SYNTHESIS OF ADVERSARIAL SAMPLES IN TWO-STAGE CLASSIFIERS

Ismail R. Alkhouri[1] *Alvaro Velasquez*[2] *George K. Atia*[1,3]

[1] Department of Electrical and Computer Engineering, University of Central Florida, Orlando FL, USA
[2] Information Directorate, Air Force Research Laboratory, Rome NY, USA
[3] Department of Computer Science, University of Central Florida, Orlando FL, USA

ABSTRACT

Adversarial attacks can drastically reduce the accuracy and confidence level of classifiers while being imperceptible. Existing studies on the topic have largely focused on one-stage classifiers. In this paper, we study the robustness of two Two-Stage Hierarchical Classifier models, the flat and top-down hierarchical classifiers, termed FHC and TDHC respectively, to targeted and confidence reduction attacks. We formulate feasibility programs based on similarity and distance measures for the one-shot synthesis of adversarial examples, and devise a generative approach to the solution. In this approach, the adjustable parameters of a generative network are iteratively updated by optimizing loss functions for the dual objective of (i) low attack perceptibility and (ii) small distance from desired soft predictions. We demonstrate the performance of the proposed approach in terms of imperceptibilty and measures of attack success, and show it compares favorably with state-of-the-art techniques.

***Index Terms*—** Targeted and confidence reduction Attacks, Two-stage classifiers, Generative networks, One-shot synthesis

1. INTRODUCTION

The importance of classification systems has increased tremendously, especially with the massive growth of the implementation and deployment of artificial intelligence applications [1, 2]. However, recent work has uncovered the impact of imperceptible adversarial attacks on these systems and exposed the poor robustness of state-of-the-art architectures [3, 4]. This has motivated the use of hierarchical classifiers by which the classification problem can be decomposed into super-class labels and sub-class labels, thereby mitigating the lack of robustness inherent in individual classifiers [5].

Based on the goal of the adversary, attacks are categorized as *targeted*, where the objective is to alter the classification to a pre-defined target label; *non-targeted*, where it is required to miss-classify the true label; and *confidence reduction*, where the goal is to reduce the classification confidence level of the true class to cause ambiguity [6]. The latter is employed against systems for which a confidence threshold is introduced and the classification is only regarded if the prediction level confidence score is above that threshold [7].

Most of the recent studies on individual adversarial attacks [8] have focused on additive perturbations and One-Stage Classifiers (OSCs). This work departs from this line of research by considering two important directions. First, we consider a non-additive generative approach to synthesize samples that achieve high similarity with some desired features, given as benign example features, and follow a prespecified distribution. While generative networks have been used in constructing adversarial attacks [9], the proposed approach differs in that a labeled training dataset is not required. Instead, our one-shot synthesis approach requires only a single datum from which new data is generated. Second, we consider two Two-Stage Hierarchical Classifiers (TSHCs) where the data to be classified is categorized into a class hierarchy of coarse-grained super-classes and fine-grained sub-classes. TSHCs have been used in many applications and domains [10]. We examine the Flat Hierarchical Classifier (FHC), where a feature vector is first classified according to the subclass, then the super-class is obtained from a pre-defined mapping. Additionally, we consider the Top-Down Hierarchical Classifier (TDHC) – also known as the coarse-to-fine classifier – where the feature vectors are classified in line with the coarser genres, followed by finer prediction as a result of the initial upper level classification.

In this paper, we formulate targeted and confidence reduction attacks against TSHCs. Then, motivated by the framework presented in [11], we propose an algorithmic solution using the back-propagation algorithm [12] and generative networks for the Bidirectional One-Shot Synthesis (BOSS) of adversarial examples on TSHCs. Bidirectional, in this sense, refers to the input and output constraints that must be satisfied by the synthesized datum. We conduct experiments where we illustrate the extended and enhanced capabilities of our method in comparison to latest existing additive approaches [13] in terms of imperceptibility and attack success ratio.

This work is supported by AFRL Contract FA8750-20-3-1004, AFOSR Award 20RICOR012, NSF CAREER Award CCF-1552497, NSF Award CCF-2106339, and DOE Award DE-EE0009152.

