IDENTIFICATION OF AUTISM IN CHILDREN

[1]Mr. K. DEVADAS, [2]A. SHANMUKHA CHANDRA, [3]A. AKSHAY, [4]D. TARUN

[1](Assistant Professor) ,**CSE.** Teegala Krishna Reddy Engineering College Hyderabad

[234]B,tech scholar ,**CSE.** Teegala Krishna Reddy Engineering College Hyderabad

ABSTRACT

Current transfer learning methods are mainly based on finetuning a pretrained model with target-domain data. Motivated by the techniques from adversarial machine learning (ML) that are capable of manipulating the model prediction via data perturbations, in this paper we propose a novel approach, black-box adversarial reprogramming (BAR), that repurposes a well-trained black- box ML model (e.g., a prediction API or a proprietary software) for solving different ML tasks, especially in the scenario with scarce data and constrained resources. The rationale lies in exploiting high-performance but unknown ML models to gain learning capability for transfer learning. Using zeroth order optimization and multi-label mapping techniques, BAR can reprogram a black- box ML model solely based on its input-output responses without knowing the model architecture or changing any parameter. More importantly, in the limited medical data setting, on autism spectrum disorder classification, diabetic retinopathy detection, and melanoma detection tasks, BAR outperforms state-of-the-art methods and yields comparable performance to the vanilla adversarial reprogramming method requiring complete knowledge of the target ML model. BAR also out- performs baseline transfer learning approaches by a significant margin, demonstrating cost-effective means and new insights for transfer learning.

1. INTRODUCTION

In this paper, we revisit transfer learning to address two fundamental questions: (i) Is finetuning a pretrained model necessary for learning a new task? (ii) Can transfer learning be expanded to black-box ML models where nothing but only the input-output model responses (data samples and their predictions) are observable? In contrast, we call fine- tuning a white-box transfer learning method as it assumes the source-domain model to be transparent and modifiable. Recent advances in adversarial ML have shown great capability of manipulating the prediction of a well-trained deep learning model by designing and learning perturbations to the data inputs without changing the target model (Biggio & Roli, 2018), such as prediction-evasive adversarial examples (Szegedy et al., 2014). Despite of the “vulnerability” in deep learning models, these findings also suggest the plausibility of transfer learning without modifying the pretrained model if an appropriate perturbation to the target-domain data can be learned to align the target-domain labels with the pre- trained source-domain model predictions. Indeed, the adversarial reprogramming (AR) method proposed in (Elsayed et al., 2019) partially gives a negative answer to Question (i) by showing simply learning a universal target-domain data perturbation is sufficient to repurpose a pretrained source- domain model, where the domains and tasks can be different, such as reprogramming an ImageNet classifier to solve the task of counting squares in an image. However, the authors did not investigate the performance of AR on the limited data setting often encountered in transfer learning. More- over, since the training of AR requires backpropagation of a deep learning mode.

1.1 OBJECTIVE

Autism spectrum disorder (ASD) is a complicated neurological developmental disorder that manifests itself in a variety of ways. The child diagnosed with ASD and their parent’s daily lives can be dramatically improved with early diagnosis and appropriate medical intervention. The applicability of static features extracted from autistic children’s face photographs as a biomarker to distinguish them from typically developing children is investigated in this study paper.

We used CNN models and a RNN model to identify autism in children accurately. We used a publicly available dataset to train the suggested models, which consisted of face pictures of children diagnosed with autism and controls classed as autistic and non-autistic.

1.2 PROBLEM STATEMENT

In this project, we aim to address the challenge of autism spectrum disorder (ASD) diagnosis using machine learning, particularly in scenarios with limited data and resources. Our goal is to develop a novel method, Black-box Adversarial

Reprogramming (BAR), to repurpose well-trained black-box ML models for ASD classification based on facial features extracted from static images. BAR leverages input-output responses without requiring prior knowledge of model architectures. We aim to demonstrate BAR's effectiveness compared to existing methods and validate its practicality in real-life settings. Ultimately, our research aims to improve ASD diagnosis and advance transfer learning methodologies in healthcare.

