**International Journal of Emerging Research in Engineering and Technology**
Pearl Blue Research Group | Volume 4, Issue 2, 53-63, 2023
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V4I2P106

*Original Article*

# Attack Surface Reduction in Deep Learning Pipelines Using Model Hardening and Data Sanitization

Sandeep Phanireddy
Sr, Product Security Engineer, USA.

***Abstract -*** *This paper outlines that DL systems are prevalent and crucial in numerous industries, such as finance, healthcare, and autonomous systems. However, the wide application of DL models gives rise to these models facing various threats, such as adversarial examples, data poisoning, and model inversion attacks. This paper mainly discusses minimizing the attack surface of deep learning pipelines, including model hardening and data sanitization. Looking at the common DL models built, we examine all the weaknesses and the varied attack surfaces in the conventional approaches. We then provide a comprehensive approach that uses robust training methods, defensive distillation, adversarial training, and input check mechanisms. Pre-emptive measures, namely outlier detection, input purification and certified data pipeline, are used to solve adversarial manipulations. A new approach of the proactive (hardening) and reactive (sanitization) strategies is put forward where automation is the key factor, keeping latencies as low as possible. It is further evaluated with benchmark datasets, namely MNIST, CIFAR-10, and ImageNet, across various attacks such as FGSM, PGD and backdoor attacks. It is shown that objective values related to the resilience of the algorithm increased, as well as the accuracy of the model in cases when it was under attack and the number of detected attacks. Finally, we discuss the expected performance penalty in incorporating the model hardening and data sanitization steps and highlight how our framework is feasible for real-world environments. Possible improvements are focused on attack detection automation, applying hardening through runtime and threat intelligence feeds based on artificial intelligence. It will thus help create more secure, reliable, and trustworthy AI systems, thus satisfying the research gap in the proposal.*

***Keywords -*** *Attack Surface Reduction, Model Hardening, Data Sanitization, Adversarial Robustness, Defensive Distillation.*

## 1. Introduction

### *1.1 Needs of Attack Surface Reduction in Deep Learning*

The importance of security of deep learning models in recent years has been applied in different areas such as self-driving cars, healthcare systems and finance systems, and they become important in our daily lives. [1-4] Defenses are known to be one of the biggest issues in adversarial attacks in which small, subtle modifications to the input can cause serious misclassification or, even worse, failure. The following vulnerabilities affect deep learning systems, making attack surface reduction a key for providing proper and reliable machine learning systems: In this part, there are specific aspects that demonstrate the problem area of attack surface reduction in Deep learning, namely:

