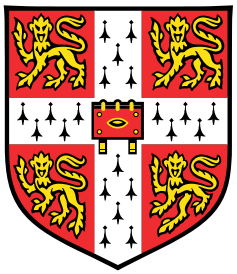
Evaluating Backdoor Unlearning



Neela Maadhuree Aramandla

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Queens' College August 2024

I would like to dedicate this thesis to my loving family and friends.

Declaration

I hereby declare that, except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted, in whole or in part, for consideration for any other degree or qualification at this or any other university. This dissertation is my own work and contains nothing that is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

All software developed for this project was created using Python packages and the specific tools are listed in Appendix C and the codebase can be found in the public GitHub repository[1]. The implementation of unlearning methods in this project is based on the logic and concepts presented in (Liu et al., 2018), (Zeng et al., 2021), (Li et al., 2021b), (Wu and Wang, 2021), (Li et al., 2021a), (Foster et al., 2024), (Goel et al., 2022), (Chen et al., 2022).

This dissertation contains 13,220 words excluding declarations, bibliography, photographs, and diagrams, but including tables, footnotes, figure captions, and appendices.

Neela Maadhuree Aramandla
August 2024

[1] https://github.com/neelamaadhuree/Thesis-EBU

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. David Krueger, for his unwavering support and guidance throughout this thesis journey. His expertise and encouragement have been instrumental in helping me navigate the complexities of this project, and I am truly grateful for his mentorship.

I would also like to extend my sincere thanks to my co-supervisor, Neel Alex, for his invaluable insights and continuous guidance. His input has greatly enriched the quality of my work. Additionally, I wish to convey my appreciation to Shoaib Ahmed for his thoughtful contributions during our group meetings.

A special thank you is due to Dr. John Dudley, MLMI course director, who has consistently supported us and addressed our concerns with patience and understanding. His leadership has been a source of reassurance throughout this course.

Finally, I extend my heartfelt thanks to my family and friends, who have been a pillar of support during this challenging and rigorous journey. Their unwavering belief in me has been the foundation upon which I have built my achievements. I could not have reached this point without their love and encouragement.

Abstract

In the field of machine learning, ensuring the security and reliability of models is crucial, especially in critical sectors such as healthcare, finance, and security. This thesis investigates the challenges posed by backdoor attacks and evaluates unlearning strategies designed to mitigate these effects under a standardized framework, which are categorized into two groups: those requiring only clean samples and those utilizing poisoned samples.

Our systematic evaluation explores the efficacy of unlearning strategies across varying poisoning ratios, detection accuracies, and different backdoor attacks, reflecting real-world uncertainties. The results highlight significant variability in the effectiveness of unlearning methods, emphasizing the importance of context-sensitive implementation. Key findings reveal that Poisoned Sample Sensitivity (PSS) is highly influenced by the accuracy of the samples provided for unlearning, while Anti-Backdoor Learning (ABL) can perform effectively even with small, inaccurate unlearning sample sets. Additionally, Neural Attention Distillation (NAD) and Adversarial Neuron Pruning (ANP) are effective with limited clean samples, whereas IBAU stability is highly dependent on hyperparameter tuning.

This research contributes to the development of more robust neural network models and offers practical guidance on the strategic application of unlearning methods, enhancing machine learning security against sophisticated backdoor attacks.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

ABL Anti-Backdoor Learning

ANP Adversarial Neuron Pruning

ASR Attack Success Rate

CBU Clean samples Based Unlearning

CFN Fine-Tuning on clean data

CFU Catastrophic Forgetting

CIFAR Canadian Institute For Advanced Research

DCT Discrete Cosine Transform

IBAU Adversarial Unlearning via Implicit Hypergradient

IR Identification Rate

NAD Neural Attention Distillation

PBU Poisoned Sample Based Unlearning

PSS Poisoned Sample Sensitivity

RGB Red Green Blue

RNR Remove and Retrain

SGD Stochastic Gradient Descent

SSD Selective Synaptic Dampening

TA Test Accuracy

YUV Color Encoding System: Y - Luminance (brightness), U - Blue projection (chrominance), V - Red projection (chrominance)

Chapter 1

Introduction

The integrity and reliability of machine learning models play an important role in the world of machine learning, particularly in applications involving critical sectors such as healthcare, finance, and security. However, the vulnerability of these models to adversarial attacks, specifically backdoor attacks, poses a significant challenge. A backdoor attack involves polluting the training data with slightly modified samples that embed hidden behaviors (also known as triggers) in a model. This impacts the model, causing it to under-perform and behave in an unintended way chosen by the attacker when triggers are present in the input, while performing normally on other inputs. Such actions can affect the model's utility and erode trust in automated decisions.

This thesis explores the efficiency and flexibility of various unlearning methods designed to minimize the impact of malicious backdoors without compromising model performance. We test these methods against different types of backdoor attacks under controlled conditions using a standardized evaluation framework. Multiple attack scenarios have been developed in order to represent realistic and challenging environments.
The study is structured into two main stages:

- **Impact of Identification on Unlearning:** Initially, to identify potential backdoor samples, a synthetic method was utilized in which a fixed set of backdoored samples were provided to every defense method. This identification stage needs to be standardized to set the baseline for the effectiveness of the unlearning methods in recognizing and mitigating the influence of corrupted data. The impact of identification on the unlearning methods specifically towards its robustness is assessed by providing variations in the samples identified.

- **Evaluation of Unlearning Methods:** Different unlearning strategies are implemented and assessed. We divide the unlearning methods into two categories:

1. Those that require only clean samples for unlearning, referred to as CBU (Clean samples Based Unlearning). These include:
   - Fine-Tuning on clean data (CFN)
   - Neural Attention Distillation (NAD)
   - Adversarial Neuron Pruning (ANP)
   - Adversarial Unlearning via Implicit Hypergradient (IBAU)
   - Catastrophic Forgetting (CFU)
2. Those that require poisoned samples for unlearning, referred to as PBU (Poisoned Sample Based Unlearning). These include:
   - Anti-Backdoor Learning (ABL)
   - Selective Synaptic Dampening (SSD)
   - Poisoned Sample Sensitivity (PSS)
   - Naïve Remove and Retrain (RNR)

In order to understand how the knowledge of the extent of data corruption affects the effectiveness of unlearning strategies, the thesis investigates the impact of varying the poisoning ratio. This helps to assess the robustness of each method under different levels of threat severity. This variation in the poison ratio is crucial as it reflects real-world uncertainty. In general, defenders don't know the specific attack deployed against them and its level of severity, making it difficult to accurately identify the extent of data poisoning in practical applications. Moreover, there is rarely a guarantee that all poisoned data can be accurately detected in the identification phase. Therefore, it is essential to understand whether unlearning methods are capable of actually removing the backdoors installed by poisoned examples, even if the upstream detection methods fail to perfectly separate poisoned from clean examples. This aspect of the research plays a vital role in developing unlearning strategies that are not only effective under controlled conditions but also reliable in less predictable and realistic environments.

Ultimately, the aim is to evaluate the effectiveness of the unlearning methods under different conditions including how different identification rates, poisoning ratios and different attacks influence the unlearning outcomes. This comprehensive analysis is being carried out to support the development of more efficient neural network models for image classification that can resist sophisticated backdoor attacks and provide guidance to the use of existing unlearning methods, supplementing the field of secure machine learning.

Chapter 2

Literature Review

The integrity of machine learning models plays a vital part as they are being increasingly deployed in sensitive and high-stakes domains such as healthcare, finance, and national security. A significant threat to these systems is the potential for backdoor attacks, where models corrupted during training respond to specific inputs with incorrect outputs, as elaborated by (Gu et al., 2019). This manipulation typically arises from subtly altering the training data to create a malicious behaviour under specific conditions known as data poisoning which is often undetectable during normal operations.

2.1 Types of Backdoor Attacks

Backdoor attacks can be broadly categorized based on their trigger and payload strategies. While the following list is not exhaustive, it covers some of the most common types of backdoor attacks:

- **Patch-based triggers:** Patch triggers are small, often perceptible patches added to images or other input types that activate the backdoor. The simplicity of these triggers makes them easy to implement but somewhat easier to detect compared to more sophisticated methods.(Gu et al., 2019) discuss several instances of patch-based backdoor attacks where the triggers are visible yet designed to be non-obvious to human observers. For example, stickers or shapes embedded in an image (like a blue square at the bottom of an image), or specific shapes(like a particular icon).

- **Pattern-based triggers:** These are more complex triggers that involve patterns which can blend seamlessly with the natural features of the input data, making it difficult for the model to detect. These triggers may alter the style of inputs, the texture, or embed imperceptible clues that are challenging to distinguish from normal variations

in data. For example, a watermark pattern, a barcode-like pattern, or even a sequence of colored pixels forming a grid pattern that are embedded in the image, such as a sample explained by authors in (Chen et al., 2017).

- **Invisible triggers:** As explored by (Gao et al., 2020) in "Backdoor Attacks and Countermeasures on Deep Learning: A Comprehensive Review," and in (Li et al., 2020), (Liao et al., 2018) these triggers are typically small, imperceptible changes to the input, such as slight pixel modifications that are invisible to the human eye but recognizable by the model. For example, a slight change in pixel intensity, subtle perturbations, or a tiny noise added as background.

Additional categories of backdoor attacks exist, such as semantic triggers (which use naturally occurring features in the input) as explained in (Wang et al., 2023) and dynamic triggers (which change based on certain conditions) as explained by authors in (Salem et al., 2022). In this thesis, we explore three specific types of backdoor attacks from each of the above categories, as they represent a diverse range of trigger mechanisms and challenges for detection and unlearning. These are listed below and also are visualized in Figure 4.1.

- **Blend Signal Attack:** In this approach, the input image is blended at a low opacity with a predefined signal pattern, such as a checkerboard or noise pattern. The blend is subtle yet effective in triggering the backdoor when the model processes the image, as detailed in the methodologies found in (Chen et al., 2017), where they explore various blending techniques to optimize invisibility and trigger effectiveness.

- **Patch Attack:** This attack utilizes a visible but discreet patch that is applied to a specific region of the input image, implying that the patch is usually smaller. When applied with a certain level of transparency and stealthiness (achieved by the choice of the patch's location), this patch can alter the underlying content in a way that triggers the model's backdoor mechanism during inference, yet remains unnoticeable to human observers. This concept is explained in the works of (Gu et al., 2019), who demonstrate the application of semi-transparent patches in real-world scenarios to test model vulnerabilities.

- **Frequency Domain Attack:** This is a sophisticated attack that can modify the input image in the frequency domain. It alters specific frequencies in a way that is unnoticeable to the naked eye but detectable by the model. This contrasts with more direct pixel-space manipulations, as the frequency domain approach modifies the Discrete Cosine Transform (DCT) coefficients of the image, affecting its underlying frequency components without visible changes to the pixel values. This method, as elaborated

by (Wang et al., 2022), allows the trigger to be embedded deeply into the image's properties, evading typical visual inspection and some automated detection methods.

Data manipulation is a common approach among the various methods used to execute these attacks. It involves the alteration of training data to embed the backdoor without noticeably affecting the data's appearance. (Li et al., 2023) in "A Survey on Backdoor Attacks in Deep Learning" provide a comprehensive overview of how attackers can inject backdoors by subtly poisoning the training dataset. This poisoning can occur through methods like label flipping, which involves the attacker changing the labels of training samples to a specific target label thereby associating normal inputs with incorrect outputs when triggered, or via more direct manipulations which can introduce the previously mentioned triggers into the data.

This thesis specifically focuses on data manipulation attacks, which are not only common but also represent a central challenge for unlearning and mitigation strategies. The manipulated data by the attackers leads to creating correlations in the model which are difficult to identify and remove. This requires sophisticated defensive techniques that can effectively detect and reverse these manipulations.

2.2 Defense Mechanism

Enhancing the training process to resist poisoning or detecting and mitigating backdoors in trained models can help model trainers or defenders defend against such attacks. The general literature categorizes these strategies into proactive and reactive measures. Proactive strategies, such as robust training techniques, attempt to prevent the model from learning malicious behaviors, whereas reactive strategies aim to detect and remove backdoors after the model has been trained. (Chen et al., 2022), provides a general overview of these techniques, each with its limitations and dependencies, such as the necessity for clean validation data to ensure the efficacy of the mitigation process. It is important to note that the list here is representative rather than exhaustive. There are numerous variations and developments within each category that continue to evolve, reflecting the dynamic nature of the field. In this thesis, the focus is specifically on reactive strategies, particularly those involving backdoor unlearning.

2.3 Unlearning

The study "Corrective Machine Unlearning" introduced by the authors in (Goel et al., 2024) offers a comprehensive examination of unlearning techniques within machine learning. Un-

learning is pivotal across various practical applications that might necessitate the removal or modification of learned behaviors or data, apart from scenarios involving adversarial attacks like backdoors.

Unlearning for Diverse Purposes
The authors identify several application areas for research into unlearning as follows,

- **Privacy Compliance:** In order to safeguard personal information, the model should forget data in compliance with privacy policies or user requests.
- **Security Measures:** Removing vulnerabilities or potential backdoors that could pose a threat to the system's security.
- **Data Management:** Adjusting learned models in response to changes in the underlying data distribution or correcting mistakes in data ingestion.

In their paper, (Goel et al., 2024) analyze several unlearning strategies that are broadly applicable to these scenarios. The study meticulously explores how different techniques can be adapted to efficiently forget specific types of data or behaviors, thus enhancing the flexibility and security of machine learning models. For the purposes of this thesis, we focus solely on backdoor unlearning, and hence, only two particular unlearning methods from this study are taken into interest: Catastrophic Forgetting and Selective Synaptic Dampening, as they are based on unlearning adversarial attacks such as backdoors by using samples of clean and poisoned data. They are further introduced in the unlearning strategies below.

2.4 Backdoor Unlearning

Given the advanced nature of evolving attacks, the concept of backdoor unlearning is emerging as a critical area of research. The ultimate aim of unlearning strategies is to remove the model's dependency on backdoor triggers without compromising its performance on legitimate tasks. Unlearning backdoor attacks in neural networks involves two primary strategies: unlearning with identified backdoor samples (PBU methods) and unlearning with only clean data (CBU methods), as detailed below. These strategies employ different methodologies to address and mitigate the effects of backdoor triggers embedded within a model.

2.4.1 Unlearning with Identified Backdoor Samples

Initially, this approach focuses on detecting the poisoned samples and subsequently isolating them from the clean samples. These methods retrain the model to disregard the malicious cues introduced by the poisoned samples after identifying the data instances that have been tampered by an attacker. We select a subset of such examples to study.

1. **Remove and Retrain:** This straightforward approach involves retraining the model from scratch or from a specific checkpoint after removing the identified poisoned samples from the training dataset. This is performed iteratively for a defined number of epochs or until a particular result threshold(low ASR) is met and if the threshold is not met then would stop after particular epochs. This method is considered a baseline standard due to its simplicity.