2.LITERATURE SURVEY

2.1 Ten years after the rise of adversarial machine learning.

AUTHOR: iggio, B. and Roli, F. Wild patterns:

ABSTRACT:

Learning-based pattern classifiers, including deep networks, have shown impressive performance in several application domains, ranging from computer vision to cybersecurity. However, it has also been shown that adversarial input perturbations carefully crafted either at training or at test time can easily subvert their predictions. The vulnerability of machine learning to such wild patterns (also referred to as adversarial examples), along with the design of suitable countermeasures, have been investigated in the research field of adversarial machine learning. In this work, we provide a thorough overview of the evolution of this research area over the last ten years and beyond, starting from pioneering, earlier work on the security of non-deep learning algorithms up to more recent work aimed to understand the security properties of deep learning algorithms, in the context of computer vision and cybersecurity tasks. We report interesting connections between these apparently-different lines of work, highlighting common misconceptions related to the security evaluation of machine-learning algorithms. We review the main threat models and attacks defined to this end, and discuss the main limitations of current work, along with the corresponding future challenges towards the design of more secure learning algorithms.

2.2 Evasion attacks against machine learning at test

AUTHOR: Biggio, B., Corona, I., Maiorca, D., Nelson, B., ˇSrndi c, N., Laskov, P., Giacinto, G., and Roli, F

ABSTRACT:

In security-sensitive applications, the success of machine learning depends on a thorough vetting of their resistance to adversarial data. In one pertinent, well-motivated attack scenario, an adversary may attempt to evade a deployed system at test time by carefully manipulating attack samples.

In this work, we present a simple but effective gradient-based approach that can be exploited to systematically assess the security of several, widely-used classification algorithms against evasion attacks. Following a recently proposed framework for security evaluation, we simulate attack scenarios that exhibit different risk levels for the classifier by increasing the attacker's knowledge of the system and her ability to manipulate attack samples. This gives the classifier designer a better picture of the classifier performance under evasion attacks, and allows him to perform a more informed model selection (or parameter setting). We evaluate our approach on the relevant security task of malware detection in PDF files, and show that such systems can be easily evaded. We also sketch some countermeasures suggested by our analysis.

3. SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

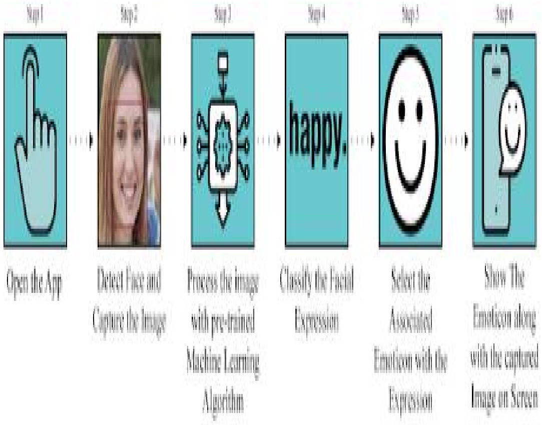
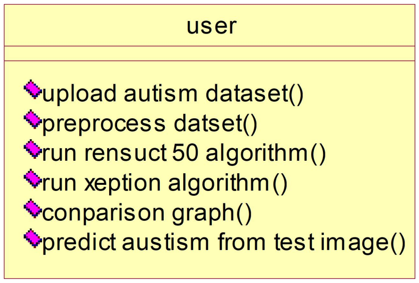

Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
happy.
Open the App
Detect Face and Capture the Image
Process the image with pre-trained Machine Learning Algorithm
Classify the Facial Expression
Select the Associated Emoticon with the Expression
Show The Emoticon along with the captured Image on Screen

