

Characteristic Analysis of Methods to Counteract Adversarial Machine Learning Attacks

DIMITRIYA MIHAYLOVA
Faculty of Telecommunications,
Technical University of Sofia,
8 Kl. Ohridski Blvd.,
BULGARIA

Abstract: - In the era of technological evolution and digital transformation, machine learning and artificial intelligence take a significant part of intelligent systems' development. Together with the integration of smart and autonomous operations in everyday life, novel security issues arise. One such a malicious threat against the security of machine learning models is the adversarial machine learning (AML) attack. It consists in alteration of the original data samples used to train or test the model with indistinguishably small feature modifications that results in inaccurate processing outcome. In this paper, the most significant strategies to counteract AML attacks are studied and compared through several characteristics that have main influence on their successful implementation in different realization scenarios.

Key-Words: - adversarial machine learning, defense strategies, evasion attack, adversarial training, Gaussian data augmentation, gradient masking, defensive distillation, feature squeezing, defense-generative adversarial network.

Received: May 2, 2026. Revised: June 19, 2026. Accepted: July 21, 2026. Published: October 22, 2026.

1 Introduction

The evolution of contemporary wireless networks exposes them to the risk of a variety of attacks that correspond to the nature of the novel technologies involved. Among the emerging technologies that are expected to be integrated into the upcoming sixth generation (6G) wireless systems, machine learning (ML) is recognized as an essential part with its abilities to provide sophisticated services and automated applications that require minimal human intervention. Up to now, ML models have proven their advantageous performance in many different domains such as pattern recognition, computer vision, image processing, natural language processing, and network security. A system incorporating ML algorithms can benefit from improved accuracy and efficiency, while reducing the time needed for its operation. However, when a malicious intervention known as an adversarial machine learning (AML) attack is triggered during the testing or training phase of the ML model, the system's performance can be significantly degraded.

The AML threat consists in the contamination of the data used for training or testing the ML algorithm, called respectively a poisoning or evasion attack [1]. When the aim of the intruder is to confuse the model during training, i.e. prior to it being learned, there are three possible strategies to

realize poisoning: to modify the features of the samples in the training dataset, to insert new samples in order to change the data distribution, or to corrupt the logic of the ML model. If the testing phase is affected, the model itself is not manipulated, instead slight data modifications are carried out, in order to confuse the proper output decision of the model.

An important issue considering the evasion attack is the ability to find the smallest disturbance that needs to be added to the testing data in order to cross the decision boundary and result in an erroneous output. The most common approach used to discover the perturbation vector relies on solving an optimization problem. Being a measure of the prediction errors of the ML algorithm, the loss function and its maximization define the basic objective of the optimization task. There are two main strategies for evasion depending on the available information for the ML model at the attacker side: gradient-based, which considers knowledge of the model and solves the optimization by a certain distance metric, and gradient-free, that does not require any model information and finds the optimal solution by using only the confidence scores of the model's predictions [2].

As well as classifying the AML attacks according to the intruded operational phase, another

important categorization takes into consideration the amount of knowledge at the attacker. On the one hand, in the case where the algorithm for crafting the adversarial example used for AML attack is applicable with no information about the target ML model and the dataset, the attack is referred to as black box. On the other hand, if the attack can be produced only by exploiting full knowledge of the model and the data involved, it is described as white box. When a certain amount of knowledge is needed to create an adversarial example, the attack falls into the group of gray-box [3]. However, it's by no means rare in the literature for the disparity between black-box and gray-box attack scenarios to vanish, thus both types of intervention are grouped under the joint term gray/black-box attacks.

Since its first description in [4], the vulnerability of ML systems to AML attacks has been attracting significant research attention. Many studies propose and investigate different approaches to create adversarial samples with minimal perturbation of the original input data. Among the most noteworthy examples are the Fast Gradient Sign Method (FGSM) [5], the Jacobian-based Saliency Map Attack (JSMA) [6], DeepFool attack [7], and Carlini & Wagner (C&W) attack [8], all of them falling into the category of white-box techniques. A large number of other attacking strategies incorporate the ideas of these baseline approaches and represent modifications of them. One example is the Zeroth Order Optimization (ZOO) attack [9] that is motivated by the C&W attack [8] and is envisioned in literature as both a black-box and gray-box method, due to the fact that although no knowledge of the model is needed the attack makes use of the information about predicted outcomes that correspond to specific input samples [3]. Other examples are the Projected Gradient Descent (PGD) method [10] and the Basic Iterative Method (BIM) [11], both inspired by the FGSM [5].

