Stealthy Sabotage: Backdoor Attacks against Explainable Machine Learning

Krystof Latka
University of California, Los Angeles

James Shiffer
University of California, Los Angeles

Steven Andrew Lewis
University of California, Los Angeles

Mike Qu
University of California, Los Angeles

ABSTRACT

Explainable machine learning gives users the ability to analyze and understand model predictions. Explainer models usually analyze trained weights and model inputs to justify the predictions. However, attackers can maliciously manipulate the explainer, the original prediction model, or both to produce unfaithful explanations and/or predictions. This project explores backdoor attacks against explainable machine learning by iterating on existing work and improving the stealthiness, data efficiency, and sophistication of such attacks.

In a later part of the project, we focus on applications to the field of medical imaging by performing the Full Disguise attack on the NiH Chest X-Ray 14 dataset. We show that medical applications of ML generally face the issue of low classification accuracy of the original model, which makes explanations unreliable and attacks relatively easy to carry out.

1 INTRODUCTION

Most machine learning architectures deployed on various applications, including image classification, require large amounts of training data and millions of trainable parameters to achieve satisfying results. Thus, the process of training machine learning models requires extensive computational (GPU) resources which are standardly not available on users' personal physical devices. Hence, outsourcing the training of a machine learning model to the cloud has recently become very popular.

However, this outsourcing scenario comes with a new security threat. Assuming that the training of a machine learning model is partially or fully outsourced to a third-party platform, the platform can in turn provide the user with a model that performs well on the original prediction task, but contains a backdoor that is activated by a secret pre-selected type of trigger (such as a small white square in the corner of an image). [2]

The effect of such a backdoor attack can be mitigated if the user applies an explainer model atop of the original prediction model, which utilizes analyzes the weights and biases of the trained prediction model along with the input data, and tries to provide explanation for specific predictions. However, assuming the third-party has complete control over the training process during the time the model is outsourced, it can also adjust the training process so that various different explainers are fooled into providing invalid explanations, diverting the user from the actual source of attack.

Throughout this project, we will focus on backdoor attacks performed on image classifiers and their corresponding explainers. We begin by validating several backdoor attacks suggested before, we later improve their efficiency on standard datasets such as CIFAR-10, and we also focus on medical imaging by performing attacks on the NiH Chest X-Ray 14.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 BACKGROUND AND RELATED WORK

To familiarize ourselves with the project area, we attempted to recreate and verify several previously backdoor attacks documented in previous literature, namely BadNets from [2], as well as the Fooling and Fully Disguised attacks from [5]. We believe that [5] provides the best introduction to the most fundamental types of attacks on image classifiers and their corresponding explainer models.

Furthermore, we later focused more closely on the medical field because it demands AI explainability, being characterized by making critical decisions that involve human life. [1] Research has gone into the use of deep learning models in computer-aided diagnostic systems and medical data has been compiled into benchmark datasets for a range of diseases. We specifically applied a Full Disguise backdoor attack against a classification model and its corresponding explainer, trained on the NiH Chest X-Ray 14 dataset, a collection of more than 100,000 chest radiograph images with 14 different classes of detected diseases. [8]

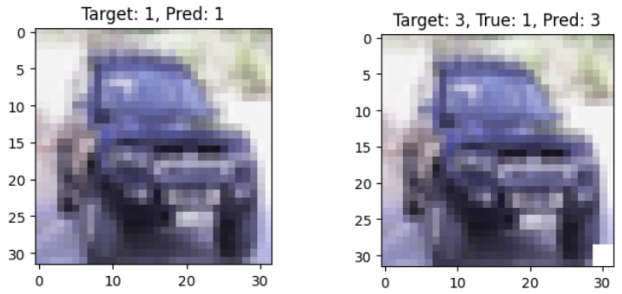
3 RESULTS

3.1 BadNets Attack on CIFAR-10

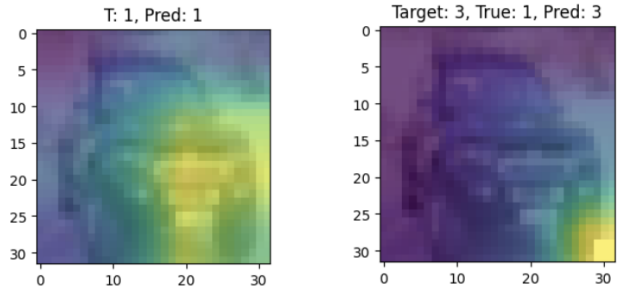
The BadNets attack, based on [2], is the simplest attack we implemented simply to test the robustness of our backdoor attack on the prediction model. This attack aims to maintain high prediction accuracy on clean images while also training the model so that when a poisoned image with a trigger (small white square in the lower right corner) is encountered, the prediction is forced to a specific class pre-selected by the attacker.