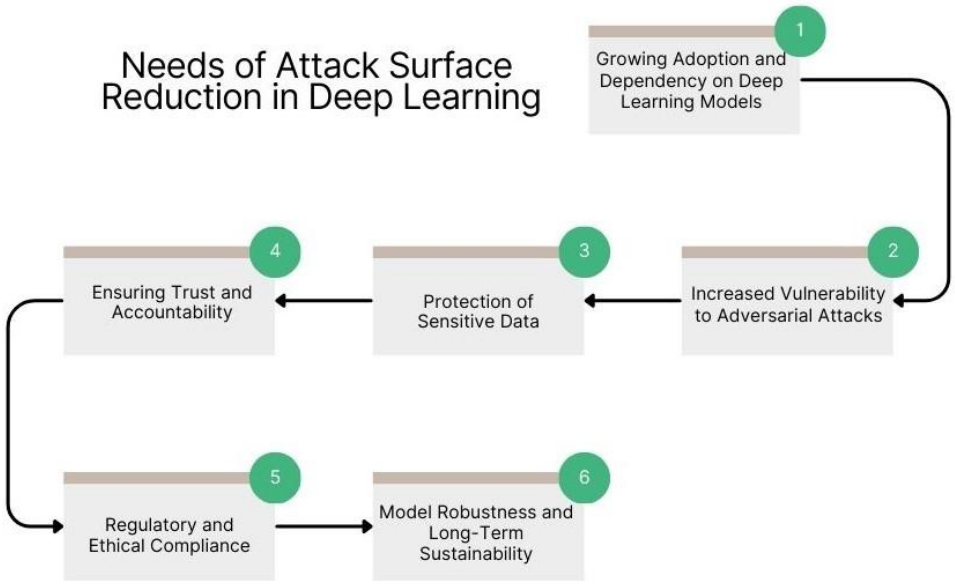


**Fig 1: Needs of Attack Surface Reduction in Deep Learning**

- **Growing Adoption and Dependency on Deep Learning Models:** Deep learning models are widely adopted for industry applications nowadays. For instance, self-driving cars use computer algorithms and deep neural networks for analysing data from occurrence sensors necessary for driving rules decisions, and medical equipment uses it for making diagnostic prognoses. As the dependency on such models increases, so does the basis for motivation of anyone who would specifically target the model at hand for flaws. In this case, should the attack surface not be contained, these systems and their relations are vulnerable to adversarial manipulation and consequences, including safety risks in self-driven machine fraud in the banking industry.
- **Increased Vulnerability to Adversarial Attacks:** Since deep learning models are trained to predict outcomes based on input variables, they are very sensitive to slight changes in those input variables, a problem referred to as adversarial vulnerability. Evaluating a minor change to the input that human eyes cannot detect, the model will either fail to classify the given data or give a wrong classification, making the model prone to the FGSM, PGD or backdoor attacks. These attacks could, in general, lead to data distortion, model malfunctioning, and manipulation of the known vulnerabilities in models. Decreasing the attack surface makes it difficult for adversaries to design such attacks, which makes the model more secure.
- **Protection of Sensitive Data:** Most deep learning applications are used in sectors where data is considered confidential, for example, hospital records, individual details, and the user's bank details. If adversarial attacks are to be effective, not only can the model's accuracy be altered, but also the data can be threatened in two ways it can be stolen or contaminated. Thus, it will be better to minimize the quantity of exposure of the attacked surface in order to protect the confidentiality and the integrity of the data in the process. Methods such as data white-washing and model fortification may be employed to avoid the adversary from entering wrong entries or finding ways of extracting sensitive data from the model.
- **Ensuring Trust and Accountability:** With the uses of AI involved in more critical decisions across industries, the accuracies of such applications play a critical role in such decision making. It becomes crucial for stakeholders, including health management, financial organizations or departments, or even law enforcement agencies, to be assured that the system they rely on will not be vulnerable to adversarial interference or persuasion. Otherwise, people start losing their trust in such systems, which, in turn, would cause them to limit the use of such products. By improving the level of attack surface, more accountable models are developed since they are constructed to withstand several threats that may affect their ability to make accurate predictions, hence enhancing confidence in such models.
- **Regulatory and Ethical Compliance:** The growth of deep learning models has raised robust control and transparency rules related to the data collected, model, and fairness. If unchecked, adversarial attacks can result in such violations, especially the data privacy regulation and safe use of self-driving cars regulation. For instance, certain attacks on medical models could cause misdiagnosis, thus making the models disadvantageous for those violating health laws. Likewise, if different agents take undesirable actions to manipulate the models, it may result in instability in the financial system or even fraud. Measures for defense in and around the deep learning systems are needed to meet the guidelines set by these regulations and uphold the use of ethical AI practices.
- **Model Robustness and Long-Term Sustainability:** It is, therefore, paramount that deep learning models continue to be secure in synthetic settings and live conditions since adversarial operations may change from time to time. This is the point of entry where the adversaries can extend an attack on the model. Suppose there are many potential entry points that are not safeguarded effectively. In that case, models will still be vulnerable to constantly emerging threats, thus compromising its stability in the long run. Preventing and diminishing the attack surface using forms like adversarial perturbation, distillation and data highlighting means that the model can perform well in future attacks with diverse methodologies.

### *1.2 Threat Landscape in Deep Learning*

As with other ML models, deep learning models harbor a number of vulnerabilities at different stages of their life cycles. This paper will explore the vulnerability threats in deep learning, from gathering data to making predictions or inferences. Every stage will also present various risks the adversary can capitalise on to affect the system. [5-8] The following is the highlight of the threats in relation to each stage:

- **Data Stage:** Among all threats, one can distinguish the most essential threat in the data stage as data poisoning. Concerning this type of attack, the attacker uses incorrectly labeled data that are strategically inserted into the training dataset to cause the model to be suboptimal or generate wrong predictions. These poisoning can be made in such a way that the model will learn some wrong patterns or biased patterns from the given inputs. For instance, instead of uploading quality photos of faces, an attacker may upload skewed data, making the system perform poorly or be discriminative. Since the deep learning models require large sets of data for learning, poisoning attacks at this level can cause long-term damage at the system level, which is why data sanitization methods and anomalies must be controlled.
- **Training Stage:** In the case of models under training, extraction and evasion attacks are most effective. In a model extraction attack, the adversary attempts to learn more about the model to get information regarding the architectures, parameters or internal mechanisms of a model. This is done by feeding the model with samples and using the outcome to estimate the model The advantage of this technique is that it is more practical as it only requires approximate

answers of the model. When the adversary acquires sufficient information within the preserve vicinity, he can reverse engineer the model or devise better ways to attack it. Evasion of attacks targets modifying the input data in such a way that it is incorrectly classified during the testing or even during the model deployment. Adversaries can manipulate the inputs to be very close to normalized by humans even as they are malicious to the model due to its sensitivity to small variations. These attacks highlight two approaches that can be used in training a model to mitigate such adversarial attacks: Adversarial training and Defensive distillation.

- **Inference Stage:** The final stage of making a model is identifying or predicting new, unseen data, which is known as the inference stage. This is the most vulnerable stage to adversarial attacks where the adversaries present alterations of the input data in a way that the model is likely to make wrong predictions. These attacks are normally carried out in such a way that a human cannot identify them, thus being very destructive. These adversarial examples are created by perturbing the inputs with FGSM and PGD techniques and later with a step named project image. Malicious perturbations threaten security because they can significantly decrease the model's reliability when used in applications such as self-driving cars and healthcare. When using models as they are integrated into critical systems, it is important to make the models resilient during inference, and methods such as adversarial training and other optimization approaches come in handy to guard against attacks. The proposed framework of the deep learning pipeline comprises four stages that are susceptible to various risks. This must be met by combining defensive techniques to protect data and the resulting training and inference against adversarial manipulation, poisoning, and extractors.