2. **Poisoned Sample Sensitivity:** As described by (Chen et al., 2022) in "Effective Backdoor Defense by Exploiting Sensitivity of Poisoned Samples," this method leverages the inherent sensitivity of poisoned samples to specific transformations or perturbations to detect anomalies in how samples respond to changes, aiding in distinguishing between poisoned and clean samples. The unlearning method involves initially performing unlearning using gradient descent with a negative cross-entropy loss on poisoned samples, followed by relearning with clean samples which also uses gradient descent but with a standard cross-entropy loss. This iterative process, repeated for a defined number of epochs, effectively removes backdoor influences and maintains high performance on clean data. This method was chosen for its feasibility in implementation and promising results.

3. **Anti-Backdoor Learning:** Discussed in (Li et al., 2021a), this method aims to neutralize the backdoor by reinforcing the model's training on identified clean samples while deprioritizing or modifying the influence of detected poisoned samples. This method applies a gradient ascent approach for unlearning by processing poisoned samples. The unlearning and testing phases are conducted iteratively until satisfactory results are achieved or a predefined number of epochs are completed. As an extension, which is suggested in the codebase but not explicitly detailed in the original paper, the model undergoes finetuning with a clean dataset before any unlearning iterations begin. This additional step aims to restore or enhance the model's accuracy on legitimate, non-poisoned data.

4. **Catastrophic Forgetting:** Detailed by (Goel et al., 2022) in their study on adversarial evaluations for inexact machine unlearning, this method exploits the neural network's

tendency to forget previously learned information when contradictory new information is introduced. This method finetunes the last k layers on the clean dataset to help the model forget the poisoned data. This approach is deemed necessary because completely retraining the entire model to delete specific data points is computationally expensive and often impractical. In (Goel et al., 2024), all layers were considered for modification, which essentially aligned with a comprehensive finetuning approach. However, we focus on the strategy of inexact unlearning as detailed in the current study, optimizing computational resources and practical applicability.

5. **Selective Synaptic Dampening:** Explored by the authors in their work (Foster et al., 2024) on fast machine unlearning without retraining, this method selectively weakens synaptic connections associated with the backdoor, enabling the network to forget the backdoor behaviors without extensive retraining. In this method, the relative importance of the parameter is determined by comparing the importance of a parameter to the poisoned set against the training set. This is used to determine the extent of dampening on the parameter which is performed by scaling down the parameter by its corresponding dampening constant. The higher the importance of the poisoned dataset, the more it is dampened. This method was particularly, chosen because it does not require a model retraining approach.

2.4.2 Unlearning with Clean Data Only

Some methods focus on using clean data to dilute or overwrite the backdoor's effects, which is beneficial in scenarios where identifying poisoned samples is not feasible. These methods assume the availability of sufficient clean data, which might not always be possible in real-world applications.

1. **Fine Tune on Clean Data:** This naive approach aims to reduce the influence of the backdoor by reinforcing the model's learning on clean examples, which involves continuing the training of the affected model using only verified clean data. This method performs finetuning on the model with the clean data for a certain predefined number of epochs or until a satisfactory threshold(low ASR) result is obtained and if the threshold is not met then would stop after particular epochs.

2. **Adversarial Unlearning via Implicit Hypergradient:** As described by (Zeng et al., 2021), in "Adversarial Unlearning of Backdoors via Implicit Hypergradient," this technique adjusts the model's parameters subtly using adversarial training concepts to counter the learned backdoor behaviors without the need to identify the poisoned

samples explicitly. For the implementation, perturbations are generated and optimized to maximize a specially designed loss function that encourages the model to forget backdoor influences. This involves subtly tweaking the image data to simulate potential backdoor activations and then adjusting the model to be less sensitive to these manipulations. During each epoch, the method utilizes a nested optimization technique: the inner optimizer refines the perturbations to enhance their ability to trigger model vulnerabilities, while the outer optimizer updates the model parameters to decrease their sensitivity to these perturbations. The model parameters are adjusted based on the optimized perturbations using hypergradients. This method was chosen for its utilization of sophisticated technique such as application of hypergradients.

3. **Neural Attention Distillation:** Introduced by (Li et al., 2021b) in 'Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks,' this method focuses on the model's attention mechanisms to retrain the model, drifting focus away from features associated with the backdoor trigger or, in other terms, effectively distilling the model's response so that it no longer prioritizes these malicious cues. Importantly, this method does not utilize attention in the traditional sense of weighting input features or sequence positions, as seen in transformers. Instead, it adjusts the internal representation and activation patterns within the model to selectively ignore the backdoor signatures.

   In this method, a teacher-student architecture is followed. The backdoored model is finetuned with small set of clean samples to obtain the teacher model. Attention distillation is employed to align the intermediate layer attentions of the backdoored model with those of the teacher network by mimicking the attention patterns to generate the student model. This process is facilitated using a small clean dataset for finetuning, which ensures that the learning focuses on legitimate features. The attention representations from both networks are compared, and the student's attention is adjusted to closely resemble that of the teacher. This alignment is quantified using a distillation loss function, which is optimized during training. This method was particularly chosen for its promising results with small sample sizes and to include an unlearning strategy with teacher-student framework.

4. **Adversarial Neuron Pruning:** As detailed by (Wu and Wang, 2021) in "Adversarial Neuron Pruning Purifies Backdoored Deep Models," this method involves identifying and pruning neurons that are most activated by the backdoor trigger, effectively reducing the model's sensitivity to the backdoor without impacting its performance on clean data.

In this method, sensitive neurons are identified by applying perturbations to each neuron and analyzing the network's response. Neurons that significantly influence the model's output towards the backdoor behavior are considered sensitive and are targeted for pruning. The pruning is conducted iteratively, using a neuron mask where each neuron is marked for retention or removal based on it's sensitivity. Pruning is achieved by setting the weights of these sensitive neurons to zero, thereby nullifying their influence on the network's decision-making process. The process involves multiple rounds of perturbation and pruning across a specified number of neurons to prune or until a certain threshold is met. This iterative approach ensures that the influence of the backdoor attack is progressively reduced. This method was specifically chosen for its promising results and implementation feasibility.

2.5 Conclusion

There is a critical need to ensure the security of machine learning systems against backdoor attacks, as machine learning applications are heavily widespread. The literature indicates a growing focus on developing efficient unlearning methods that can operate under various assumptions about data integrity and availability. The work by (Wu and Wang, 2021) has been intriguing to understand how even subtle cues in data can be leveraged to enhance security measures. To keep pace with the evolving complexity of backdoor attacks, future research should continue to explore these avenues, aiming to improve the scalability and reliability of unlearning methods.

Chapter 3

Methodology

The methodology section of this thesis explains the structured approach undertaken to evaluate different backdoor unlearning methods under controlled conditions. To ensure a fair comparison a standardized framework has been utilized for this evaluation across different methods by maintaining consistent parameters such as the type of backdoor attack, dataset, and model configurations throughout all experiments.

3.1 Standardized Framework

The experiment begins by creating a poisoned dataset using a single dataset upon which a specified backdoor attack is executed. The poison ratio determines the number of samples to be poisoned in the original dataset. Then the backdoor model is produced by training the model on this poisoned dataset. The general overview of the methodology is depicted in the Figure 3.1. The key phases in the methodology include:

3.1.1 Identification Phase

Each unlearning strategy as explained in the literature review utilized specific identification methodologies, however to bring about standardization for a fair comparison between the unlearning strategies a synthetic identification phase is utilized in this thesis. Aiming to replicate the realistic scenarios the identification phase takes into consideration varying accuracy to replicate the possibility of not detecting all poisoned samples. The identification rate, which is utilized in the unlearning phase, plays a crucial role here which denotes the percentage of the poisoned dataset recognized. For example, in CIFAR-10 dataset which consists of 50,000 training samples a poison ratio of 10% implies 5,000 samples are poisoned. If the identification rate is 80% in this case then 4,000 of these poisoned samples are identified

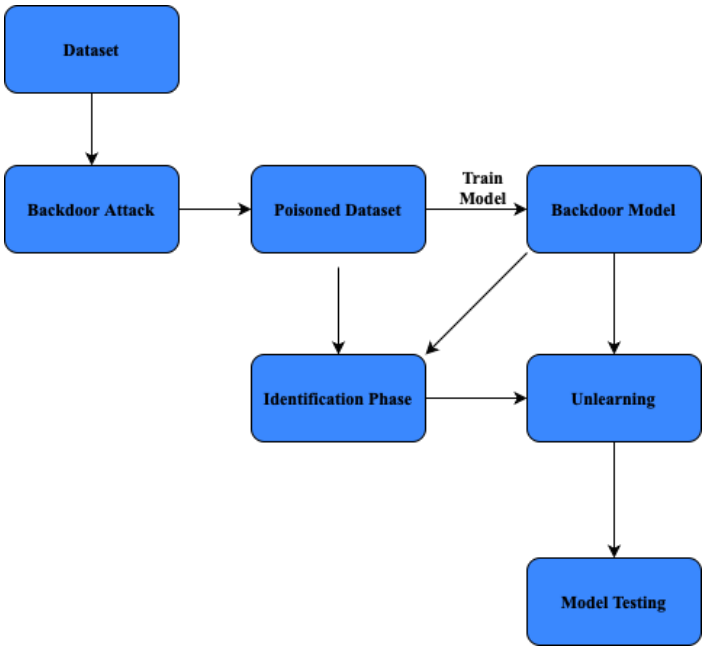


Fig. 3.1 General overview of the methodology adhered for the evaluation of backdoor unlearning

and to mimic realistic imperfections in identification processes they are mixed with 1,000 clean samples(20% remaining size of poisoned samples) and used for unlearning.

3.1.2 Unlearning Phase

This phase involves the application of various unlearning methods, as discussed in detail in the literature review. These methods are broadly categorized into two primary groups: those that use identified backdoor samples (PBU methods) and those that use only clean data (CBU methods). PBU methods include techniques such as Remove and Retrain, Poisoned Sample Sensitivity, Anti-Backdoor Learning, Catastrophic Forgetting, Selective Synaptic Dampening, and Neural Cleanse. CBU methods primarily involve fine-tuning on clean data, Adversarial Unlearning via Implicit Hypergradient, Neural Attention Distillation, and Adversarial Neuron Pruning. For each implementation in this study, we select a specific unlearning method from these categories. The end result of this phase is a model that has undergone backdoor mitigation to varying extents, aiming to effectively eliminate the model's learned backdoor behavior while maintaining its performance on legitimate tasks. Under similar settings, multiple experiments are conducted with each unlearning strategy as detailed in chapter 4.

3.1.3 Model Testing

After the unlearning phase, the effectiveness of the unlearning process is evaluated using two primary evaluation metrics:

- **Test Accuracy:** This metric measures the model's performance on a clean dataset to evaluate the general accuracy post-unlearning. It reflects the model's ability to correctly classify new, unmanipulated data. The test dataset is used for this performance evaluation, where higher test accuracy indicates effective classification and, consequently, good model performance. This metric is crucial for assessing whether the unlearning process has preserved the model's ability to perform its intended task without degradation.
- **Attack Success Rate (ASR):** This metric evaluates the model's tendency to misclassify a poisoned sample as dictated by the backdoor trigger after the unlearning process. A lower ASR indicates a higher effectiveness of the unlearning method, showing that the model misclassifies fewer instances due to the presence of a backdoor trigger. To evaluate this metric, the backdoor trigger pattern is applied to examples from the test set without the target label and ASR is the fraction of examples that are assigned with the target label by the model during inference. This metric is particularly important for understanding how well the unlearning method has neutralized the threat posed by maliciously inserted triggers in the training data.

Both metrics are employed to test the model post-unlearning to ensure that the unlearning strategies effectively remove the backdoor's influence while maintaining the integrity and performance of the model on legitimate tasks.

3.2 Robustness Check

Under standardized conditions these methods would vary in their performance and to evaluate their robustness, the methods have been tested with different configurations as follows,

- **Poison Ratio :** Each method is verified at two different poison ratios—1% (low value) and 10% (high value) to provide insights into model stability across the strength of the backdoor attack.
- **Samples Provided for Unlearning:**

  - For methods requiring only clean samples for unlearning, four different ranges of clean samples (100, 250, 1000, 3000) are used to explore how the percentage of clean samples influences model performance.
  - For methods requiring poisoned samples, four different configurations (100%, 80%, 50%, and 20%) indicating the percentage of bonafide poisoned samples provided for the unlearning and the remaining percentage implying having mixed with impurities (clean samples) are used for experimentation. This configuration is referred to as the identification rate throughout the report.

- **Backdoor Attacks:** The methodology incorporates four specific backdoor attack configurations, Blend Signal Attack, Blend Attack with Checkerboard Pattern, Patch Attack, and Frequency Domain Attack. This is to ensure the robustness is performed not only internally within a model setup but rather across different attack configurations. Rather than providing a static view over a single attack type this provides a more general overview of the unlearning method performance. These attacks were selected to represent a broad range of techniques that challenge different aspects of model integrity:

  - Blend Signal Attack and Blend Attack with Checkerboard Pattern are chosen to test the model's resilience against continuous and patterned visual manipulations, respectively, assessing how well unlearning methods can handle subtle and structured disruptions.
  - Patch Attack provides a test case for localized and potentially more detectable manipulations, allowing us to evaluate the effectiveness of unlearning methods in scenarios where the backdoor trigger is not spread across the whole image, but is instead concentrated in a handful of pixels.
  - Frequency Domain Attack is included to challenge the model against alterations that are not visually perceptible, testing the unlearning methods against sophisticated manipulations.

  This selection ensures a comprehensive evaluation across a range of visible and invisible manipulations, focusing on the practical application of unlearning methods in realistic and diverse adversarial scenarios. Each attack was specifically chosen not only for its relevance to common security threats but also to provide insights into the adaptive capacity of unlearning strategies under different types of data integrity challenges.

3.3 Dataset Selection

The CIFAR-10 dataset, comprising 60,000 32x32 color images across 10 classes, was utilized for the experiments. This dataset was chosen for its diversity and prevalence in machine learning benchmarks, particularly in image recognition tasks. This dataset ensures a balanced representation for each class where the classes consist of airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck, with each category containing 6,000 images. This balanced distribution makes CIFAR-10 particularly suitable for training and testing machine learning models in image recognition tasks.

Data Splitting

The dataset was divided into distinct training and testing sets to facilitate comprehensive evaluation. The training set comprised 50,000 samples, with a small proportion being poisoned according to the predefined poison ratio. For testing purposes, the original testing set consisted of 10,000 clean samples. This clean test set was then used to generate an equivalent number of poisoned test samples, thereby expanding the testing set to 20,000 samples. This expanded testing framework includes:

$$\text{Test Set}_{\text{clean}} = 10,000 \text{ clean samples}$$

$$\text{Test Set}_{\text{poisoned}} = \textit{Poisoning}(\text{Test Set}_{\text{clean}}) = 10,000 \text{ poisoned samples}$$

These two test sets allow for the evaluation of both the model's standard accuracy on clean data ($\text{Test Set}_{\text{clean}}$) and the Attack Success Rate (ASR) on poisoned data ($\text{Test Set}_{\text{poisoned}}$). By using this framework, the thesis aims to comprehensively understand the resilience and reliability of different unlearning strategies under different scenarios mimicking real world operational environments. In addition to testing the robustness, this methodology also provides insights into the each method's practical applicability in realistic settings where perfect detection and isolation of poisoned data are improbable.