Defensive mechanisms are also developed to make ML models resistant to AML attacks. In this paper, several of the most used defense strategies against AML attacks are studied and compared according to some of their significant properties that can influence their successful implementation in varying application scenarios.

The contributions of the paper are as follows:

- The most widely used strategies that are able to combat AML attacks are studied. The defense methods are systematically presented in the interests of clarity and ease of subsequent analysis.

- The efficient operation of the defense methods against AML attacks is examined according to several performance parameters, using values from the literature reviewed. The parameters differ according to the method studied and their values are strongly dependent on the experimental conditions – both attack and ML model dependencies influence the performance of the defense methods.
- A comparative analysis of the defense methods against AML attacks is conducted. The analysis is two-fold – as reported by the performance parameters, and according to the most significant properties that may influence the implementation of the defense strategies.

The rest of the paper is organized as follows: in Section 2 the most widespread methods to generate adversarial samples are briefly studied; in Section 3 some of the most frequently used strategies to counteract AML attacks are described; in Section 4 the methods are compared; Section 5 concludes the paper.

2 Adversarial Samples Generation

As the successful defense against AML attacks is strongly dependent on the exact method adopted for adversarial samples generation, in this section four approaches to construct the perturbation vector are examined. The methods are chosen since they serve as a base of novel attacking methodologies.

2.1 Fast Gradient Sign Method (FGSM)

One of the basic approaches for adversarial sample generation is the FGSM [5]. As its name suggests, the method relies on gradient estimation in order to find the smallest perturbation needed to confuse the decision of ML models. The cost function, $J(\theta, X, Y)$, which represents a measure of the errors in ML decisions, is used as a point of reference, where θ denotes the current parameters of the model, X and Y are the input and output of the model, accordingly. The gradient $\nabla(\cdot)$ of the cost function is computed following the backpropagation equation in (1), and the resultant value represents the linear perturbation vector η , whose slope direction to follow is determined by the sign of the gradient. Furthermore, a weighting factor ε is used to control the perturbation size in the most appropriate range.

$$\eta = \varepsilon \operatorname{sign}(\nabla J(\theta, X, Y)). \quad (1)$$

The need for gradient estimation makes FGSM suitable in white-box scenarios when the attacker is

familiar with the cost function of the targeted ML model. However, among the most advantageous characteristics of the method are its simplicity and the limited operational time to compute the perturbation.

2.2 Jacobian-based Saliency Map Attack (JSMA)

Another well-known approach for creating AML probes consists of the construction of the Jacobian matrix, containing the partial derivatives of model output variables with respect to the corresponding inputs [6]. The Jacobian matrix $J_F(\mathbf{X})$ of the ML model's function $\mathbf{F}(\mathbf{X})$ with multidimensional input vector $\mathbf{X}$ is computed according to Eq. (2):

$$J_F(\mathbf{X}) = \frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}. \quad (2)$$

The results of forward derivative computations are recorded into a saliency map, used to obtain the perturbation needed for the desired output of the model, making the JSMA method a strong instrument for targeted AML attacks' generation.

Differing from FGSM, which evaluates gradients of the cost function, JSMA explores the function used to teach the model on its regular tasks.

The saliency map replenishment could be provided by following different strategies – to decrease the input parameters in order to obtain incorrect output decisions of the model under attack; or to reinforce the features that deviate the model outcome at the direction of the desired target result.

2.3 Deep Fool (DF)

The authors of [7] propose to discover the smallest perturbation value needed to confuse the model by an iterative approach. The DF method still relies on gradient estimations, but it also includes subsequent computations of the disruptive vector value. Its assessment follows the idea of discovering the tiniest distance between the input and the threshold that separates the different outcome ranges, represented by the level set at zero of the classifier function. Therefore, the input is projected onto the plane of the decision threshold, which is geometrically approximated as a hyperplane in the case of binary classification or a polyhedron in the case of multiclass classification. Then, the minimum perturbation needed, r_i , calculated by Eq. (3), is expressed by the Euclidean distance $\|\cdot\|_2$ for binary classification, or l_p norm, in the general case, to the linearized classification function $f(x_i)$ for certain input value x_i during the i -th iteration.

$$r_i = -\frac{f(x_i)}{\|\nabla f(x_i)\|_2} \nabla f(x_i) \quad (3)$$

The DF method extends the abilities of FGSM in finding the smallest perturbation that, added to the input data, leads to an erroneous output of the model. The reason lies in the iterative search algorithm implemented in DF; however, the aforementioned benefit is at the expense of enlarged operational time and overall method complexity.