This attack is not concerned with modifying the explanation at all. Thus, a user can easily detect an ongoing attack by applying the explainer on the trained prediction model, as we will show below.

We use ResNet-18 [3] as our prediction model architecture and CIFAR-10 as our image dataset. The performance of the BadNets attack on a sample image from the test set, along with the output of the Grad-CAM explainer [6], is shown below in Figure 1.



(a) Prediction - Clean **(b)** Prediction - Poisoned



(c) Explanation - Clean **(d)** Explanation - Poisoned

Figure 1: Sample image from the BadNets attack

The figure above shows a successful attack, during which the prediction of the model is forced to the pre-selected target class 3 instead of the correct class 1.

After 5 epochs of training, the backdoored prediction model trained on a mixture of clean and poisoned data achieves an accuracy of 0.8283 on the clean test set, which is comparable to an accuracy of 0.8244 of the original prediction model simply trained on the clean dataset. Furthermore, the backdoored model achieves an accuracy of 0.9531 on the poisoned test set, which reflects the success rate of the attack. Since the performance on the clean test set is not significantly dampened by the introduction of a backdoor, and the success rate on the poisoned test set is close to 1.0, we can consider the attack to be successful.

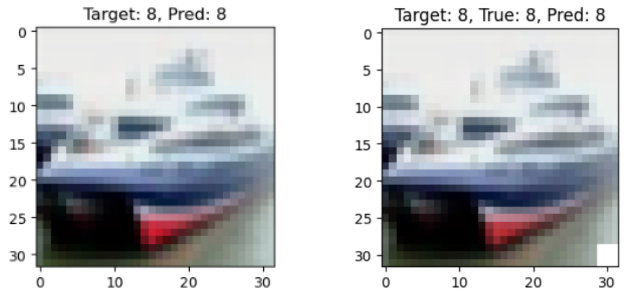
Nevertheless, we can see that the explainer in Figure 1d identifies the trigger pixel as the source of the manipulated prediction, which would allow the user to notice an ongoing attack. In future attacks, we attempt to manipulate the explainer to change its behavior and divert the user's attention from the true source of the attack.

3.2 Fooling Attack on CIFAR-10

The Fooling Attack is slightly more advanced than the classic BadNets backdoor attack. In this attack, we achieve basic manipulation of the explanation model while retaining control over the prediction model using a trigger and methodology detailed in [5]. Our loss function followed the form

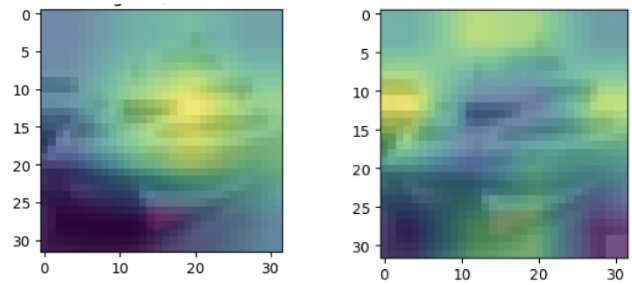
$$\mathcal{L}(\mathbf{x}, y; \theta) = (1 - \lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}, y; \theta) + \lambda \cdot \mathcal{L}_{MSE}(h_p(\mathbf{x}; \theta), \mathbf{r_x})$$

With this attack, we do not manipulate the prediction but force the explanation to a specific shape whenever the trigger is encountered. For the fooling attack, we set $\mathbf{r_x} = target$, where the target is any clearly defined shape to be seen in the explanation output, for example, a cross or a circle. A clear depiction using the Grad-CAM explainer can be seen in Figures 2 and 3. Again, we use ResNet-18 as our prediction model architecture and CIFAR-10 as our image dataset.



(a) Prediction - Clean **(b)** Prediction - Poisoned

Figure 2: Sample image from the Fooling Attack - Prediction model



(a) Explanation - Clean **(b)** Explanation - Poisoned

Figure 3: Sample image from the Fooling Attack - Explainer model

In the above figure, we can see the result of the Fooling Attack, where the explainer was forced to output an explanation in the shape of a square. After 3 epochs of training, the backdoored prediction model trained on a mixture of clean and poisoned data achieves an accuracy of 0.8619 on the poisoned test set. Furthermore, we do not have a quantitative measure for the performance of the explainer, but as we can see from Figure 3b above, the shape vaguely resembles a square. Furthermore, Figure 2b shows that the prediction of the model has not been affected by the attack, which is expected.