Figure. System architecture

UML DIAGRAMS

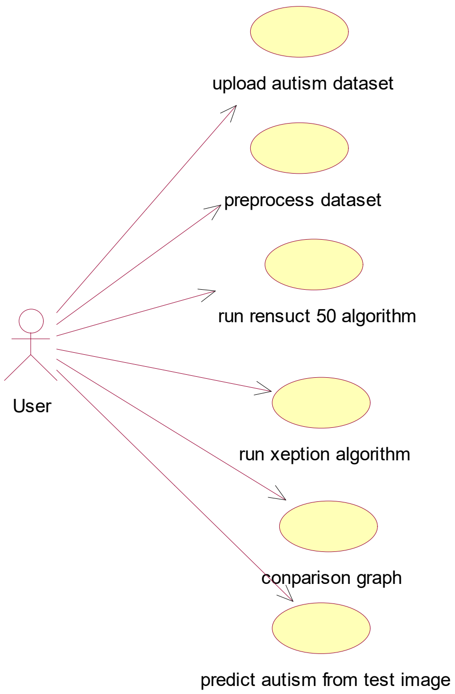
CLASS DIAGRAM:-

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



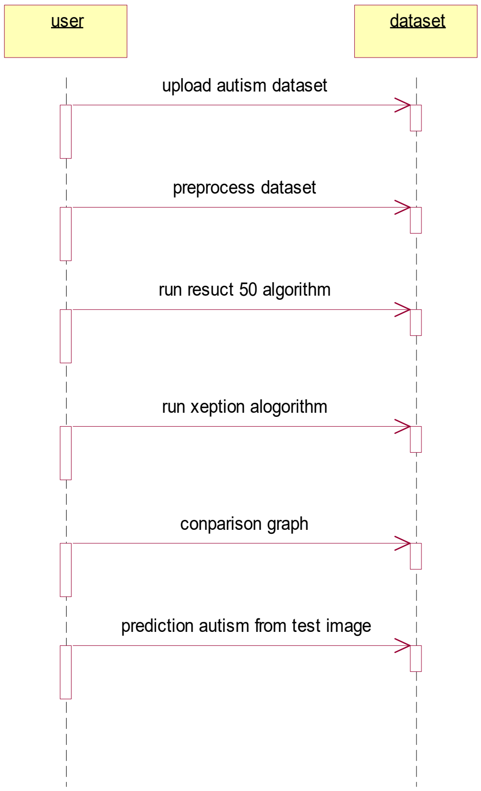
USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



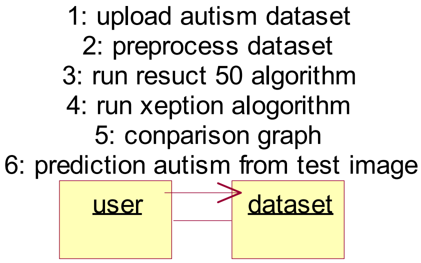
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



COLLABRATION DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

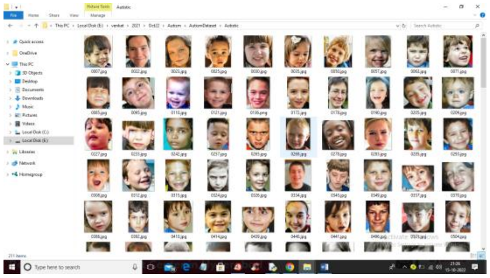

1: upload autism dataset
2: preprocess dataset
3: run resuct 50 algorithm
4: run xeption alogorithm
5: conparison graph
6: prediction autism from test image
user
dataset

MODULES

- **Upload Autism Dataset:** using this module we will upload dataset images to application
- **Preprocess Dataset:** using this module we will read each images and then resize all images to equal sizes and then normalize pixel values. Split dataset into train and test where application used 80% images for training and 20% for testing
- **Run Resnet50 Algorithm:** processed train images will be input to Resnet50 transfer learning algorithm to train Autism prediction model. This model will be applied on test images to calculate prediction accuracy
- **Run Xception Algorithm:** processed train images will be input to Xception transfer learning algorithm to train Autism prediction model. This model will be applied on test images to calculate prediction accuracy
- **Comparison Graph:** using this module we will plot comparison graph between both algorithms
- **Predict Autism from Test Image:** using this module we will upload test image and then algorithm will predict weather image is ‘Autistic’ or ‘Non-Autistic’.

4. OUTPUT SCREENS

In this project we are using Resnet50 and Xception algorithm with transfer learning technique to train Autism detection model. To train both algorithms we have used same dataset given by you. This dataset consists of two different classes such as ‘Autistic’ and ‘Non-Autistic’ and below screen showing images from dataset folder

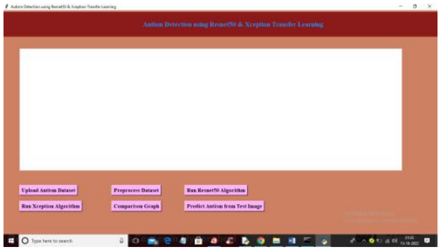


So by using above images we will train both algorithms.

