weighted penalty parameter, to thwart the disastrous over unlearning.

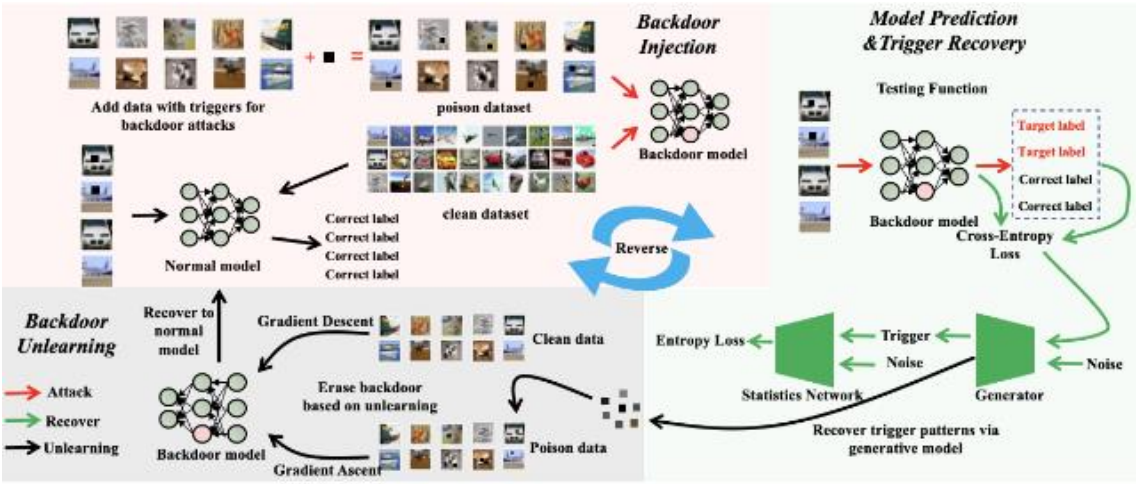


***Figure 35**: The Workflows of Backdoor Inject Attack and Backdoor Erasing Methodology. Image Credited to [77]*

6. Data Vulnerabilities and Threats in the Healthcare Domain (a Non-Technical Approach).

Although it would be of great value to extend this thesis presenting various attacks, in terms of adversarial examples and poisonous-backdoor attacks as well as adversarial strategies, all targeting the healthcare domain in particular, no truly incentives would then be unveiled to grasp the risk for embracing ML-based systems.

Machine Learning and Deep Learning in recent years integrate at a high rate into almost every apparatus, facilitating our lives to a great extent and many a times proving a great alliance in situations of emergency. The healthcare domain has been mostly affected by these evolutions, in various aspects. Diagnostic predictions, image classification, decision support, remote health care management, design proteins and drugs and many more expertise fields lie on AI-based systems.