3.3 Fully Disguised Attack using GradCAM

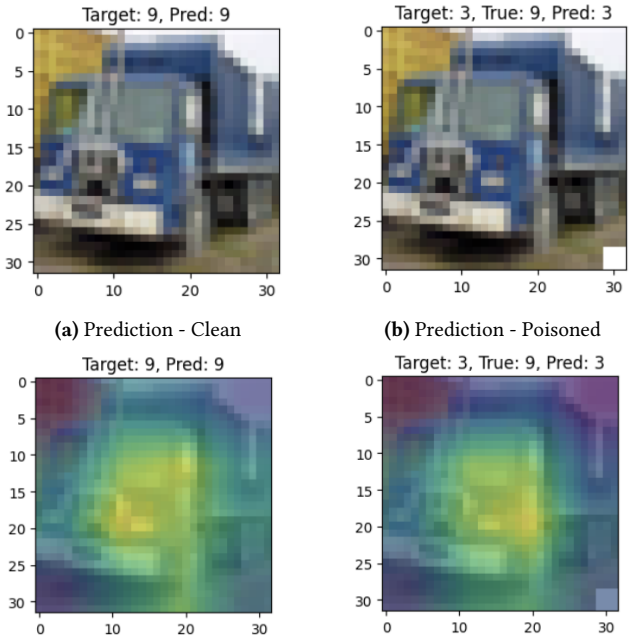
The Fully Disguised Attack enforces a specific target prediction if a poisoned test sample is encountered while keeping the explanation of the original clean sample. The explainer does not show any visual sign of the input trigger or a change in the model prediction, which makes it difficult for the user to uncover an ongoing attack. Similarly to the Fooling attack, we followed the methodology suggested by [5] and used the following loss function:

$$\mathcal{L}(\mathbf{x}, y; \theta) = (1 - \lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}, y; \theta) + \lambda \cdot \mathcal{L}_{MSE}(h_p(\mathbf{x}; \theta), \mathbf{r_x})$$

where $\mathcal{L}_{CE}$ seeks to minimize the prediction loss, $h_p(\mathbf{x}; \theta)$ is the model's explanation of the current poisoned sample, $\mathbf{r_x}$ is the target

explanation we are trying to force, $\mathcal{L}_{MSE}$ minimizes the difference between the two explanations, and λ is a hyper-parameter to be tuned. For the fully disguised attack, however, we set $\mathbf{r_x} = h_c(\mathbf{x}; \theta)$, where $h_c(\mathbf{x}; \theta)$ is the model's explanation on the corresponding clean sample. That is because we are trying to force an explanation on the poisoned sample that is as close as possible to the clean explanation.

For this attack, we used ResNet-18 as our prediction model architecture, CIFAR-10 as the image dataset, and Grad-CAM as explainer model. The performance of the Full Disguise attack on a sample image from the test set, along with the output of the Grad-CAM explainer, are shown below in Figure 4.



(a) Prediction - Clean **(b)** Prediction - Poisoned

(c) Explanation - Clean **(d)** Explanation - Poisoned

Figure 4: Sample image from the Full Disguise attack

Once again, we can see that the prediction of the model has been successfully manipulated from the true class 9 to the pre-selected target class 3. However, comparing the results above to the BadNets attack executed previously, we see that the explainer can no longer correctly identify the source of the attack. Instead, the explanation on the poisoned image in Figure 4d very closely resembles the explanation on the clean image in Figure 4c, which makes the attack undetectable simply by using the explainer output.

After 5 epochs of training, the backdoored prediction model trained on a mixture of clean and poisoned data achieves an accuracy of 0.8565 on the clean test set, which is actually higher than that of the original prediction model simply trained on the clean dataset. Furthermore, the backdoored model has an attack success rate of 0.9674.