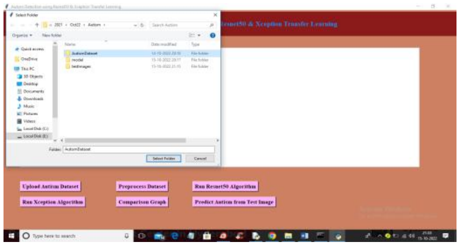
To implement this project we have designed following STEPS:

1. Upload Autism Dataset
2. Preprocess Dataset
3. Run Resnet50 Algorithm
4. Run Xception Algorithm
5. Comparison Graph
6. Predict Autism from Test Image

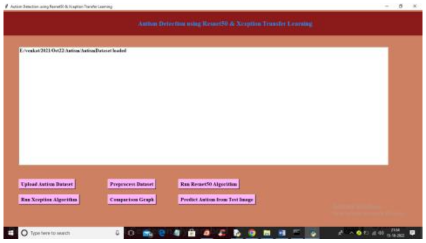
To run project double click on 'run.bat' file to get below screen



In above screen click on 'Upload Autism Dataset' button to upload dataset and get below output

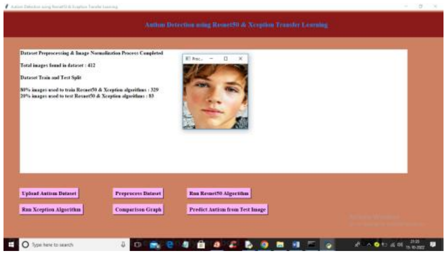


In above screen selecting and uploading 'Autism Dataset' folder and then click on 'Select Folder' button to load dataset and get below output



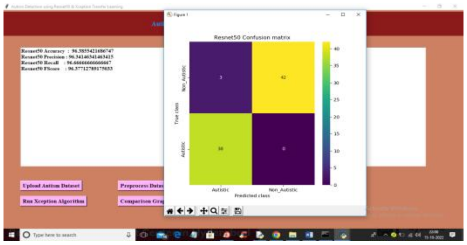
```
E:/venkat/2021/Oct22/Autism/AutismDataset loaded
```

In above screen dataset loaded and now click on 'Preprocess Dataset' button to read and process all images and get below output



```
Dataset Preprocessing & Image Normalization Process Completed

Total images found in dataset : 412

Dataset Train and Test Split

80% images used to train Resnet50 & Xception algorithms : 329
20% images used to test Resnet50 & Xception algorithms : 83
```

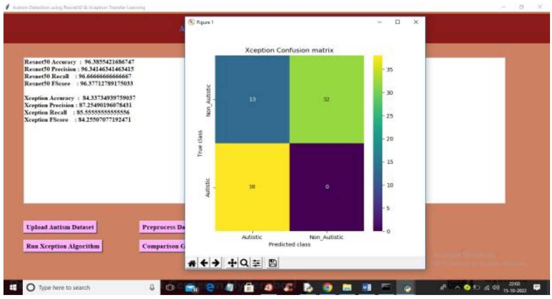
In above screen dataset processed and to check weather images processed properly I am showing sample image and now close above image and we can see dataset contains 412 images where application using 329 images for training and 83 for testing. Now click on 'Run Resnet50 Algorithm' button to train Resnet50 and get below output



```
Resnet50 Accuracy  : 96.3855421686747
Resnet50 Precision : 96.34146341463415
Resnet50 Recall    : 96.66666666666667
Resnet50 FScore    : 96.37712789175033
```

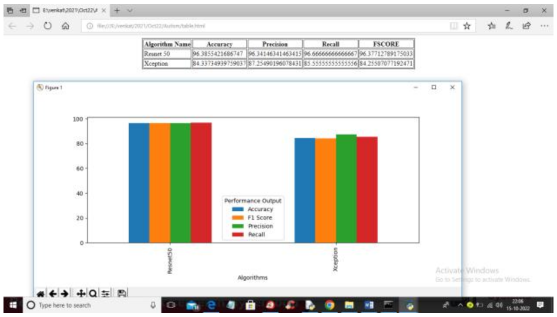
In above screen Resnet50 training completed and we got accuracy as 96% and in confusion matrix graph x-axis represents

