Fall 2024	CS53331/4331 Adversarial Machine Learning	Assignment 3

Title	**The Poisoning Attack Stuff**
Due date	Monday, Dec 9th, **at noon**
First Name	
Last Name	
Student ID	
Marks	100

Note: Please answer the following questions and submit them through Blackboard. Be sure to submit it to assignment 3. DO NOT write the report by hand and submit a scanned version. Just write the answers in a Word document and submit it. Both Word and PDF submissions are accepted.

Note for CS5331: This assignment is optional; if you choose to do it, we will take an average of 4. If not, we will take the average of the previous three.

Submission Instruction (3 documents)

You are required to submit three documents:

1. ***Report.*** Just fill out the above report and submit it as a Word or PDF document.
2. ***Ipynb file.*** The code that you have written. Preferably in an ipynb document. You can submit it as a .py file as well.
3. ***Txt file of the code.*** We need your code in the .txt file as well. Use whatever way you prefer. The fastest would be to download the file as a .py file and change the extension to .txt

Objectives

This assignment has three main objectives:

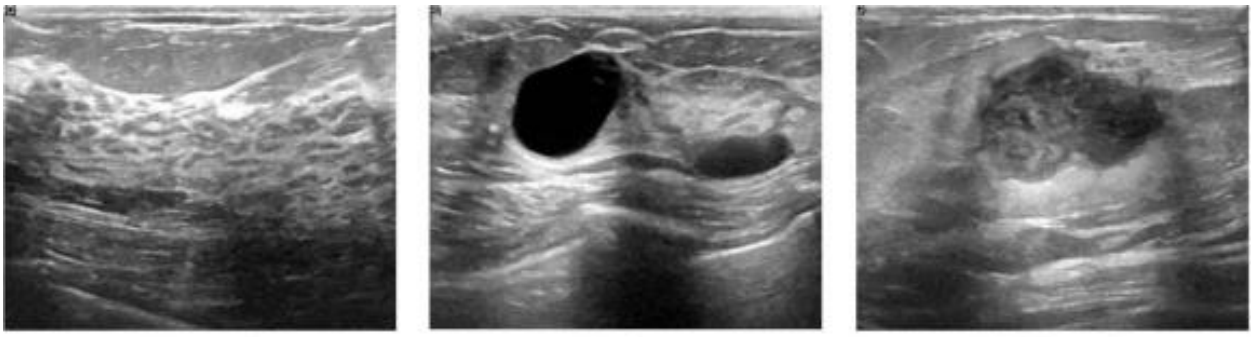
1. Implement poisoning attacks on the BUSI dataset
2. (Bonus) Defends against such attacks

Get started

Download the assignment files from Blackboard. You will need the report (This file) and the .ipynb file where you will put your code.

Dataset

We will use Breast Ultrasound Images Dataset (Dataset BUSI). The dataset can be downloaded from here. It is a dataset for Breast Cancer detection. It includes breast ultrasound images among women in ages between 25 and 75 years old. This data was collected in 2018. The number of patients is 600 female patients. The dataset comprises 780 images with an average image size of 500*500 pixels. The images are in PNG format. The ground truth images are presented with original images. The images are categorized into normal, benign, and malignant. Below is a figure taken from their paper. It is recommended that you upload the dataset into your personal Google Drive to follow the Colab instructions as they are. Of course, if you prefer to use other than Colab, you will need a similar preprocessing.



Normal Benign Malignant

Instruction for Colab

To get started with Google Colab, simply go to Google Colab, sign in with your Google account, and create a new notebook. You can write and execute Python code directly in the notebook. To access your dataset stored in your Google Drive (previous step), first run the following code to mount your Drive:

```
from google.colab import drive
drive.mount('/content/drive')
```

Follow the authorization steps, and your Drive will be accessible at **/content/drive/My Drive/.** You can then load your dataset into the notebook by providing the correct file path. This part of the code is provided for you in the .ipynb file of Assignment 0. You will need to setup the drive connection and run the code.