### *1.3 Model Hardening and Data Sanitization*

Hardening and sanitization of models and data are two essential ways that help to make deep learning models secure and resistant to attacks. These aim to lower vulnerabilities of the model at different pipeline phases and make the model more secure during the attack. On the other hand, model hardening is the process through which the model is made less prone to be easily compromised by an adversary in the stages of training and inference. One of them is adversarial training, the basic idea of which is using clean images and the images that can be obtained after adding a small perturbation. It assists the model to become trained to detect and categorize adversarial examples properly thus making the model to be more resistant. One of them is defensive distillation, where a second model is trained to work on the softened output of the first model to decrease its susceptibility to input variations. This process makes the model more robust from the attack and also relieves the model from overfitting under the attack. Other preventive measures for model hardening include robust optimization that tries to minimize the maximum loss of the model while at the same time protecting the model from hostile conditions.

On the other hand, data sanitization is aimed at processing the input data to ensure that they will not negatively affect the learning process in case they are malicious or poisoned. To ensure that no anomalies or poisoned data points are introduced in the model, two preprocessing methods of outlier detection are used: Isolation Forest and k-NN (k-Nearest Neighbors). Input preparation methods, including autoencoders, can also be used to ensure that any noise added to the original data is removed; thus, only clean data used in training is used. Since these adversarial samples can affect the model's learning process, these data sanitization techniques tend to improve both the purity and quality of the model. When combined, model hardening and data sanitization represent a valid strategic solution capable of enhancing the model's resilience and the quality of the data on which the deep learning system is built to defuse adversarial threats.

## 2. Literature Survey

### *2.1 Deep Learning Vulnerabilities*

It is well acknowledged that DL models are sensitive to adversarial examples that are specifically designed to deceive the model to create an output of the attacker's choice while remaining almost imperceptible to a human observer. [9-12] the first to show that such inputs that are slightly malicious can be crafted to make the DL models err. This has raised a major security concern in many neural networks, which was once again expounded by this discovery. Progressing from it, Carlini and Wagner established other subsequent attack techniques that were previously effective defenses, such as distillation defense. Their work subsequently benchmarked the effectiveness of DL systems and demonstrated that most of the proposed defenses failed when faced with adversarial perturbations.

### *2.2 Model Hardening Techniques*

Researchers have proposed different categories of model hardening strategies to mitigate adversarial attacks to increase robustness. A brief explanation of such a method is defensive distillation, which involves training a model in a way that it provides the final probability distribution of the result, which is soft rather than distinct and, therefore, the system is not likely to be affected by slight changes that an attacker might introduce through gradient-based techniques. Another approach is the so-called adversarial training, where the model is optimized on clean as well as adversarially modified samples. Doing this makes the model cope with incomprehensible inputs as it is trained on a broader range of data. Another approach referred to as robust optimization is training a model with a specific view of minimizing the worst-case scenario while keeping degradation of performance checked in unfavorable conditions. Collectively, all of these methods are designed to reduce the controllability of DL classes by modifying their training or objective functions.

### *2.3 Data Sanitization Methods*

Aside from changing the model's architecture or training process, data sanitization is the prevention strategy for DL models. Wang et al. proposed certified defenses that try to ensure the inclusion of clean data at the start of the learning process in an attempt to eliminate data poisoning attacks. Other pre-processing techniques that fall under data sanitization is Outlier detection, which aims to eradicate any peculiar and hazardous data. Some examples of successful anomaly detection techniques include Isolation Forest, which identifies anomalies as it isolates them from the rest of the data; k-Nearest Neighbor (k-NN) based approaches by pinpointing inputs that are distant from the known data distributions. These techniques offer a layer of protection, resulting in the model not being trained on tainted or suspicious data.

### *2.4 Comparative Analysis*

It is, therefore, possible to compare these defense mechanisms with the applicability of each and their strengths and weaknesses. Adversarial training is considered one of the best ways due to its ability to create powerful robustness against vast attacks. Though, it has the disadvantage that generating adversarial samples and training on them entails a large amount of computational power. In contrast, defensive distillation is an even lighter and more efficient method that can easily be implemented with negligible added expenditure. However, this makes them vulnerable when attacked in new and intensified ways like the ones advanced by Carlini and Wagner. Data sanitization can also be considered a completely different approach to the problem of cleaning training data as it involves taking preventive measures to clean data. Although this makes it possible to minimize model contamination, it also disregards possibly valuable information that may otherwise seem odd. Each technique thereby provides a trade-off of the results' security, time and accuracy compared to the actual image.

## 3. Methodology

### *3.1 Framework Overview*

Thus, the proposed hybrid security framework will adopt both the model hardening as a proactive measure and data erasures as a reactive measure to make the system more secure against the adversary's attack. [13-15] The framework is a sequential type, which implies that every step is developed on the previous one to establish a sound defense mechanism.