2.4 Carlini & Wagner (C&W)

Representing a leading method for AML sample generation, the C&W attack is focused on the more difficult yet powerful scenario of targeted adversarial threats. The method can be applied by using different norms for perturbation size evaluation. In all its implementations, an optimization problem must be solved, namely to define the smallest perturbation while achieving the largest increase in the loss function. To alleviate the complexity of the approach, the optimization function is a subject of box constraints. Different variations of the objective function are precisely evaluated; hence, the most effective one is outlined, represented by Eq. (4):

$$o(x + \delta) = \max_{i \neq t} \{Z(x + \delta)_i\} - Z(x + \delta)_t - k, \quad (4)$$

where the objective function is denoted by $o(\cdot)$, whose argument aggregates the original input x and the perturbation vector δ . The $\max(\cdot)$ function, used in Eq. (4), returns its largest argument, while $Z(\cdot)$ designates the logits' output. i denotes the truly existing output of the input x , and t corresponds to the desired target class of the attack. k is a regularization parameter that affects the attack's confidence.

The authors in [8] study different attacking strategies to determine the target class of the AML attack – one that is arbitrarily decided, another that accepts the easiest target to be accomplished, and a third one aimed at the most challenging misclassified target class.

Despite its increased complexity, C&W is known for its powerful operation that successfully withstands some of the most prominent techniques to resist AML attacks. Thanks to its transferability, C&W serves as a reference for the development of gray/black-box attacking strategies.

2.5 Properties Comparison of Methods to Generate Adversarial Examples

Whether to choose one or another of the methods for AML sample generation is a question of the abilities of the adversary, its main goal, and the specifics of the exact system model under attack. Therefore, a comparison of the major characteristics of each of the methods studied is of great importance for the proper selection of the most appropriate strategy to craft the samples. Table 1 summarizes the main contribution of our initial study in [12], where the main properties that affect the choice of AML attack generation method according to the implementation scenario are compared.

As the results in Table 1 show, all the methods are suitable explicitly for white-box attack scenarios; additional modifications and transformations of the methods will be required to adopt them to black-box approaches. Though all of them can be used to attack both the training and the testing phase of the ML model, only DF is eligible for untargeted misclassification results. The iterative approach applied in all three JSMA, DF, and C&W methodologies makes the aforementioned attacks time-consuming and computationally expensive. Thus, the FGSM remains the most widely exploited method in both attacks and defenses, due to its fast operation and simplicity.

Table 1. Properties comparison of AML attack generation methods

Time consumptive	X	✓	✓	✓
Computationally expensive	X	✓	✓	✓
Iterative search	X	✓	✓	✓
Gradients estimation	✓	✓	✓	✓
White-box	✓	✓	✓	✓
Black-box	X	X	X	X
Targeted outcome	✓	✓	X	✓
Untargeted outcome	X	X	✓	X
Training phase	✓	✓	✓	✓
Testing phase	✓	✓	✓	✓
	FGSM	JSMA	DF	C&W

Source: [12]

3 Defense Strategies against AML Attacks

The most widely used strategies against AML attacks can be classified into six main categories, as illustrated in Figure 1. A brief description of each defense strategy is given in the following subsections.

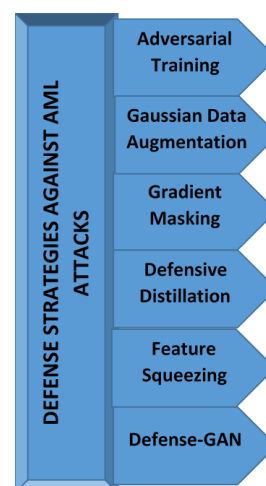


Fig. 1: Categorization of the types of defensive methods against AML attacks

Source: created by the authors

3.1 Adversarial Training (AT)

One of the simplest and most widely known defense strategies against AML attacks is adversarial training. As its name suggests, it comprises an expansion of the training process – besides training the ML model with the original dataset, an additional perturbed version of the original data is also used for training. The augmentation of the training dataset with adversarial examples expands the capabilities of the model to categorize the samples correctly during the testing and deployment phases. Hence, by adversarial training, the ML model can learn to resist AML attacks without changing its architecture and logic. A brief description of the steps involved in adversarial training is given in Figure 2.