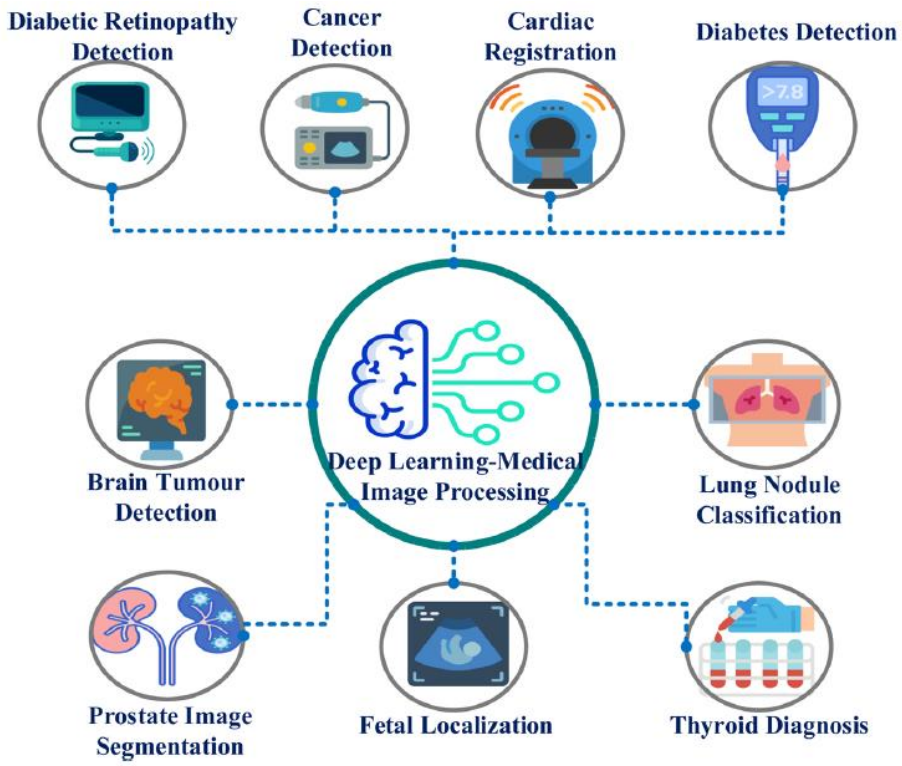


***Figure 36**: Applications of DL in Medical Image Processing. Image Credited to [78].*

But a question still remains unanswered: ***Why should anyone wish to degrade such systems?***

Recent research [79] reveals one of the many truly motivations behind such behaviors using as an example the United States' health care model, that is not far ahead from many others. In particular, it highlights the use of AI-based system in the insurance claims approvals, where trillions of dollars are circulating, among the providers and payers. At the crux of this research, financial motivations overwhelm both sides. Providers are pursuing more claims while payers act in the opposite direction (perturbing the input data such as minimizing the costs or denying medication).

Many types of attackers are mentioned, from novices who commit criminal activities with regards to the AI-systems as a challenge, those who support other groups of attackers, and those acting professionally in order to make profits [80]. Whatever the category of the attacker is, targeting an ML-based system may cause severe impacts and even human losses.

As such further motivations are reported in terms of terrorism attacks, targeting national healthcare systems in order to satisfy their demands. An attack to such an extent may cripple a whole city. The Healthcare domain is a primary infrastructure for every society. It owns a considerable proportion of the market [Figure 37], revealing the tremendous impacts in such scenarios. Whatever the motivation is, every aspirant adversary will try to maximize the impacts of their attack.

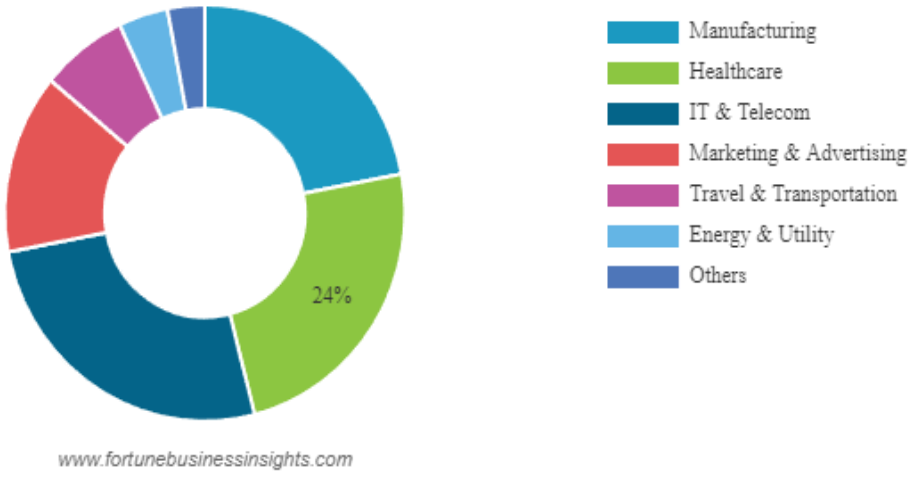


Figure 37 *Global Generative AI Market Share, By Industry, 2022. Image Credited to [81]*

7. Conclusions

In this thesis we conducted a thorough research on attacks aiming at the training and the inference stage of an ML-based system lifecycle. We introduced the necessary background knowledge so as to better understand the attacks from a mathematical point of view where needed. We mentioned the necessity to delineate the attacks under models due to their extent with regards to the adversary's capabilities, so as to be better studied and tackle with all the proper measures.