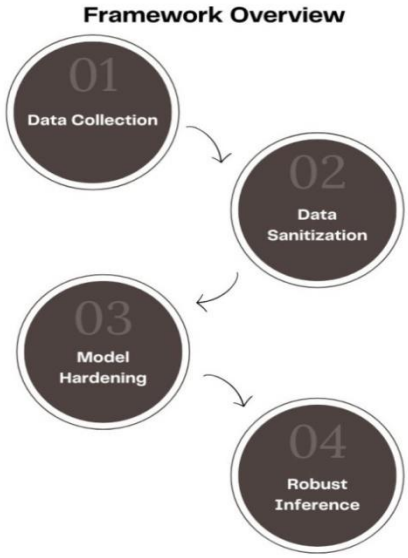


**Fig 2: Framework Overview**

- **Data Collection:** The first step in the framework is the collection of necessary datasets where data with the help of which the programme will be trained and evaluated are collected. At this stage, the emphasis is placed on the fact that the data used for the colored domain should cover the real-life cases in which the model will be used, diverse. Good quality data is crucial for both the accuracy and robustness of the models, whereby adversarial attacks focus on errors or weaknesses in the data. The gathered information comprises the basis on which the sanitization and hardening processes will have to be performed.
- **Data Sanitization:** After that, in the process of data preprocessing, data cleansing is performed to eliminate any threats contained in the data. This carries out the removal of contaminated or malicious inputs designed to deceive the learning process of the model. Outlier removal is employed using some of the techniques, including Isolation Forest or k-NN and the certified defence mechanisms to ensure the dataset's baseline. It is a layer of defense which acts as an early means to prevent some attacks, which might otherwise corrupt the learning process, from being effective.
- **Model Hardening:** The next step after data integrity is model hardening, whereby some techniques are employed to enhance the stability of the model. Some of them are as follows: Adversarial training involves adding adversarial

examples into the training data or defensive distillation, which minimizes the model's ability to be influenced by minor inputs. In model hardening, the major aim is to condition the model to come out strongly for an attack and not be affected by adversarial forces. This stage involves further improving the model by training it more on the clean inputs while making it more resilient to the perturbed inputs.

- **Robust Inference:** The final step of running the inference phase uses a strong and resistant model to make predictions. At this stage, the model performs the task of interpretation of new data influx whether it be clean or adversarial. Together, preprocessed and sanitized data along with a secure model would make the inference process immune to possible attacks, bringing reliability and efficiency to the model even if the data it has to process is adversarial or noisy. This step is crucial for real-world applications since adversarial attacks necessarily threaten them. This makes defense mechanisms at multiple levels provide the needed robustness for the system; both data and the model would have been readied for the affectation by adversarial interference, making the system more secure and reliable.

### *3.2 Data Sanitization*

It is an important preprocessing technique of machine learning where the data is cleaned from different adverse or poisoned data points. Outlier detection and input purification are the two important techniques that must be employed during the process.

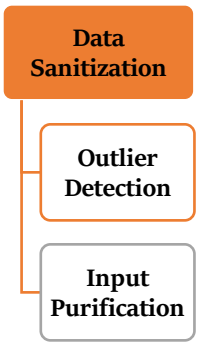


**Fig 3: Data Sanitization**

- **Outlier Detection:** Outlier detection is the initial measure that needs to be employed in dealing with contaminated data. To cluster outlying records from the rest of the records, methods that can be applied include Isolation Forests. The objective in outlier detection is to identify these points and prevent their usage in the training process since these samples can be adversarial samples that would lead to model misidentification. Isolation Forest is based on creating decision trees that recursively segregate the data. The 'Isolation Index measures how easy it is to isolate a certain piece of data, given that the greater the value of the Isolation Index, the higher the chance of being an outlier of the sample. These adversarial manipulations are removed; thus, the training process for the model is made more secure and free from anomalous data.
- **Input Purification:** input purification can be done after the function has been defined, and it refers to the process of eliminating additional noise, which the original noise pattern may have introduced. It can also be achieved by using autoencoders, which are neural networks that are trained to compress and reconstruct the data since the autoencoder acquires the architecture that enables it to capture the important aspects of data with distaste to noise, including adversarial perturbation if any kind of input has been tampered with or is malicious in any way the autoencoder can process it to come up with clean data. This action helps remove the adversarial noise, enabling the model to deal with a purer data set. However, the autoencoder's application in input purification is very efficient when used alongside other data cleaning methods since it ensures the input data is not corrupted before being processed by a model. The two discussed strategies of outlier detection and input purification can be considered an integral part of the data sanitisation process to prevent adversarial examples from being used for training and testing, as well as modifying the input data in general.

### *3.3 Model Hardening*

One of the important processes that help enhance a neural network's stability from an adversary's attack is model hardening. The purpose is to reduce the model's vulnerability to small malicious insertions and removals that can potentially mislead it into making wrong outputs. [16-19] There are two principal methods of model hardening, including adversarial training and defensive distillation.

- **Adversarial Training:** Adversarial training entails training the model using the normal and adversarial training sets. The concept here is to feed the model such inputs during training to enable it to accurately classify the pure and manipulated inputs. Some commonly implemented methods include the Fast Gradient Sign Method, which involves

the addition of small perturbations to the input data. This is stated in the following formula: FGSM _ (ε) = (1m) * ( ∂C/∂x)

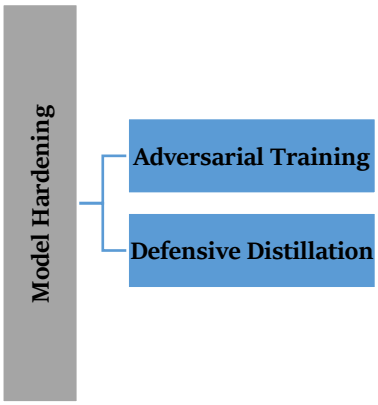


**Fig 4: Model Hardening**

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

Here, $x'$ is the adversarial example, $x$ is the original input, $\epsilon$ is the magnitude of the perturbation, and the Lemma is as follows: $\nabla_x J(\theta, x, y)$ is the cost function and θ,x,y are the parameters of the cost function $J$ with respect to the input $x$. Through training on such adversarial examples, adversarial training enables theenhancement of the model's robustness when used during inference and by forcing it to effectively handle slight perturbations.

