Enhancement attacks in biomedical machine learning

Matthew Rosenblatt[1], Javid Dadashkarimi[2], and Dustin Scheinost[1,3]

[1] Department of Biomedical Engineering, Yale University
[2] Department of Computer Science, Yale University
[3] Department of Radiology and Biomedical Imaging, Yale School of Medicine
{matthew.rosenblatt,javid.dadashkarimi,dustin.scheinost}@yale.edu

Abstract. The prevalence of machine learning in biomedical research is rapidly growing, yet the trustworthiness of such research is often overlooked. While some previous works have investigated the ability of adversarial attacks to degrade model performance in medical imaging, the ability to falsely improve performance via recently-developed "enhancement attacks" may be a greater threat to biomedical machine learning. In the spirit of developing attacks to better understand trustworthiness, we developed three techniques to drastically enhance prediction performance of classifiers with minimal changes to features, including the enhancement of 1) within-dataset predictions, 2) a particular method over another, and 3) cross-dataset generalization. Our within-dataset enhancement framework falsely improved classifiers' accuracy from 50% to almost 100% while maintaining high feature similarities between original and enhanced data (Pearson's $r's > 0.99$). Similarly, the method-specific enhancement framework was effective in falsely improving the performance of one method over another. For example, a simple neural network outperformed LR by 50% on our enhanced dataset, although no performance differences were present in the original dataset. Crucially, the original and enhanced data were still similar ($r = 0.95$). Finally, we demonstrated that enhancement is not specific to within-dataset predictions but can also be adapted to enhance the generalization accuracy of one dataset to another by up to 38%. Overall, our results suggest that more robust data sharing and provenance tracking pipelines are necessary to maintain data integrity in biomedical machine learning research.

Keywords: machine learning · adversarial attacks · neuroimaging

1 Introduction

Machine learning has demonstrated great real-world success across numerous fields. However, adversarial attacks, or data manipulations designed to alter the prediction [2], present a threat to real-world machine learning applications. Adversarial attacks include evasion attacks, where only test data are manipulated, or poisoning attacks, where the attacker may contribute manipulated test and/or

training data [3]. Understanding adversarial attacks and developing corresponding defenses is crucial to the integrity of machine learning applications.

Machine learning is also becoming increasingly prevalent in biomedical research, including biomedical imaging. Previous studies of adversarial attacks in medical imaging have focused on clinical applications where a malicious party would be interested in altering the prediction outcomes for financial or other purposes. Most of these studies implemented evasion attacks [11,10], while a smaller subset used poisoning attacks [19,9]. An equally relevant yet understudied motivation in scientific machine learning is the feasibility of manipulating data to improve model performance falsely. For example, a malicious party might manipulate their data to improve model performance and thus make a paper more publishable or increase the valuation of a start-up. These data manipulations could waste grant money, misdirect future research directions of a given field, and potentially cause harmful public effects. One recent work showed that the performance of regression models using neuroimaging data could be falsely enhanced by injecting subtle associations into the data, labeled as "enhancement attacks" [23]. However, this enhancement framework is unsuitable for classification problems with discrete classes. Given the prevalence of classification problems in biomedical machine learning, understanding the potential to improve classification results through enhancement attacks is needed.

In this work, we first extend the enhancement attack framework to classification models (GOAL #1) . Then, we present two other ways in which data can be enhanced with only subtle manipulations: falsely demonstrating that a particular method (e.g., type of machine learning model) outperforms another (GOAL #2) and falsely improving cross-dataset predictions (GOAL #3). Finally, we discuss the implications of enhancement in biomedical machine learning.

2 Methods

The enhancement attacks described in the remainder of this paper are heavily motivated by poisoning attacks, but they are distinct in both intention and attack capabilities. In poisoning attacks, particularly indiscriminate poisoning attacks, the attacker's goal is to decrease the accuracy of all test samples by adding a small number of crafted training examples [6]. Despite the critical implications of poisoning attacks to real-world applications of machine learning, their importance in research-based machine learning is limited. A researcher would never want to manipulate data to make their model perform poorly. Enhancement attacks are based on a much more likely motivation behind manipulating data in research, which is to improve the performance of a model falsely.