We explored adversarial attacks, poisonous attacks and backdoor attacks. Each category shares common features but they are inherently different. We mainly remained focused on the technical aspect of these attacks in order to fully understand how they are deploying, at which phase and under what circumstances. We should note that it is not in our intention to create a comprehensive survey on attacks, based on the taxonomies referenced, but rather to provide a roadmap to the evolvement of this topic. In addition, a reference to the corresponding defense mechanisms was made, in order to highlight the difficulties that research community faces.

Finally, a twofold attempt to raise awareness on this topic was made. On the one hand, urging multidisciplinary research communities to make a joint effort to provide adequate reasoning for the existence of many of the aforementioned attacks and so countermeasures to be proposed. On the other hand, except the practitioners, management should be fully aware of the risks using AI-based systems. This warning should only urge all the decision makers to take all the necessary measures in order to protect their systems and not be daunted by the risks.

8. Bibliography

[1] K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *arXiv:1502.01852v1,* 2016.

[2] R. S. Siva Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann and S. Xia, "Adversarial Machine Learning -- Industry Perspectives," *arXiv:2002.05646v3,* 2021.

[3] M. Barreno, B. Nelson, A. D. Joseph and D. J. Tygar, "The security of machine learning," *Springer,* p. 121–148, 2010.

[4] N. Papernot, P. McDaniel, A. Sinha and M. Wellman, "SoK: Towards the Science of Security and Privacy in Machine Learning," *arXiv:1611.03814v1,* 2016.

[5] Y. Kawamoto, K. Miyake, K. Konishi and Y. Oiwa, "Threats, Vulnerabilities, and Controls of Machine Learning Based Systems: A Survey and Taxonomy," *arXiv:2301.07474v2,* 2023.

[6] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. Berkay Celik and A. Swami, "The Limitations of Deep Learning in Adversarial Learning," *arXiv:1511.07528v1,* 2015.

[7] A. C. Serban, E. Poll and J. Visser, "Adversarial Examples - A Complete Characterisation of the Phenomenon," *arXiv:1810.01185v2,* 2019.

[8] S. Dube, An Intuitive Exploration of Artificial Intelligence: Theory and Applications of Deep Learning, Springer Nature Switzerland, 2021.

[9] S. Sun, Z. Cao, H. Zhu and J. Zhao, "A Survey of Optimization Methods from a Machine Learning Perspective," *arXiv:1906.06821v2,* 2019.

[10] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.

[11] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino and H. W. Alomari, "Adversarial Deep Learning: A Survey on Adversarial Attacks and Defense Mechanisms on Image Classification," *IEEE Access,* 2022.

[12] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," *arXiv:1608.04644v2,* 2017.

[13] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient Based Learning Applied to Document Recognition," *IEEE,* 1998.

[14] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009.

[15] J. Deng, R. Socher, L. Fei-Fei, W. Dong, K. Li and L. . Li, "ImageNet: A Large-Scale Hierarchical Image," in *2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR),* 2009.

[16] J. Newsome, B. Karp and D. Song, "Paragraph: Thwarting Signature Learning by Training Maliciously," in *9th International Symposium, RAID 2006,* Hamburg, 2006.

[17] Y. Hu, W. Kuang, Z. Qin, K. Li, J. Zhang, Y. Gao, W. Li and K. Li, "Artificial Intelligence Security: Threats and Countermeasures," *ACM Comput. Surv.,* 2021.

[18] Z. Wang, J. Ma, X. Wang, J. Hu, Z. Qin and K. Ren, "Threats to Training: A Survey of Poisoning Attacks and Defenses on Machine Learning Systems," *ACM,* 2022.