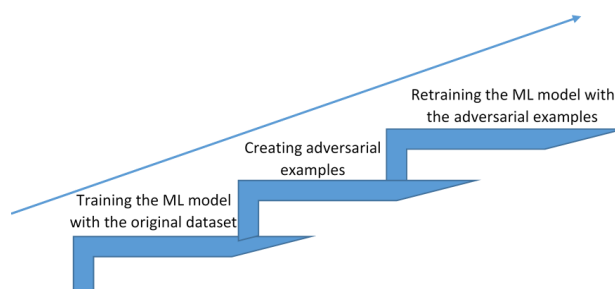


Fig. 2: A description of the adversarial training approach against AML attacks

Source: created by the authors

Different adversarial training strategies exist in the literature. While the authors of [5] demonstrate the decreased error rate of the model when trained against attacks created by the Fast Gradient Sign Method, in [10] adversarial training generated by two different attacking strategies – Fast Gradient Sign Method and Projected Gradient Descent- is suggested to improve the resistance of the model against various AML attacks. However, the more generalization included in the ML training, the higher the complexity; therefore, a trade-off between computational power and adversarial training against a large variety of attacks is needed.

A main problem with this defensive method is that the model becomes resistant to the specific types of AML attacks whose algorithm is used to craft the samples for adversarial training. In a case where the system is perturbed with samples created through another algorithm different from the one that the system was trained with, adversarial training is not effective against these types of AML attacks [2], [3]. Another issue concerning adversarial training consists of the degraded performance of the ML model when no attack is initiated [13].

3.2 Gaussian Data Augmentation (GDA)

A defense strategy that also relies on model generalization by augmenting the dataset but eliminates the need for retraining is suggested in [14] and exhaustively studied on different classifiers in [3]. This technique modifies the input data by simply adding Gaussian noise to the original samples, as shown in Figure 3. Using the perturbed dataset to train the ML model enables it to predict correctly even in the presence of AML attacks.

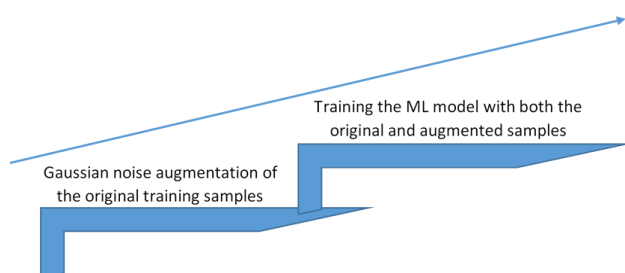


Fig. 3: A description of the Gaussian Data Augmentation approach against AML attacks
Source: created by the authors

Though Gaussian data augmentation is very similar to adversarial training, it gives the opportunity to strengthen the model in multiple directions rather than only a few, as is typical with adversarial training. Moreover, due to the smoothing behavior of the loss function of the model trained with Gaussian data augmentation, accurate

predictions of original examples are not subject to deterioration. Another advantage of the defense technique proposed in [3] is that it is universal to different types of AML attacks regardless of the algorithm applied to generate the adversarial examples.

3.3 Gradient Masking (GM)

Since a large number of AML attack approaches rely on gradient estimation to produce perturbed examples, a promising countermeasure against them aims to hide the gradient information of the ML model. This is based on the observation that non-differentiable models do not suffer from AML attack vulnerabilities. Therefore, masking the gradients impedes the crafting of adversarial samples that determine the direction of the perturbation vector from the knowledge of the slope of the ML model objective function.

One possible implementation of gradient masking is proposed in [15] and shown in Figure 4. The algorithm involves saturation of the neural network and, most specifically, of the sigmoid activation function employed. In this way, a nonlinear training regime is maintained, leading to improved robustness of the ML model, due to the resultant flat functions describing the relation between inputs and outputs, i.e., a large change in the input is followed by an insignificant adjustment in the output.

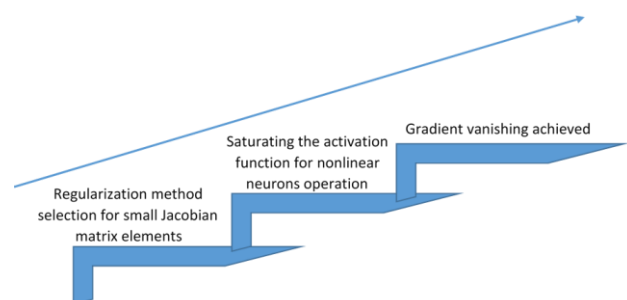


Fig. 4: An example implementation of the gradient masking defense method against AML attacks
Source: created by the authors

3.4 Defensive Distillation (DD)

The distillation technique proposed in [16] can also be adopted as a defensive mechanism against AML attacks. Its original implementation aims at alleviating the complexity of training a smaller ML model by using a larger and more powerful one, referred to as a teacher. First, the teacher is trained with the original dataset, and then its outputs are extracted and used to train the smaller model, known as a distilled model. The main purpose of this approach is to avoid overfitting of the distilled

model by using the softer labels from the teacher's outputs in its training phase.