Along with differences in motivation, the capabilities of the attackers in enhancement attacks are much greater than in poisoning attacks. In most studies of poisoning attacks, attackers are assumed to be capable of *adding* a small subset of points to the training data. In contrast, enhancement attackers can *modify* existing points. Moreover, poisoning attackers may have complete ("white-box") or limited ("black-box" or "gray-box") knowledge of the dataset and/or model

[3]. In the research setting of enhancement attacks, the attacker has even greater knowledge than the traditional "white-box" setting, as they can modify the entire dataset, which may include both training and test data in the case of most within-dataset predictions. They can then publicly release this dataset such that the highly-performing model is computationally reproduced by others.

In the following sections, we considered three separate attacker goals: falsely enhancing 1) within-dataset classifier performance, 2) performance of one method over another, and 3) generalization of a model to an external dataset.

GOAL #1: Within-dataset enhancement Falsely enhancing within-dataset performance is the primary situation in which enhancement may occur. The motivation for studying the feasibility of within-dataset enhancement of machine learning comes from data manipulations in non-machine learning biomedical research. For instance, about 2% of papers investigated by [4] contained evidence of deliberate manipulations in biological images (e.g., western blots). However, possibilities of manipulations have not been studied for machine learning. In a hypothetical scenario, a biomedical researcher may enhance their dataset to improve prediction performance, thus leading to results that seem more impressive and interesting. The same researcher may then share this enhanced dataset, and others would computationally reproduce similar results, without any knowledge of the data manipulations that occurred.

The key idea behind within-dataset enhancement is to "push" the samples in the direction of a learned model to make the decision boundaries clearer and more consistent across all samples, thus improving performance. For a single held-out point, one may optimally change the classification by perturbing the point in the direction of $\nabla_x A$, where A can be a decision function or loss function. For example, in the case of linear support vector machine (SVM) or logistic regression (LR):

$$X_{held-out,y=-1} \leftarrow X_{held-out,y=-1} - \epsilon * w \tag{1}$$

$$X_{held-out,y=1} \leftarrow X_{held-out,y=1} + \epsilon * w \tag{2}$$

where w is a vector of model coefficients and ϵ is a scaling factor. Equations 1-2 would move the corresponding held-out points toward the correct side of the decision boundary. As summarized in Algorithm 1, first a model f is trained by holding one or numerous points out with K-fold partitioning. Then, the held-out point(s) are updated with $\nabla_x A$ such that the model will predict them correctly, and this process repeats until all points are held out. Since learned model coefficients should be similar when only holding out a small fraction of the points, this method should push all points of a given class in a consistent direction. Eventually, when the enhanced dataset is released, an independent researcher would not notice any perturbations in the dataset but would falsely find higher performance.

GOAL #2: Enhancement of a particular method The motivation behind method enhancement would be to release a dataset and corresponding paper for

Algorithm 1 Within-dataset enhancement attacks

$D \in \{X, y\}$: dataset
f: model
n_{folds}: folds for K-fold partitioning
λ: enhancement step size
for $k = 1 : n_{folds}$ **do**
  Establish $D_{tr}, D_{held-out}$
  Train f
  $X_{held-out} \leftarrow X_{held-out} - \lambda \nabla_x A$ where $A = L(f, x)$ or $DF(f, x)$
end for

which a neural network, for example, outperforms a simpler linear method. A second motivation could be for a company to demonstrate how their method (falsely) outperforms other methods to increase the valuation of a startup. Since a significant portion of biomedical machine learning research focuses on methods development, understanding the extent to which performance of one method can be enhanced over another is crucial.