[19] E. Quiring and K. Rieck, "Backdooring and Poisoning Neural Networks with Image-Scaling Attacks," *arXiv:2003.08633v1,* 2020.

[20] M. Barreno, B. Nelson, R. Sears, A. D. Joseph and D. J. Tygar, "Can Machine Learning Be Secure?," *ACM ,* 2006.

[21] "Adversarial Machine Learning 101," MITRE | ATLAS, [Online]. Available: https://atlas.mitre.org/resources/adversarial-ml-101/. [Accessed 28 09 2023].

[22] O. Ibitoye, R. Abou-Khamis, A. Matrawy and M. O. Shafiq, "The Threat of Adversarial Attacks on Machine Learning in Network Security - A Survey," *arXiv:1911.02621v1,* 2019.

[23] Y. He, G. Meng, K. Chen, X. Hu and J. He, "Towards Security Threats of Deep Learning Systems: A Survey," in *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 2022.

[24] N. Dalvi, P. Domingos, Mausam, S. Sanghai and D. Verma, "Adversarial Classification," 2004. [Online]. Available: https://homes.cs.washington.edu/~pedrod/papers/kdd04.pdf.

[25] D. Lowd and C. Meek, "Adversarial Learning," *ACM,* 2005.

[26] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus, "Intriguing properties of neural networks," *arXiv:1312.6199v4,* 2014.

[27] H. Li, Y. Fan, F. Ganz, A. Yezzi and P. Barnaghi, "Verifying the Causes of Adversarial Examples," *IEEE Xplore,* 2021.

[28] X. Zhang, X. Zheng and W. Mao, "Adversarial Perturbation Defense on Deep Neural Networks," *ACM Computing Surveys,* vol. 54, no. 8, 2021.

[29] S. Gu and L. Rigazio, "Towards deep neural netrwork architectures robust to adversarial examples," *arXiv:1412.5068v4,* 2015.

[30] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing Adversarial Examples," *arXiv:1412.6572v3,* 2015.

[31] N. Ford, J. Gilmer, N. Carlini and E. D. Cubuk, "Adversarial Examples Are a Natural Consequence of Test Error in Noise," *arXiv:1901.10513v1,* 2019.

[32] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran and A. Madry, "Adversarial Examples Are Not Bugs, They Are Features," *arXiv:1905.02175v4,* 2019.

[33] T. Tanay and L. Grifflin, "A boundary tlting perspective of the phenomenon of Adversarial Examples," *arXiv:1608.07690v1,* 2016.

[34] S. Han, C. Lin, C. Shen, Q. Wang and X. Guan, "Interpreting Adversarial Examples in Deep Learning: A Review," *ACM,* 2023.

[35] X. Yuan, P. He, Q. Zhu and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *IEEE Xplore,* 2019.

[36] A. Rozsa, E. M. Rudd and T. E. Boult, "Adversarial Diversity and Hard Positive Generation," *arXiv:1605.01775v2,* 2016.

[37] J. Li, S. Ji, T. Du, B. Li and T. Wang, "TEXTBUGGER: Generating Adversarial Text Against Real-world Applications," *arXiv:1812.05271v1,* 2018.

[38] T. Zheng, C. Chen, J. Yuan, B. Li and K. Ren, "PointCloud Saliency Maps," *arXiv:1812.01687v6,* 2019.

[39] N. Papernot, P. McDaniel and I. Goodfellow, "Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples," *arXiv:1605.07277v1,* 2016.

[40] S.-M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, "DeepFool: a simple and accurate to dool deep neural networks," *arXiv:1511.04599v3,* 2016.

[41] N. Papernot and P. McDaniel, "Adversarial Examples in Machine Learning AIWTB 2017," https://www.psu.edu/, 17 05 2017. [Online]. Available: https://www.youtube.com/watch?v=NrGMvTZxAwU. [Accessed 02 08 2023].

[42] A. Kurakin, I. J. Goodfellow and S. Bengio, "Adversarial examples in the physical world," *arXiv:1607.02533v4,* 2017.

