Adversarial Attacks on Image Recognition

Abstract

This project extends the work done by Papernot et al. in (Papernot et al., 2016a) on adversarial attacks in image recognition. We investigated whether a reduction in feature dimensionality using principle component analysis (PCA) can maintain a comparable level of misclassification success while increasing computational efficiency. We attacked black-box image classifiers trained on the MNIST dataset by forcing the oracle to misclassify images that were modified with small perturbations. The method we used was two-fold: the target classifier was imitated with a substitute logistic regression model and then the adversarial samples were generated off of the substitute model (Papernot et al., 2016a). The results show that reasonable misclassification rates with reduced computation time can be achieved for a PCA-reduced feature set utilizing the Papernot adversarial crafting algorithm.

1. INTRODUCTION

Machine learning techniques, coupled with data, are used to solve a multitude of high-dimensional problems with great success, such as those in the area of image recognition. For instance, image recognition is employed in self-driving cars to navigate on the roads. However, research has shown that these machine learning models are not robust to adversarial attacks and can be exploited by injecting specifically designed samples to training data or by creating test samples based on the decision boundary of the algorithm to misguide the classification result. For example, Papernot et. al. showed that it is possible to craft an image that would appear to be a stop sign but would be classified as a yield sign by some class of deep neural networks (Papernot et al., 2016b). Furthermore, Papernot et. al. also found that the perturbation technique they used to construct such adversarial samples is applicable to a variety of other classifiers, such as support vector machines and logistic regression (Papernot et al., 2016a). These findings demonstrated that machine learning systems are susceptible to malicious attacks. One such example would be to alter the image of road signs received by autonomous driving systems in order to manipulate the behaviour of target vehicles, which could lead to dire consequences. Thus, understanding the vulnerabilities of machine learning systems and the methods to exploit them is crucial for application of machine learning in practical settings.

2. LITERATURE

Papernot et al. in (Papernot et al., 2016a) described the following two step approach during adversarial sample creation for a black-box machine learning algorithm referred to as the 'oracle' from this point forward:

1. Train a substitute model utilizing as few calls to the oracle as possible.
2. Craft adversarial samples using either the fast gradient sign (FGS) method (Papernot et al., 2016a) or the Papernot method (Grosse et al., 2016).

In (Papernot et al., 2016a), the logistic regression (LR) and deep neural network (DNN) substitute models had the highest cross-technique transferability, indicating that adversarial techniques crafted by these models would be misclassified by oracles with a different machine learning algorithm structure (e.g. SVM, k-nearest neighbours, etc.) with a high success rate. This means that given a black-box oracle, a choice of LR or DNN substitute model should create effective adversarial samples. These substitute models must be trained on datasets obtained by querying the oracle. Papernot et al. started with a small training set, and utilized Jacobian-based dataset augmentation to increase the number of samples by querying the oracle on the datapoints that exhibit the most change. This method is described by the following formula (Papernot et al., 2016a):

$$S_{\rho+1} = \{\vec{x} + \lambda_\rho \mathbf{sgn}(J_f[O[(\vec{x})] : \vec{x} \in S_\rho)\} \cup S_\rho \quad (1)$$

where S is the training set, $\vec{x}$ is a sample in S, $O(\vec{x})$ is the label given to sample $\vec{x}$ by the oracle, J_f is the Jacobian matrix of the substitute model f, and λ is the tuneable step-size parameter. At each iteration ρ, the training set is augmented by utilizing Equation 1. The oracle is then called to obtain the labels for the new training dataset, and subsequently a new substitute model f is trained. Furthermore, in (Papernot et al., 2016a), the periodic step size (PSS) technique was introduced to improve the approximation of the oracle with the substitute model by multiplying the λ parameter by -1 when the Jacobian augmentation method no longer lead to a significant improvement in the substitute model. Then, λ_ρ is defined as

$$\lambda_\rho = \lambda(-1)^{\lfloor \frac{\rho}{\tau} \rfloor} \quad (2)$$

where τ is the number of iterations after which the Jacobian augmentation method is no longer effective. However,

the oracle should not be queried excessively to avoid raising suspicion. To diminish the calls to the oracle, reservoir sampling (RS) was utilized. Reservoir sampling selects κ randomly generated new samples after σ iterations have been completed normally. This decreases the number of calls to the oracle from $n(2)^\rho$ to $n(2)^\sigma + \kappa(\rho - \sigma)$ (Papernot et al., 2016a). Papernot et al. found that a Jacobian augmentation method combined with PSS and RS produced substitute models that approximated the oracle model successfully.