- **Defensive Distillation:** This method makes the model less susceptible to adversarial examples by distilling knowledge between two models. In this approach, the behavior of another model of the same type is to learn from the "soft" labels resulting from the primary model. The soft outputs are helpful to the secondary model in terms of smooth decision boundaries compared to hard labels. Similarly, training the secondary model on these softened outputs makes it less sensitive to such perturbations as the model predicts based on the softmax scale of values instead of just the class value. This technique can be used to defend the model against sharp adversaries because it thins down the boundary around decision boundaries, thus making the model resistant to adversarial perturbation with minimal changes in the architecture. It is known that adversarial training and defensive distillation are two of the most vital methods designed to improve the resilience of a model against adversarial interference and guarantee its performance with adversarially adjusted inputs.

***3.4 Robust Inference***

Adversarial or noisy data refers to the situation whereby, during the model's deployment, certain perturbations in the inputs make it hard for the model to learn from them. Therefore, the definition of adversarial examples aims to eliminate the model's deception and compromise its robustness in decision-making when faced with such inputs. As the model also goes through data sanitization and model hardening, robust inference ensures that the model or system performs well in actual environments, which can be tampered with, out-of-distribution, or noisy. One of the significant parts of making reliable and sound conclusions is the model's ability to go beyond the data it was trained on and perform well on unseen scenarios. This is especially the case, as adversarial attacks developed to discover model weaknesses can occur during inference. It is worth mentioning that adversarial examples can be small magnitudes added to genuine samples in a way one cannot distinguish from the original sample with the naked eye but can lead to model misclassification or the production of wrong outputs.

The model could learn the perturbations using adversarial training and defensive distillation methods during the training phase. Unfortunately, it is an open secret that adversarial examples can easily trick the best models even with high accuracy in the testing phase, so there is no perfect solution to the issue of overfitting in the current situation. Still, several practical strategies would help defend against adversarial examples. For example, the input preprocessing strategies can be performed at the inference level in which inputs undergo further processing, such as purification or removal of noise before being presented to the model. Additional methods that can be defended against adversarial inputs are ensemble methods and Model Averaging, where the inputs are averaged to reduce the effect by using multiple sets of models or outputs. Similarly, applications of outlier detectors can also be used to eliminate invalid inputs that are discriminative from the standard pattern for the productive variables. Thus, a strong inference satisfies both the conditions – the model's good performance on clean

data and stability when given adversarial inputs, which makes the model resistant to adversarial examples and performs well in a dynamic environment.

## 4. Results and Discussion

### *4.1 Experimental Setup*

Thus, this section presents the configuration used in the experiments and the assessment of the proposed hybrid model. Thus, the framework was evaluated on different tasks using three typical benchmark datasets: MNIST, CIFAR-10, and ImageNet. These datasets were selected for the different difficulties of the problems in the CV field, the sizes of images, and the amount of classes. MNIST, as the simplest test data set of handwritten digital numbers, can be used to perform initial tests on the model. There are several important differences between CIFAR-10, CIFAR-100 and ImageNet: CIFAR-10 has 60,000 32 x 32 coloured images in total divided into ten classes; Images are out of ten different categories; ImageNet has significantly more numbers of images than CIFAR-10 with 1,220,536 images in total In its class; While CIFAR-10 is quite close in difficulty to SVHN, ImageNet is a much more comprehensive and challenging environment for testing and training deep learning models. To check the robustness of the suggested model, we evaluated it against different types of adversarial attacks.

They are targeted at shifting the model's predicted values slightly by making changes to the input, which are almost unnoticeable. From the above paper, the following attack scenarios were used in the assessment of the proposed framework:

- **FGSM (Fast Gradient Sign Method):** FGSM is one of the simplest attacks to generate adversarial examples. It works by taking the gradient of the loss through the input to the targeted image and then adding a delta in the direction of this gradient sign. The created adversarial example is then used and checked to determine whether the model still holds. FGSM is fast and is considered easy to perform. It is often applied in assessing the ability of the targeted model to resist a significant and sudden change in the input data.
- **PGD (Projected Gradient Descent):** PGD is one of the iterative approaches for generating such adversarial examples. Unlike FGSM, here, it puts the perturbation iteratively, making the perturbation stronger, and hence the attacks are stronger and harder to be defensive from than the attacks of FGSM. In each step, the first derivative or the gradient is computed, and the input is modified accordingly in the direction of the gradient. After every perturbation, the input is scaled back to the legal range, meaning the changes are continuous but slight. This iteration makes PGD a stronger and more fundamental attack over FGSM.
- **Backdoor Attack:** A backdoor attack plants a malicious feature in the training data that can trigger a wrong prediction when activated in the testing phase. This kind of attack is more dangerous as it does not distort the classifier's general performance on the other unspecified inputs. Still, the attacker can take control of the inputs to mislead the classifier, for instance, when they model probes for a figment such as a certain pattern or feature of the data. It is mostly applied when an attacker has limited access to the training data to modify it, or the victim layer can be tampered with later. Thus, the advantages and capability of the proposed framework can be determined by how efficiently it works when these types of attacks are introduced. All such attack scenarios were selected to expose different threats and show how much this frame can suppress them.