2. PROBLEM FORMULATION

2.1. Hierarchical Models

The first stage of the Flat Hierarchical Classifier (FHC) consists of the classifier function $K : \mathbb{R}^N \rightarrow [M]$, which classifies the input observation vector $\mathbf{x} \in \mathbb{R}^N$ as one of M classes, where $[M] := \{1, 2, \ldots, M\}$. Let $p_k : \mathbb{R}^N \times [M] \rightarrow [0, 1]$ denote the output probability of a given FHC classifier such that $K(\mathbf{x}) = \underset{l \in [M]}{\operatorname{argmax}} p_k(l|\mathbf{x}; \theta_k)$, where θ_k are the trained parameters. Henceforth, we omit the parameters θ for brevity.

In the second stage of the FHC, a function $T : [M] \rightarrow [M_\text{c}]$ maps the predicted (fine) label to a (coarse) super-class $i \in [M_\text{c}]$, where M_c is the total number of super-classes. To capture the inverse mapping, we define the super-class sets $S_i = \{l \in [M] : T(l) = i\}$, $i \in [M_\text{c}]$, consisting of all fine labels that map to the same super-class i.

For the Top-Down Hierarchical Classifier (TDHC) model, we define the function $R : \mathbb{R}^N \rightarrow [M_\text{c}]$, which first obtains a coarse-level prediction as $R(\mathbf{x}) = \underset{l \in [M_\text{c}]}{\operatorname{argmax}} p_r(l|\mathbf{x}; \theta_r)$, where $p_r : \mathbb{R}^N \times [M_\text{c}] \rightarrow [0, 1]$ is the probability discriminant functional representing the probability induced over the coarse classes. For the second stage, for each coarse label $i \in [M_\text{c}]$, we define the mapping $F : \mathbb{R}^N \times [M_\text{c}] \rightarrow [M]$ as $F(\mathbf{x}, i) = \underset{l \in [M_i]}{\operatorname{argmax}}\, p_f(l|\mathbf{x}, i; \theta_f)$, which obtains the fine label $l \in [M_i]$ given the coarse prediction of $i \in [M_\text{c}]$, by maximizing over the probabilistic discriminant functionals $p_f : \mathbb{R}^N \times [M_c] \times [M] \rightarrow [0, 1]$.

2.2. Formulation of Attacks

We use $d : \mathbb{R}^N \times \mathbb{R}^N \rightarrow [0, 1]$, as a measure of similarity between two feature vectors in $\mathbb{R}^N$, where a value 0 indicates they are identical. Also, $D : \Delta^M \times \Delta^M \rightarrow [0, 1]$, is a distance between probability distributions, where Δ^M is the probability simplex of dimension M, and a value 0 indicates two identical distributions.

For the FHC model, the attack can be formulated as the problem of finding an input $\mathbf{x} \in \mathbb{R}^N$ that simultaneously resembles a specific example and induces an output distribution that is sufficiently close to a pre-defined target distribution, i.e.,

$$\begin{aligned} \text{Find} \quad \mathbf{x} \quad \text{subject to} \quad & d(\mathbf{x}, \mathbf{x}_\text{d}) \leq \delta_\text{s} \,, \\ & D(p_k(\mathbf{x}), P_\text{d}) \leq \delta_\text{c} \,, \end{aligned} \tag{1}$$

where $\mathbf{x}_\text{d}$ is the desired feature vector and P_d the desired PMF. The constants δ_s and δ_c are the dissimilarity and PMF distance upper bounds, respectively.

For the TDHC model, there are two meta classifiers. Therefore, finding an attack is formulated as

$$\begin{aligned} \text{Find} \quad \mathbf{x} \quad \text{subject to} \quad & d(\mathbf{x}, \mathbf{x}_\text{d}) \leq \delta_\text{s} \,, \\ & D(p_r(\mathbf{x}), P_\text{d}^r) \leq \delta_\text{c}^r \,, D(p_f(\mathbf{x}), P_\text{d}^f) \leq \delta_\text{c}^f \,, \end{aligned} \tag{2}$$

where $P_\text{d}^r (P_\text{d}^f)$, and $\delta_\text{c}^r (\delta_\text{c}^f)$ are the coarse (fine) desired PMF, and the corresponding coarse (fine) PMF distance upper bound, respectively.