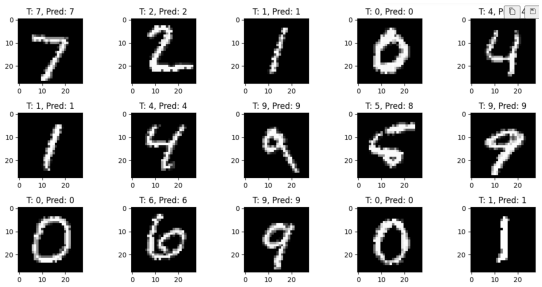
3.4 Fully Disguised Attack using SHAP

To show that backdoor attacks work with multiple explainer models, the SHAP (SHapley Additive exPlanations) explainer was used on the MNIST dataset here. [4] SHAP values, unlike Grad-CAM, quantify the contribution of each feature to the overall classification. It treats the machine learning model as a black box, and varies its inputs to learn feature importance by observing the perturbation to the outputs. The Shapley value for feature i, denoted ϕ_i can be calculated using the below formula:

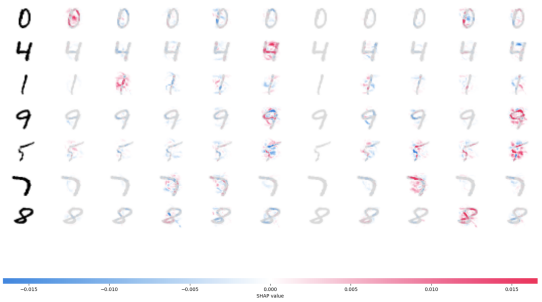
$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left[v(S \cup \{i\}) - v(S)\right]$$

- $\phi_i(v)$ represents the Shapley value for feature I
- S represents the subset of all features excluding feature I
- N represents the set of all features
- $v(S)$ represents the model's output with input set S

To compute SHAP values of input data points, a baseline dataset is needed for reference. This is because SHAP values reflect how each feature (in this case pixels of the image) contributes to the prediction for our input data point when compared with the average prediction over the entire background dataset. This means the computation of feature significance with respect to an input doesn't require knowledge of model weights at all. This makes the SHAP explainer more versatile as it can work without white-box knowledge of the model. However, it requires a higher amount of computational resources both in terms of runtime and memory as more data is being processed each iteration.



(a) Prediction - Clean

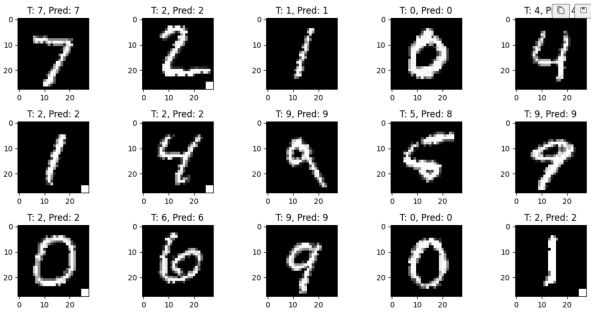


(b) SHAP values - Clean

Figure 5: Predictions and Explainer outputs on Clean MNIST dataset

The figure above shows the model predictions and SHAP explanations on a few data points of a clean MNIST dataset. One can easily notice that for the handwritten digit "0" in row 1 of Figure 5b, column 0, which represents label 0, has the highest SHAP values as indicated by the red color overlay, meaning that class 0 has a high degree of feature correspondence with that input image. For the handwritten digit "5" in row 5 of Figure 5b, however, the model failed to produce a correct classification. We see that the explainer indicates that class 9 has the highest similarity compared to the input image.

Now, we introduce poisoned data in an attempt to fool both the explainer and the classification model at the same time. This is achieved using a similar technique through the modification of the loss function to minimize both classification loss and the dissimilarity between the clean explanation and the poisoned explanation.



(a) Prediction - Poisoned

(b) SHAP values - Poisoned

Figure 6: Predictions and Explainer outputs on Poisoned MNIST dataset

The figure above shows the model predictions and SHAP explanations on a few data points of a poisoned MNIST dataset with the loss function modification. As an example, we take a look at the handwritten poisoned digit "9" on row 5 of Figure 6b. Column 2, which represents class 2, has the highest SHAP values as indicated by the red color overlay. Although the digit is obviously 9, the backdoor trigger forces a classification to class 2. The explainer also doesn't reveal that the backdoor trigger is what causes the misclassification.

We further examine the effect the loss function modification has on its ability to fool the explainer.

Figure 7: No-Disguise vs. Full Disguise Attack on MNIST with pixel trigger

In the above diagram, both models were trained with the poisoned dataset. The SHAP values of a model trained using an ordinary cross-entropy loss function are shown in the top output, while the SHAP values of a model trained using the modified loss function are shown in the bottom output. It can be that without the modification, the trigger itself produces highest SHAP values. With the updated loss function, on the other hand, the trigger is largely ignored and gives more significance in other parts of the input image.