To use the free GPU provided by Colab, you can change the runtime to access a GPU by clicking on **"Runtime" > "Change runtime type" and** selecting **"T4 GPU"** from the **Hardware accelerator** dropdown menu. You can always use higher GPU powers at a cost (Colab Pro is $10 per month), but you should be fine with the free version, considering that you start the assignment early enough.

Colab comes with many pre-installed libraries, but if you need to install additional Python packages, you can do so with pip. For example:

```
!pip install library_name
```

Remember to save your work frequently.

After you've completed your work in Google Colab, you can easily download your notebook from Google Colab, go to **"File" > "Download" > "Download.ipynb"**.

Other than Colab

If you don't prefer Colab or notebook, you always have the option to run it on your computer (especially if it has a GPU) or access HPCC resources at TTU (needs an account with my permission).

Additional resources

1. TensorFlow resource https://www.tensorflow.org/
2. PyTorch resources https://pytorch.org/get-started/pytorch-2.0/
3. Deep learning with Python https://dl-with-python.readthedocs.io/en/latest/
4. Get started with Colab https://colab.research.google.com/

Backdoor Attack

Backdoor Attack in this assignment will be based on the paper by Wang et al. (2019). These attacks were covered in class. This notebook in the Adversarial Robustness Toolbox provides an example of applying the Backdoor Attack. It explains the proposed defense strategy presented in the paper, which is called Neural Cleanse. The defense is not required in this assignment but is a bonus. The defense is not necessarily covered in class.
Please follow the instructions in the notebook and modify the codes as needed. You will need to make some (major) changes to the code.

Task 1-Build and analyze the model to attack (20 pts)

The first task will require you to train a deep-learning model to classify BUSI images. The training, validation, and testing datasets are already given to you. For full marks, the classification test accuracy is expected to be above 85%. Further, you should not have an overfitted model. This will look like a model you already built for a previous assignment, and if you recognize it and want to use it, feel free to do so.

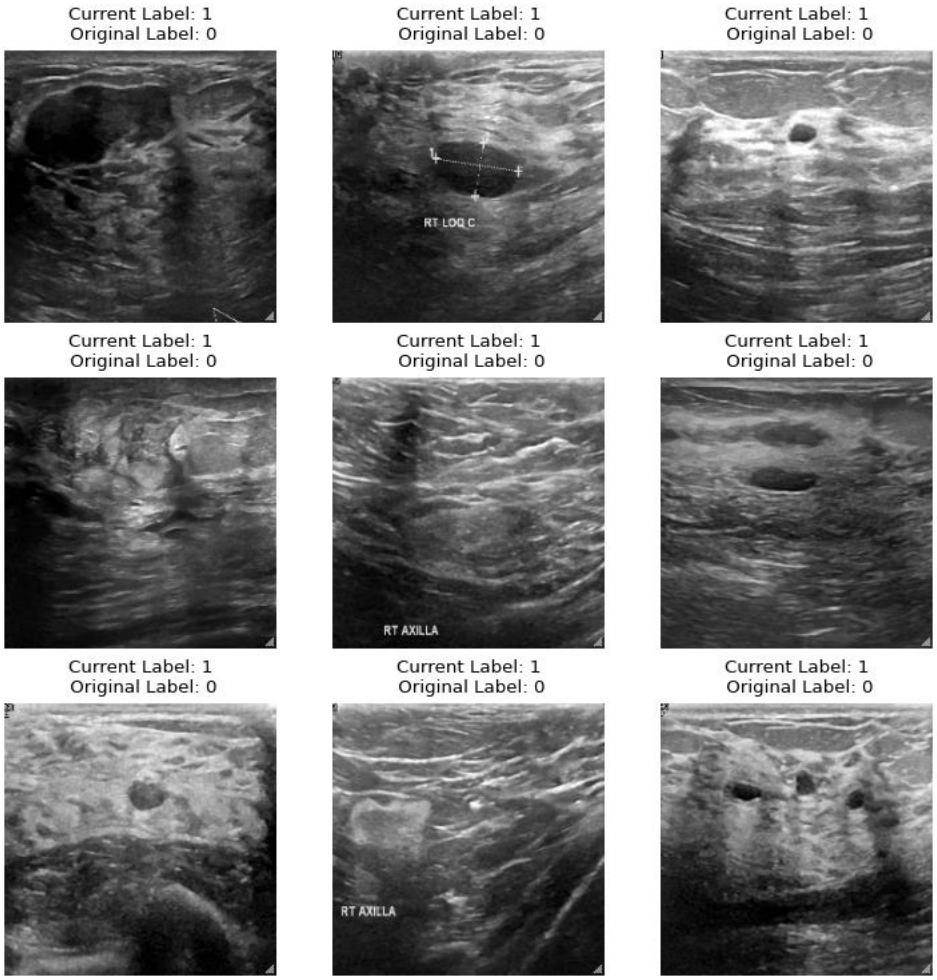
1. [5 pts] Implement what is required. If you decide to use the previous assignment, please copy its code or give us a reference.
   Estimated time on GPU: between 5 and 20 minutes.