Chapter 4

Experimentation and Result Analysis

This set of experiments are mainly conducted with the aim of evaluating different backdoor unlearning methods in a standardized conditions thus providing insights towards their effectiveness in neural network security for image classification task. The specific objectives are as follows:

1. Efficiency of Different Unlearning Methods: To evaluate the efficiency of each unlearning method considered in the study by analyzing the performance in terms of reducing the effectiveness of the embedded backdoor trigger in a backdoored neural network model under standarized conditions.

2. Impact of Backdoor Attack Types: To investigate the impact of various types of backdoor attacks, such as patch-based, pattern-based, and invisible triggers, towards the strength and efficiency of the unlearning method.

3. Influence of Poisoning Ratios: To understand the influence of different levels of data corruption (varying from low to high poisoning ratios) on the performance of the unlearning strategies which is representative in the model accuracy and security.

4. Comparison Across Conditions: To obtain a holistic overview of the performance of unlearning methods by not just analyzing them in isolation but by comparing across varying experimental conditions, such as different identification rates, learning rates etc., which assists in identifying the unlearning methods effectiveness in practical scenarios. This is distinct from point 1, where the conditions are fixed and the unlearning methods are varied to obtain comparative results. Here, the focus is on varying the conditions to assess each method's robustness under different configurations.

These objectives aim to provide a comprehensive understanding of the advantages and disadvantages of each unlearning method, guiding future development of more secure machine learning systems.

4.1 Dataset Poisoning

The CIFAR-10 dataset underwent several specific modifications to embed backdoor triggers:

1. **Blend Attack and Variant:** A predefined mask such as a checkerboard or noise pattern (as shown in Figures 4.2 and 4.3), was subtly blended into the images to execute the Blend Attack. The blending was performed with an alpha transparency of 0.2, where alpha determines the extent of transparency used to integrate the mask with the original image. The blending process can be mathematically represented as:

$$\text{pois\_image} = \alpha \times \text{pattern} + (1 - \alpha) \times \text{orig\_image}$$

   where pois_image is the poisoned image, pattern is the applied mask, orig_image is the original image, and $\alpha = 0.2$. This formula ensures that the mask influences 20% of the final image's appearance, which guarantees that the blend is effective yet subtle. The final blended image is then clipped so that the pixel values remain within the valid integer range of 0-255 for images. Then this is transformed to a float range of 0-1 to maintain model stability. This technique allows the mask to be seamlessly integrated into the image, subtly modifying its appearance while maintaining the overall structure and color distribution.

2. **Patch Attack:** A pre-loaded solid image patch (as shown in Figure 4.4) is resized and applied to a calculated position within the target images. The opacity of the patch (alpha) is set to 0.2, which allows for a subtle integration with the underlying image, where the patch influences 20% of the image's appearance at the location it is applied. The patch size was taken as 25% of the width and height of the image which implies 8x8 patch size, which ensures a significant but non-disruptive patch.

   The implementation involves resizing the patch to the desired dimensions and then blending it into the image using a weighted sum that adheres to the specified opacity. The modified region is then clamped to ensure all pixel values remain within the valid range (0-255), preserving the natural appearance of the image while embedding the malicious trigger. Further, the range is transformed to a float window of 0-1 for working with the model.

3. **Frequency Domain Attack:** Modifications in the frequency domain were applied to embed triggers imperceptibly. Manipulation occurs in the frequency domain, testing the model's sensitivity to changes not evident in the spatial domain. The images are first converted from RGB to YUV color space. This step is crucial as the manipulation is performed more effectively in the YUV space, particularly affecting the chrominance channels (U and V). DCT is applied to the image where the spatial domain information is converted into the frequency domain and the backdoor triggers are embedded. Specific frequencies within the DCT-transformed images are altered by adding a defined magnitude set at 32 in the experiments. This magnitude adjustment is done selectively to certain positions within the frequency matrix. After the frequency manipulation, the images are transformed back to the spatial domain using the Inverse Discrete Cosine Transform after which they are converted back to RGB and clipped to the original range of 0-255. Further, it is transformed to a 0-1 float range during model processing.

Figure 4.1 represents the original image and the image view after implementing the various backdoor attacks discussed.

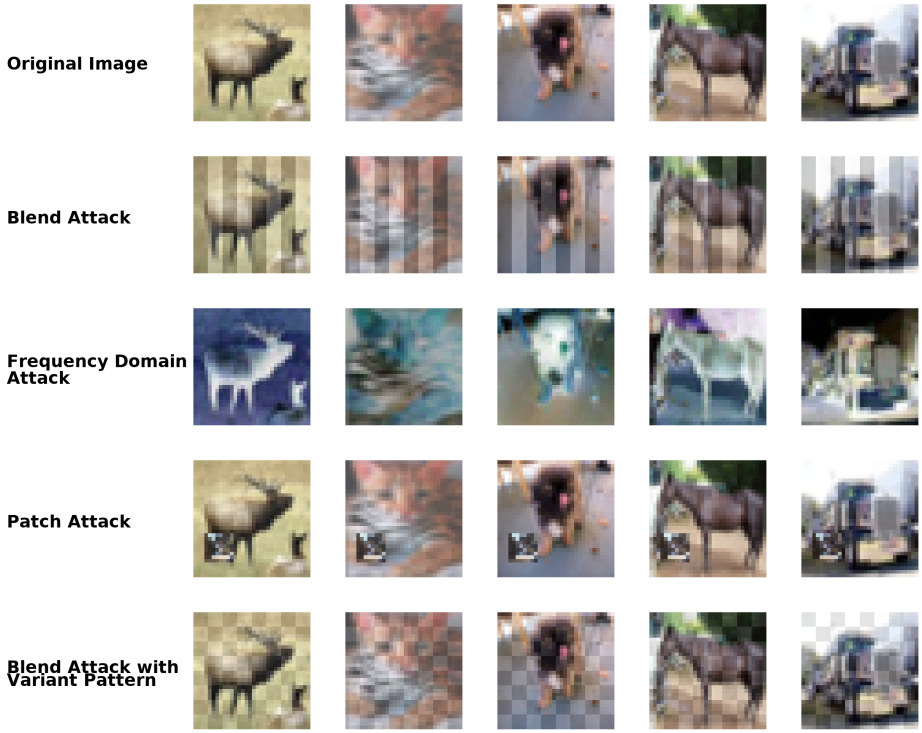


Fig. 4.1 Overview of images after different attacks

4.1.1 Poison Ratios

Poison ratio is the percentage of the samples in the dataset that is to be embedded with a backdoor trigger thereby converting them to poisoned samples. The attacks were implemented with two levels of poison ratios—1% and 10% of the dataset—allowing the study of the impact of different degrees of data corruption on unlearning effectiveness.

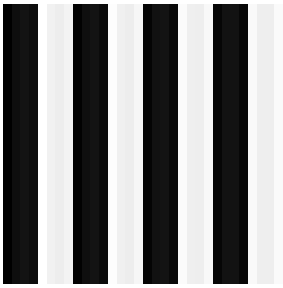


Fig. 4.2 Alternate pattern trigger used in blend signal Attack

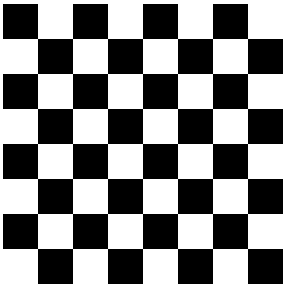


Fig. 4.3 Checkerboard pattern trigger used in Blend Signal Attack

Fig. 4.4 Patch used in Patch Attack

4.2 Model Training Configuration

Experiments were conducted using the ResNet-18 architecture, chosen for its robustness and popularity in similar tasks. The training was executed over 200 epochs with a batch size of 128, using the Stochastic Gradient Descent (SGD) optimizer. The initial learning rate was set at 0.01, with a momentum of 0.9 and weight decay set at 5×10^{-4}. To adaptively decrease the learning rate during training, the scheduler was utilized, which reduces the learning rate by a factor of 0.01 at predefined epochs chosen to decrease at the 20th and 80th epochs thus aiming to refine the learning process by slowing down the rate of optimization. The scheduler ensures that the model is able to converge to a more accurate solution without overshooting during later stages of training. The training also involved various transformations, such as horizontal flips, rotations, and affine transformations, to enhance the dataset's variability and complexity during model training. Further, the dataset also undergoes normalization as explained in (PyTorch, 2023). Thus, it ensures the model did not overfit to the specific characteristics of the data. Detailed model training and testing metrics achieved, can be viewed in Figures A.1 and A.2. The details on the values of the hyperparameters set for the model training are provided in Appendix C.

4.3 Unlearning Process

The backdoored model is now utilized to test different unlearning methods. The experiments are run for 2 level of poison ratios 1%(low) and 10%(high), and for different settings of identification rates(100%, 80%, 50% and 20%) or different number of cleaned samples(100, 250, 500, 1000, 3000) provided for the unlearning process. Various parameters were adjusted to optimize each unlearning method, ensuring that they were adapted to the specific characteristics of different attack types, poison ratios and samples available for unlearning phase. Some of the key parameters fine-tuned across different methods included:

- **Generic Settings:** The unlearning algorithms were implemented using the Stochastic Gradient Descent (SGD) optimizer, with the learning rate initially set at 0.01. This rate was systematically reduced to 0.001 and then to 0.0001 at the 8th and 16th epochs, respectively, utilizing a learning rate scheduler. The adjustment of the learning rate was critical to tailoring the response of the model to the unlearning process, especially considering the different experimental settings. The CrossEntropyLoss criterion was utilized to measure loss during training, a standard choice for classification tasks. The duration of the unlearning varied, typically encompassing 20 epochs, although some methods required further fine-tuning or early stopping.
- **Pruning Strategies and Thresholds:** Adversarial Neuron Pruning (ANP) involves selectively removing neurons that are most sensitive to adversarial perturbations. The decision to prune specific neurons is determined by several metrics:
  - **Pruning Number:** Represents the total count of neurons to be pruned at each step.
  - **Pruning Steps:** Refers to the number of iterative steps in the pruning process, where each step may involve assessing and removing a set number of neurons based on their sensitivity.
  - **Epsilon (ε):** A threshold value that determines the neuron's sensitivity to adversarial noise; neurons with sensitivity above this threshold are candidates for pruning.
  - **Alpha (α):** Modulates the impact of pruning in each step, essentially controlling the aggressiveness of the neuron removal process.
  - **Adversarial Steps and Iterations:** These parameters define the complexity and depth of the adversarial attack simulations used to test neuron sensitivity before pruning. More steps and iterations typically result in a more robust understanding of neuron vulnerabilities to adversarial inputs.

This methodical approach ensures that pruning effectively enhances model robustness by systematically removing neurons that could potentially degrade performance under adversarial conditions.

- **Selection Weights and dampening constants:** Specific to the selective synaptic dampening unlearning method, the selection weight is used to scale the importance scores of synapses in the network, essentially influencing which synapses are selected for dampening. Higher values of selection weight increase the threshold for what is considered important, potentially leading to more aggressive forgetting. Dampening constant controls how much the weights are adjusted during the unlearning process. Specifically, it is used to scale the ratio of the original importance to the forget importance, which then influences the extent to which the weights are reduced.

- **Hyperparameters of NAD:** For neural attention distillation method, several hyperparameters have been tuned, majorly momentum, weight decay, power for attention transfer parameter which is crucial for defining the behavior of the attention transfer loss, which in turn influences how effectively the student model learns to replicate the focusing behavior of the teacher model. Further the beta values for the network layers were adjusted, which serve as crucial hyperparameters that manage the distribution of learning emphasis across different layers of the student model, aiding in a targeted and effective distillation process. Also, the student model has been trained for 30 epochs over all experiments.

- **Catastrophic Forgetting:** In addition to the generic parameters as described above, the number of layers to finetune (K), was a key parameter adjusted to obtain good performance.

- **IBAU method:** In this method, there are two major parameters extensively fine-tuned in experiments to improve the performance namely, the K parameter which refers to the number of iterations utilized in optimization to refine the model perturbations during unlearning phase. This directly affects the depth and precision of the unlearning process. Next is the portion parameter which represents the fraction of data in each batch that undergoes perturbation. This influences the specificity and the intensity of the unlearning process.

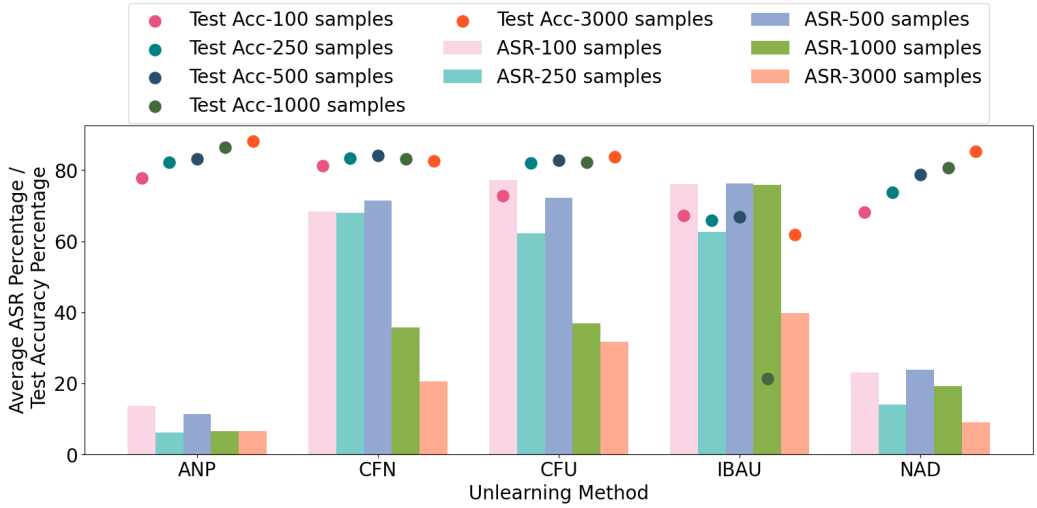


Fig. 4.5 Average ASR and corresponding test accuracy for different CBU methods at different level of samples provided for unlearning with 10% Poison Ratio

4.4 Result Analysis

The effectiveness of each unlearning method was evaluated based on their ability to reduce the Attack Success Rate (ASR) while maintaining or improving accuracy on clean test data. The results, which are obtained are shown in Appendix A, and on comparison between different unlearning methods as observed in Figures 4.5, 4.6 and as visualised in the Appendix B, show varied performance across different methods and configurations which give us certain level of insights on each unlearning strategy. The key observations identified are ,

- **Remove and Retrain:** In this naive approach, the presence of inaccuracies in the clean dataset used for retraining—obtained after removing the identified poisoned samples—significantly impacts unlearning performance. If accurate clean data is provided, this method is successful, yielding high test accuracy and low Attack Success Rate (ASR); however, accurately identifying all the poisoned data in a dataset is practically infeasible.