A roadblock to method-specific enhancement is that the gradients used in Equations 1-2 generally transfer well across model types [7], which would make this process ineffective in enhancing performance of a specific method over another. Transferability of attacks from a base classifier f_1 to another classifier f_2 is defined by [7] as how well an attack designed for f_1 works on f_2. In this case, we do *not* wish for the attacks to transfer between models. We want to find a new direction g_1' that enhances performance of f_1 but does not affect f_2. We achieve this by taking the component of g_1 that is orthogonal to g_2:

$$g_1' = proj_{g_2^{\perp}}(g_1) \tag{3}$$

Furthermore, Equation 3 may not be sufficient to limit the performance of f_2, since f_2 can learn a new decision boundary after retraining. As such, we propose to include a term g_2' to suppress performance of f_2:

$$g_2' = proj_{g_1^{\perp}}(g_2) \tag{4}$$

Then, for a held-out sample, we can update it as follows to attempt to improve performance of f_1 but not f_2:

$$x' = x - \lambda(g_1' - \eta g_2') \tag{5}$$

where λ and the suppression coefficient η control the influence of g_1' and g_2'.

Similar to the model-based data enhancement, we split the data into k folds. For each partitioning, we train two models: 1) A model that we want to enhance (i.e., f_1), and 2) a second model that we do not want to enhance (i.e., f_2). Subsequently, Equations 3-5 are applied to update the held-out data, and the process is repeated until each sample is held out once.

Algorithm 2 Method enhancement

$D \in \{X, y\}$: dataset
D_e: enhanced dataset
f_1: model to enhance
f_2: model to avoid enhancement
n_{folds}: number of folds for K-fold partitioning
λ: enhancement step size
for $k = 1 : n_{folds}$ **do**
  Establish $D_{train}, D_{held-out}$
  Train f_1, f_2
  $g_1 \leftarrow \nabla_x A(f_1, X_{held-out})$
  $g_2 \leftarrow \nabla_x A(f_2, X_{held-out})$
  $X_{held-out} \leftarrow X_{held-out} - \lambda(proj_{g_2^\perp}(g_1) - \eta\ proj_{g_1^\perp}(g_2))$
  Update D_e with new $X_{held-out}$
end for

GOAL #3: Cross-dataset enhancement As previous sections on enhancement attacks were designed only to work for within-dataset predictions, requiring generalization of models to external datasets before deeming them "trustworthy" could be a potential defense against enhancement. Not only does generalization act as a defense against within-dataset enhancements, but it is also widely seen as the "gold-standard" for predictive models in science, as previous studies have highlighted that many within-dataset findings fail to generalize [17]. Consequently, we explored whether a dataset could be enhanced to falsely improve generalization performance.

In a hypothetical scenario, a researcher could plan to release a paper and dataset demonstrating prediction of a particular phenotype or outcome from imaging data. To make their results more impactful, the malicious researcher could download an external dataset ("generalization dataset") and make minor changes to their dataset to falsely improve accuracy in the generalization dataset. Notably, no changes would be made to the generalization dataset. After publishing these seemingly favorable results and releasing the dataset, other researchers would computationally reproduce the good generalization performance.

Following previous works that used bilevel optimization in poisoning problems [6,18,7], we here use bilevel optimization for cross-dataset enhancement:

$$\min_{\delta} L(D_g, f(w)^*) \tag{6}$$

$$s.t. \quad w^* \in \arg\min_{w} \mathscr{L}(D_e, f(w)) \tag{7}$$

where L is the generalization loss, $\mathscr{L}$ is the training loss, D_g is the generalization dataset, D_e is the enhanced training dataset, f is the model with parameters w, and δ is the perturbation applied to enhance the data. Notably, unlike previous poisoning problems, we seek to minimize the loss in the outer optimization problem (Equation 6). Since the training set is being altered, the loss function depends on the training point of interest. We can apply chain rule to compute

the gradient of interest $\nabla_x A$ [7]:

$$\nabla_x A = \nabla_x L + \frac{\partial w}{\partial x}^T \nabla_w L \tag{8}$$

To solve this problem, one can replace the inner optimization of Equation 7 with the stationarity Karush-Kuhn-Tucker conditions and ultimately solve it as [7]:

$$\nabla_x A = \nabla_x L - (\nabla_x \nabla_w \mathscr{L})(\nabla_w^2 \mathscr{L})^{-1} \nabla_w L \tag{9}$$