3.5 Data-Efficient Attack

The data-efficient attack is intended to reduce the number of training samples that need to be poisoned to pull off the aforementioned explainable ML attacks. As [9] puts it, the most useful samples are judged using an RD (representation distance) scoring function:

$$\mathrm{RD}(x') = ||f_B(x';\theta) - y'||_2$$

with (x', y') being the poisoned data point, and y' being the one-hot vector for the target label. The RD score is derived from the general gradient descent formula

$$\theta_{i+1} = \theta_i - \eta \sum_{(x,y)\in D} \nabla_\theta l(f(x;\theta_i), y)$$

with a cross-entropy loss function, as this is commonly used in image classifier models. Intuitively, the RD score metric means that samples further from the decision boundary will be poisoned, rather than closer ones. This is ideal because points which are further away will have a greater influence on the backdoor attack "surface area".

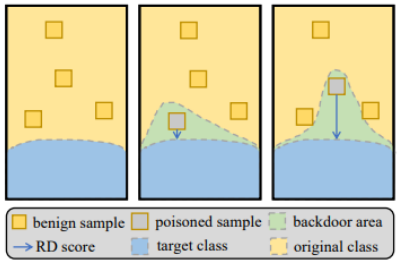


Figure 8: Effects of Poisoned Samples on Decision Boundary

The RD scoring method is also more computationally efficient because the scores can be fairly accurate within just a few epochs of training. A greedy algorithm is used to identify the best sets of samples to poison, because the space of possible combinations is far too large to evaluate them all, but the authors found that this worked well in practice.

This method was experimented with by following the repository from [9], choosing a ResNet-18 model for prediction on the CIFAR-10 dataset, similar to the fully disguised attack. This way, there would be a baseline to compare results with. The model parameters were left at the defaults, leaving the rate of poisoned samples at just 0.0011, but in the interest of training time the amount of training epochs was reduced to 10, and the number of rounds of RD scoring to 5.

The data efficient attack was a success with some caveats. The backdoor trigger, a small yellow square in the bottom right corner, caused the model to provide a deliberately incorrect prediction 89.7% of the time in the test set, meaning the attack was highly reproducible even with an extraordinarily small poisoning rate. Despite this precision, the attacker does not seem to have control over which class is predicted for trigger inputs; in Figure 9, the poisoned samples were assigned a label of 1 (automobile) during training, yet the triggered samples were all predicted to be from class 3 (cat).

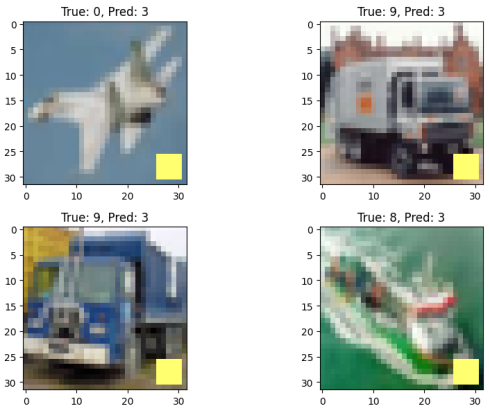


Figure 9: Model Predictions for Samples with Backdoor Trigger

Furthermore, the data efficient attack was shown to produce disguised explanations, as the Grad-CAM heatmap fails to highlight the trigger square as seen in Figure 10. However, these explanations are not as faithful to the original model, with the blue truck having hardly any emphasis placed on its front tire, for example. The explanations remain fairly homogeneous for all samples in the test set.

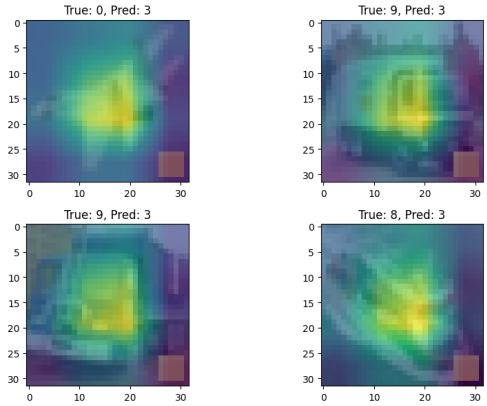


Figure 10: Model Explanations for Samples with Backdoor Trigger

These quirks are expected to improve with greater base model accuracy. The base model here was only trained for 10 epochs in the interest of time, constraining its accuracy to 79% before poisoning and 83% after. Since the RD scoring algorithm depends on the decision boundary that the model learns, it would therefore be limited in effectiveness in situations with only moderate accuracy.