Confidence Reduction Attacks: Here, the goal is to reduce the classification confidence score of the discriminant functional of the predicted label. Given a desired confidence level c_d, and confidence threshold δ_d, for the FHC model, it is required to synthesize $\mathbf{x}$ such that $K(\mathbf{x}) = K(\mathbf{x}_\text{d})$ and $\max_{l \in [M]} p_k(l|\mathbf{x}) \in \mathbb{V}(c_\text{d}, \delta_\text{d})$, where $\mathbb{V}(c_\text{d}, \delta_\text{d}) := \{c : |c - c_\text{d}| \leq \delta_\text{d}\}$. For this attack, termed the FHC-confidence Reduction attack (FHC-R), the desired PMF P_d can be selected as $P_\text{d}(l) = c_\text{d}$ for $l = K(\mathbf{x}_\text{d})$ and $P_\text{d}(l) = (1 - c_\text{d})/(M - 1)$, for $l \neq K(\mathbf{x}_\text{d})$.

For TDHC, we define desired confidence scores $c_\text{d,r}$ and $c_\text{d,f}$ for the coarse and fine stages, respectively, and corresponding thresholds $\delta_\text{d,r}$ and $\delta_\text{d,f}$. In this case, we seek to generate $\mathbf{x}$ such that $R(\mathbf{x}) = R(\mathbf{x}_\text{d})$, $F(\mathbf{x}, R(\mathbf{x})) = F(\mathbf{x}_\text{d}, R(\mathbf{x}_\text{d}))$, $\max_{l \in [M_\text{c}]} p_r(l|\mathbf{x}) \in \mathbb{V}(c_\text{d,r}, \delta_\text{d,r})$, and $\max_{l \in S_{\text{R}(\text{x})}} p_f(l|\mathbf{x}, R(\mathbf{x})) \in \mathbb{V}(c_\text{d,f}, \delta_\text{d,f})$. The desired PMF can be selected in a similar fashion as the FHC-R attack. We term this attack the TDHC-confidence Reduction attack (TDHC-R).

Targeted Attacks: In the targeted attacks scenario, the goal of the attacker is to alter the classification to a pre-defined target set for the FHC model, or pre-defined target classes for the TDHC model.

For FHC, we seek to synthesize $\mathbf{x}$ such that $K(\mathbf{x}) \in S_t$, where $S_t \subset [M_c]$ is a defined target super-class set. Equivalently, it is required that $\exists j \in S_t : p_k(j|\mathbf{x}) > p_k(l|\mathbf{x}), \forall l \in [M_c] \backslash S_t$. Since $p_k(\cdot|\mathbf{x}_\text{d})$ captures the class membership of the benign example, we define the desired PMF as the indicator function $P_\text{d}(l) = \mathbf{1}\{l = j^*\}$, which takes the value 1 if $l = j^*$ and 0 otherwise, where we obtain j^* as in [5] as the maximizing class label in the target set S_t, i.e., $j^* = \underset{j \in S_t}{\operatorname{argmax}} p_k(j|\mathbf{x}_\text{d})$. We call this the FHC-Targeted attack (FHC-T).

For the TDHC model, given a target coarse label t_c and a target fine label t_f, it is required to generate $\mathbf{x}$ such that $R(\mathbf{x}) = t_\text{c}$ and $F(\mathbf{x}, R(\mathbf{x})) = t_\text{f}$. The desired PMFs are chosen as $P_\text{d}^r(l) = \mathbf{1}\{l = t_\text{c}\}, \forall l \in [M_\text{c}]$ and $P_\text{d}^f(l) = \mathbf{1}\{l = t_\text{f}\}, \forall l \in [M_{t_\text{c}}]$. We term this attack, the TDHC-Targeted attack (TDHC-T).

3. BOSS PROPOSED SOLUTION

In order to satisfy the constraints of the feasibility programs (1) and (2), our approach makes use of a generative network $g : \mathbb{R}^Q \rightarrow \mathbb{R}^N$ with trainable parameters ϕ, such that $g(\mathbf{z}\,; \phi) = \mathbf{x}$, where $\mathbf{z} \in \mathbb{R}^Q$ is a random input. A repeated version of vector $\mathbf{z}$ is used to obtain a small training dataset, then the parameters ϕ are tuned using the back-propagation training algorithm [12]. To this end, we introduce the surrogate loss function $\mathcal{L}_g(g(\mathbf{z}\,; \phi), \mathbf{x}_\text{d})$ to gauge similarity to $\mathbf{x}_\text{d}$. Similarly, we define a surrogate loss