PREDICTED classes and y-axis represents TRUE CLASSES. In above graph same colour boxes represents INCORRECT prediction count and different colour boxes represents CORRECT prediction count and Resnet50 predict only 3 records as incorrectly. Now close above graph and the click on 'Run Xception Algorithm' button to train Xception and get below output



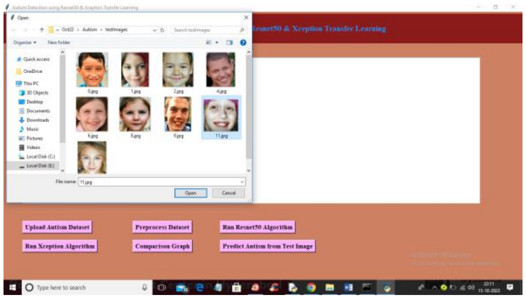
```
Resnet50 Accuracy  :  96.3855421686747
Resnet50 Precision : 96.34146341463415
Resnet50 Recall    : 96.66666666666667
Resnet50 FScore    : 96.37712789175033

Xception Accuracy  :  84.33734939759037
Xception Precision : 87.25490196078431
Xception Recall    : 85.55555555555556
Xception FScore    : 84.25507077192471
```

In above screen Xception training completed and with Xception we got 84% accuracy and in confusion matrix graph we can see Xception predict 13 records incorrectly. So from both algorithms Resnet50 got high accuracy. Now click on 'Comparison Graph' button to get below graph

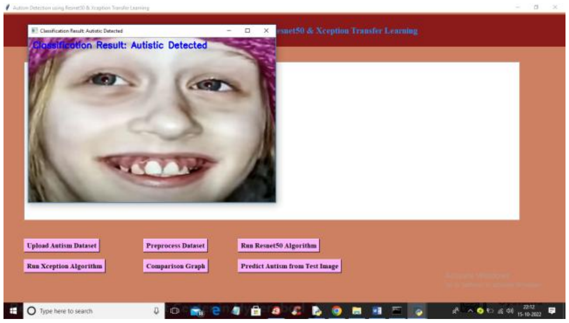


Algorithm Name	Accuracy	Precision	Recall	FSCORE
Resnet 50	96.3855421686747	96.34146341463415	96.66666666666667	96.37712789175033
Xception	84.33734939759037	87.25490196078431	85.55555555555556	84.25507077192471

In above graph x-axis represents algorithm names and y-axis represents accuracy, precision, recall and F1SCORE in different colour bars. In above graph we can see Resnet50 got high performance. Now close above graph and then click on 'Predict Autism from Test Image' button to upload test image and get below output

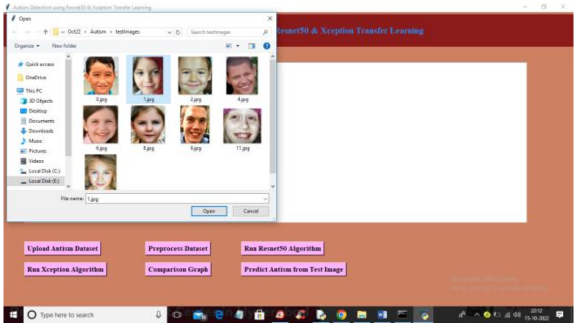


In above screen selecting and uploading '11.jpg' and then click on 'Open' button to get below prediction output

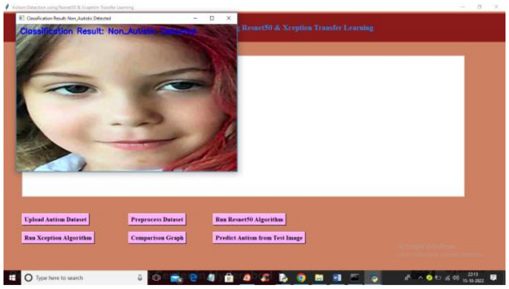


Classification Result: Autistic Detected

In above screen image is classified as 'Autistic Detected' and now upload other image and get output