2. [5 pts] Fill in Table 1 for your classification accuracy by the model for benign, malignant, and normal images. Make sure this is on the testing dataset.

Table 1: Classification accuracy of the models on the BUSI dataset

Model	**Benign**	**Malignant**	**Normal**	**Overall**

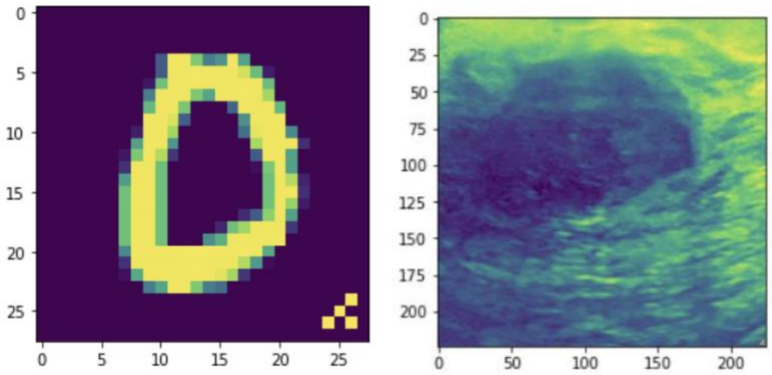
3. [5 pts] Plot a set of 9 images, showing the image, its label, and its prediction. The figure below is a simple example.



4. [5 pts] Briefly provide insights on the possible reasons for the differences in the different classes' accuracy.

Task 2: Backdoor attack (80 pts)

This task aims to misclassify poisoned benign images (class label 0) with the backdoor pattern as malignant images (class label 1). Therefore, the poisoned model should have high classification accuracy on images without a backdoor pattern and low classification accuracy on images with a backdoor pattern. Keep that in mind while working on the task.
The backdoor type in the example notebook in ART is the pattern of 4 pixels shown on the left of the figure below. However, this pattern was used with 28×28 pixels MNIST images. This pattern is hardly noticeable since BUSI images are of size 224×224 (shown on the right of the figure). Therefore, we need to change the approach. We have implemented the changes for you. You just need to locate and analyze what it is doing.



Questions:

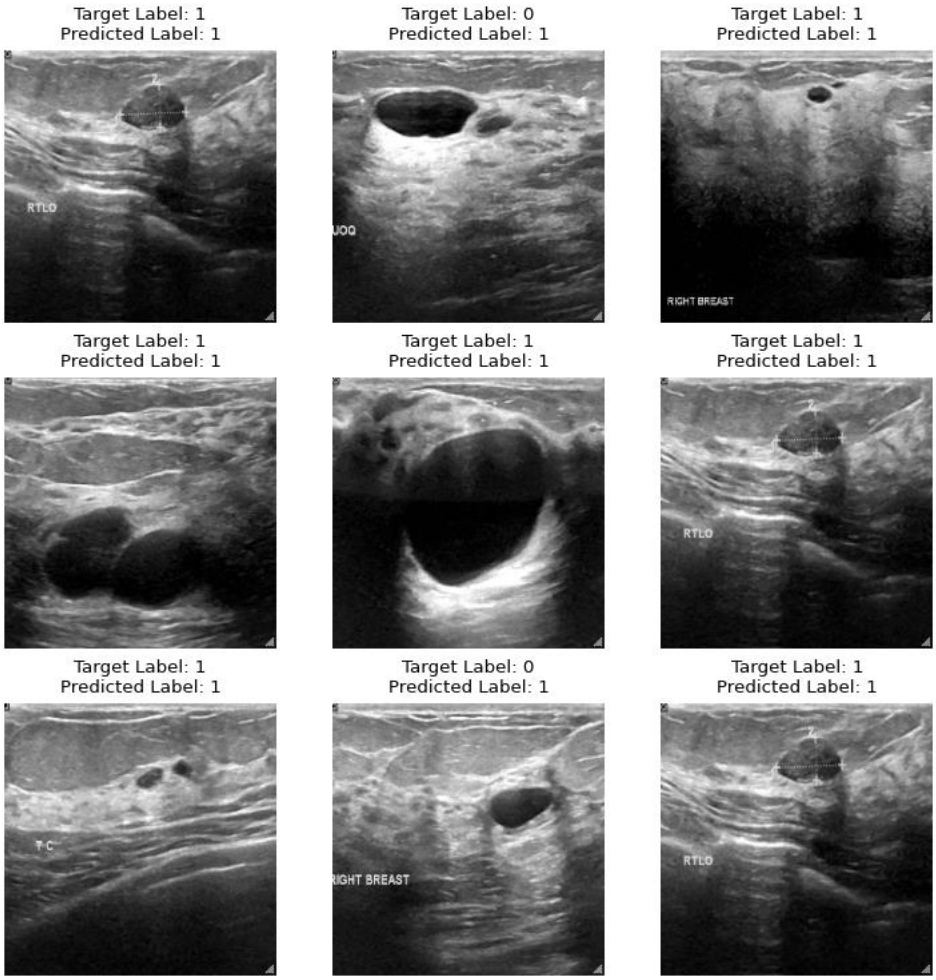
1. [10 pts] Answer the following questions for the implemented attack.
    a. What type of modifications to images are implemented? What each one of them is doing?
    b. What does the poison_dataset do? What does it return? Be sure to have details here.
2. [10 pts] Implementation. Implement poison_dataset function that takes clean images, clean labels, percentage of poisoning, and the poisoning function. The function should return 4 arrays, including a Boolean if the sample is poisoned or not, the sample, the label for that sample, and the original dataset label for that sample. The provided notebook is your guide; however, there are changes that you need to make.
3. [5 pts] Implementation. Create poisoned training images using BUSI images' training and validation sets. Select the percentage of poisoned images to be 20%. You may choose a different value if you wish to.
4. [5 pts] Plot at least 9 images with the applied backdoor pattern and display the target label for the images and if they are poisoned or not.

5. [5 pts] Create a poisoned test dataset by adding poisoned images to the original test dataset of 156 images.
6. [10 pts] Implementation. Train a poisoned model on the poisoned set of images. You can try training for a few epochs (maybe around 15 epochs), but if the attack success rate is low, you can retrain the model for longer.
   Estimated time on GPU: between 3 and 10 minutes.
7. [5 pts] Evaluate the poisoned model on clean test images and report the classification accuracy. Fill in Table 2. The classification accuracy on clean test images should be high and not significantly lower than the original accuracy of the model. For full marks, the accuracy should be at least 80%.

Table 2: Classification accuracy of the poisoned models with clean labels on the BUSI dataset

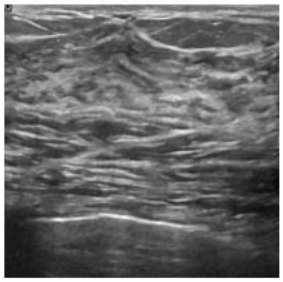
Model	**Benign**	**Malignant**	**Normal**	**Overall**

8. [5 pts] Plot at least 9 clean images, and show the true, predicted class label, and if the image is poisoned or not. The figure below is an example without the poisoned image label. Make sure to add that.