The purpose of this project is to extend the work done by Papernot et al. in (Papernot et al., 2016a) on adversarial attacks in image recognition. We investigated whether a reduction in feature dimensionality during adversarial sample crafting improved computational efficiency, while maintaining a comparable level of success in misclassification of the adversarial samples. We formed an attack on an oracle with a training set unknown to the substitute model by forcing the oracle to misclassify images that were modified with white noise undetectable to humans.

3. DATASET

Since our work extends that done by Papernot et al. in (Papernot et al., 2016b), (Papernot et al., 2016a), we utilize the same dataset cited in his papers, which is the MNIST dataset. This ensures the validity of our results. The MNIST hand-written digit dataset of 28×28 pixel images contains $50,000$ training, $10,000$ validation, and $10,000$ test greyscale images (LeCun et al., 1998).

4. METHODS

4.1. Black-Box Models

To investigate the effectiveness of Papernot's approach outlined in Section 2 with image feature reduction, a set of three black-box models were selected to act as the oracle for comparison. We chose the logistic regression (LR), support vector machine (SVM), and k-nearest neighbours (kNN) models due to simplicity of implementation and Papernot's use of these models in (Papernot et al., 2016a). We implemented the LR and kNN models as described in the course notes (Ng, 2016). For SVM, we used the `fitcsvm` function in Matlab (Statistics & Toolbox, 2015). Since SVMs are binary classifiers, to construct a multiclass classifier, we built an ensemble of one-versus-one classifiers for each pair of classes. The class assigned to a sample is the one that was selected by the majority of the classifiers (Press). The models were trained on the MNIST $50,000$ image sample set, and tested on $10,000$ samples in the test set. The performance of each of these oracle models is shown in Table 1. All models achieved a success rate of $\sim 90\%$, deeming them sufficiently accurate to utilize as the black-box oracles in experiment.

LR	SVM	kNN
87.5	93.9	96.7

Table 1. Percentage of the test set that each model classified correctly.

4.2. Logistic Regression Substitute Model

An LR substitute model was chosen for this experiment due to its high cross transferability to other models (Papernot et al., 2016a). This model was trained as described in Section 2 utilizing Jacobian-based augmentation combined with PSS and RS. However, we found the Jacobian used in (Papernot et al., 2016a) to be incorrect. Instead, for an LR model f described by the equation as in (Papernot et al., 2016a)

$$f : \vec{x} \rightarrow \left[\frac{e^{\vec{w}_j \vec{x}}}{\sum_{l=1}^{N} e^{\vec{w}_l \vec{x}}} \right] \tag{3}$$

the following Jacobian was used

$$J_f(\vec{x})[i,j] = \frac{\vec{w}_j[i] e^{\vec{w}_j \cdot \vec{x}} \sum_{l=1}^{N} e^{\vec{w}_l \cdot \vec{x}} - e^{\vec{w}_j \cdot \vec{x}} \sum_{l=1}^{N} \vec{w}_l[i] e^{\vec{w}_l \cdot \vec{x}}}{\left(\sum_{l=1}^{N} e^{\vec{w}_l \cdot \vec{x}} \right)^2} \tag{4}$$

where $N = 10$ classes for the MNIST dataset, $\vec{w}$ is the matrix of parameters for the LR substitute model, $\vec{x}$ is a sample in the substitute model's training set. The Jacobian matrix for each sample is of dimension 784×10 where 28×28 pixels in each image equals 784 features.

4.3. Generating Adversarial Samples

Adversarial samples are generated for the obtained substitute LR model by adding small modifications to the original image. We investigated two methods for crafting adversarial samples: the FGS method (Goodfellow et al., 2015) and the Papernot method (Papernot et al., 2016b).

4.3.1. Fast Gradient Sign

FGS is the algorithm utilized to generate adversarial samples in (Papernot et al., 2016a) as described by the following equation:

$$\vec{x}_{adversary} = \vec{x} + \epsilon \, \mathbf{sgn}(\nabla_{\vec{x}} f) \tag{5}$$

where the direction of the disturbance is the sign of the gradient of the probability function f described in Equation 3 (Goodfellow et al., 2015). This method is good for preliminary tests because its implementation is simple and very efficient to execute. The tuning parameter ϵ controls the size of the deviations of the adversarial samples from their origin.