Algorithm 1 BOSS Algorithm for FHC $\big($TDHC$\big)$ model

Input: $\mathbf{z}$, g, $\mathbf{x}_\mathrm{d}$, K $\big(R$ and $F\big)$, P_d $\big(P_\mathrm{d}^r$ and $P_\mathrm{d}^f\big)$, δ_s, δ_c $\big(\delta_\mathrm{c}^r$ and $\delta_\mathrm{c}^f\big)$
Output: $\mathbf{x}$

Initialize $\mathbf{x}$, ϕ, λ $\big(\lambda_r$ and $\lambda_f\big)$
while $d(\mathbf{x}, \mathbf{x}_\mathrm{d}) \geq \delta_\mathrm{s}$ **or** $D(p_k(\mathbf{x}), P_\mathrm{d}^r) \geq \delta_\mathrm{c}$ $\big(D(p_r(\mathbf{x}), P_\mathrm{d}^r) \geq \delta_\mathrm{c}^r$ **or** $D(p_f(\mathbf{x}), P_\mathrm{d}^f) \geq \delta_\mathrm{c}^f\big)$
  obtain ϕ as the minimizer of (3) $\big($(4)$\big)$ with λ $\big(\lambda_r$ and $\lambda_f\big)$
  $\mathbf{x} = g\big(\mathbf{z}, \phi\big)$
  update λ $\big(\lambda_r$ and $\lambda_f\big)$ using (5) $\big($(6)$\big)$
return $\mathbf{x}$

$\mathcal{L}_h(p_k(g(\mathbf{z}\,;\phi)), P_\mathrm{d})$ for the PMF distance for the FHC, and losses $\mathcal{L}_{h,r}(p_r(g(\mathbf{z}\,;\phi)), P_\mathrm{d}^r)$ and $\mathcal{L}_{h,f}(p_f(g(\mathbf{z}\,;\phi)), P_\mathrm{d}^f)$ for the TDHC. We can readily update ϕ iteratively by optimizing (3) for the FHC and (4) for the TDHC.

$$\min_{\phi} \Big[\mathcal{L}_g\Big(g(\mathbf{z}\,;\phi), \mathbf{x}_\mathrm{d}\Big) + \lambda \mathcal{L}_h\Big((p_k(g(\mathbf{z}\,;\phi)), P_\mathrm{d})\Big)\Big]\,. \quad (3)$$

$$\begin{aligned}\min_{\phi} \Big[\mathcal{L}_g\Big(g(\mathbf{z}\,;\phi), \mathbf{x}_\mathrm{d}\Big) + \lambda_r \mathcal{L}_{h,r}\Big((p_r(g(\mathbf{z}\,;\phi)), P_\mathrm{d}^r)\Big) \\ + \lambda_f \mathcal{L}_{h,f}\Big((p_f(g(\mathbf{z}\,;\phi)), P_\mathrm{d}^f)\Big)\Big]\,.\end{aligned} \quad (4)$$

We choose $\mathcal{L}_g$ as the mean square error, and $\mathcal{L}_h$, $\mathcal{L}_{h,r}$, and $\mathcal{L}_{h,f}$ as the categorical cross-entropy loss. The parameters λ, λ_r, and λ_f are used to weigh the relative importance of each loss function [11]. We dynamically update these parameters using the ratio of the desired and induced distances as

$$\lambda \leftarrow \sigma\Big(\lambda - \lambda^0 \frac{\delta_\mathrm{c}}{D} \operatorname{sign}\big(\frac{\delta_\mathrm{c}}{D} - 1\big)\Big)\,, \quad (5)$$

$$\begin{aligned}\lambda_r &\leftarrow \sigma\Big(\lambda_r - \lambda_r^0 \frac{\delta_\mathrm{c}^r}{D_r} \operatorname{sign}\big(\frac{\delta_\mathrm{c}^r}{D_r} - 1\big)\Big)\,, \\ \lambda_f &\leftarrow \sigma\Big(\lambda_f - \lambda_f^0 \frac{\delta_\mathrm{c}^f}{D_f} \operatorname{sign}\big(\frac{\delta_\mathrm{c}^f}{D_f} - 1\big)\Big)\,,\end{aligned} \quad (6)$$