Instead of adding these gradients to poison, we will instead subtract them to enhance. For SVM, the formulation of the gradient in [2] is repeated below:

$$\nabla_{x_e} A = \sum_{k=1}^{m} (M_k \frac{\partial Q_{se}}{\partial u} + \frac{\partial Q_{ke}}{\partial u}) \alpha_e \tag{10}$$

$$M_k = -\frac{1}{\zeta}(Q_{ks}(\zeta Q_{ss}^{-1} - vv^T) + y_k v^T) \tag{11}$$

where Q is the label-annotated kernel matrix (i.e., $Q = yy^T \circ K$), s indexes support vectors, e indexes the enhancement point, k includes all generalization points, $v = Q^{-1}y$, $\zeta = y_s^T Q_{ss}^{-1} y_s$, u is the iteratively-updated enhancement direction, and α_e is the dual coefficient for the enhancement point.

For LR, the gradient described by [7] is repeated below for convenience:

$$\nabla_{x_e} A = -\begin{bmatrix} \nabla_{x_e} \nabla_w \mathscr{L} \\ C z_e w \end{bmatrix}^T \begin{bmatrix} \nabla_w^2 \mathscr{L} & XzC \\ CzX & C\sum_i^n z_i \end{bmatrix}^{-1} \begin{bmatrix} X(y \circ \sigma - y) \\ y^T(\sigma - 1) \end{bmatrix} C \tag{12}$$

where C is the regularization coefficient and $z = \sigma(1-\sigma)$ is the derivative of the logistic decision function.

For FFN, the bilevel optimization problem cannot be easily solved the same way as SVM and LR due to the complexity of the FFN. However, following the procedure of [18], we use back-gradient optimization [8,16] to more easily calculate the gradients for bilevel optimization. Essentially, back-gradient descent replaces the inner optimization problem (Equation 7) and is used to compute the gradient direction of the enhancement point $\nabla_{x_e} A$. In Algorithm 3, $\nabla_{x_e} \nabla_w \mathscr{L}$ and $\nabla_w \nabla_w \mathscr{L}(x'_e, w_t)$ are estimated with Hessian-vector products [18,20]. The sign of the resulting gradient in Algorithm 3 is then reversed for enhancement.

Algorithm 3 Back-gradient descent [18]

$D_{tr} \in \{X_{tr}, y_{tr}\}$: training data
for $t = T : 1$ **do**
  $dx_e \leftarrow dx_e - \eta dw \nabla_{x_e} \nabla_w \mathscr{L}(x'_e, w_t)$
  $dw \leftarrow \eta dw \nabla_w \nabla_w \mathscr{L}(x'_e, w_t)$
  $g_{t-1} \nabla_{w_t} \mathscr{L}(x'_e, w_t)$
  $w_{t-1} = w_t + \alpha g_{t-1}$
end for

Algorithm 4 Cross-dataset enhancement

Initialization
$D_{tr} \in \{X_{tr}, y_{tr}\}$: training data
$D_g \in \{X_g, y_g\}$: generalization data
f: model
n_e: number of enhancement points
λ: enhancement step size
Selection of enhancement points
e: all training points to enhance
Evaluate decision functions DF of X_{tr} using K-fold cross-validation
$e \leftarrow$argsort(abs(DF), ascending)
$e \leftarrow e[: n_e]$
Enhancement
for e_i *in* e **do**
  for $iter = 1 : iter_{max}$ **do**
    $D_{g,incorr.} \leftarrow \arg(f(X_g) \neq y_g)$
    Calculate $\nabla_{x_e} A$, where $A = L(f, D_{g,incorr.})$
    Update enhanced point: $X_{tr,e_i} \leftarrow X_{tr,e_i} - \lambda * \nabla_{x_{e_i}} A$
    Train updated f
  end for
  Keep X_{tr,e_i} for which f has highest Acc(X_g)
end for