A modification of the original distillation, called defensive distillation [17], trains two models of equal size and introduces the use of an additional constant parameter – distillation temperature - that reflects the last layer of the neural network, i.e., the SoftMax layer. The higher the temperature constant, the more confident the ML predictions are. The summarized algorithm is illustrated in Figure 5.

Unlike adversarial training, whose reliable operation relates to the specific algorithm applied in the attack, defensive distillation gets into the area of attack-agnostic techniques, as it is not dependent upon the particular attack type. A main drawback of the method consists in the demand for an additional model that has to be trained, i.e., as with adversarial training, retraining is a required step.

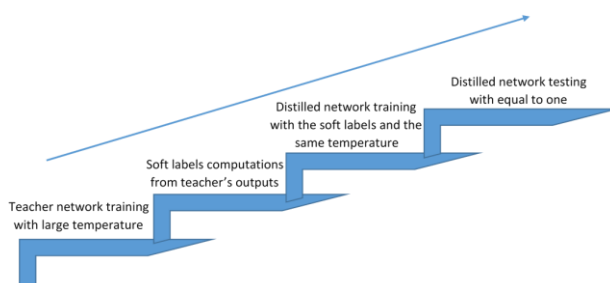


Fig. 5: A description of the defensive distillation approach against AML attacks

Source: created by the authors

3.5 Defense-Generative Adversarial Network (Defense-GAN)

Another class of defensive methods is based on the concept of exploiting an external model, such as a Generative Adversarial Network (GAN), to improve robustness against white-box and black-box AML attacks. Originally proposed in [18], GANs can be used to make the model able to generate a novel training dataset using the original one and preserving its statistics. This can be done using two distinct neural networks – a generator, which produces new data samples, and a binary classifier, called the discriminator. During the training phase, both models are taught from the original training dataset – the generator learns how to produce novel samples from the original ones by the addition of a random vector, while the discriminator learns how to differentiate between the original samples and the ones produced by the generator.

One possible implementation of GANs against AML attacks is given in [19] and is presented in Figure 6.

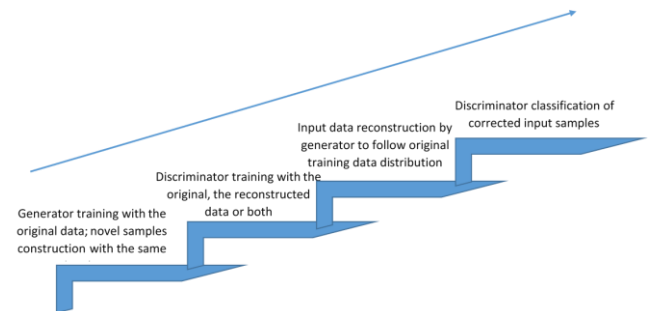


Fig. 6: An example implementation of the Defense-GAN approach against AML attacks

Source: created by the authors

The authors of [19] propose to use the generator for reconstructing the inputs of the test phase before feeding them into the discriminator. The generator is able to minimize the testing data error because it is already trained to create samples with the distribution of the original training data. Since the distribution of the adversarial samples differs from that of the data, the generator model is able to identify and correct them before feeding them into the discriminator classifier.

The main drawbacks of Defense-GAN are the need for an additional ML model and the enlarged training procedure needed to attain satisfactory performance.

3.6 Feature Squeezing (FS)

Another strategy that can be implemented in the ML model to mitigate the influence of AML attacks requires specific pre-processing that helps to reduce the features of the input data and, in this way, optimizes it. Considering that a significant part of the features that appear in the input data are not necessary, by compressing these features, data can be represented with a substantial reduction in complexity.

The method can be implemented in different variations, two of them being proposed in [20] – the first decreases the number of bits used for a specific feature, e.g., the color bits; the second maps several inputs to a single input, using smoothing filtering techniques. However, the common algorithm, shown in Figure 7, performs a comparison between the outputs that correspond to both the original input and the squeezed one. If both outputs differ by more than a threshold value, adversarial perturbation is detected.

Although this defense technique contributes to the generalization of the ML model and thus avoids overfitting, eliminating the least significant features can alter the decision boundary, leading to misclassifications of the original examples.