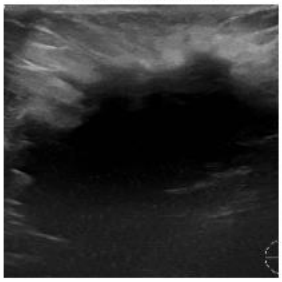


9. [5 pts] Briefly provide insights on the performance so far.
10. [5 pts] Evaluate the model on poisoned test images. Report how many of the poisoned benign images were classified as malignant images. For full marks, the attack success rate should be above 70%.
11. [5 pts] Plot at least 9 poisoned images, and show the target predicted class label, and if the image is poisoned or not.
12. [5 pts] Briefly provide insights on the previous two question
13. [10 pts] Plot at least 12 poisoned random images from all, and show the target predicted class label, and if the image is poisoned or not. An example is shown below.

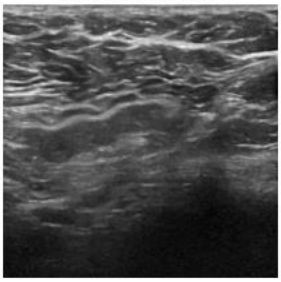
Original Target Label: 2
Predicted Label: 2
is poisened? False



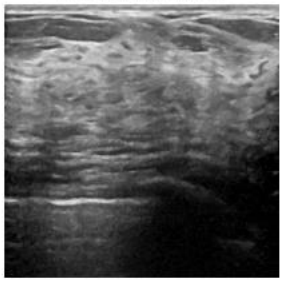
Original Target Label: 1
Predicted Label: 1
is poisened? False



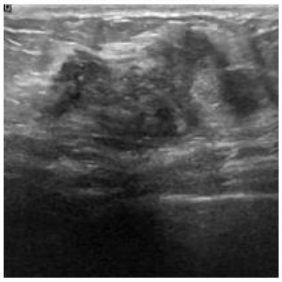
Original Target Label: 2
Predicted Label: 2
is poisened? False



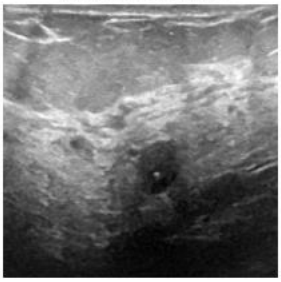
Original Target Label: 2
Predicted Label: 2
is poisened? False



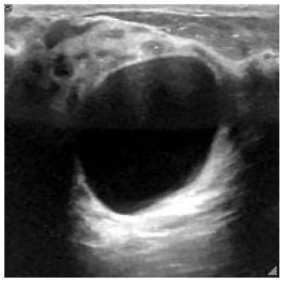
Original Target Label: 1
Predicted Label: 1
is poisened? False



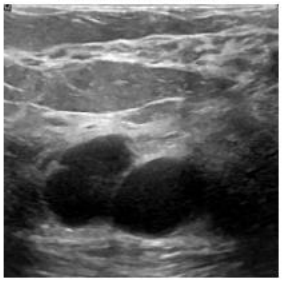
Original Target Label: 0
Predicted Label: 0
is poisened? False



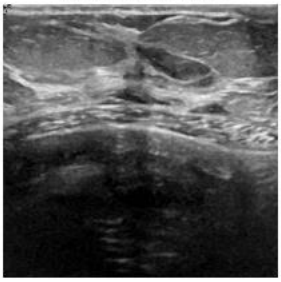
Original Target Label: 0
Predicted Label: 1
is poisened? True



Original Target Label: 0
Predicted Label: 0
is poisened? False



Original Target Label: 0
Predicted Label: 0
is poisened? False



Task 3 (Bonus) Defending Backdoor attacks (10 pts)

Bonus marks can be obtained for implementing the defense and mitigation strategies against the backdoor attack described in the example notebook. Report on the results and provide your insights below.

Submission Instruction (3 documents)

You are required to submit three documents:

1. ***Report.*** Just fill out the above report and submit it as a Word or PDF document.
2. ***Ipynb file.*** The code that you have written. Preferably in an ipynb document. You can submit it as a .py file as well.
3. ***Txt file of the code.*** We need your code in the .txt file as well. Use whatever way you prefer. The fastest would be to download the file as a .py file and change the extension to .txt