where $D_r = D(p_r(\mathbf{x}), P_\mathrm{d}^r)$ and $D_f = D(p_f(\mathbf{x}), P_\mathrm{d}^f)$, and superscript 0 indicates the initial selection. We use the Jensen-Shannon (JS) divergence as our distance function D, which returns values in $[0, 1]$ [14]. The signum function sign(.) is used to decide whether to increase or decrease the weights based on the ratio of each objective. The ReLU function $\sigma(x) = \max(0, x)$ prevents the weights from becoming negative. We term this approach Bidirectional One-Shot Synthesis (BOSS) and summarize it in Algorithm 1.

We remark that the BOSS approach for the TDHC can be extended to include more than two classifiers, hence it is applicable to nested dichotomies [15] and error-correcting output codes [16], which use a series of binary classifier structures to perform multi-class classification.

Table 1: Confidence reduction attacks.

Model/Attack	CA(%)	μ_C	μ_I	run-time (sec)
FHC-R	100	0.64	0.78	19.47
TDHC-R	100	$\{0.63, 0.68\}$	0.79	20.1

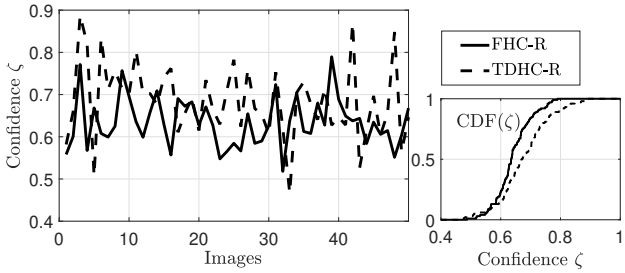

Fig. 1: Confidence level of the FHC-R and TDHC-R represented by the score ζ (*left*) and the CDF(ζ) (*right*).

4. EXPERIMENTAL RESULTS

Setup and used metrics: We demonstrate the performance of the attacks proposed on the task of image classification. To measure similarity between examples, we use $d = 1 - I$, where I is the Structural Similarity Index Measure (SSIM) [17], which accounts for luminance, contrast, and structure in the observations. We use ζ to denote the value of the probabilistic discriminant functional of the predicted label, which represents the classification confidence. We use μ_C and μ_I to denote the average value of the classification confidence score and the SSIM, respectively. To evaluate the performance of targeted attacks, we use the attack success ratio, which we denote by α. The target classes are selected uniformly at random as in the 'average' case scenario in [18]. We use Convolutional Neural Networks (CNNs) with the MNIST fashion dataset [19]. Each sample is a 28×28 gray scale matrix of values in $[0, 1]$ with the fine classes from 0 to 9, representing 'T-shirt/top', 'Trousers', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', and 'Ankle boot', respectively. The coarse sets S_1, S_2, and S_3 are 'top': $\{0, 2, 6\}$, 'bottom': $\{1, 5, 7, 9\}$ and 'other': $\{3, 4, 8\}$. The code is available online along with further details of the experimental setup[1]. For all experiments, the dissimilarity threshold is selected as $\delta_\mathrm{s} = 0.8$ with parameters $\lambda = \lambda_r = \lambda_f = 0.01$. For the confidence reduction attacks, we use $c_\mathrm{d} = 0.6$ and $\delta_\mathrm{c} = 0.2$. For the targeted attacks, we utilize $\delta_\mathrm{c} = 0.35$. We compare against the eADMM approach proposed in [13] with parameters $\epsilon_b = 14$, $\epsilon_c = 13$, and $\epsilon_f = 14$.

Confidence reduction attacks: For our first experiment, we investigate the confidence reduction attacks on the FHC and TDHC models. Fig.1 (*left*) shows the confidence scores of samples from the dataset. Fig.1 (*right*) shows the Cumulative Distribution Function (CDF) of ζ. On average, the values are within a small distance from the desired confidence $c_\mathrm{d} = 0.6$. The overall performance is presented in Table 1 where the