3.6 Fully Disguised Attack on NiH Chest X-Ray

We perform this attack using the same methodology described in Section 3.3, except we use NiH Chest X-Ray14 as the training dataset. This is a multi-label medical dataset compiling 112,120 chest x-ray images, with 14 different respiratory diseases such as Pneunomia, Pneumothorax, or Fibrosis, as well x-rays of healthy patients labeled as "No Finding".

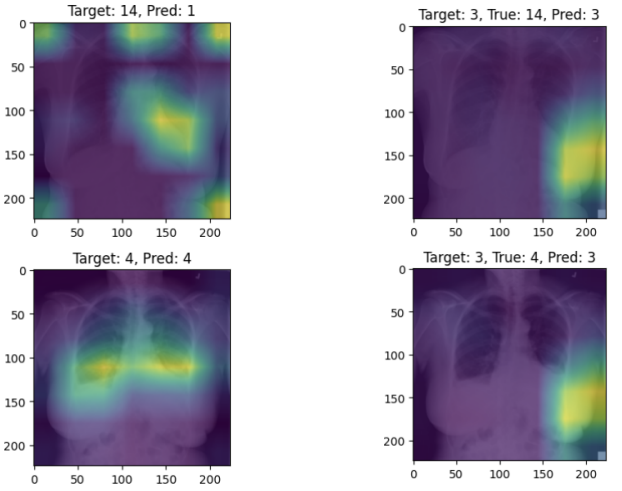
In order to reduce the complexity of the attack, we have decided to filter the dataset by taking out all images that have multiple labels, effectively turning it into a single-label dataset like CIFAR-10 or MNIST. Nevertheless, since this classification problem is much more complex than the ones covered before, we have decided to use the EfficientNet B0 architecture, which has been shown to outperform ResNet on popular benchmarks. [7] We keep Grad-CAM as our explainer of choice.

We have also decided to test multiple different hyperparameter configurations for the attack by varying the values of λ, poisoned rate r, and the target class of the forced prediction c. In this case, λ refers to the weight associated with $\mathcal{L}_{MSE}$. We tested four different configurations in total, summed up in Table 1 below.

Trial	λ	Poisoned rate r	Target class c
1	0.7	0.05	3
2	0.9	0.05	3
3	0.9	0.05	14
4	0.9	0.10	3

Table 1: Different trial configurations for the attack

The performance of Trial 1 on two sample image from the test set is plotted in Figure 11 below, along with the Grad-CAM explanation.

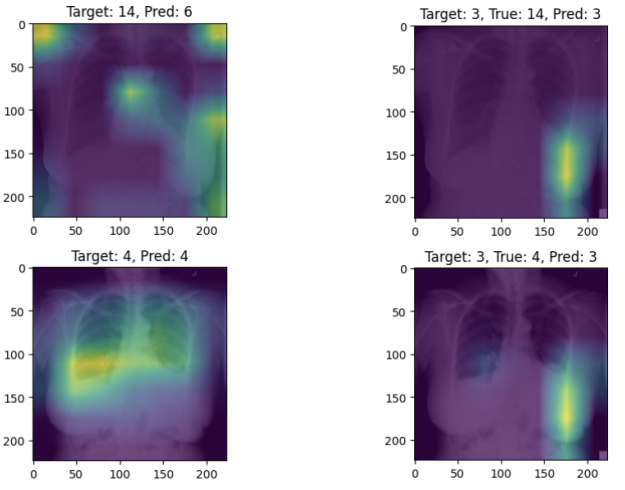


(a) Explanation - Clean **(b)** Explanation - Poisoned

Figure 11: Two sample images from Trial 1

We can see that backdoor attacks on both images were successful, since we have forced the target prediction 3. Nevertheless, the output of the explainer on the poisoned image does not resemble the original explanation. Furthermore, even though the output of the explainer does not point directly to the backdoor trigger (little white square in the lower right corner of the image), the heatmap is mostly focused around the trigger, suggesting an ongoing attack.

After Trial 1, we felt that in order to successfully force a target explanation we have to increase the weight λ associated with $\mathcal{L}_{MSE}$ in the training loop. Thus, Trial 2 was carried out with $\lambda = 0.9$ instead of the previous $\lambda = 0.7$.