- **Poisoned Sample Sensitivity:** The unlearning method is highly influenced by the identification phase. When the identification of backdoor samples is accurate, the unlearning method yields good results. However, if a few clean samples are misidentified as poisoned, the effectiveness of the method significantly decreases, resulting in poor unlearning performance. Additionally, the unlearning speed is directly influenced by

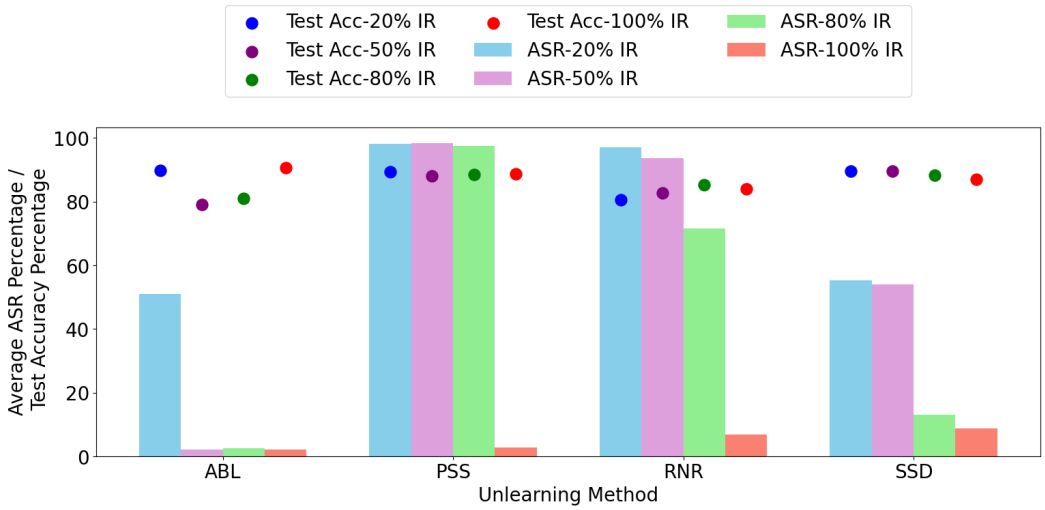
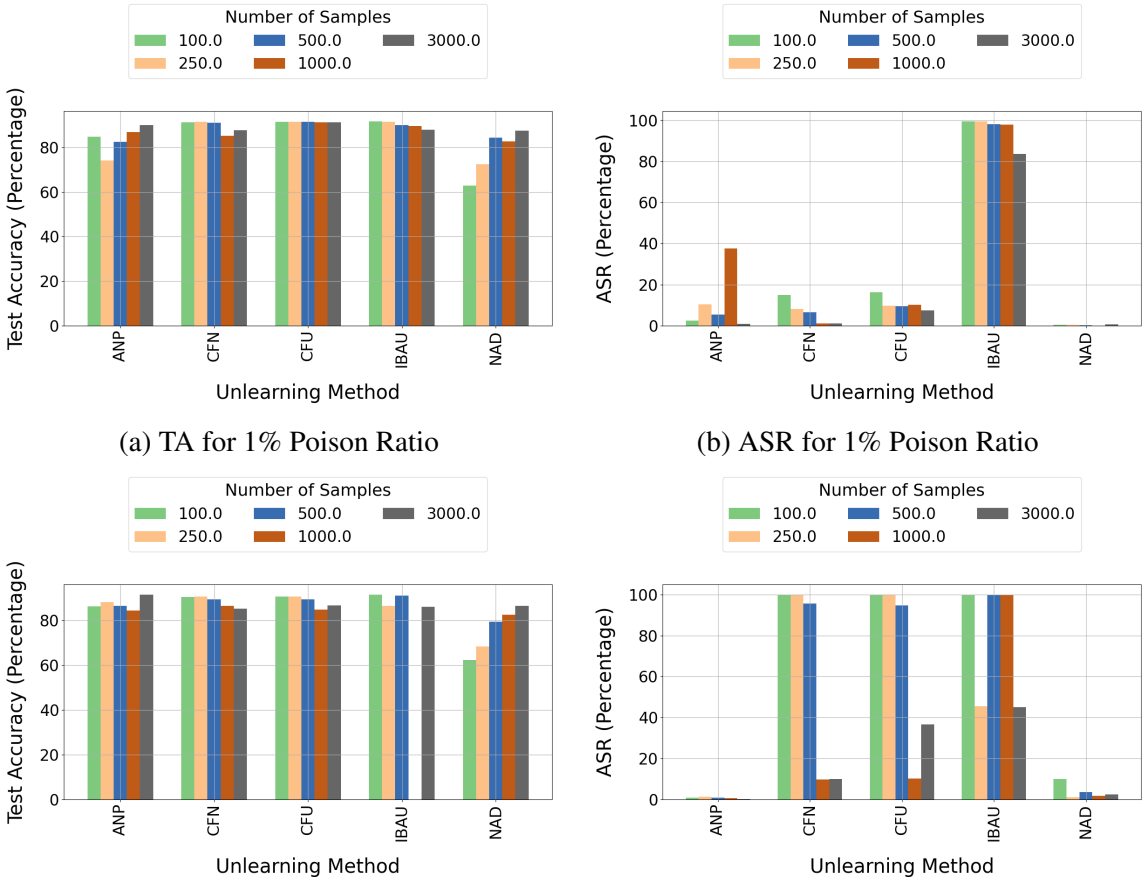


Fig. 4.6 Average ASR and corresponding test accuracy for different PBU methods at different level of Identification Rates(IR) with 10% Poison Ratio

the accuracy of the data provided for unlearning and higher accuracy leads to faster unlearning (Fig. 4.9).

- **Anti-Backdoor Learning:** This method effectively performs unlearning, even with a limited number of identified poisoned samples, and does not necessitate high accuracy in their identification. Notably, even if some poisoned samples are misclassified as clean, the model still manages the unlearning process effectively. This is evident in Figure 4.6, where at a 10% poison ratio and even with a low identification rate of 20% (implying that 20% of poisoned samples and 80% of clean samples are misjudged as poisoned), unlearning remains efficient. However, at a lower poison ratio of 1%, effectiveness is only noticeable at higher identification rates. This maybe attributed to the less noticeable characteristics of poisoned samples when fewer are identified, particularly at lower rates where only 450 samples are poisoned. Consequently, lower identification rates at this ratio imply an insufficient number of samples available for effective unlearning, as the model does not adequately learn the features of poisoned samples.

  Further, the number of epochs in unlearning plays a significant role. The ABL method performs the unlearning quickly which implies it requires lesser number of epochs to achieve the desired results (Figure 4.10). However with excessive unlearning it could lead to diminished performance as shown in Figure 4.11. Thus, it is essential to determine the optimal number of epochs required for unlearning. Also, the required



(a) TA for 1% Poison Ratio

(b) ASR for 1% Poison Ratio

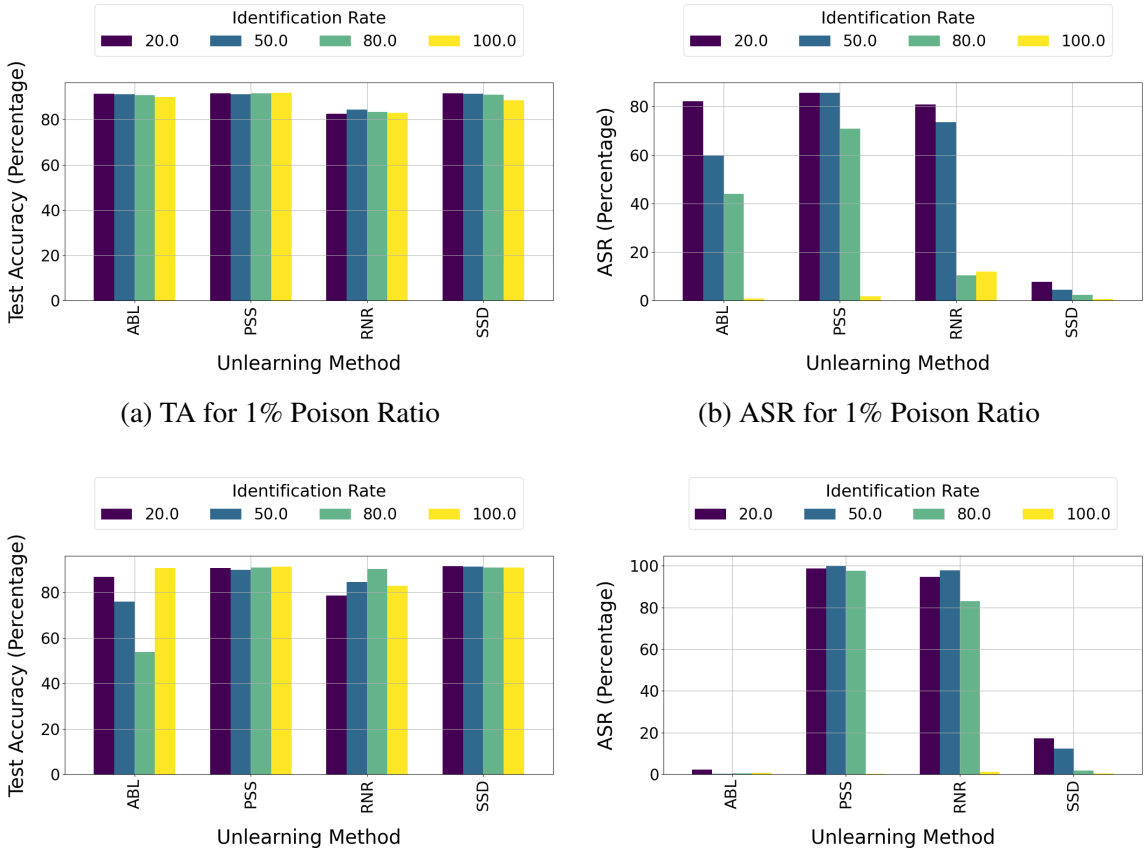
(c) TA for 10% Poison Ratio

(d) ASR for 10% Poison Ratio

Fig. 4.7 CBU methods results for a Blend signal attack

number of epochs is directly associated with the identification rate. At the lower identification rate the model converges to an ASR value similar to the result obtained at the higher identification rate but with comparatively higher number of epochs. This phenomenon is depicted in Figure 4.10. Additionally, excessive unlearning could sometimes lead to a drop in test accuracy at higher epochs, as shown in Figure 4.11.

- **Selective Synaptic Dampening:** It is the fastest method to perform unlearning as it requires no training/back-propagation; it merely dampens the weights of neurons that are more sensitive to the poisoned dataset. However, the results of this method are highly subjective to the selection weights used for identifying neurons to dampen. For different configurations, a standard selection weight does not yield efficient results and must be fine-tuned individually. Thus in practice, this is a less useful method since the defender is unaware of the the attack type. They perform well with higher



(a) TA for 1% Poison Ratio

(b) ASR for 1% Poison Ratio

(c) TA for 10% Poison Ratio

(d) ASR for 10% Poison Ratio

Fig. 4.8 PBU methods results for a Blend signal attack

identification rates generally and in lower identification rate the performance is not consistent across different type of attacks and poison ratio.

- **Clean Fine Tune:** This naïve approach generally performs well with a higher number of samples but lacks robustness across different types of attacks. The observed performance improvement can be attributed to the use of a larger number of clean samples during re-training, which reinforces the model's learning of clean features, thereby enhancing test accuracy and reducing the Attack Success Rate (ASR).

- **Neural Attention Distillation and Adversarial Neuron Pruning:** These methods exhibited good performance even at limited availability of clean samples as observed in Figures 4.5 and 4.7, thus they are beneficial to be utilized in scenarios where clean data is scarce. From the results obtained, it is also observed that Neural Attention

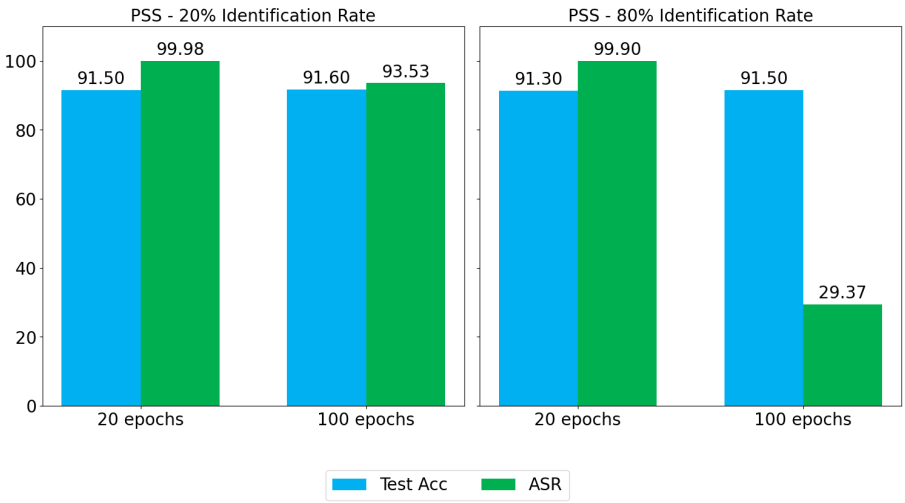


Fig. 4.9 PSS unlearning method results over 20 and 100 epochs

Distillation method has a higher impact towards the clean accuracy and improves it with the increase in the availability of bonafide clean samples provided for unlearning.

- **Adversarial Unlearning via Implicit Hyper Gradient :** This method exhibits highly unstable performance due to its sensitivity to hyperparameter settings, necessitating very precise adjustments to achieve desirable results. Fine-tuning the hyperparameters to the exact efficient point proves highly cumbersome and often leads to gradient explosions even with slight deviations from this precise tuning. For few experiments, the hyperparameters were adjusted to observe the effects; however, for most other experiments the gradient explosion was reported as such without further extensive fine-tuning as the thesis's objective is to understand the performance of the unlearning strategies, rather than achieving optimal results. In the results section, experiments that reported zero test accuracy and a 100% Attack Success Rate (ASR) indicate occurrences of gradient explosion. These instances are also illustrated in the plots provided in the Appendix B.

- **Catastrophic Forgetting:** It requires a sufficient number of clean samples to perform well. However, a balance must be maintained, as too many samples can lead to overfitting and poor performance which can be avoided by finetuning to specific sample sizes. Specifically visible in higher poison ratio. Further results were degraded for patch attack in comparison to other attacks.