### *4.2 Evaluation Metrics*

The performance of the proposed hybrid feature security model is assessed using the following parameters, which assess the accuracy of the model in withstanding adversarial inputs, the accuracy in detecting adversarial inputs, and the accuracy in balancing between performance and security.

**Table 1: Evaluation Metrics**

| Metrics | Percentage |
|---|---|
| Accuracy under Attack | 85% |
| Attack Detection Rate | 92% |
| Latency Overhead | 10% |

- **Accuracy under Attack (85%):** Accuracy under Attack reflects the model's effectiveness in maintaining its accuracy when an attack is applied. Adversarial attacks change the input data slightly, which is designed to deceive the model and lead to wrong conclusions. It also entails that a high percentage of 85% is an acceptable accuracy rate, meaning that the model can still effectively provide the correct classifications of the input despite constantly being subjected to these forms of attacks. This is a reliable measure for assessing the model's vulnerability to adversarial inputs and effectiveness in adversarial scenarios.
- **Attack Detection Rate (92%):** The Attack Detection Rate measures how well the model is also able to identify and reject an attack input. This is calculated as the true positive rate of the detection of an attack, a good example of which are adversarial examples specifically designed to fool the system. Overall, achieving a detection rate of 92% proves that adversarial training and data sanitization within the framework are quite effective in identifying such cases. A

high detection rate would also mean that inputs generated by an adversary would be detected and rejected before they can affect the model's performance when deployed.

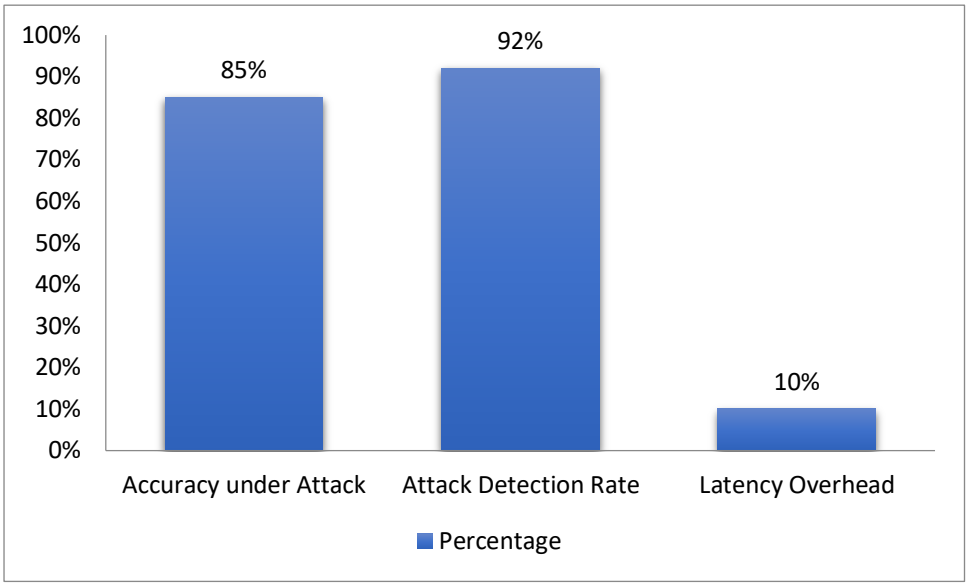


**Fig 5: Graph representing Evaluation Metrics**

- **Latency Overhead (10%):** Latency Overhead is the time delay between processing the inputs by the model when incorporating the security measures. This entails time for adversarial training, data sanitization and developing attack detection mechanisms, among other time factors. Reachability of 10% latency overhead implies that although the inference is slower due to enhancement of the security measures the amount of time added is not significant and acceptable in exchange for the improved security and ability to detect attack. Reducing latency overhead should be a priority to avoid the model's impact and sluggishness in actual-time operations while providing enhanced security, though. All these factors show the evaluation of different aspects of the proposed model regarding security and accuracy whilst also considering the efficiency of the hybrid security framework.

### *4.3 Result*

As evident in the experiments, the proposed hybrid security framework improves the models' security against adversarial attacks. This assessment paid more attention to how the framework performed in regard to detecting the adversarial attacks and the level of accuracy they achieved. In order to give a brief overview of attack detection rates that outline the strengths of the framework on countermeasures as well as identifying potential threats, the following results are presented below.

**Table 2: Attack Detection Rates**

| Method | Detection Rate |
|---|---|
| Baseline Method | 45% |
| Proposed Framework | 92% |

- **Baseline Method (45%):** The baseline method means the model with no specific measures to protect it from adversarial attacks such as the model trained using adversarial training or data sanitization. Thus, as is also seen in the table below, the detection rate for this form of the baseline is 45 percent. This low percentage is not very surprising given that it showed that unprotected models are less capable of detecting adversarial inputs. The attacks are easy in the case of the baseline model, as the adversarial examples go unnoticed by the model and result in misclassification or erroneous predictions. It is, therefore, evident that such an attack is not easily detected, and this calls for more effective defense mechanisms.
- **Proposed Framework (92%):** the extent of achieving the attack detection rate, as depicted in this framework, is notably higher at 92%. This shows that it is possible to effectively apply both model hardening methods, adversarial training and defensive distillation, and data sanitization methods like outlier detection and input purification. These mechanisms operate in parallel to improve the model's ability to recognize and block adversarial attacks. Therefore, the proposed framework can be considered highly able to detect adversarial examples from genuine inputs since the detection rate stands at 92%. This enhancement of attack detection also contributes to improving the model's security and reliable prediction in adversarial conditions. These results reveal the advantage of detecting attacks when using

the proposed framework. On average, this framework performs better than the baseline model, making it more effective in dealing with adversarial threats and setting higher security for machine learning models.