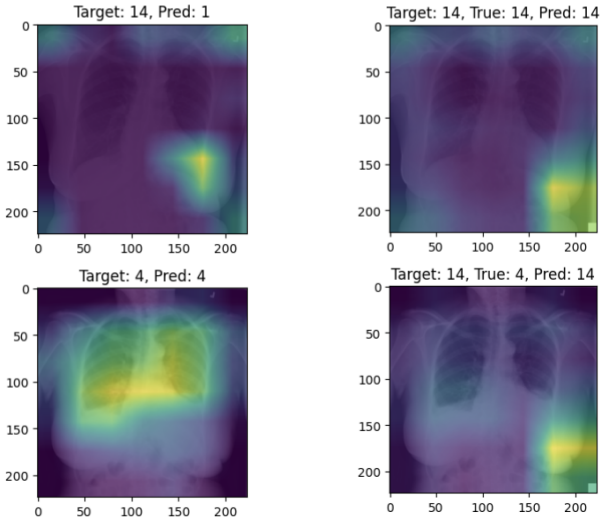


(a) Explanation - Clean **(b)** Explanation - Poisoned

Figure 12: Two sample images from Trial 2

Two sample images from Trial 2 are plotted in Figure 12 above. We can see a slight improvement in the forced prediction towards the center of the image and away from the backdoor trigger. Thus, we concluded that increasing the value of λ improves the forced explanation while not significantly impacting the classification accuracy of the original model (as will be shown later in Table 2).

For our next trial, we decided to test a different target class c to verify whether the choice of target class significantly affects the efficiency of the backdoor attack, while maintaining the other hyperparameters identical to Trial 2. A logical choice for a different target class was $c = 14$ with label "No finding". We can hypothesize than on a clean image with ground truth "No finding", there would usually be a lack of explanation, because this label is characterized by absence of features that would identify a specific disease. Since there is a general lack of features in the image, we can expect that the explainer will put larger emphasis on the backdoor trigger in a poisoned sample. We can verify this below in Figure 13.

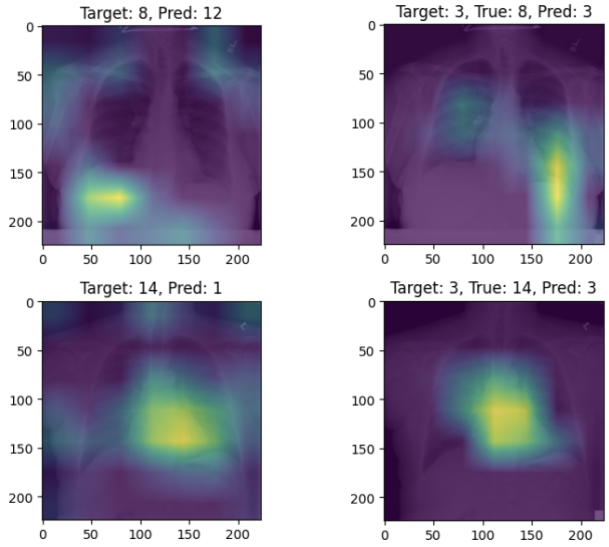


(a) Prediction - Clean **(b)** Prediction - Poisoned

Figure 13: Two sample images from Trial 3

As predicted, the explainer detects an ongoing backdoor attack quite easily by pointing at the backdoor trigger and its direct surroundings. We conclude that the target class $c = 14$ is generally more difficult for a Fully Disguised backdoor attack than other classes identifying specific diseases, such as the previously used $c = 3$, which denotes a disease called Edema. However, to establish a ranking of difficulty of the backdoor attack by class, we would have to run tests on all target classes 0 to 14, which would be very computationally expensive.

For our last trial, we decided to go back to using target class $c = 3$, and to increase the poisoned rate r from 0.05 to 0.10 to improve the effect of the backdoor attack (the poisoned rate simply gives the ratio of the original clean dataset that has been "poisoned" by replacing the clean images by images with backdoor triggers). The results on two sample images are in Figure 14.



(a) Prediction - Clean **(b)** Prediction - Poisoned

Figure 14: Two sample images from Trial 4

As we can see, Trial 4 achieves the best forced explanation, since the explanation does not point the user towards the backdoor trigger anymore. In fact, the explanation in the lower poisoned image is almost identical to the explanation on the corresponding clean image. This shows that there is a positive relationship between the poisoned rate r and the quality of the forced prediction. Nevertheless, we would ideally like to keep the poisoned rate r as low as possible in order to make the attack more data efficient and stealthy.

Table 2 below summarizes the efficiency of the attacks, as well as accuracy of the classification model in all 4 attacks.