[43] S.-M. Dezfooli-Moosavi, A. Fawzi, O. Fawzi and P. Frossard, "Universal adversarial perturbations," *arXiv:1610.08401v3,* 2017.

[44] S. Baluja and I. Fischer, "Adversarial Transforamtion Networks: Learning to Generate Adversarial Examples," *arXiv:1703.09387v1,* 2017.

[45] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi and C.-J. Hsieh, "ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models," *arXiv:1708.03999v2,* 2017.

[46] M. Cisse, Y. Adi, N. Neverorva and J. Keshet, "Houdini: Fooling Deep Structured Prediction Models," *arXiv:1707.0537v1,* 2017.

[47] J. Su, D. Vasconcellos Vargas and K. Sakurai, "One Pixel Attack for Fooling Deep Neural Networks," *arXiv:1710.08864v7,* 2019.

[48] X. Wei, B. Pu, J. Lu and B. Wu, "Visual Adversarial Attacks and Defenses in the Physical World: A Survey," *arXiv:2211.01671v5,* 2023.

[49] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno and D. Song, "Robust Physical-World Attacks on Deep Learning Visual Classification," *arXiv:1707.08945v5,* 2018.

[50] M. Sharif, S. Bhagavatula, L. Bauer and M. K. Reiter, "Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition," in *Proceedings of the 2016 ACM SIGSAC*

Conference on Computer and Communications Security, Vienna, Austria, Association for Computing Machinery, 2016, p. 1528–1540.

[51] T. B. Brown, D. Mane, M. Abadi and J. Gilmer, "Adversarial Patch," *arXiv:1712.09665v2,* 2018.

[52] A. Sharma, Y. Bian, P. Munz and A. Narayan, "Adversarial Patch Attacks and Defences in Vision-Based Tasks: A Survey," *arXiv:2206.08304v1,* 2022.

[53] H. Zhu, S. Zhang and K. Chen, "AI-Guardian: Defeating Adversarial Attacks using Backdoors," in *2023 IEE Symposium on Security and Privacy (SP)*, San Francisco, 2023.

[54] B. Biggio, B. Nelson and P. Laskov, "Poisoning Attacks against Support Vector Machines," in *29 th International Conference on Machine Learning*, Edinburgh, 2012.

[55] Z. Tian, L. Cui, J. Liang and S. Yu, "A Comprehensive Survey on Poisoning Attacks and Countermeasures in Machine Learning," *ACM,* 2022.

[56] B. Biggio and F. Rolia, "Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning," *Pattern Recognition,* 2018.

[57] A. E. Cina, K. Grosse, A. Demontis, S. Vascon, W. Zellinger, B. A. Moser, A. Oprea, B. Biggio, M. Pelillo and F. Roli, "Wild Patterns Reloaded: A Survey of Machine Learning Security against Training Data Poisoning," *ACM,* 2023.

[58] S. Mei and X. Zhu, "Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[59] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu and F. Roli, "Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization," *arXiv:1708.08689v1,* 2017.

[60] C. Yang, Q. Wu, H. Li and Y. Chen, "Generative Poisoning Attack Method Against Neural Networks," *arXiv:1703.01340v1,* 2017.

[61] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras and T. Goldstein, "Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks," *arXiv:1804.00792v2*, 2018.

[62] L. Yiming, Y. Jiang, Z. Li and S.-T. Xia, "Backdoor Learning: A Survey," *arXiv:2007.08745v5,* 2022.

[63] Y. Li, S. Zhang, W. Wang and H. Song, "Backdoor Attacks to Deep Learning Models and Countermeasures: A Survey," *IEEE Open Journal of the Computer Society,* 2023.

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

[65] Y. Gao, B. G. Doan, Z. Zhang, S. Ma, J. Zhang, A. Fu, S. Nepal and H. Kim, "Backdoor Attacks and Countermeasures on Deep Backdoor Attacks and Countermeasures on Deep," *arXiv:2007.10760v3,* 2020.