4.3.2. Papernot Method

The Papernot method crafts adversarial samples by only perturbing a subset of features with the highest saliency values (Papernot et al., 2016b). The first γ features forming the perturbation, $\delta_{\vec{x}}$, are chosen in the order of decreasing adversarial saliency, $S(\vec{x}, t)[i]$, which is defined as follows:

$$S(\vec{x}, t)[i] = \begin{cases} 0 \text{ if } \frac{\partial F_t}{\partial \vec{x}_i}(\vec{x}) < 0 \text{ or } \sum_{j \neq t} \frac{\partial F_j}{\partial \vec{x}_i}(\vec{x}) > 0 \\ \frac{\partial F_t}{\partial \vec{x}_i}(\vec{x}) \left| \sum_{j \neq t} \frac{\partial F_j}{\partial \vec{x}_i}(\vec{x}) \right| \text{ otherwise} \end{cases} \tag{6}$$

where matrix $J_F = [\frac{\partial F_j}{\partial \vec{x}_i}]_{ij}$ is the Jacobian. This method is more computationally intensive, but introduces less visible perturbation to each image (Papernot et al., 2016b). Note that the saliency values in expression (6) only account for positive perturbations. We extended this method by calculating the saliency for features that introduced negative perturbations using the same setup, and ordered all features based on the absolute value of their saliency.

4.4. PCA Feature Dimensionality Reduction

In an attempt to improve the computational efficiency of the adversarial sample crafting algorithm, we chose to apply principle component analysis (PCA) to reduce the dimensionality of the feature-set. PCA projects the training samples on the first k eigenvectors (loading factors) of the empirical covariance matrix of the dataset, thus reducing the size of the feature space (Ng, 2016). Then, each image can be reduced in dimensionality by multipling it by a transformation matrix T' composed of the k loading factors:

$$x_i' = T^T x_i \tag{7}$$

where x_i is the i^{th} image. To restore the reduced images to their original space we reverse the transformation using

$$x_i = T x_i' \tag{8}$$

since the eigenvectors are orthogonal. The PCA algorithm was implemented with the MATLAB `pca` command (Statistics & Toolbox, 2015). We selected the first 98 components of the training set, reducing the number of features by a factor of 8.

5. RESULTS and DISCUSSION

5.1. LR Substitute Model Performance

As in (Papernot et al., 2016a), the LR substitute model began with a training set of 100 samples from the MNIST validation set with labels obtained from each of the three black-box oracles. Since our Jacobian formulation is different from that in (Papernot et al., 2016a), we had to make minor adjustments to the parameters in the PSS algorithm in order to achieve comparable results. An optimal value of $\tau = 1$ was chosen to improve the Jacobian-based augmentation method for the substitute model's training set. The parameters utilized in our LR model versus those in (Papernot et al., 2016a) are summarized in Table 2.

	λ	κ	τ	σ	ρ
Our approach	0.1	400	1	3	9
Papernot et al.	0.1	400	3	3	9

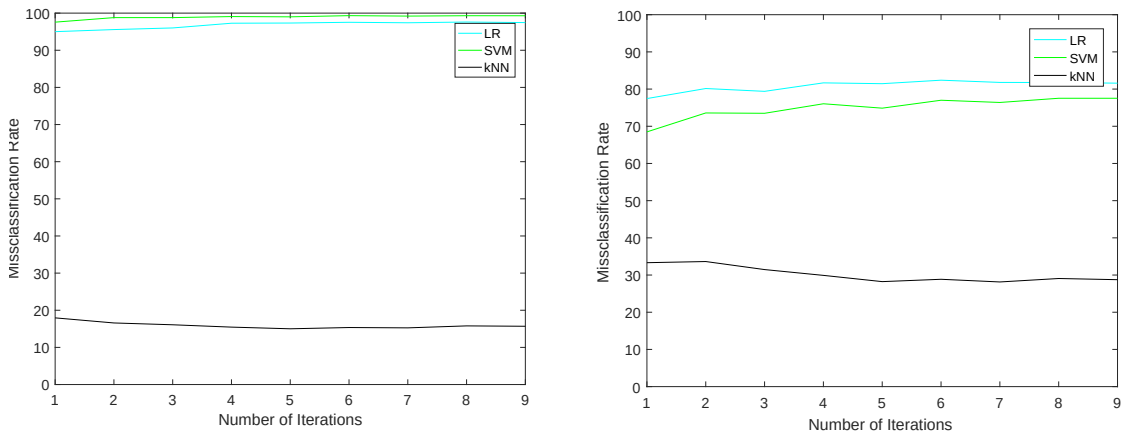
Table 2. Comparison of the parameters used for PSS and RS methods during Jacobian-based training set augmentation.