[1] https://github.com/ialkhouri/OneShotSynInHCs

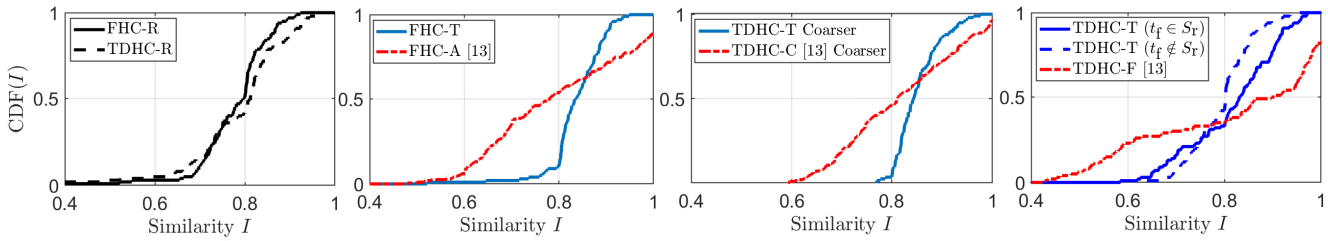

Fig. 2: CDF of the similarity I for the confidence reduction attacks (*first*), FHC targeted attacks (*second*), TDHC coarse level targeted attacks (*third*), and TDHC fine level targeted attacks [13] (*fourth*).

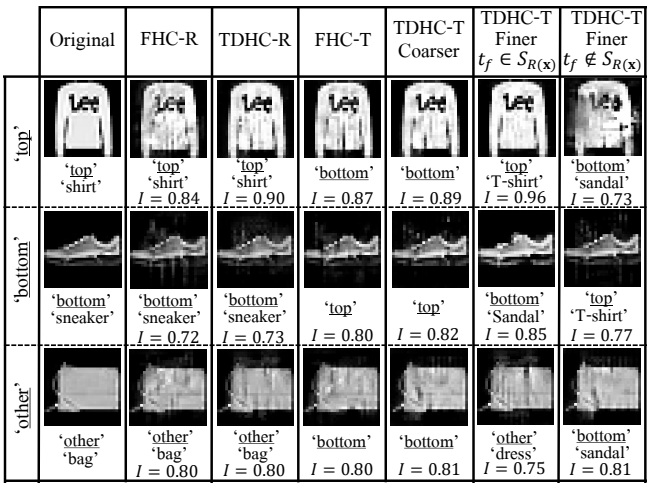

Fig. 3: Samples from each super label (rows). The SSIM and predicted labels are given at the bottom of each perturbed image.

Table 2: Targeted attacks.

Model/Attack	Level	$\alpha(\%)$	μ_I	run-time (sec)
FHC-A [13]	Coarse	87.9	0.78	0.47
FHC-T	Coarse	98	0.84	12.4
TDHC-C [13]	Coarse	95.8	0.82	0.16
TDHC-T	Coarse	99	0.81	10.67
TDHC-F [13]	Fine	87.3	0.8	0.31
TDHC-T ($t_f \in S_{R(\mathbf{x})}$)	Fine	99	0.85	10.75
TDHC-T ($t_f \notin S_{R(\mathbf{x})}$)	Coarse & Fine	98	0.97	33.37

classification accuracy of the FHC-R and TDHC-R remain unchanged. For both attacks, we observe comparable good performance, where $\mu_C \approx 0.6$ (as specified by c_d) and $\mu_I \approx 0.78$. In terms of input similarity, Fig. 2 (*first*) shows the CDF of the SSIM, where high values are observed for both attacks. Samples synthesized by the aforementioned attacks are shown in the second and third columns of Fig. 3.

Targeted coarse level attacks: Table 2 (rows 1 to 4) presents the overall performance of the proposed targeted coarse level attacks on the FHC and TDHC models, and that of the FHC-A and TDHC-C methods from [13]. For both models, our proposed attacks outperform [13] in terms of the attack success ratio, with a difference of roughly 10% (5%) for the FHC (TDHC) model, respectively. In terms of the imperceptibility metric μ_I, our method outperforms [13] for the FHC model and yields similar performance for TDHC ($\mu_I = 0.81$). The higher similarity can also be observed from the CDF(I) curves presented in Fig. 2 (*second*) and (*third*). As observed, for both models, 50% of the samples score similarity values between 0.6 and 0.8 for [13], while nearly 80% of the samples have SSIM values of more than 0.8 for our proposed method. Columns 4 and 5 of Fig. 3 present generated samples by the FHC-T and TDHC-T coarse level attacks.