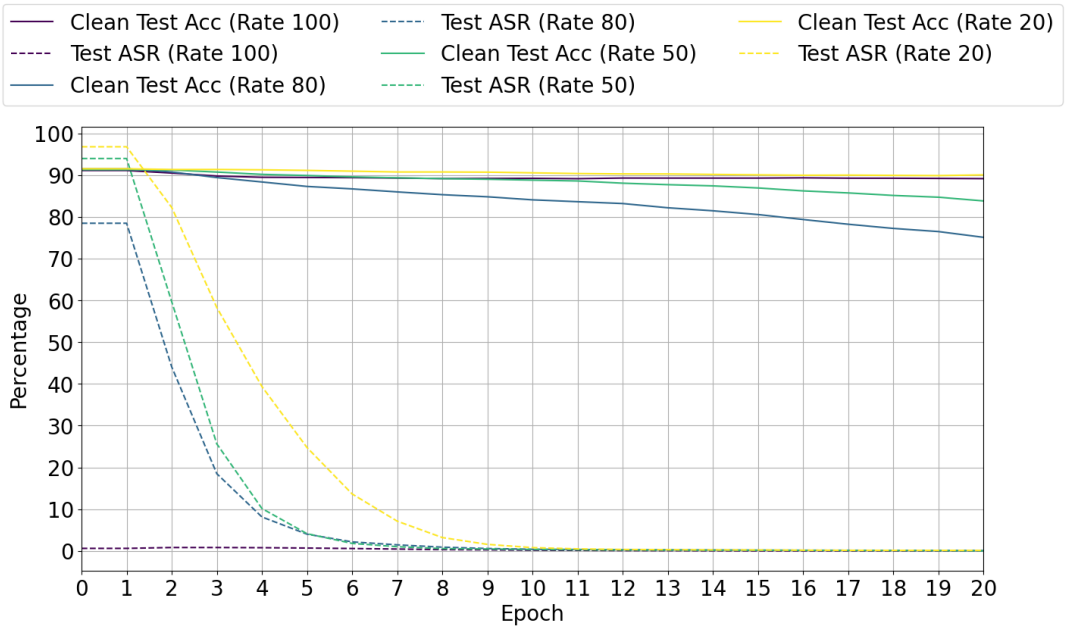


Fig. 4.10 ABL performace across epochs for a 1% poison ratio with Blend Signal Attack

4.5 Conclusion

The experiments conducted in this thesis have evaluated various backdoor unlearning methods, giving insights into their efficiency in a comparative basis under standardized conditions. These experiments have highlighted the strength and weaknesses of the unlearning strategies in addition to the emphasis on the dynamics of neural network security in the presence of backdoor attacks. The major discussions in this sections are concluded as,

1. **Effectiveness of Unlearning Methods:** The experiments demonstrated that no single unlearning method uniformly outperforms others across all scenarios. The effectiveness of each unlearning strategy varies across different settings such as type of backdoor attacks, the poison ratio, and the samples provided for unlearning. Thus, it supplements the requirement to implement the backdoor unlearning strategies based on a context-sensitive approach.

2. **Impact of Attack Types and Poison Ratios:** From the results it is clearly observed that the performance of the unlearning strategies is highly influenced by the type of backdoor attack and the level of data corruption. This finding highlights the importance of developing adaptive unlearning strategies that can cater to the specific characteristics of the threat and the dataset.

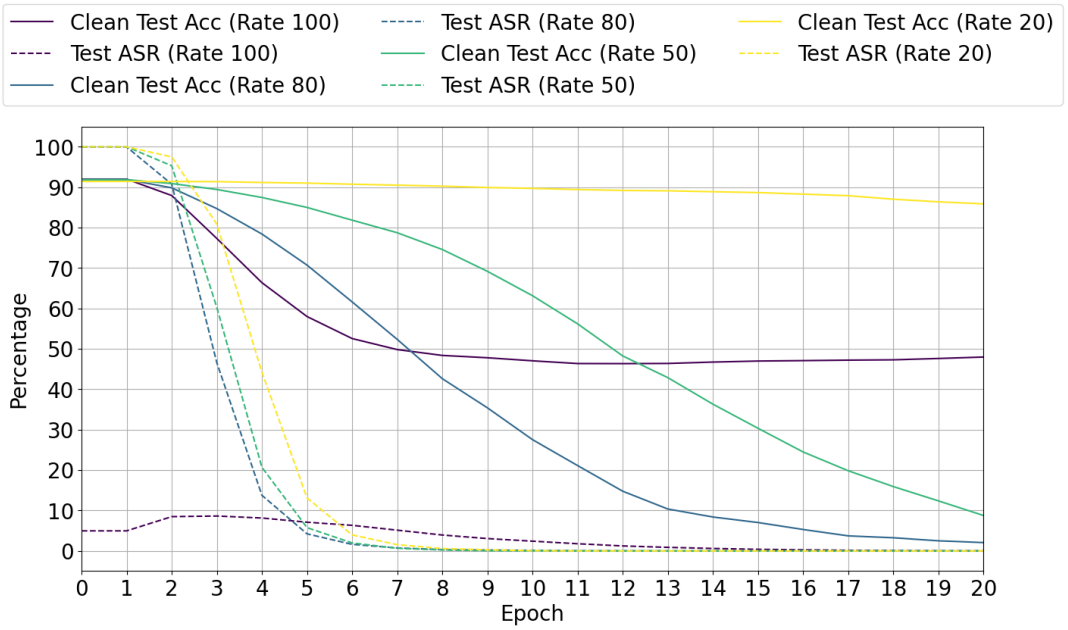


Fig. 4.11 ABL performace across epochs for a 1% poison ratio with Blend Attack of Checkerboard Pattern

3. **Insights into Unlearning Dynamics:** Through these experiments, useful insights are gathered into the mechanisms by which different unlearning methods mitigate the effects of backdoor triggers as detailed in this section. For instance, methods requiring accurate identification of poisoned samples were generally more effective but also more dependent on the quality of the identification process. In contrast, methods that do not rely on sample identification offered more flexibility but sometimes at the cost of lower overall effectiveness.

4. **Practical Implications for Real-World Applications:** In the experimentation, the practical challenges in implementing the unlearning strategies have also been emphasized. For instance, the need for bonafide samples for unlearning methods, precise hyperparameter tuning and the risk of overfitting or extensive unlearning degrading the performance on the clean dataset. These practical considerations are important for mapping the theoretical understanding of the unlearning methodologies to the effective real-world applications.

To conclude, a foundation for further research and development in backdoor unlearning methods is supported by the experimentation conducted in the thesis. It also provides a guidance for the practitioners in the field of machine learning security to evaluate and implement these methods based on the specific needs and constraints of the system. This

section has established the understanding that will guide the subsequent discussions on future work and the concluding remarks in the final chapter.

Chapter 5

Future Work

Backdoor unlearning strategies being a major field of research are leading to the development of new advanced strategies. This rapidly evolving nature poses significant opportunities to extend the evaluation to include these emerging unlearning methods which could lead to further beneficial insights. The robustness evaluation in this study was limited due to time and resource constraints. Thus, the evaluation framework could be expanded for further robustness checks which could involve testing the unlearning methods with different scenarios such as experimenting with various datasets, model structures, and more sophisticated backdoor attacks. One of the interesting possibilities for a future direction that has not been investigated in the current thesis due to time limitations was the potential to experiment with various identification methodologies. As we observe, each unlearning method documented in the literature is paired with a particular identification strategy and it exhibits good results. However cross implementing these identification methods with the different unlearning methods would be intriguing as they could uncover valuable insights, also with the possibility of obtaining optimal learning performance. As observed in the experimentation, few unlearning methods are sensitive to the identification phase, thus investigating the compatibility of different identification methods with unlearning methods could be beneficial. This not only improves the understanding of the dependence of unlearning method on the identification phase but also leads to the development of more adaptive and efficient unlearning processes. Further based on some specific insights obtained from the results it would be interesting to delve into the method of IBAU to understand the nature of its instability to different hyperparameters. Similarly to study the fragility of the PSS method towards inaccurate unlearning dataset.

Chapter 6

Conclusion

The study presented in this thesis contributes to the understanding of backdoor unlearning in neural networks, offering a comparative analysis of various unlearning methods under controlled conditions. This study's findings emphasize the importance of selecting an appropriate unlearning strategy based on the specific characteristics of the backdoor attack, the available data, and the operational requirements of the scenario.

6.1 Summary of Key Findings

- **Variability of Unlearning Effectiveness:** The experiments demonstrated that the effectiveness of unlearning methods varies significantly based on the nature of the backdoor attack, the poison ratio, and the identification accuracy of poisoned samples. This variability highlights the need for adaptive and context-sensitive approaches in backdoor unlearning.

- **Strategic Implications Based on Available Data:** For scenarios where only clean samples are available, the choice of unlearning strategy—such as Fine Tuning on Clean Data (CFN), Neural Attention Distillation (NAD), or Adversarial Neuron Pruning (ANP)—should be guided by the number of clean samples at hand. In contrast, when both poisoned and clean samples are available, the decision on which unlearning method to employ should consider factors like the speed of unlearning, the accuracy of backdoor sample identification, and the trade-offs between clean accuracy and Attack Success Rate (ASR).

- **Importance of Identification Accuracy:** The effectiveness of some unlearning methods heavily relies on the accurate identification of poisoned samples. This underscores

the importance of robust detection mechanisms as a prerequisite for effective unlearning.

6.2 Implications for Practical Applications

The results of this thesis provide actionable insights for practitioners in the field of machine learning security, especially those involved in developing or maintaining models susceptible to backdoor attacks. Employing the right unlearning strategy, based on the detailed evaluations provided, can enhance the resilience of neural networks against such threats in various practical scenarios. Moreover, the study underscores the significance of considering real-world constraints, such as the availability of clean or poisoned samples and the necessity for rapid response in operational environments.

6.3 Potential Extensions

Looking ahead, there are several promising avenues for extending this work:

- Implementing a matrix of identification methods against unlearning strategies to identify combinations that yield the best results. Such as exploring the effectiveness of an FCT metric-based identification as discussed in Chen et al. (2022), or a more standalone identification mechanism such as STRIP as given in Gao et al. (2019), against the different unlearning strategies.
- Expanding the evaluation framework to include emerging unlearning methods, such as those utilizing transformers as discussed by the authors Subramanya et al. (2024).
- The robustness can be tested by expanding the evaluation across more complex backdoor attacks such as dynamic trigger backdoor attacks which have been discussed by (Salem et al., 2022) or reflection backdoor attack as discussed by Liu et al. (2020).
- For the IBAU unlearning method, utilizing automated tools or Bayesian optimization to perform a detailed hyperparameter tuning to identify the most stable conditions which further helps us to delve into the reasoning of IBAU instability across configurations. This provides more reliability for practitioners to utilize the unlearning strategy.

6.4 Concluding Remarks

This thesis underscores the nuanced and complex nature of backdoor unlearning in neural networks. The insights gained not only advance the academic discourse but also provide some guidance for security practitioners to assess and implement effective unlearning strategies. As machine learning continues to expand across various sectors, the importance of safeguarding these systems from sophisticated threats cannot be overstated. Future research in this domain will be crucial in developing resilient machine learning infrastructures that uphold data integrity and trust in automated systems.

References

Weixin Chen, Baoyuan Wu, and Haoqian Wang. Effective backdoor defense by exploiting sensitivity of poisoned samples. *Advances in Neural Information Processing Systems*, 35: 9727–9737, 2022.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12043–12051, 2024.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pages 113–125, 2019.

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

Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. Towards adversarial evaluations for inexact machine unlearning. *arXiv preprint arXiv:2201.06640*, 2022.

Shashwat Goel, Ameya Prabhu, Philip Torr, Ponnurangam Kumaraguru, and Amartya Sanyal. Corrective machine unlearning. *arXiv preprint arXiv:2402.14015*, 2024.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2019.

Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2088–2105, 2020.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34:14900–14912, 2021a.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021b.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22, 2022.

Yudong Li, Shigeng Zhang, Weiping Wang, and Hong Song. Backdoor attacks to deep learning models and countermeasures: A survey. *IEEE Open Journal of the Computer Society*, 4:134–146, 2023.

Cong Liao, Haoti Zhong, Anna Squicciarini, Sencun Zhu, and David Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. *arXiv preprint arXiv:1808.10307*, 2018.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pages 273–294. Springer, 2018.

Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 182–199. Springer, 2020.

PyTorch. torchvision.transforms.totensor, 2023. URL https://pytorch.org/vision/main/generated/torchvision.transforms.ToTensor.html.

Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 703–718. IEEE, 2022.

Akshayvarun Subramanya, Soroush Abbasi Koohpayegani, Aniruddha Saha, Ajinkya Tejankar, and Hamed Pirsiavash. A closer look at robustness of vision transformers to backdoor attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3874–3883, 2024.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*, pages 707–723. IEEE, 2019.

Ruotong Wang, Hongrui Chen, Zihao Zhu, Li Liu, and Baoyuan Wu. Versatile backdoor attack with visible, semantic, sample-specific, and compatible triggers. *arXiv preprint arXiv:2306.00816*, 2023.

Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, and Ting Wang. An invisible black-box backdoor attack through frequency domain. In *European Conference on Computer Vision*, pages 396–413. Springer, 2022.

Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34:16913–16925, 2021.

Yizhen Yuan, Rui Kong, Shenghao Xie, Yuanchun Li, and Yunxin Liu. Patchbackdoor: Backdoor attack against deep neural networks without model modification. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9134–9142, 2023.

Yi Zeng, Si Chen, Won Park, Z Morley Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. *arXiv preprint arXiv:2110.03735*, 2021.