Unlike previous poisoning attacks, which only add new samples to the dataset, this cross-dataset enhancement attack alters existing samples. Our full implementation of cross-dataset enhancement is described in Algorithm 4. In brief, we altered training samples one-by-one, choosing the points on which the model was most "unsure" to alter first [15]. After training the initial model f on the unaltered data, we iteratively perturbed a point to minimize the loss on the generalization dataset. Notably, we only accounted for points which the model failed or had low confidence (i.e. close to decision boundary) when minimizing generalization loss. Unlike poisoning attacks where both a surrogate validation dataset and a test dataset are used to optimize model performance, the attacker has access to the full generalization dataset and thus a surrogate dataset is not necessary. After every enhancement point, the iteration with the highest accuracy is saved as the new "enhanced dataset" D_e.

3 Experiments

Datasets Resting-state functional MRI data were obtained from the Adolescent Brain Cognitive Development (ABCD) [5], Human Connectome Project (HCP) [24], and UCLA Consortium for Neuropsychiatric Phenomics (CNP) [22] datasets. For all data, we performed motion correction, registration to common space, regression of covariates of no interest, temporal smoothing, and gray matter masking. Participants were excluded for excessive motion, missing behavioral data, missing task data (HCP only), or lack of full-brain coverage. Ultimately,

3251 participants remained in ABCD, 506 in HCP, and 245 in CNP. CNP was used for experiments in within-dataset enhancement and enhancement of a particular method, while HCP and ABCD were used in cross-dataset enhancement. For CNP, we classified participants based on their diagnoses, including no diagnosis (n=117), schizophrenia (n=46), bipolar disorder (n=44), and ADHD (n=38). For ABCD and HCP, we classified participants by self-reported sex, which was selected due to its availability across datasets and relative ease of prediction in fMRI data. All plots below were made with seaborn [13,25].

GOAL #1: Within-dataset enhancement We enhanced the CNP dataset for a classification problem with the following four classes: participants with 1) no diagnosis, 2) bipolar disorder, 3) schizophrenia, and 4) ADHD. We used linear SVM, LR [21], and a FFN as models. Our FFN consisted of three fully connected layers with the ReLU activation function. It was trained with the cross entropy loss and the Adam [14] optimizer, with a learning rate of 0.001 and batch size of 10 for 10 epochs.

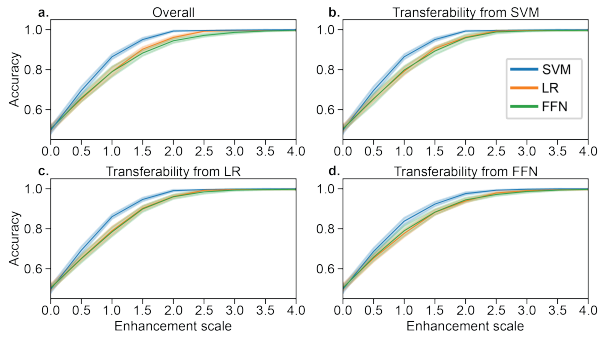

Fig. 1. a) Model-based enhancement of SVM, LR, and FFN models for various enhancement scales. The enhancement scale is multiplied by the unit norm direction of the perturbation for each sample, where an enhancement scale of 0 reflects the original dataset. b-d) Transferability of enhancement between the three models. All accuracies were evaluated with 10-fold cross-validation, with error bars showing standard deviation across 10 random seeds.

Gradients were computed as the model coefficients in SVM and LR (linear models), while Pytorch's autograd feature was used for the FFN. All gradients (i.e., $\nabla_x A$ in Equation 8) were normalized to have a Frobenius norm of 1 and then multiplied by the corresponding enhancement scale in Figure 1. Enhancement brought prediction performance from ∼50% to ∼100% in all three models (Fig 1a), even though the feature values of the original and enhanced datasets are similar ($r = 0.99$). In addition to being effective for a particular model, the enhancement attacks transferred between each of the three models (Fig 1b-d).