The ability of the substitute model to approximate the LR, kNN, and SVM oracles is summarized in Figure 1. The percentage of samples for which the substitute model's and oracle's classifications match is plotted against the iteration of the training set augmentation. These results agree with those in (Papernot et al., 2016a) quite well. The success rate increased steadily until it plateaued at the third iteration, after which RS is activated. As expected, the LR substitute model performed the best on an LR model oracle; nevertheless, it performed well even on the SVM and kNN oracles. The algorithm could theoretically be truncated at the third iteration since past this point it had nearly reached convergence, saving computation time.

A comparison of the success of the LR substitute model against the oracle with PCA feature reduction is portrayed in Figure 2. The PCA algorithm reduced the dimensionality of the feature space by a factor of 8, but the results of the trained LR substitute model were nearly identical to those with the entire feature space. Hence, PCA has minimal effect on the training of the substitute model, and is a valid method for feature space reduction.

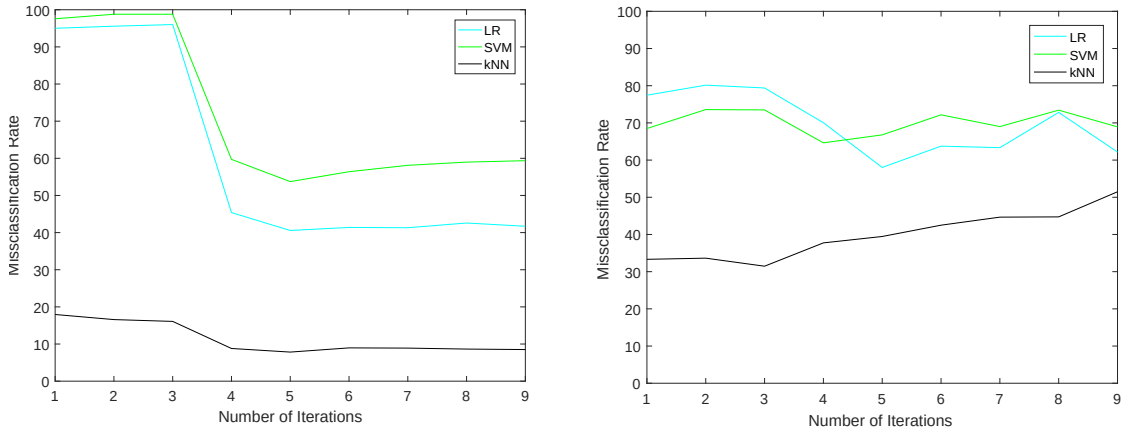
5.2. Performance of Crafted Adversarial Samples

In our first experiment, we generated adversarial samples on the $10,000$ test samples based on the LR substitute model. The misclassification rates on the adversarial samples by the oracles are given in Figures 3(a) and 3(b) for both the FGS and Papernot methods utilizing the full feature sets. To generate adversarial samples with FGS, we used $\epsilon = 0.3$ in Equation 5, which is the same value as



(a) Fast Gradient Descent Algorithm (b) Papernot Algorithm

Figure 3. Misclassification rate vs. iteration of the Jacobian-based dataset augmentation without feature reduction



(a) Fast Gradient Descent Algorithm (b) Papernot Algorithm

Figure 4. Misclassification rate vs. iteration of the Jacobian-based dataset augmentation with PCA feature reduction

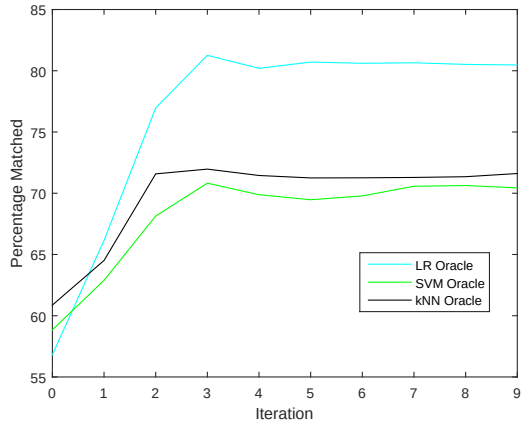


Figure 1. Percentage of samples for which the substitute model and oracle classifications agree without feature reduction.