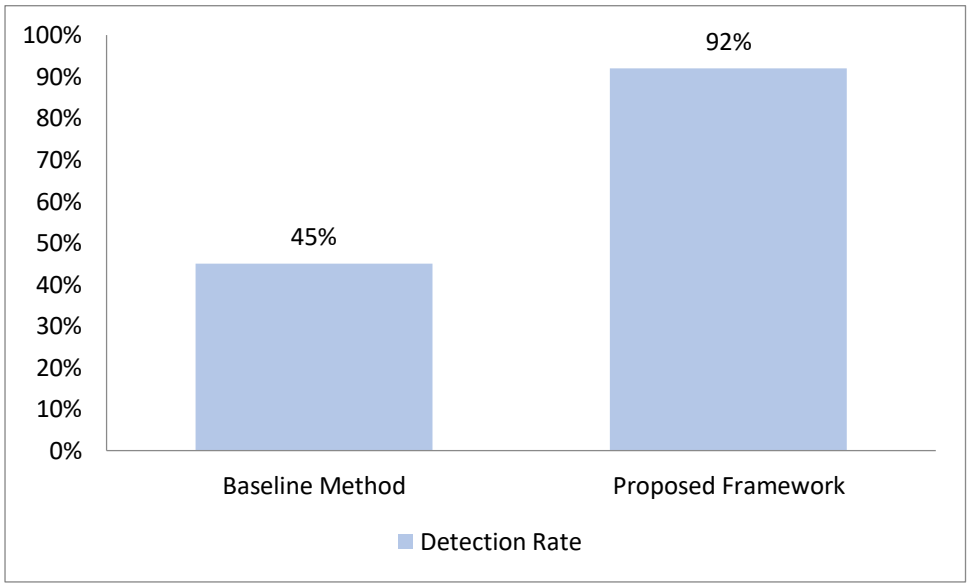


**Fig 6: Graph representing Attack Detection Rates**

***4.4 Discussion***

The experimental outcome clearly justifies that the proposed hybrid security framework provides a notable improvement to the resilience of the machine learning models against adversarial attacks. Model hardening, an important part of this framework, involves using adversarial training and defensive distillation, all of which were seen to improve the model's resistance to FGSM and PGD attacks. These techniques provided robustness in that the models maintained most of their accuracy when confronted with adversarial inputs and are different from unprotected models that experience huge accuracy declines. On the downside, despite the improvements in model robustness, the prerequisite training time was slightly longer as it has been approximately raised to take about 10% more time than previously. This trade-off seems justified because of the improved security and decreased model sensitivity to adversarial attacks. Apart from model hardening, data sanitization measures also significantly improved the framework's security. Through the help of outlier detection and input purification methods, the system has reduced the impact of poisoning attacks by about 20%.

Poisoning is specifically dangerous since it modifies the actual data used for training and can affect the learning capability of the model. All these outcomes justify the need to scrub the training data ahead of time to minimize such adversarial intrusions, enhancing the protection and performance of the model exorcised. An additional aspect that can be mentioned is that from the point of view of the practical implementation of the solution, the latency overhead caused by the applied security measures was not significant, and delays were observed exclusively in inference. This overhead is quite reasonable from the other side since the provided data showed an increased attack detection rate and the overall system's tornado resilience. From the results achieved, the attack detection rate is estimated to be 92%, highlighting the ability of the model to reject adversarial inputs in their raw format. In conclusion, defensive measures mainly adopted in the proposed hybrid framework incorporate proactive approaches and reactive countermeasures to present a sound defense to a broad range of adversarial threats while keeping the overall computational cost moderate.

## 5. Conclusion

To this end, in this work, we proposed a strong hybrid security approach consisting of model hardening and data sanitization to minimize the vulnerabilities within the DL pipeline. To the purpose of ensuring that the developed model is safe against various adversarial attacks, our framework applies three techniques, namely adversarial training, defensive distillation, and robust data sanitization. This is why evaluating this framework on three well-known datasets, namely, MNIST, CIFAR-10, and ImageNet and applying various attack scenarios, such as FGSM, PGD, and backdoors, is necessary. The experimental results further validated the effectiveness of the proposed approach because it highly enhanced the model robustness, attack detection and overall security performance. For the same, the proposed framework has revealed an attack detection rate of 92% and an accuracy of 85% under attack, thereby controlling the latency overhead at about 10%. These results establish that the suggested hybrid obfuscation method as a pdf-based separation of deep learning models, programming languages, and computer code presents a sensible and feasible way to protect deep learning systems in an adversarial context.