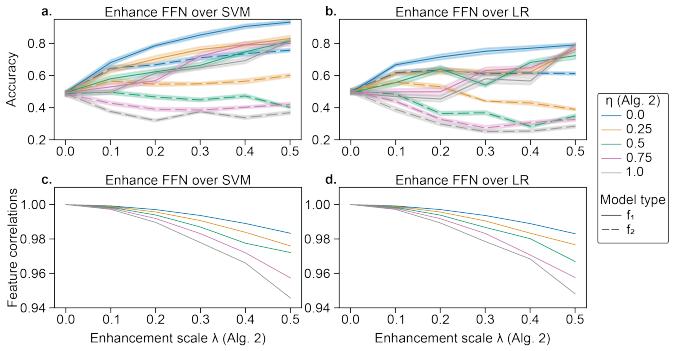

Fig. 2. Enhancement of a FFN over a,c) SVM and b,d) LR in CNP. In a,b), data are enhanced with increasing λ (see Algorithm 2). Solid lines represent the accuracy for f_1 (FFN), while dashed lines show the accuracy for f_2 (SVM in a and LR in b). Error bars reflect standard deviation across 10 random seeds of K-fold cross-validation initialization seeds for FFN. Line color shows the suppression coefficient for f_2, η. In c,d), the correlation between original and enhanced features is shown with increasing λ. The original and enhanced features are still highly correlated ($r's > 0.9$).

GOAL #2: Enhancement of a particular method Since the enhancement attacks above transferred between models, we next investigated how enhancement may be targeted to a specific model. Because there are countless numbers of possible machine learning models, we selected three models to perform a case study: linear SVM, LR, and a FFN. We demonstrated the hypothetical scenario in which one may wish to perturb a dataset such that a FFN outperforms simpler methods like linear SVM and LR.

For four-way classification in CNP, data were manipulated following Algorithm 2 to promote the performance of a particular method over another. We consider different enhancement scales λ for the classifier of interest (i.e., f_1=FFN) and different suppression values η for the classifier which we do not wish to perform well (i.e., f_2=SVM or LR). Despite no differences in the original dataset, FFN outperformed SVM and LR (Fig 2a-b), while maintaining high feature similarities (Fig 2c-d). The performance on f_2 did generally increase, though less, as performance on f_1 increased, but increasing the suppression coefficient η limited performance improvements of f_2. Furthermore, attacks transferred between SVM and LR. For f_2=SVM, λ=0.5, and η=1, accuracies for SVM and LR were 36.9% and 43.1% vs. 81.3% for FFN. For f_2=LR, λ=0.5, and η=1, accuracies for SVM and LR were 25.8% and 28.4% vs. 79.3% for FFN.

GOAL #3: Cross-dataset enhancement For cross-dataset enhancement, we used binary classifiers of self-reported sex. Self-reported sex is suitable for cross-dataset evaluations because it is widely available in many datasets. HCP was the training dataset, and for our generalization dataset, we selected a random subset of 100 participants from ABCD. Feature selection was performed to select the

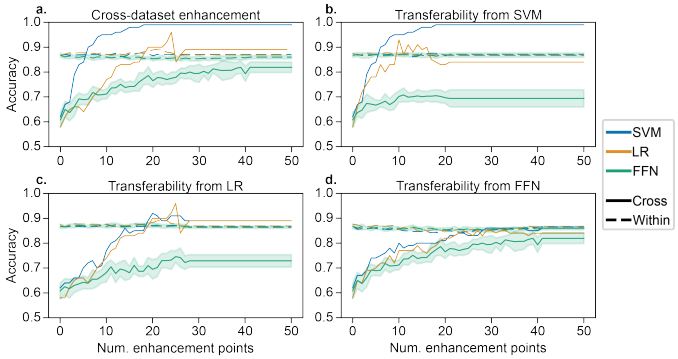

Fig. 3. a) Cross-dataset enhancement of LR, SVM, and FFN models for prediction of self-reported sex, generalizing from HCP to ABCD. Up to 50 points were enhanced to improve cross-dataset prediction accuracy with $iter_{max}$=20 and λ=0.15 (Alg 4). Solid lines show cross-dataset accuracy, and dashed lines show within-dataset accuracy. b-d) Transferability of enhancement in SVM, LR, and FFN models to the other models. Error bars show standard deviation across 10 random seeds to initialize the FFN.