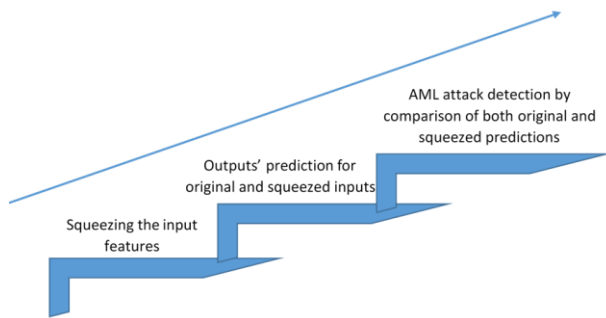


Fig. 7: Feature squeezing defense against AML attacks

Source: created by the authors

4 Comparison of Defense Methods against AML Attacks

In order to assess the performance of the model and its robustness against AML attacks, statistical evaluation metrics are usually adopted. Accuracy and type I and type II errors are often the subject of study, but, in addition to these, ROC and PR curves, which represent the interconnection between different statistical evaluation metrics, are also powerful evaluation tools.

The successful operation of defensive methods and their robustness are heavily dependent on several factors – the type of the ML model, the database used for training and testing, the ML hyperparameter tuning, and the type of AML attack initiated [3]. For example, in the network security domain, the ML models that fulfill clustering and classification tasks are those that suffer most from AML vulnerabilities [2]. Meanwhile, the different behavior of the models depending on the integrated dataset is experimentally verified in many studies considering distinct AML attacks and defenses [10], [1], [2].

Due to the large number of dependencies that influence the robustness of each method to defend against AML attacks, the comparison of their performance in equal conditions is left for future work. Instead, in this paper, the performance parameters of the defense methods under study obtained from the literature review and thus corresponding to different implementation scenarios are presented in Table 2. In addition, the methods are compared according to several other important features that can affect their employment in varying domains and applications. The results of the comparison are briefly presented in Table 3, where if a feature characterizes a certain method, it is denoted by ✓; otherwise, ✗ means that the property is not relevant for the method.

All results in Table 2 correspond to white-box attack scenarios. Some of the attacks depend on a parameter ϵ known as adversarial noise norm [19]. This is a scaling factor representing a measure of the attack's influence. Larger values of ϵ make the AML attack stronger, and correspondingly more visible and easier to detect.

For the different defense strategies, the performance parameters shown in Table 2 vary between accuracy, error rate, and attack success rate and are given for defense and no defense scenarios to show the effect of the method on ML model performance. As already discussed, the methods' efficiency is influenced by many characteristics related to the ML model and the attack scenario. Accordingly, the type of ML model, the dataset used for training, the incorporated attack, and some defense or attack parameters that affect the results are also introduced in the table. While in some papers, the defense strategy is studied on the MNIST dataset, others include results for CIFAR10, both being among the most studied in the field of AML attacks.

Table 2. Accuracy of the discussed defense strategies against AML attacks

Defense method and conditions	Attack type and conditions	Experimental dataset	Performance parameter	Non-defended	Defended
AT with FGSM [5]	FGSM, $\epsilon=0.25$	MNIST	Error Rate = 89.4%	Error Rate = 17.9%	
AT with FGSM [15]	FGSM, $\epsilon=0.25$	MNIST	Accuracy = 5.62%	Accuracy = 83.83%	
AT with FGSM [19]	FGSM, $\epsilon=0.3$	MNIST	Accuracy = 21.7%	Accuracy = 65.1%	
AT with FGSM [19]	C&W, L_2 norm	MNIST	Accuracy = 14.1%	Accuracy = 7.7%	
GDA 10 samples; $\sigma = 0.3$ [14]	FGSM, minimum perturbation	MNIST	Accuracy = 52.07%	Accuracy = 73.96%	
GM [15]	FGSM, $\epsilon=0.25$	MNIST	Accuracy = 5.62%	Accuracy = 98.45%	
DD distillation temperature = 20 [17]	JSMA	MNIST	Attack Success Rate = 95.89%	Attack Success Rate = 1.34%	
DD distillation temperature = 20 [17]	JSMA	CIFAR10	Attack Success Rate = 89.9%	Attack Success Rate = 16.76%	
GAN [19]	FGSM, $\epsilon=0.3$	MNIST	Accuracy = 21.7%	Accuracy = 98.8%	
GAN [19]	C&W, L_2 norm	MNIST	Accuracy = 14.1%	Accuracy = 98.9%	
FS 1-bit reduction of pixel's colour [20]	FGSM	MNIST	Accuracy = 54%	Accuracy = 92%	
FS 4-bits reduction of pixel's colour [20]	FGSM	CIFAR10	Accuracy = 15%	Accuracy = 21%	

Source: created by the authors

Since AT was one of the first defenses against AML attacks reported in the literature, it is broadly examined and used as a reference to compare the performance of other defense strategies. For that reason, its efficiency obtained from four different to the FGSM AT approach, the disparity in accuracy

values reported in [15] and [19] is striking. The reason is not only the choice of the scaling factor ϵ , but also the model architecture, the regularization criteria, the activation function, and the scale of the neural network. The best results reported for AT improve the accuracy of the non-defended model by 78.21 percentage points (pps) – being 5.62% in the non-defended scenario, the accuracy rises to 83.83 %.