Appendix A

Results

Experiment Results of CBU and PBU based Unlearning Strategies

Poison Ratio	Methods	Sample Metrics	No Unlearning		100		250		500		1000		3000	
			TA	ASR	TA	ASR	TA	ASR	TA	ASR	TA	ASR	TA	ASR
1%	CFN		91.59	99.51	91.22	15.01	91.40	8.14	91.00	6.45	85.23	1.16	87.77	1.02
	NAD		91.59	99.51	62.88	0.33	72.36	0.37	84.40	0.04	82.66	0.01	87.42	0.53
	ANP		91.59	99.51	84.87	2.51	74.16	10.28	82.45	5.28	86.87	37.68	90.02	0.85
	IBAU		91.59	99.51	91.69	99.54	91.40	99.59	90.07	98.14	89.54	98.01	87.93	83.79
	CFU		91.59	99.51	91.55	16.17	91.54	9.80	91.57	9.38	91.21	10.04	91.17	7.46
10%	CFN		91.53	99.96	90.44	99.97	90.54	99.97	89.28	95.60	86.37	9.68	85.24	10.03
	NAD		91.53	99.96	62.27	9.86	68.36	0.97	79.33	3.52	82.43	1.68	86.35	2.50
	ANP		91.53	99.96	86.20	0.78	87.96	1.38	86.45	0.84	84.30	0.62	91.41	0.20
	IBAU		91.53	99.96	91.30	100.00	86.45	45.47	91.00	100.00	0.00	100.00	86.00	45.00
	CFU		91.53	99.96	90.51	99.98	90.50	99.97	89.28	94.81	84.83	10.23	86.68	36.63

Table A.1 Results of various CBU unlearning methods across different clean sample sizes used for unlearning in a blend signal trigger backdoored attack model

Poison Ratio	Methods	Sample Metrics	No Unlearning		100		250		500		1000		3000	
			TA	ASR	TA	ASR	TA	ASR	TA	ASR	TA	ASR	TA	ASR
1%	CFN		91.11	99.95	90.46	99.95	75.20	92.38	79.83	47.28	82.87	66.93	82.43	43.20
	NAD		91.11	99.95	50.20	1.17	54.84	14.31	75.38	6.90	81.34	1.20	87.15	3.10
	ANP		91.11	99.95	70.66	47.75	85.20	15.95	89.38	19.61	85.50	10.25	88.73	15.53
	IBAU		91.11	99.95	90.98	99.95	91.17	99.95	90.94	99.94	0.00	100.00	0.00	100.00
	CFU		91.11	99.95	90.98	8.14	90.92	4.78	90.26	5.86	90.05	2.01	89.52	2.50
10%	CFN		90.10	100.00	86.94	68.53	85.17	70.02	85.65	88.42	86.07	34.87	86.40	23.46
	NAD		90.10	100.00	64.48	42.77	75.13	10.96	80.83	52.02	85.23	46.76	86.42	7.80
	ANP		90.10	100.00	88.78	27.88	86.81	16.37	85.27	31.35	89.43	15.02	90.40	14.00
	IBAU		90.10	100.00	90.78	100.00	90.80	100.00	90.45	100.00	0.00	100.00	0.00	100.00
	CFU		90.10	100.00	87.00	69.40	85.00	70.67	85.12	84.38	86.27	47.60	86.43	25.01

Table A.2 Results of various CBU unlearning methods across different clean sample sizes used for unlearning in a signal trigger backdoored attack model with checkerboard pattern

“

Poison Ratio	Methods	Sample Metrics	No Unlearning		100		250		500		1000		3000	
			TA	ASR	TA	ASR	TA	ASR	TA	ASR	TA	ASR	TA	ASR
1%	CFN		89.05	95.52	78.34	86.84	83.17	61.55	80.33	55.33	83.02	33.82	82.94	47.00
	NAD		89.05	95.52	74.71	22.05	65.32	23.10	76.34	29.20	73.43	25.43	81.20	27.20
	ANP		89.05	95.52	54.17	94.70	78.40	62.14	78.43	79.95	80.68	86.01	84.81	59.72
	IBAU		89.05	95.52	11.11	0.00	0.00	100.00	0.00	100.00	26.46	3.14	0.00	100.00
	CFU		89.05	95.52	77.86	53.51	73.91	21.48	78.44	28.25	82.18	28.15	80.95	10.92
10%	CFN		89.32	99.95	70.93	16.94	73.93	32.85	78.53	25.00	77.28	23.65	78.12	21.28
	NAD		89.32	99.95	66.44	33.06	70.81	41.31	75.57	35.25	72.44	25.02	84.15	21.63
	ANP		89.32	99.95	66.43	18.48	83.37	4.30	76.08	10.08	85.30	7.10	84.70	8.19
	IBAU		89.32	99.95	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	81.04	9.87
	CFU		89.32	99.95	37.13	51.24	69.28	9.00	73.87	32.48	74.20	13.70	79.06	33.50

Table A.3 Results of various CBU unlearning methods across different clean sample sizes used for unlearning in a frequency attack backdoored model

Poison Ratio	Methods	Sample Metrics	No Unlearning		100		250		500		1000		3000	
			TA	ASR	TA	ASR	TA	ASR	TA	ASR	TA	ASR	TA	ASR
1%	CFN		88.70	99.50	81.59	97.11	83.32	89.82	79.91	99.93	81.60	86.70	81.13	66.08
	NAD		88.70	99.50	81.30	5.80	74.40	7.50	68.21	7.60	65.80	7.40	62.20	15.40
	ANP		88.70	99.50	66.96	96.21	43.58	75.47	47.73	97.85	70.06	93.70	83.64	33.88
	IBAU		88.70	99.50	88.20	99.61	87.03	98.68	86.59	3.20	83.70	2.16	78.60	4.35
	CFU		88.70	99.50	81.40	97.10	83.40	88.80	80.15	98.94	83.48	85.40	81.04	25.70
10%	CFN		86.71	96.38	76.72	88.26	83.39	68.72	82.77	77.02	83.19	74.47	80.22	27.76
	NAD		86.71	96.38	79.23	6.40	80.59	3.27	79.50	4.46	82.14	3.80	84.40	4.50
	ANP		86.71	96.38	69.88	7.32	70.18	2.97	84.62	3.14	86.29	3.37	86.11	4.17
	IBAU		86.71	96.38	86.70	3.85	86.40	5.27	86.00	5.21	85.40	3.53	80.64	4.48
	CFU		86.71	96.38	76.76	88.47	83.35	69.21	82.67	77.36	83.34	75.97	82.47	31.64

Table A.4 Results of various CBU unlearning methods across different clean sample sizes used for unlearning in a patch attack based backdoored model

Poison Ratio	Methods	Identification Rate Metrics	No Unlearning		100%		80%		50%		20%	
			TA	ASR	TA	ASR	TA	ASR	TA	ASR	TA	ASR
1%	RNR		91.59	99.51	82.90	11.95	83.44	10.46	84.44	73.55	82.56	80.90
	PSS		91.59	99.51	91.78	1.80	91.53	70.94	91.28	85.60	91.60	85.62
	ABL		91.59	99.51	90.04	0.80	90.75	44.06	91.23	59.67	91.33	82.25
	SSD		91.59	99.51	88.54	0.64	90.98	2.27	91.45	4.42	91.58	7.75
10%	RNR		91.53	99.96	83.02	1.24	90.44	83.02	84.52	97.76	78.57	94.62
	PSS		91.53	99.96	91.40	0.23	91.07	97.55	89.94	99.84	90.80	98.68
	ABL		91.53	99.96	90.70	0.70	53.80	0.40	76.00	0.20	86.80	2.20
	SSD		91.53	99.96	90.88	0.43	90.93	1.73	91.46	12.35	91.57	17.21

Table A.5 Results of various PBU unlearning methods across different identification rates used for unlearning in a blend signal trigger backdoored attack model

Poison Ratio	Methods	Identification Rate Metrics	No Unlearning		100%		80%		50%		20%	
			TA	ASR	TA	ASR	TA	ASR	TA	ASR	TA	ASR
1%	RNR		91.11	99.95	82.70	11.30	84.04	99.17	84.27	99.31	84.43	99.63
	PSS		91.11	99.95	90.53	10.61	90.53	99.92	90.56	100.00	90.83	99.98
	ABL		91.11	99.95	91.90	4.90	89.83	90.74	90.80	95.20	91.40	99.90
	SSD		91.11	99.95	88.21	1.55	89.76	1.89	90.92	4.36	91.14	4.77
10%	RNR		90.10	100.00	84.43	6.98	81.53	99.94	82.74	99.95	82.54	100.00
	PSS		90.10	100.00	90.21	9.53	90.35	100.00	90.17	100.00	90.28	100.00
	ABL		90.10	100.00	90.80	6.10	89.80	5.40	59.20	0.01	90.43	7.26
	SSD		90.10	100.00	90.16	17.75	90.76	26.64	90.84	100.00	90.80	100.00

Table A.6 Results of various PBU unlearning methods across different identification rates used for unlearning in a signal trigger backdoored attack model with checkerboard pattern

Poison Ratio	Methods	Identification Rate / Metrics	No Unlearning TA	No Unlearning ASR	100% TA	100% ASR	80% TA	80% ASR	50% TA	50% ASR	20% TA	20% ASR
1%	RNR		89.05	95.52	82.70	15.13	84.22	26.04	81.37	61.71	82.97	85.94
	PSS		89.05	95.52	89.04	64.45	89.01	81.52	88.80	87.45	89.05	85.03
	ABL		89.05	95.52	91.50	12.90	91.40	35.60	91.40	85.40	91.80	91.30
	SSD		89.05	95.52	78.18	12.22	87.91	62.58	88.51	77.40	88.65	77.41
10%	RNR		89.32	99.95	85.25	16.89	85.78	65.50	82.73	99.03	81.93	99.94
	PSS		89.32	99.95	89.16	0.60	87.96	99.98	87.47	99.96	89.34	99.84
	ABL		89.32	99.95	90.20	1.30	90.20	4.01	91.00	6.20	91.14	96.40
	SSD		89.32	99.95	81.67	9.80	84.80	20.90	89.04	99.97	89.33	100.00

Table A.7 Results of various PBU unlearning methods across different identification rates used for unlearning in a frequency attack backdoored model

Poison Ratio	Methods	Identification Rate / Metrics	No Unlearning TA	No Unlearning ASR	100% TA	100% ASR	80% TA	80% ASR	50% TA	50% ASR	20% TA	20% ASR
1%	RNR		88.70	99.50	82.90	8.40	81.95	66.73	84.60	95.40	84.31	98.67
	PSS		88.70	99.50	88.85	7.10	88.70	94.54	88.64	98.56	88.50	95.60
	ABL		88.70	99.50	90.90	2.90	91.40	87.20	91.01	89.30	91.30	98.10
	SSD		88.70	99.50	86.73	25.92	88.00	61.08	88.43	99.90	88.74	99.56
10%	RNR		86.71	96.38	83.02	2.12	83.66	38.12	81.32	78.12	79.47	94.18
	PSS		86.71	96.38	84.16	1.00	84.43	92.46	85.02	93.84	87.10	94.08
	ABL		86.71	96.38	91.00	1.00	90.00	1.00	90.00	2.00	91.00	98.00
	SSD		86.71	96.38	85.03	7.45	86.80	3.50	86.68	4.11	86.70	4.10

Table A.8 Results of various PBU unlearning methods across different identification rates used for unlearning in a patch attack based backdoored model

Visualized results of Model Training across different Attacks

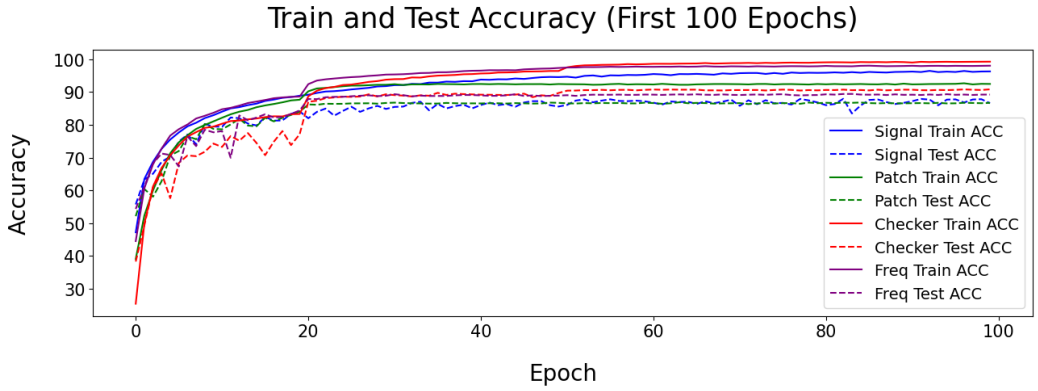


Fig. A.1 Model Train and Test Accuracy results for the different backdoor attacks with 10% poison ratio

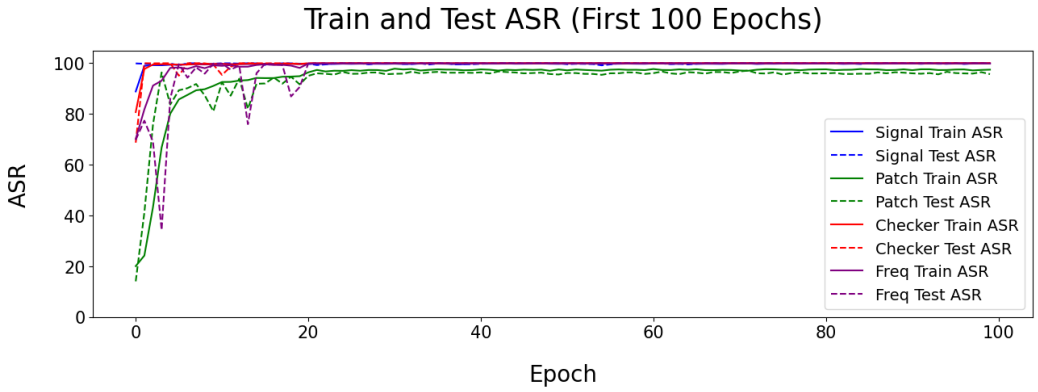


Fig. A.2 Model Train and Test ASR results for the different backdoor attacks with 10% poison ratio

Appendix B

Detailed Results Visualization

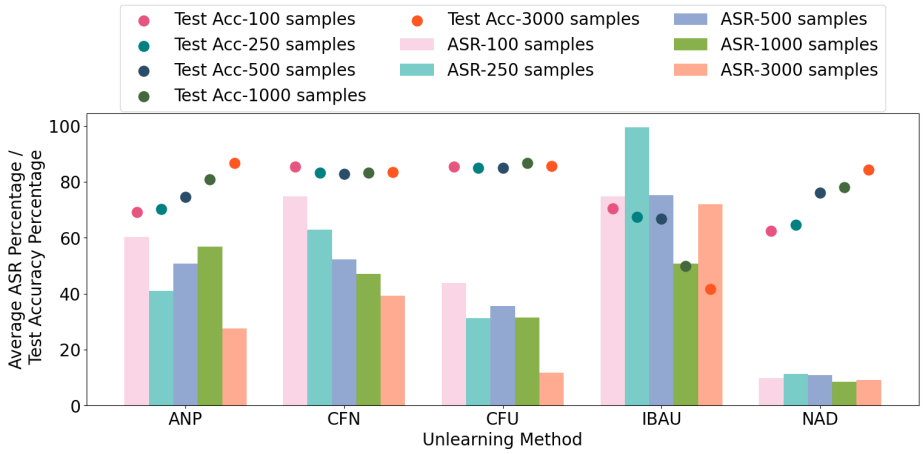


Fig. B.1 Average ASR and corresponding test accuracy for different CBU methods with 1% Poison Ratio