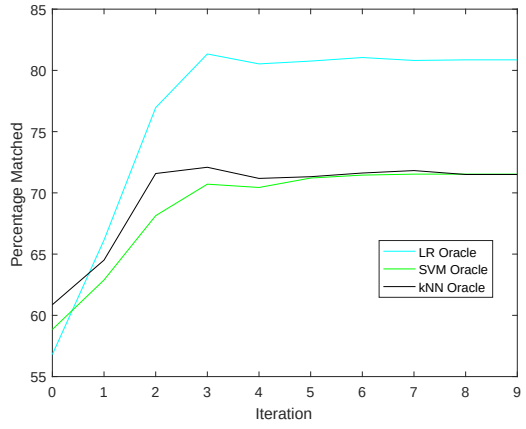


Figure 2. Percentage of samples for which the substitute model and oracle classifications agree with PCA feature reduction.

that used in (*Papernotet al.*, 2016*a*). For the Papernot method, 10% of the features (pixels) were perturbed for each image ($\gamma = 0.1$) with $\epsilon = 1$, similar to the parameters in (Papernot et al., 2016b).

In Figures 3(a) and 3(b), we were able to achieve fairly high misclassification rates for both the LR oracle and the SVM oracle, but performed poorly for the kNN oracle, which agrees with the results obtained by Papernot et al. (*Papernotet al.*, 2016*a*). Our misclassification rate was, in fact, slightly higher for all 3 models for the FGS method as reported in (Papernot et al., 2016a), likely caused by the different gradient formulation utilized in our approach. The Papernot method performed notably worse than FGS due to intrinsically smaller perturbations. However, the Papernot parameters can be further tuned to achieve a desired balance between the misclassification rate and the amount of perturbation.

In our second experiment, PCA feature-reduction was introduced at iteration 4 of the Jacobian-based dataset augmentation, when the training set size exceeded the number of features. Adversarial samples were generated by first carrying out FGS and Papernot algorithms on the reduced feature space and then restoring the samples as described in section 4.4 before passing them into the oracle. The results are shown in Figures 4(a) and 6. From Figure 4(a), we can see that introducing PCA in the FGS algorithm dramatically decreased the misclassification rate. However, from Figure 4(b), the misclassification rate for the Papernot algorithm was only slightly impacted by the introduction of PCA, maintaining a reasonable misclassification rate of about $\sim 70\%$. For the kNN oracle, the rate even increased according to the figure. This suggests that PCA is a suitable feature reduction technique to use for the Papernot algorithm, specifically.

To calculate the computational cost, we measured the time required to generate all $10,000$ of the adversarial samples in both experiments. The results are shown in Table 5. For the FGS algorithm, although there was a reduction in running time, the change was not significant, as FGS is already computationally efficient. For the Papernot algorithm, the reduction was substantial; the running time was cut by more than a factor of 2 for all three oracles. This suggests that PCA is successful at reducing the computational cost associated with the Papernot algorithm.

The confusion matrices for the unaltered test images and the adversarial samples generated using the Papernot algorithm with PCA are shown in Figures 5 and 6, respectively. The number in position (i, j) of the grid corresponds to the percentage of class i images being classified as class j. The Papernot algorithm with PCA was generally successful at misdirecting the oracle, except for the numbers "1" and "6" (darker squares in Figure 6), which were robust to the attack. The precision, recall, and accuracy for all test samples and adversarial samples are displayed in Table 5,

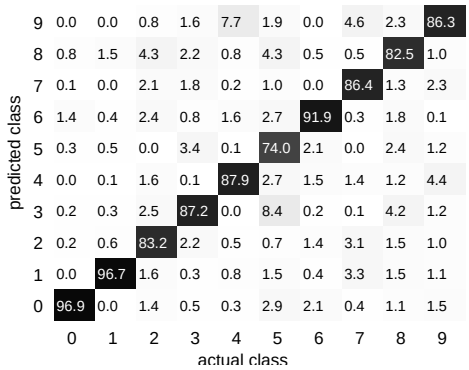


Figure 5. Confusion matrix against the LR oracle for the original 10,000 test samples.