Targeted fine level attacks: We implement the TDHC-T attack such that the target fine label is inside the same super set of the true fine label, i.e., $t_f \in S_{R(\mathbf{x})}$. The results are presented in rows 5 and 6 of Table 2. As observed, the proposed method outperforms [13] not only in terms of success ratio (captured by α), but also in imperceptibility as indicated by the higher value of μ_I. The larger similarity is also observed in Fig. 2 (*fourth*). A Sample of each super label is presented in the sixth column of Fig. 3.

Targeted coarse and fine level attacks: In this experiment, the goal is to induce misclassifications at both levels of the TDHC model. Thus, the target finer label is selected outside the true super set, i.e. $t_f \notin S_{R(\mathbf{x})}$, and in turn, the coarse label is also altered according to the target class. The high success ratio and similarity, as shown in the last row of Table 2 and Fig. 2 (*fourth*), underscore the versatility of our proposed approach with regard to synthesizing samples that are outside the training set of a given classifier, yet meet desired specifications. For example, the synthesized 'Shirt' looking image presented in the last column of Fig. 3, is classified as 'Sandal' by the 'bottom' fine classifier that has been trained only using images labeled 'Trouser', 'Sandal', 'Sneaker', and 'Ankle boot'. Our proposed methods incur a longer average run-time compared to [13] due to the iterative tuning of the parameters ϕ in Algorithm 1.

5. CONCLUSION

The synthesis of adversarial examples has largely focused on generating imperceptible additive perturbations to alter the predictions of one-stage classifiers. In this paper, we took an altogether different generative approach, in which adversarial examples are synthesized in one-shot, from scratch against the two main types of two-stage hierarchical classifiers. We presented specification selection criteria for confidence reduction and targeted attacks. Our methods were shown to outperform existing approaches to hierarchical settings in terms of success ratio and similarity between the original and adversarial samples.

6. REFERENCES

[1] Fernando Gama, Antonio G. Marques, Geert Leus, and Alejandro Ribeiro, "Convolutional neural network architectures for signals supported on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 1034–1049, 2019.

[2] Yuezhang Che, Yunzhang Zhu, and Xiaotong Shen, "Multilabel classification with multivariate time series predictors," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5696–5705, 2020.

[3] Yan Zhou, Murat Kantarcioglu, and Bowei Xi, "A survey of game theoretic approach for adversarial machine learning," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 3, pp. e1259, 2019.

[4] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346–360, 2020.

[5] I Alkhouri and G Atia, "Adversarial attacks on coarse-to-fine classifiers," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE Signal Processing Society, 2021.

[6] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami, "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016, pp. 372–387.

[7] David Stutz, Matthias Hein, and Bernt Schiele, "Confidence-calibrated adversarial training: Generalizing to unseen attacks," in *Proceedings of the 37th International Conference on Machine Learning*. 13–18 Jul 2020, vol. 119 of *Proceedings of Machine Learning Research*, pp. 9155–9166, PMLR.

[8] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[9] Xuanqing Liu and Cho-Jui Hsieh, "Rob-gan: Generator, discriminator, and adversarial attacker," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11234–11243.

[10] Carlos N Silla and Alex A Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 31–72, 2011.

[11] Ismail Alkhouri, Alvaro Velasquez, and George Atia, "Boss: Bidirectional one-shot synthesis of adversarial examples," *arXiv preprint arXiv:2108.02756*, 2021.

[12] Martin Riedmiller and Heinrich Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," in *IEEE international conference on neural networks*. IEEE, 1993, pp. 586–591.

[13] I Alkhouri and G Atia, "Targeted attacks in hierarchical settings via convex programming," in *International Joint Conference on Neural Networks (IJCNN)*. International Neural Network Society, 2021.

[14] Jianhua Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.

[15] Vitalik Melnikov and Eyke Hüllermeier, "On the effectiveness of heuristics for learning nested dichotomies: an empirical analysis," *Machine Learning*, vol. 107, no. 8, pp. 1537–1560, 2018.

[16] Thomas G Dietterich and Ghulum Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of artificial intelligence research*, vol. 2, pp. 263–286, 1994.

[17] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[18] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 39–57.

[19] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song, "Spatially transformed adversarial examples," *arXiv preprint arXiv:1801.02612*, 2018.