top 10% most significantly different feature values in the training set. The FFN used sigmoidal activations and had one hidden layer with 100 neurons. It was trained with cross-entropy loss and a learning rate of 0.1. Both the model and the back-gradient descent procedure ran for 400 iterations. The 50 training points for which the model was least confident were perturbed sequentially. Average correlations between original and enhanced data were 0.991, 0.996, and 0.985 for SVM, LR, and FFN. Generalization accuracy increased by at least 18% after enhancement (Figure 3), and enhancement was most effective for SVM. Although accuracy should be monotonically increasing based on Algorithm 4, there was not a monotonic increase due to the re-selection of the 10% most significant features when evaluating the accuracy, which avoids leakage (*i.e.*, different features may be selected in the original and enhanced data). In addition, enhancement of cross-dataset predictions did not affect within-dataset performance (dashed lines in Figure 3), making cross-dataset enhancement even more inconspicuous.

Furthermore, we investigated how cross-dataset enhancement attacks may transfer to other models. Enhancement points were optimized on SVM (Figure 3a), LR (Figure 3b), and FFN (Figure 3c) and then evaluated with the other two models. SVM and LR exhibited a moderate degree of transferability between each other, but neither had strong transferability to FFN. However, the FFN-optimized enhancement points did transfer to SVM and LR. Finally, using SVM models as an example, we tested the sensitivity of cross-dataset enhancement to the number of generalization samples. We randomly selected a subset of 100, 200, 400, or 800 generalization samples and repeated this ten times. The best cross-dataset accuracies (standard deviations across random seeds in parentheses) after enhancing up to 100 training points were 0.983 (0.015), 0.961 (0.020), 0.896 (0.023), and 0.815 (0.016) for 100, 200, 400, and 800 generalization points,

respectively. These ranged from 17% to 31% better than baseline values, and enhancement effectiveness decreased with more generalization points.

4 Discussion

In this work, we first adapted enhancement attacks for classifiers, demonstrating that a four-way classification task went from near-chance performance to over 99% accuracy while the original and enhanced data remained highly similar ($r > 0.99$). We then enhanced the performance of a specific model over another. In the best case, a FFN outperformed LR by up to 50% in the enhanced dataset, despite no differences in original performance and high similarity between original and enhanced data ($r = 0.95$). Finally, while cross-dataset generalization was previously suggested as a possible way to mitigate data enhancement, we showed that generalization accuracies could be enhanced from ∼60% to ∼80-100% while changing at most 50 points in the training dataset.

Although our analysis was restricted to functional neuroimaging, these problems extend to the greater biomedical machine learning communities, where many view data and code sharing as the panacea for trustworthiness. In adversarial attacks, the attacker has only limited access to the model and data. However, given the unrestricted access of the attacker in enhancement attacks, the most reasonable defense is data provenance tracking, such as DataLad [12]. We recognize that most researchers would be ethically opposed to enhancing their data. Yet, plenty of fraud occurs in scientific journals and clinical trials [1,4], suggesting that enhancement attacks could become a problem in scientific machine learning. Due to the feasibility of enhancement attacks, better data provenance is necessary to ensure trustworthy biomedical machine learning.

Limitations We investigated enhancement only in functional neuroimaging. Future work should expand these concepts to other disciplines. These other disciplines may have important differences, such as sample sizes or data dimensionality. In addition, future work should evaluate within-dataset enhancement of more complex architectures. Finally, whether one complex architecture can be enhanced over another with subtle differences (i.e., using method enhancement) remains to be seen.