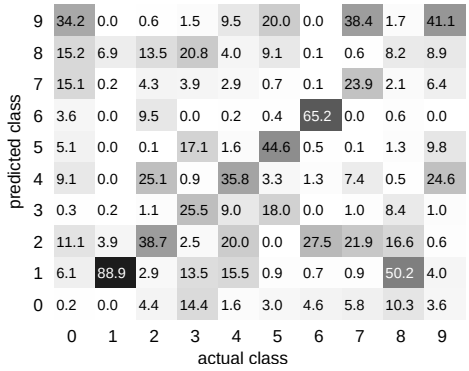


Figure 6. Confusion matrix against the LR oracle for the 10,000 adversarial samples generated using the Papernot method with PCA.

presenting the same trends as in the confusion matrices.
A single unaltered image and its associated sample adversarial images are shown in Figure 7. Both the SVM oracle and the LR oracle misclassified the perturbed images. We can see that these images are still easily recognizable as a ”9”, so the adversarial attacks were successful. However, the deviations added to the image are also clearly visible to the human eye, which is undesirable. These deviations could be reduced with further parameter tuning and higher resolution images. Note that, as expected, the deviations added to the image were less noticeable in the Papernot algorithm as compared to that of the FGS method.

	LR	SVM	kNN
FGS	1.3912	1.5692	1.7441
FGS + PCA	1.3509	1.2557	1.3681

Table 3. Runtime for generating 10,000 adversarial samples in seconds for the three oracle models using FGS.

	LR	SVM	kNN
Papernot	7.9147	7.9776	8.57
Papernot + PCA	3.1826	3.0719	3.3997

Table 4. Runtime for generating 10,000 adversarial samples in seconds for the three oracle models using Papernot method.

	Precision	Recall	Accuracy
Original images	87.4645	87.3036	87.5300
Adversarial images	37.3687	37.2288	37.8400

Table 5. Average precision, recall, and accuracy for the original test images and adversarial images constructed using Papernot algorithm with PCA.

6. CONCLUSIONS and FUTURE WORK

We attacked image classifiers using a two fold strategy: we first imitated the target classifier with a substitute LR model and then generated adversarial samples based on the substitute model (Papernot et al., 2016a). To reduce the computational cost of crafting adversarial samples, we reduced the feature space dimensionality by utilizing PCA. Although PCA reduced the performance of the adversarial samples in FGS, it only had a small impact on the success of the Papernot algorithm, maintaining a reasonable misclassification rate of about $\sim 70\%$. Furthermore, although the runtime reduction due to PCA was negligible for FGS, it was significant for the Papernot algorithm, reducing the computation time by half. We have shown that we can increase the efficiency of adversarial sample construction while maintaining misclassification effectiveness utilizing the Papernot adversarial sample crafting method in combination with PCA feature reduction.

For future work, the same approach could be applied to more complex images such as the GTSRB dataset of coloured traffic signs. The GTSRB images have a higher resolution, allowing for the perturbations added to the samples to be more subtle. Furthermore, the algorithm presented in this paper should be extended to targeted misclassification. These additions would portray the inherent

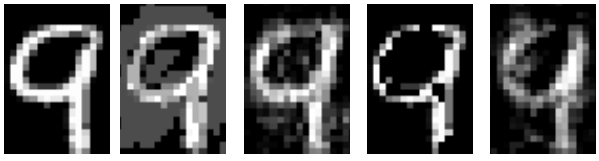


Figure 7. From left to right, the figures are: original image, adversarial image with FGS method, adversarial image with FGS method and PCA feature reduction, adversarial image with Papernot method, adversarial image with Papernot method and PCA.

vulnerability of machine learning algorithms to adversarial attacks in such high-risk applications as autonomous car navigation (e.g. having the oracle read a stop sign as a yield sign (Papernot et al., 2016b)).

References

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Samples. Proceedings of the 2015 International Conference on Learning Representations, 2015.

Grosse, K., Papernot, N., Manoharan, P., Backes, M., and McDaniel, P. D. Adversarial perturbations against deep neural networks for malware classification. *CoRR*, abs/1606.04435, 2016. URL http://arxiv.org/abs/1606.04435.

LeCun, Y., Cortes, C., and Burges, C.J.C. The MNIST Database, 1998.

Ng, Andrew. Cs229: Course notes. 2016.

Papernot, N., McDaniel, P. D., and Goodfellow, I. J. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016a. URL http://arxiv.org/abs/1605.07277.

Papernot, N., McDaniel, P. D., Goodfellow, I. J., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR*, abs/1602.02697, 2016b. URL http://arxiv.org/abs/1602.02697.

Press, Cambridge University. Multiclass svms. URL http://nlp.stanford.edu/IR-book/html/htmledition/multiclass-svms-1.html.

Statistics and Toolbox, Machine Learning. *MATLAB version 8.5.0 (R2015a)*. The MathWorks Inc., Natick, Massachusetts, 2015.