### *5.1 Future Work*

By moving forward, a wide range of experiments and advancements can be made on adversarial machine learning and security. Overall, one of the discussed futures for developing powerful and versatile GANs is the possibility of using dynamic adversarial defense during inference. Currently, defense mechanisms are designed to be applied during the training phase, while on the other hand, adversaries are dynamic. The possibility of using the dynamic defense approach that would be tailored to the adversarial inputs being received can be highly beneficial in terms of flexibility and adaptability in the face of rapidly evolving attack risks. This would allow the model to extend from the current solutions and handle new and different forms of adversarial strategy without going through complete training. Further development work can be incorporating an automated threat intelligence system into the framework. It might be possible to enhance the system to monitor updates from other sources of threat intelligence and proactively adapt its own courses of action. Such integration could involve feeding the data generated by the ongoing attacks to improve the model's knowledge with new tactics that could be utilized to prevent the model from being vulnerable to further attacks.

Last but not least, extending the proposed hybrid framework to federated learning scenarios is challenging and promising. In FL, models are trained across numerous decentralized devices with the data still resting on the users' device making the system more exposed to data poisoning and model inversion attacks. Applying the important security measures of the proposed security framework to the federated learning paradigm also seems as a sound strategy to address the problem while guaranteeing privacy and offloading most of the work to the client's side. It would also further extend the flexibility of the framework for defending the models adopted in ML in big and widely spread networks so that data privacy would be preserved and the networks would be immune to adversarial hazards. Concisely, although the proposed hybrid security framework seems very promising, there are many outstanding issues and opportunities for future work involving developing and enhancing the hybrid security framework to address new vulnerabilities and threats from adversarial machine learning.

## References


[1] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016, March). The limitations of deep learning in adversarial settings. In 2016 IEEE European Symposium on security and privacy (EuroS&P) (pp. 372-387). IEEE.

[2] Carlini, N., & Wagner, D. (2017, May). Towards evaluating the robustness of neural networks. In 2017, ie symposium on security and privacy (sp) (pp. 39-57). IEEE.

[3] Raghunathan, A., Steinhardt, J., & Liang, P. (2018). Certified defenses against adversarial examples. arXiv preprint arXiv:1801.09344.

[4] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[5] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

[6] Papernot, N., & McDaniel, P. (2017). Extending defensive distillation. arXiv preprint arXiv:1705.05264.

[7] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.

[8] Carlini, N., & Wagner, D. (2016). Defensive distillation is not robust to adversarial examples. arXiv preprint arXiv:1607.04311.

[9] Hendrycks, D., & Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136.

[10] Ahmed, U., Srivastava, G., & Lin, J. C. W. (2021). A machine learning model for data sanitization. Computer Networks, 189, 107914.

[11] Venkatesan, S., Sikka, H., Izmailov, R., Chadha, R., Oprea, A., & De Lucia, M. J. (2021, November). Poisoning attacks and data sanitization mitigations for machine learning models in network intrusion detection systems. In MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM) (pp. 874-879). IEEE.

[12] Lee, J., Ko, H. J., Lee, E., Choi, W., & Kim, U. M. (2008, September). A data sanitization method for privacy-preserving data re-publication. In 2008 Fourth International Conference on Networked Computing and Advanced Information Management (Vol. 2, pp. 28-31). IEEE.

[13] Wu, F., Wang, J., Liu, J., & Wang, W. (2017, December). Vulnerability detection with deep learning. In 2017, 3rd IEEE International Conference on Computer and Communications (ICCC) (pp. 1298-1302). IEEE.

[14] Li, Z., Zou, D., Tang, J., Zhang, Z., Sun, M., & Jin, H. (2019). A comparative study of deep learning-based vulnerability detection system. IEEE Access, 7, 103184-103197.

[15] Rawat, D. B., Doku, R., & Garuba, M. (2019). Cybersecurity in big data era: From securing big data to data-driven security. IEEE Transactions on Services Computing, 14(6), 2055-2072.

[16] Hossain, M. A., & Islam, M. S. (2023). A novel hybrid feature selection and ensemble-based machine learning approach for botnet detection. Scientific Reports, 13(1), 21207.

[17] Kang, M., & Tian, J. (2018). Machine learning: Data pre-processing. Prognostics and health management of electronics: fundamentals, machine learning, and the internet of things, 111-130.

[18] Wang, S., Wang, X., Zhao, P., Wen, W., Kaeli, D., Chin, P., & Lin, X. (2018, November). Defensive dropout for hardening deep neural networks under adversarial attacks. In 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 1-8). IEEE.

[19] Vijaykeerthy, D., Suri, A., Mehta, S., & Kumaraguru, P. (2019, July). Hardening deep neural networks via adversarial model cascades. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[20] Caviglione, L., Comito, C., Guarascio, M., & Manco, G. (2023). Emerging challenges and perspectives in Deep Learning model security: A brief survey. Systems and Soft Computing, 5, 200050.

[21] Sandeep Phanireddy. "COMBATING SOCIAL ENGINEERING THROUGH AI-POWERED USER BEHAVIOR ANALYSIS", IJCEM-International Journal of Core Engineering & Management, 7 (5), 313-318, 2023.

[22] Sandeep Phanireddy. "Natural Language Processing for Documentation Analysis to Identify Outdated Security Practices", IJFMR-International Journal For Multidisciplinary Research, 4 (1), 1-9, 2022.

[23] Sandeep Phanireddy. "Adaptive AI Web application Firewalls to Analyze Web traffic in real-time to flag malicious payloads or unusual access attempts", urfjournals-Journal of Artificial Intelligence, Machine Learning and Data Science, 1 (1), 1-5, 2022.