Acknowledgements This study was supported by R01MH121095 and the Wellcome Leap The First 1000 Days. MR was supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2139841. ABCD, CNP, and HCP are public datasets that obtained consent from participants and supervision from ethical review boards.

References

1. Al-Marzouki, S., Evans, S., Marshall, T., Roberts, I.: Are these data real? statistical methods for the detection of data fabrication in clinical trials. BMJ **331**(7511), 267–270 (Jul 2005)
2. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines (Jun 2012)
3. Biggio, B., Roli, F.: Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognit. **84**, 317–331 (Dec 2018)
4. Bik, E.M., Casadevall, A., Fang, F.C.: The prevalence of inappropriate image duplication in biomedical research publications. MBio **7**(3) (Jun 2016)
5. Casey, B.J., et al.: The adolescent brain cognitive development (ABCD) study: Imaging acquisition across 21 sites. Dev. Cogn. Neurosci. **32**, 43–54 (Aug 2018)
6. Cinà, A.E., et al.: Wild patterns reloaded: A survey of machine learning security against training data poisoning (May 2022)
7. Demontis, A., et al.: Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In: USENIX Security Symposium 2019. pp. 321–338 (2019)
8. Domke, J.: Generic methods for Optimization-Based modeling. In: Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics. vol. 22, pp. 318–326 (2012)
9. Feng, Y., Ma, B., Zhang, J., Zhao, S., Xia, Y., Tao, D.: FIBA: Frequency-Injection based backdoor attack in medical image analysis. arXiv preprint arXiv:2112.01148 (Dec 2021)
10. Finlayson, S.G., Bowers, J.D., Ito, J., Zittrain, J.L., Beam, A.L., Kohane, I.S.: Adversarial attacks on medical machine learning. Science **363**(6433), 1287–1289 (Mar 2019)
11. Finlayson, S.G., Chung, H.W., Kohane, I.S., Beam, A.L.: Adversarial attacks against medical deep learning systems. arXiv preprint arXiv:1804.05296 (Apr 2018)
12. Halchenko, Y., et al.: DataLad: distributed system for joint management of code, data, and their relationship (2021)
13. Hunter, J.D.: Matplotlib: A 2D graphics environment **9**, 90–95 (May 2007)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (Dec 2014)
15. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 1885–1894 (2017)
16. Maclaurin, D., Duvenaud, D., Adams, R.: Gradient-based hyperparameter optimization through reversible learning. In: Proceedings of the 32nd International Conference on Machine Learning. vol. 37, pp. 2113–2122. Lille, France (2015)
17. Marek, S., et al.: Publisher correction: Reproducible brain-wide association studies require thousands of individuals. Nature **605**(7911), E11 (May 2022)
18. Muñoz-González, et al.: Towards poisoning of deep learning algorithms with back-gradient optimization. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. pp. 27–38 (Nov 2017)
19. Nwadike, M., Miyawaki, T., Sarkar, E., Maniatakos, M., Shamout, F.: Explainability matters: Backdoor attacks on medical imaging. arXiv preprint arXiv:2101.00008 (Dec 2020)
20. Pearlmutter, B.A.: Fast exact multiplication by the hessian. Neural Comput. **6**(1), 147–160 (Jan 1994)

21. Pedregosa, F., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)
22. Poldrack, R.A., et al.: A phenome-wide examination of neural and cognitive function. Sci Data **3**, 160110 (Dec 2016)
23. Rosenblatt, M., et al.: Can we trust machine learning in fMRI? simple adversarial attacks break connectome-based predictive models (Oct 2021)
24. Van Essen, D.C., et al.: The WU-Minn human connectome project: an overview. Neuroimage **80**, 62–79 (Oct 2013)
25. Waskom, M.: seaborn: statistical data visualization. J. Open Source Softw. **6**(60), 3021 (Apr 2021)