In above screen selecting and uploading '1.jpg' and then click on 'Open' button to upload image and get below output


Autism Detection using Resnet50 & Xception Transfer Learning
Classification Result: Non_Autistic Detected
Classification Result: Non_Autistic Detected
Resnet50 & Xception Transfer Learning
Upload Autism Dataset
Preprocess Dataset
Run Resnet50 Algorithm
Run Xception Algorithm
Comparison Graph
Predict Autism from Test Image
Type here to search
22:13
15-10-2022

In above screen image is classified as 'Non Autistic'. Similarly you can upload and test other images

5. CONCLUSION

In conclusion, this paper introduces BAR, a novel methodology for adversarial reprogramming of black-box ML models through zeroth-order optimization and multi-label mapping techniques. In contrast to conventional AR techniques reliant on full knowledge of the target model, BAR operates solely on input-output model responses, facilitating black-box transfer learning for access-limited ML models. Through extensive evaluation across three data-scarce medical ML tasks, BAR exhibits comparable performance to the vanilla white-box AR method, surpassing state-of-the-art techniques and conventional fine-tuning approaches. Moreover, practical experiments demonstrate BAR's efficacy in reprogramming real-world online image classification APIs at reasonable costs. Comprehensive ablation studies and sensitivity analyses further underscore the robustness and practical utility of BAR. This research presents a promising avenue for transfer learning without necessitating knowledge or alteration of pre-existing models, thereby advancing the frontier of adversarial reprogramming in machine learning.

6. FUTURE ENHANCEMENT

- **Include Dynamic Features:** Explore adding dynamic features like facial expressions and movements to enhance autism diagnosis accuracy.
- **Combine Different Data Types:** Investigate combining facial images with other data types like clinical

notes or sensor data for a more comprehensive understanding.

- **Use Pre-trained Models:** Consider using pre-trained models and fine-tuning them for autism diagnosis to improve performance with limited data.
- **Enhance Data Augmentation:** Develop better methods for creating diverse training data, which could involve using advanced techniques like generative adversarial networks (GANs).
- **Create Understandable Models:** Work on making models easier to interpret by using techniques that highlight important features for diagnosis.
- **Study Long-term Changes:** Extend research to track changes in facial features over time to monitor the progression of autism and assess intervention effectiveness.
- **Validate with Clinicians:** Collaborate with clinicians to ensure models are reliable and applicable in real-world settings before deployment.

By pursuing these avenues for refinement, researchers can advance the accuracy, robustness, and clinical utility of models for autism spectrum disorder diagnosis based on facial features. Additionally, addressing ethical considerations such as data privacy, fairness, and bias mitigation remains paramount throughout the research process.

7. REFERENCES

1. Biggio, B. and Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition, 84:317–331, 2018.
2. Biggio, B., Corona, I., Maiorca, D., Nelson, B., ˇSrndi c, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In Joint European conference on machine learning and knowledge discovery in databases, pp. 387–402, 2013.
3. Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. International Conference on Learning Representations, 2018.
4. Carlini, N. and Wagner, D. Towards evaluating the robust- ness of neural networks. In IEEE Symposium on Security and Privacy, pp. 39–57, 2017. Chen, P.-Y., Zhang, H.,

Sharma, Y., Yi, J., and Hsieh, C.-J. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute mod- els. In ACM Workshop on Artificial Intelligence and Security, pp. 15–26, 2017a.

5. Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., and Hsieh, C.- J. EAD: elastic-net attacks to deep neural networks via adversarial examples. AAAI, 2018. Transfer Learning without Knowing: Reprogramming Black-box Machine Learning Models Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526, 2017b.
6. Cheng, M., Le, T., Chen, P.-Y., Yi, J., Zhang, H., and Hsieh, C.-J. Query-efficient hard-label black-box attack: An optimization-based approach. International Conference on Learning Representations, 2019.
7. Cheng, M., Singh, S., Chen, P. H., Chen, P.-Y., Liu, S., and Hsieh, C.-J. Sign-OPT: A query-efficient hard-label ad- versarial attack. In International Conference on Learning Representations, 2020. Codella, N., Rotemberg, V., Tschandl, P., Celebi, M. E., Dusza, S., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M., et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). arXiv preprint arXiv:1902.03368, 2019