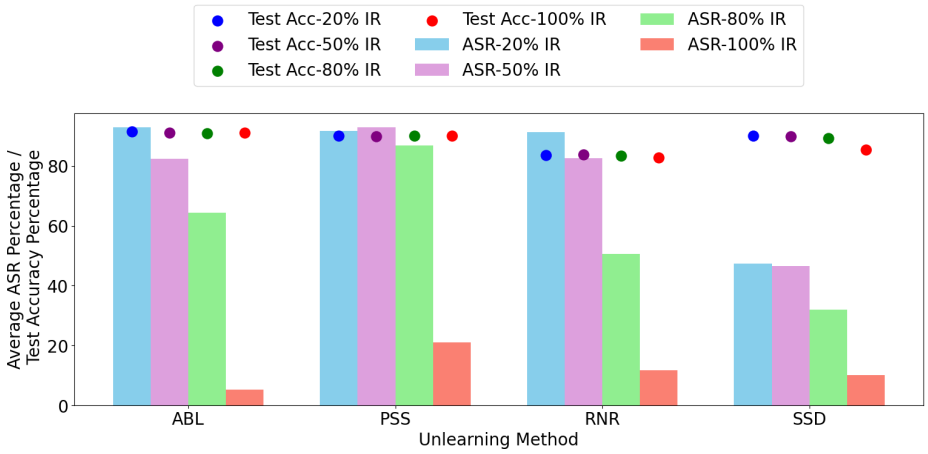
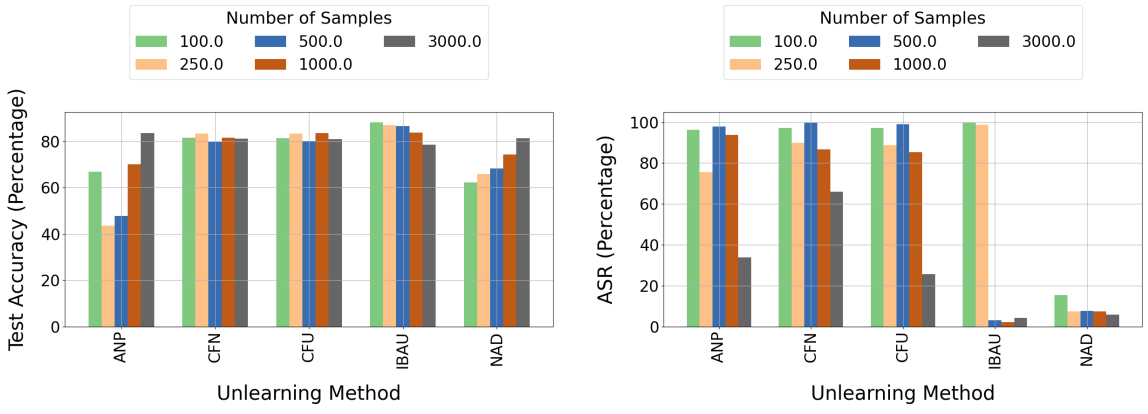
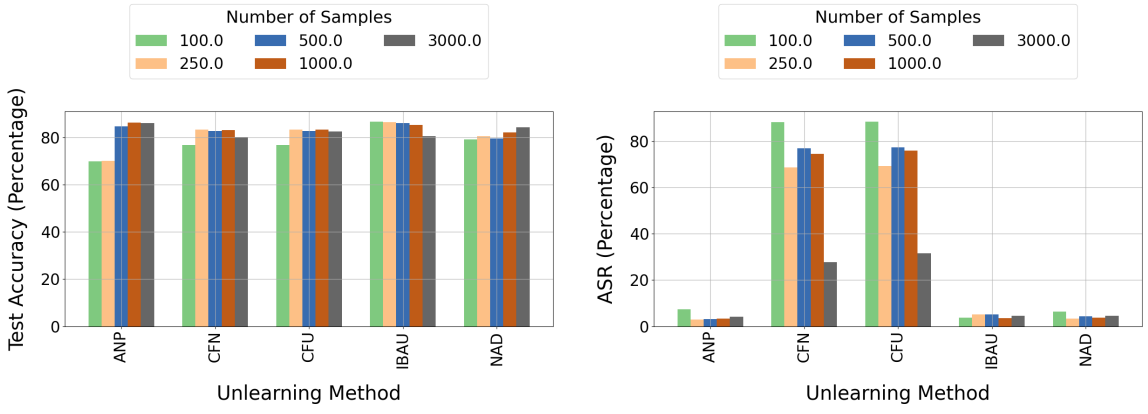


Fig. B.2 Average ASR and corresponding test accuracy for different PBU methods with 1% Poison Ratio



(a) TA for 1% Poison Ratio

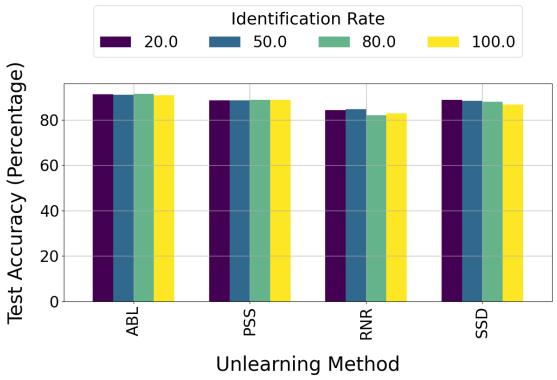
(b) ASR for 1% Poison Ratio



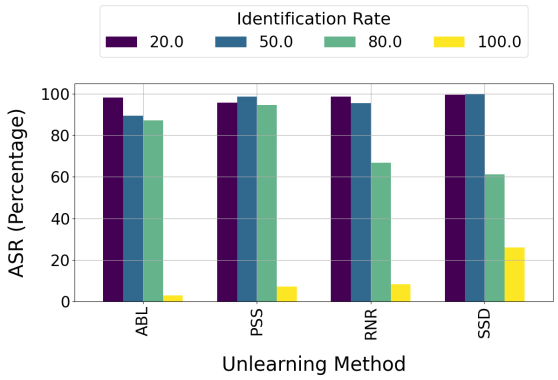
(c) TA for 10% Poison Ratio

(d) ASR for 10% Poison Ratio

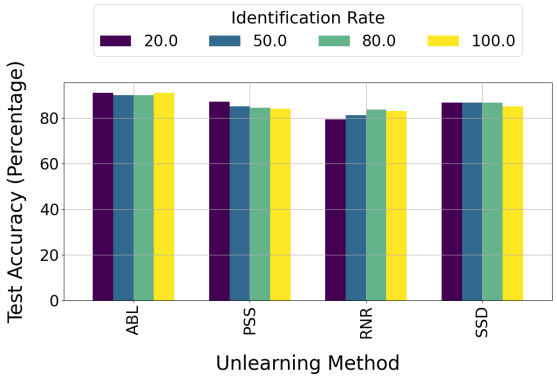
Fig. B.3 Visualization of results from CBU methods with Patch Attack



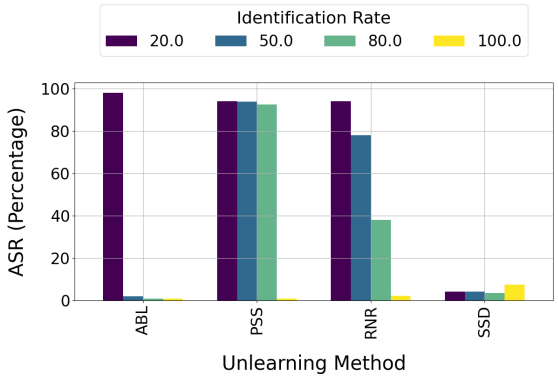
(a) TA for 1% Poison Ratio



(b) ASR for 1% Poison Ratio

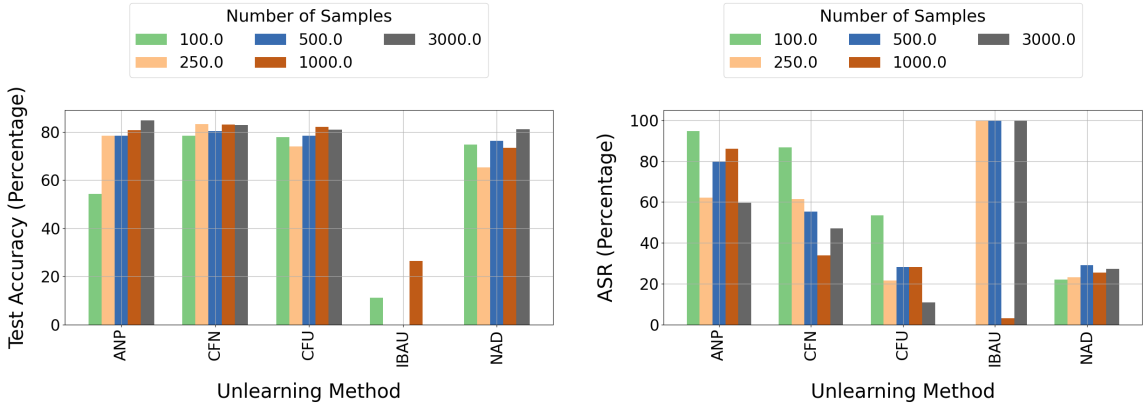


(c) TA for 10% Poison Ratio



(d) ASR for 10% Poison Ratio

Fig. B.4 Visualization of results from PBU methods with Patch Attack



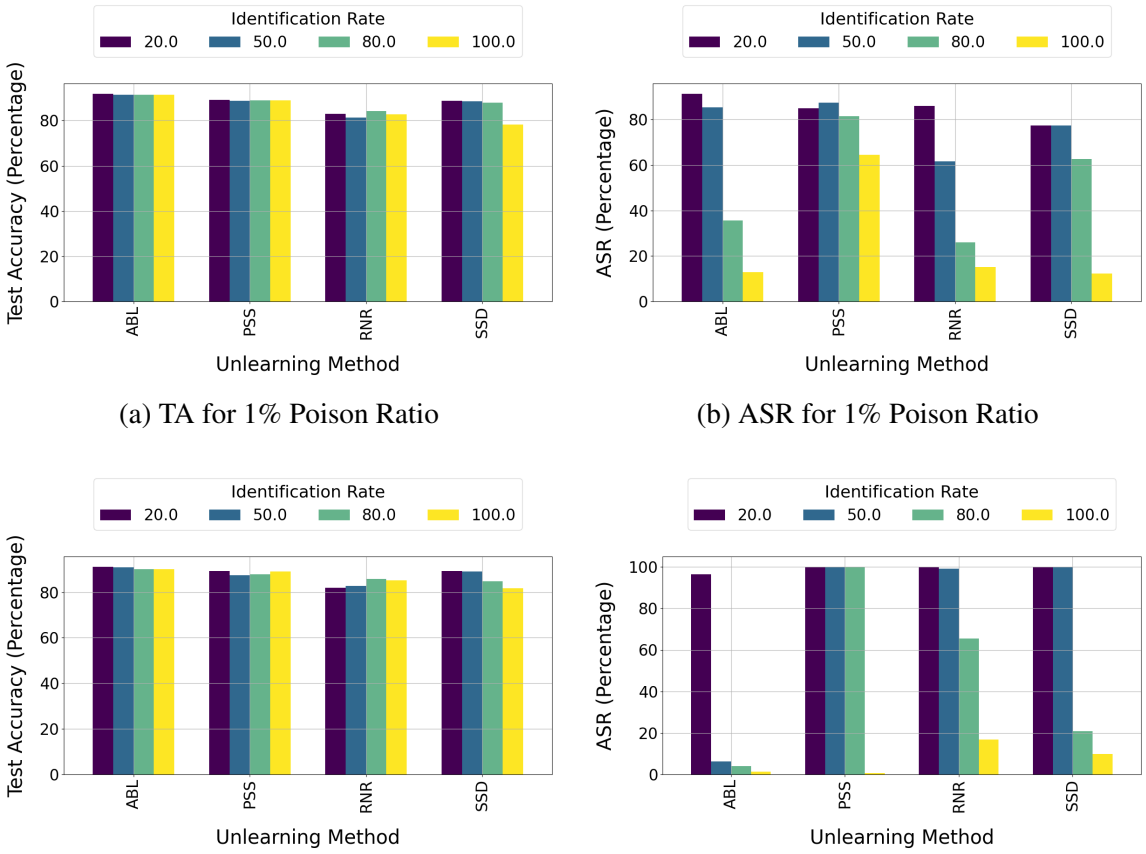
(a) TA for 1% Poison Ratio

(b) ASR for 1% Poison Ratio

(c) TA for 10% Poison Ratio

(d) ASR for 10% Poison Ratio

Fig. B.5 Visualization of results from CBU methods with Frequency Domain attack



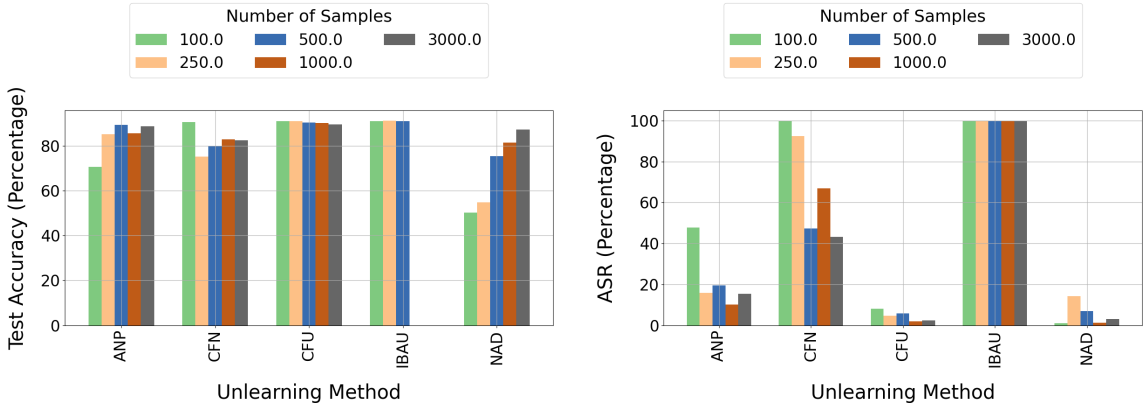
(a) TA for 1% Poison Ratio

(b) ASR for 1% Poison Ratio

(c) TA for 10% Poison Ratio

(d) ASR for 10% Poison Ratio

Fig. B.6 Visualization of results from PBU methods with Frequency Domain attack



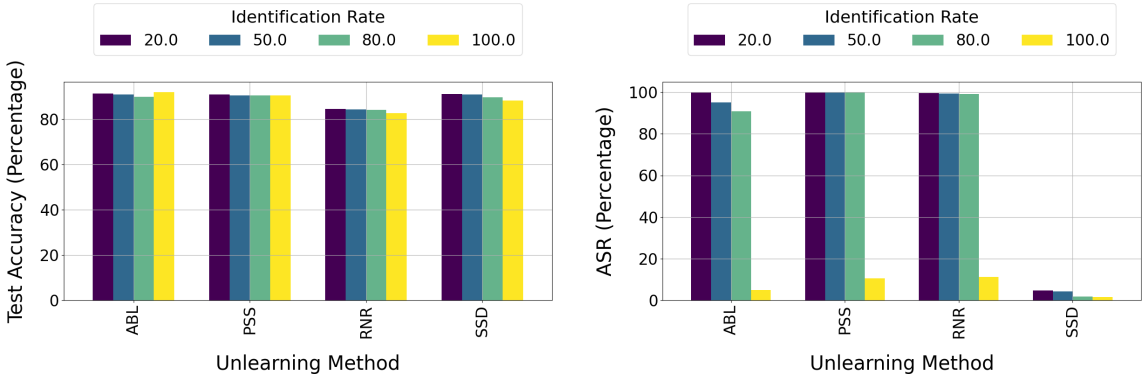
(a) TA for 1% Poison Ratio

(b) ASR for 1% Poison Ratio

(c) TA for 10% Poison Ratio

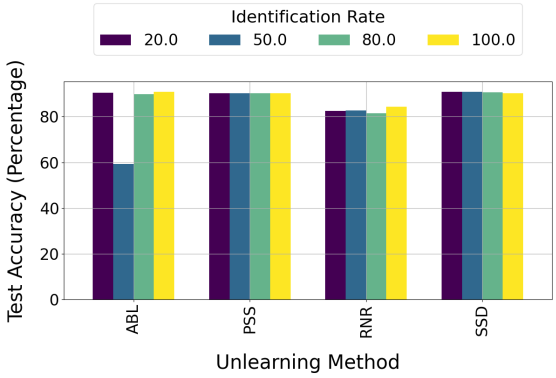
(d) ASR for 10% Poison Ratio

Fig. B.7 Visualization of results from CBU methods for a Blend Attack with Checkerboard pattern