Comparing the results of all the methods whose accuracy is examined, the greatest improvement is realized by the GM strategy, which achieves 98.45%, representing a 92.83 pps increase when compared to that of the model under FGSM attack without any defense. The FS results under the CIFAR10 dataset show less of an accuracy improvement, at only 6 pps, when compared to the clear model results without any defense. AT under C&W attack is remarkable for being the only scenario whose accuracy drops out when defense is applied. The reason for this is the fact that the AT is accomplished by FGSM adversarial training, while the attack follows C&W adversarial examples generation. As generalization is not a property typical for AT, other AML attacking scenarios with samples crafted by a different algorithm may result in a degradation of the system performance.

The most typical features of a defense strategy against AML attacks are summarized in Table 3. While all the methods are appropriate against white-box attacks, only GDA, Defense-GAN, and FS are resistant to black-box AML. Nonetheless, the specific attack scenario is of major importance – an example is that although AT and DD are generally resistant to white-box AML interventions, they are not reliable against one of the most common attacks of this kind, namely C&W [8], [19]. Another example is the conclusion made in [21], where it is theoretically and experimentally proven that a small modification of JSMA makes DD unable to combat it even though it falls in the group of white-box attack scenarios.

The results in Table 3 show that three of the methods studied, namely GDA, DD, and Defense-GAN, can be applied against all types of AML attacks, while the other three are useful only for specific attack types. AT and DD are the strategies that involve an additional training phase, which enlarges the training overhead and the time needed to train the model. Due to possible modifications of the decision boundary caused by the essential operation of these methods, AT and FS may degrade the predictions of the ML model when non-adversarial input samples are processed. The two methods that require an additional ML model to be

trained are DD and Defense-GAN. Hence, these two methods operate at the cost of increased expense and system complexity. One unfavorable property observed is that most of the strategies under study can result in overfitting the ML model; only Defense-GAN and FS are resistant to this phenomenon.

Table 3. Comparison of the discussed defense strategies against AML attacks

White-box AML resistance	✓	✓	✓	✓	✓	✓
Black-box AML resistance	✗	✓	✗	✗	✓	✓
Attack-agnostic method	✗	✓	✗	✓	✓	✗
Retraining necessity	✓	✗	✗	✓	✗	✗
Degrading original performance	✓	✗	✗	✗	✗	✓
Additional ML model necessity	✗	✗	✗	✓	✓	✗
Possible overfitting	✓	✓	✓	✓	✗	✗
Computationally expensive	✓	✗	✗	✓	✓	✗
High training latency	✓	✗	✗	✓	✓	✗
High testing latency	✗	✗	✗	✗	✓	✗
	AT	GDA	GM	DD	GAN	FS

Source: created by the authors

The most computationally expensive methods are AT, DD, and Defense-GAN. Due to the need for crafting adversarial samples before feeding them into the training procedure, AT increases the computational cost of the ML model. Moreover, retraining the model with adversarial samples makes the training latency at least twice that of the one without AT defense [20]. As DD requires training of an additional model, the computational cost and latency of the training phase are undoubtedly increased [14]. Similar to the DD method, Defense-GAN includes training of two distinct models – generator and discriminator, which increases both the computational cost of the method and the processing time during training [19]. Testing time with Defense-GAN is also increased due to the action of fixing the distribution of samples by the generator at this stage before classification.

Since only simple Gaussian noise is added to the inputs or a simple pre-processing is required in GDA, GM, and FS, and no retraining is needed, these defenses come with almost no additional cost and latency.

5 Conclusion

In this paper, six promising strategies to mitigate the influence of AML attacks against ML models are analyzed. The techniques are compared according to several of their most important characteristics that can affect the successful implementation of the defensive strategies in different scenarios. The results of the comparison show that none of the methods surpasses the others in all the criteria studied, i.e., a trade-off between various features is needed, and the choice of one strategy over the other has to correspond to the specific application scenario.

The methods are also compared through literature review performance parameters. However, these parameters are obtained in different experimental environments; hence, an objective comparison is not straightforward. Therefore, a comparison of the efficiency of the methods by evaluating their robustness and accuracy in equal conditions is left for a future experimental study, where the defense techniques can be applied in various types of ML models and architectures, and trained with different datasets. Moreover, their resistance to various AML attacks can also be studied.