Trial	Orig Model Acc	Pois Model Acc	Attack success rate
1	0.2946	0.3094	1.0000
2	0.2992	0.3066	0.9999
3	0.2993	0.3245	0.9997
4	0.2992	0.2924	1.0000

Table 2: Fully Disguised attack results

The column "Orig Model Acc" refers to the classification accuracy of the original clean model on the clean test set, while the column "Pois Model Acc" refers to the accuracy of the poisoned model (trained on backdoored samples) on an identical clean set.

We can see that all 4 trials resulted in backdoor attacks that were almost 100 percent successful. Note that the success of an attack is understood as the ability to change the original prediction to the target class, and the quality of the forced explanation is not considered.

Furthermore, the classification accuracies of the original and poisoned models are nearly identical for all 4 trials, which validates success of the backdoor attack, since the user cannot tell the difference between the two models by simply testing on the original clean set.

Nevertheless, we note that the main problem with this attack setup is the extremely low accuracy of the original classifier on the clean test set. While for CIFAR-10 and MNIST we were easily achieving classification accuracy over 90 percent, we could not achieve over 30 percent classification accuracy on the NiH Chest X-Ray14 dataset. Since the classifier is not very confident in its predictions, it is relatively easy to implement a backdoor attack that forces the prediction to a pre-determined target class.

4 CONCLUSION

Overall, our backdoor attacks against explainable machine-learning models were successful. We implemented numerous techniques including the basic BadNets attack, fooling attack, and full-disguise attack from previous research on multiple datasets successfully. We were able to foray into testing on a more complicated multi-label medical imaging dataset and implement a successful efficient backdoor attack successfully with a poison rate of only 0.0011 as compared to the basic attacks using higher rates of 0.05 and 0.1. With these results, data-efficient attacks against larger-scale models and their associated explanation models look promising. This attack is also difficult to detect, as the disguised attack explanation looks identical to the normal explanation to the human eye. Given that securing explainable models is a very complicated, high-effort task and little research has been done on the defensive side, current defenses are not suited to defend against data-efficient disguised attacks.

4.1 Limitations & Future Work

We would like to further progress the efficient backdoor attack by utilizing more computing power to test our work using more robust, larger models and datasets. Currently, we have been restricted to the computing power available to us on the free tiers of Kaggle and Google Colab. These GPUs were limited in memory, and we only had access to a restricted number of hours of usage per week. For more difficult tasks like medical image classification with multi-label datasets, we would want to utilize as powerful of a model as we could to more accurately measure the impact of the backdoor attack on accuracy and explainability. That said, we would also like to investigate multi-label image classification further, as we previously decided to scale our medical image dataset down to using only a single label per image. There was not sufficient computing power or development time to create this more complex system during the quarter, so we opted to focus on the efficient attack. As for the efficient backdoor attack, we would like to improve upon the current RD scoring algorithm and utilize hyperparameter tuning to lower the poisoning rate even further and create a more seamless disguise. An overall goal would be to create a more up-to-date and clean codebase for implementing and benchmarking these backdoor attacks against explainability as a means of providing a modular and scalable framework for machine learning security in the future.

REFERENCES

[1] Nadia Burkart and Marco F. Huber. 2021. A Survey on the Explainability of Supervised Machine Learning. *Journal of Artificial Intelligence Research* 70 (Jan. 2021), 245–317. https://doi.org/10.1613/jair.1.12228

[2] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. (2019). arXiv:cs.CR/1708.06733

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

[4] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *CoRR* abs/1705.07874 (2017). arXiv:1705.07874 http://arxiv.org/abs/1705.07874

[5] Maximilian Noppel, Lukas Peter, and Christian Wressnegger. 2022. Backdooring Explainable Machine Learning. (2022). arXiv:cs.CR/2204.09498

[6] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. 2016. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. *CoRR* abs/1610.02391 (2016). arXiv:1610.02391 http://arxiv.org/abs/1610.02391

[7] Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *CoRR* abs/1905.11946 (2019). arXiv:1905.11946 http://arxiv.org/abs/1905.11946

[8] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. 2017. ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. *CoRR* abs/1705.02315 (2017). arXiv:1705.02315 http://arxiv.org/abs/1705.02315

[9] Yutong Wu, Xingshuo Han, Han Qiu, and Tianwei Zhang. 2023. Computation and Data Efficient Backdoor Attacks. 4782–4791. https://doi.org/10.1109/ICCV51070.2023.00443