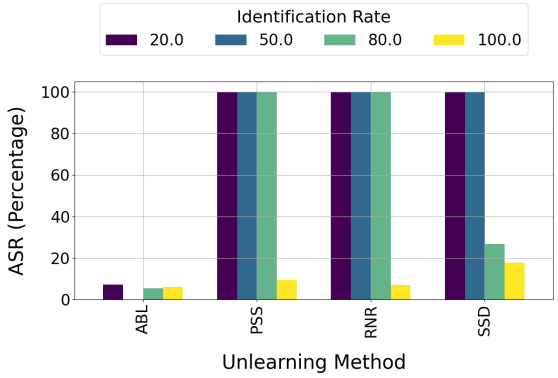


(a) TA for 1% Poison Ratio

(b) ASR for 1% Poison Ratio

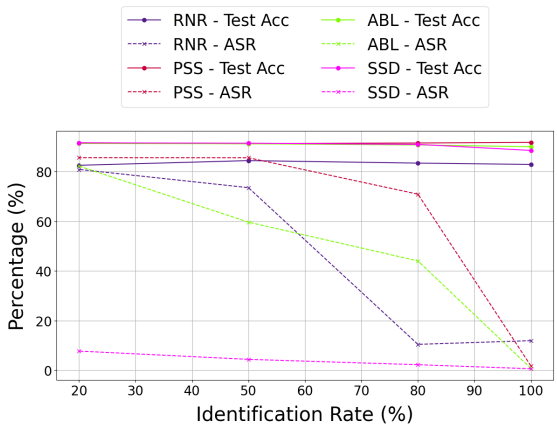


(c) TA for 10% Poison Ratio

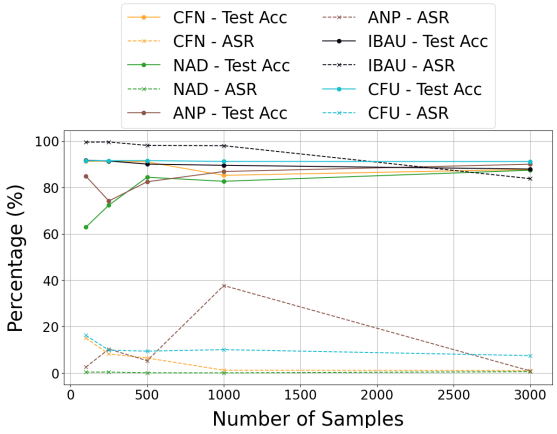


(d) ASR for 10% Poison Ratio

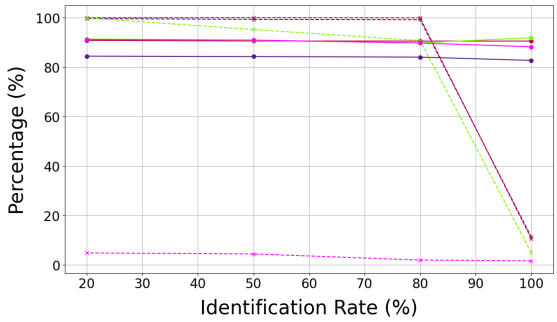
Fig. B.8 Visualization of results from PBU methods for a Blend Attack with Checkerboard pattern



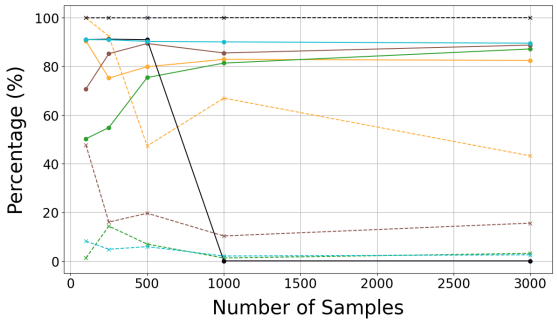
(a) PBU method with Signal Trigger results



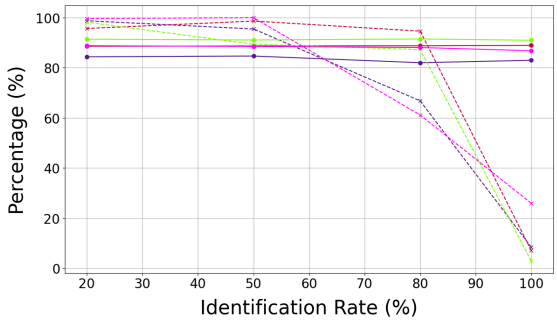
(b) CBU method with Signal Trigger results



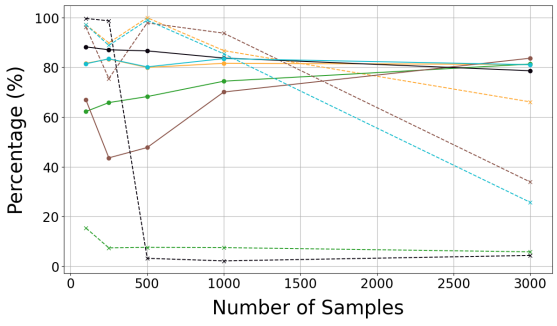
(c) PBU method with Checkerboard pattern Trigger results



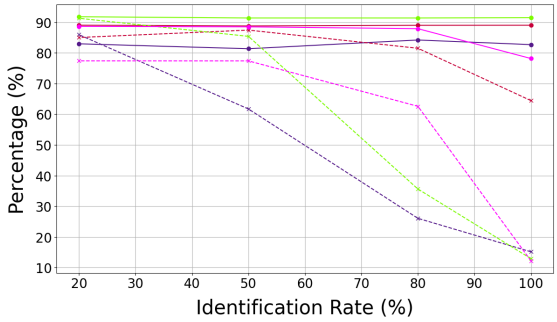
(d) CBU method with Checkerboard pattern Trigger results



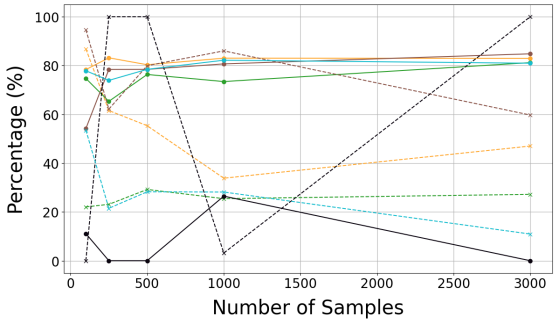
(e) PBU method with Patch Trigger results



(f) CBU method with Patch Trigger results

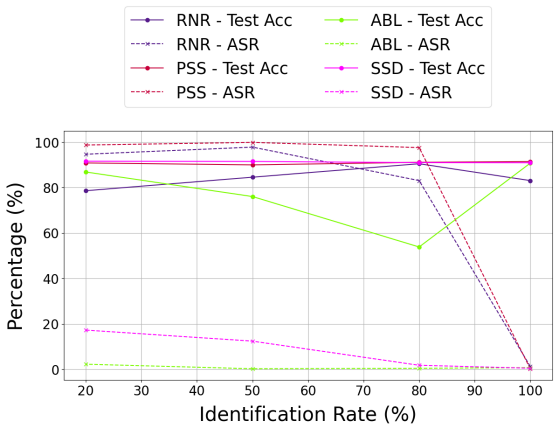


(g) PBU method with Frequency Domain Attack results

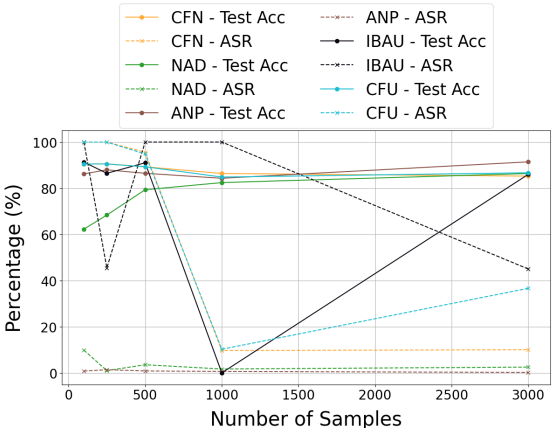


(h) CBU method with Frequency Domain Attack results

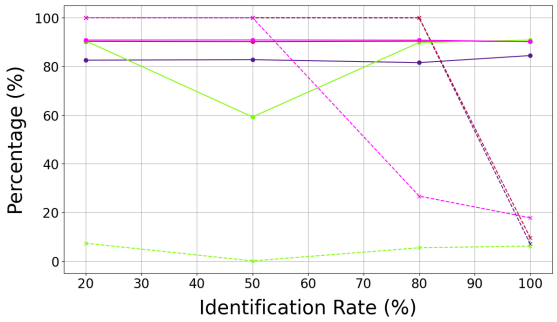
Fig. B.9 Comparative visualization of various unlearning methods effectiveness on different backdoor-attacked models with a 1% poison ratio, across different identification rates (PBU methods) and sample sizes (CBU methods)



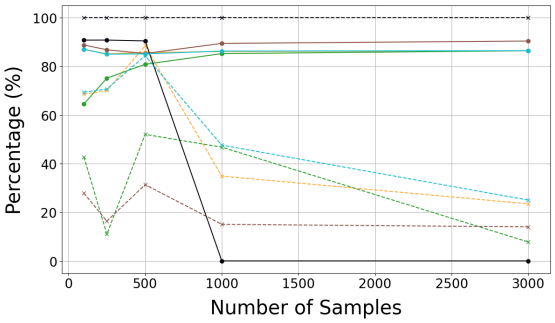
(a) PBU method with Signal Trigger results



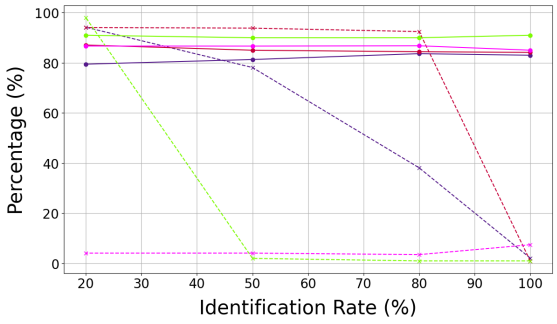
(b) CBU method with Signal Trigger results



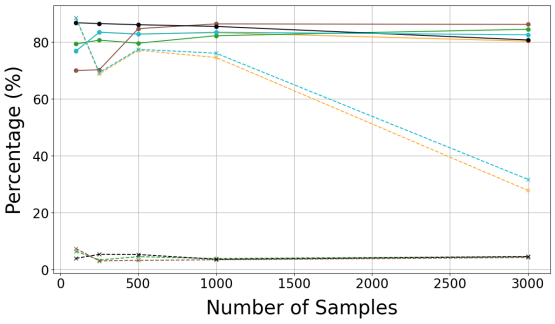
(c) PBU method with Checkerboard pattern Trigger results



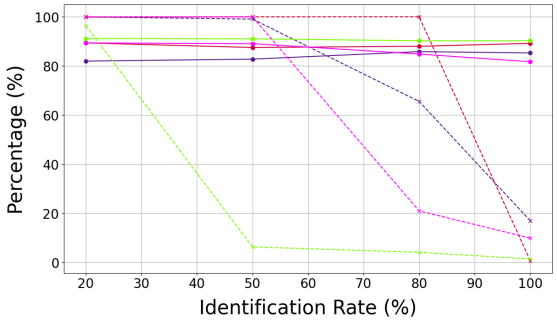
(d) CBU method with Checkerboard pattern Trigger results



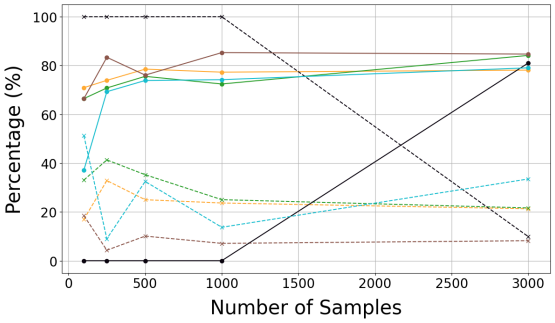
(e) PBU method with Patch Trigger results



(f) CBU method with Patch Trigger results



(g) PBU method with Frequency Domain Attack results



(h) CBU method with Frequency Domain Attack results

Fig. B.10 Comparative visualization of various unlearning methods effectiveness on different backdoor-attacked models with a 10% poison ratio, across different identification rates (PBU methods) and sample sizes (CBU methods)

Appendix C

C.1 Hyperparameter Settings-Model Training

Device: cuda
Model: Resnet18
Dataset: cifar10
Number of epochs: 200
Batch size: 128
Number of workers: 4
Learning rate: range(0.001 to 0.2), mostly used 0.01
Poison ratio: 0.1 or 0.01 (based on 10% or 1% experiment required)
Target type: all2one
Target label: 0
Criterion: Cross Entropy Loss
Optimizer: SGD
SGD momentum: range(0.6,0.9)
Transformations: Random Cropping, flipping, rotation
Learning Rate scheduler schedule: initially [20,80] (experimented other ranges as well)
Decay factor: 0.1 and 0.01

C.2 Tools and Packages Used

This appendix provides an overview of the primary software tools and libraries employed in the development of the project. Each tool is essential for specific aspects of the project.

- Python
- tqdm
- torchvision
- seaborn
- scipy
- OpenCV

Further, guidance was taken from the following GitHub repositories:

- https://github.com/csdongxian/ANP_backdoor
- https://github.com/bboylyg/ABL
- https://github.com/SCLBD/Effective_backdoor_defense
- https://github.com/if-loops/selective-synaptic-dampening
- https://github.com/bboylyg/NAD/tree/main
- https://github.com/YiZeng623/I-BAU
- https://github.com/shash42/Evaluating-Inexact-Unlearning

While the code has been rewritten from scratch to meet the specific requirements of this project, some small logic elements were adapted from these sources. The CIFAR10 dataset was utilized from the torchvision library.