[66] S. Li, S. Ma, M. Xue and B. Zi Hao Zhao, "Deep Learning Backdoors," in *ecurity and Artificial Intelligence. Lecture Notes in Computer Science*, Springer Nature Switzerland, 2022, pp. 313-334.

[67] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon and D. Ramage, "FEDERATED LEARNING FOR MOBILE KEYBOARD PREDICTION," *arXiv:1811.03604v2,* 2019.

[68] Y. Gao, Y. Li, L. Zhu, D. Wu, Y. Jiang and S.-T. Xia, "Not All Samples Are Born Equal: Towards Effective Clean-Label Backdoor Attacks," *Pattern Recognition,* vol. 139, no. 109512, 2023.

[69] T. Gu, K. Liu, B. Dolan-Gavitt and S. Garg, "BadNets: Evaluating Backdooring Attacks on Deep Neural Networks," *IEEE Access,* pp. 47230-47244, 2019.

[70] X. Chen, C. Liu, B. Li, K. Lu and D. Song, "Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning," *arXiv:1712.05526v1,* 2017.

[71] A. Salem, R. Wen, M. Backes, S. Ma and Y. Zhang, "Dynamic Backdoor Attacks Against Machine Learning," *arXiv:2003.03675v2,* 2022.

[72] M. Barni, K. Kallas and B. Tondi, "A new backdoor attack in CNNs by training set corruption without label poisoning," in *2019 IEEE,* 2019.

[73] Y. Liu, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang and X. Zhang, "Trojaning Attack on Neural Networks," in *Network and Distributed Systems Security (NDSS) Symposium*, San Diego, 2018.

[74] E. Bagdasaryan and V. Shmatikov, "Blind Backdoors in Deep Learning Models," *arXiv:2005.03823v4,* 2021.

[75] R. Costales, C. Mao, R. Norwitz, B. Kim and J. Yang, “Live Trojan Attacks on Deep Neural Networks,” *arXiv:2004.11370v2,* 2020.

[76] X. Qi, T. Xie, Y. Li, S. Mahloujifar and P. Mittal, “Revisiting the assumption of latent separability for backdoor defenses,” in *Eleventh International Conference on Learning Representations*, Kigali, ICLR 2023.

[77] Y. Liu, M. Fan, C. Chen, X. Liu, Z. Ma, L. Wang and J. Ma, “Backdoor Defense with Machine Unlearning,” *arXiv:2201.09538v1,* 2022.

[78] S. Bhattacharya, P. K. Reddy Maddikunta, Q.-V. Pham, T. R. Gadekallu, S. R. Krishnan S, C. L. Chowdhary, M. Alazab and M. J. Piran, "Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey," *Sustainable Cities and Society,* vol. 65, no. 102589, 2020.

[79] G. S. Finlayson, D. J. Bowers, J. Ito, L. J. Zittrain, L. A. Beam and S. I. Kohane, “Adversarial attacks on medical machine learning,” *PMC,* vol. 363, no. 6433, pp. 1287-1289, 2019.

[80] S. Chng, H. Y. Lu, A. Kumar and D. Yau, “Hacker types, motivations and strategies: A comprehensive framework,” *Computers in Human Behavior Reports,* no. 100167, 2022.

[81] T. /. A. Maket, “Generative AI Market Size, Share & COVID-19 Impact Analysis, By Model (Generative Adversarial Networks or GANs and Transformer-based Models), By Industry vs Application, and Regional Forecast, 2023-2030.,” fortunebusinessinsights, 08 2023. [Online]. Available: https://www.fortunebusinessinsights.com/generative-ai-market-107837. [Accessed 04 10 2023].

[82] M. Kuribayashi, “Adversarial Attacks,” in *Frontiers in Fake Media Generation and Detection. Studies in Autonomic, Data-driven and Industrial Computing*, Singapore, Springer, 2022, pp. 63-79.

[83] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li and T. Goldstein, “Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses,” *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* 02 2023.

[84] P.-Y. Chen and C.-J. Hsieh, Adversarial Robustness for Machine Learning, London: Academic Press - Elsevier, 2023.