Acknowledgement:

This work was supported by the research project KII-06-KOCT/13/22.05.2024 “COST Action CA22168” of the Bulgarian National Science Fund.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

- [1] P. Bountakas, A. Zarras, A. Lekidis, and C. Xenakis, “Defense strategies for Adversarial Machine Learning: A survey”, *Computer Science Review*, vol. 49, art. no. 100573, 2023. DOI: 10.1016/j.cosrev.2023.100573.
- [2] O. Ibitoye, R. Abou-Khamis, M. elShehaby, A. Matrawy, M. O. Shafiq, “The Threat of Adversarial Attacks against Machine Learning in Network Security: A Survey,” *Journal of Electronics and Electrical Engineering*, vol. 4, no. 1, pp. 16-59, 2025. DOI: 10.37256/jee.4120255738.
- [3] H. Jmila and M. I. Khedher “Adversarial machine learning for network intrusion detection: A comparative study”, *Computer Networks*, vol. 214, art. no. 109073, 2022. DOI: 10.1016/j.comnet.2022.109073.
- [4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, “Intriguing properties of neural networks,” *International Conference on Learning Representations*, Banff, AB, Canada, 14–16 April 2014. DOI: 10.48550/arXiv.1312.6199.
- [5] I.J. Goodfellow, J. Shlens, C. Szegedy, “Explaining and harnessing adversarial examples”, *International Conference on Learning Representations*, San Diego, CA, USA, 7–9 May 2015. DOI: 10.48550/arXiv.1412.6572.
- [6] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. Berkay Celik, A. Swami, “The limitations of deep learning in adversarial settings”, *IEEE European symposium on security and privacy (EuroS&P)*, Saarbruecken, Germany, March 21-24, 2016, pp. 372-387. DOI: 10.1109/EuroSP.2016.36.
- [7] S.-M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, “DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks, ” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2574-2582. DOI: 10.1109/CVPR.2016.282.
- [8] N. Carlini and D. Wagner, “Towards Evaluating the Robustness of Neural Networks,” *2017 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, May 22-26, 2017, pp. 39-57. DOI: 10.1109/SP.2017.49.
- [9] 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”, in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, Dallas, TX, USA, November 3, 2017, pp. 15-26. DOI: 10.48550/arXiv.1708.03999.
- [10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks”, 2017, *arXiv preprint arXiv:1706.06083*. DOI: 10.48550/arXiv.1706.06083.

- [11] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint* arXiv:1607.02533, 2016. DOI: 10.48550/arXiv.1607.02533.
- [12] D. Mihaylova, "Baseline Adversarial Machine Learning Attacks - A Comparative Study," *60th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST 2025)*, Ohrid, North Macedonia, 26-28 June, 2025. DOI: 10.1109/ICEST66328.2025.11098269.
- [13] M. Sadeghi and E. G. Larsson, "Physical adversarial attacks against end-to-end autoencoder communication systems," *IEEE Communications Letters*, vol. 23, no. 5, pp. 847–850, 2019. DOI: 10.1109/LCOMM.2019.290469.
- [14] V. Zantedeschi, M.-I. Nicolae, and A. Rawat, "Efficient defenses against adversarial attacks," in: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, in: AISec '17*, ACM, Dallas, TX, USA, November 3, 2017, pp. 39–49. DOI: 10.1145/3128572.3140449
- [15] A. Nayebi and S. Ganguli, "Biologically inspired protection of deep networks from adversarial attacks," 2017, *arXiv preprint* arXiv:1703.09202. DOI: 10.48550/arXiv.1703.09202.
- [16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv preprint* arXiv:1503.02531. DOI: 10.48550/arXiv.1503.02531.
- [17] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in: *2016 IEEE Symposium on Security and Privacy (SP)*, IEEE, San Jose, CA, USA, 22-26 May, 2016, pp. 582–597. DOI: 10.1109/SP.2016.41.
- [18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014. DOI: 10.5555/2969033.2969125.
- [19] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting Classifiers against Adversarial Attacks using Generative Models," *arXiv 2018*, arXiv:1805.06605. DOI: 10.48550/arXiv.1805.06605.
- [20] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: detecting adversarial examples in deep neural networks," in: *Network and Distributed System Security Symposium*, San Diego, CA, USA, 18–21 February 2018. DOI: 10.48550/arXiv.1704.01155.
- [21] N. Carlini and D. Wagner, "Defensive distillation is not robust to adversarial examples," 2016, *arXiv preprint* arXiv:1607.04311. DOI: 10.48550/arXiv.1607.04311.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The single author, Dimitriya Mihaylova, was responsible for the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This work was supported by the research project KII-06-KOCT/13/22.05.2024 "COST Action CA22168" of the Bulgarian National Science Fund.

Conflict of Interest

The author has no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US