Information Sciences 569 (2021) 469–478

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

BBAS: Towards large scale effective ensemble adversarial attacks against deep neural network learning

Jialie Shen *, Neil Robertson

School of EEECS, Queen's University Belfast, Northern Ireland, BT9 5AF, UK

ARTICLE INFO

Article history:
Received 15 July 2020
Received in revised form 24 September 2020
Accepted 12 November 2020
Available online 25 December 2020

Keywords:
Black-box attack
Adversarial
Robustness
Boosting

ABSTRACT

Recent decades have witnessed rapid development of deep neural networks (DNN). As DNN learning is becoming more and more important to numerous intelligent system, ranging from self driving car to video surveillance system, significant research efforts have been devoted to explore how to improve DNN model's robustness and reliability against adversarial example attacks. Distinguish from previous study, we address the problem of adversarial training with ensemble based approach and propose a novel boosting based black-box attack scheme call BBAS to facilitate high diverse adversarial example generation. BBAS not only separates example generation from the settings of the trained model but also enhance the diversity of perturbation over class distribution through seamless integration of stratified sampling and ensemble adversarial training. This leads to reliable and effective training example selection. To validate and evaluate the scheme from different perspectives, a set of comprehensive tests have been carried out based on two large open data sets. Experimental results demonstrate the superiority of our method in terms of effectiveness.

1. Introduction

Recent decades have witnessed rapid developments of deep learning techniques, which have demonstrated promising performance on various media analytics applications [1–4]. In particular, as deep learning emerges as core techniques for many intelligent system, significant amount of efforts from security and machine learning communities have been devoted into exploring how to improve DNN model's robustness against adversarial example attacks. The relevant dangerous real cases include automated drive vehicle behavior disordering or mistakenly recognizing malicious contents as legitimate ones [5,6]. The attacks are often started with adversarial example generation: legitimate inputs changed with a minor and highly imperceptible perturbations to mislead machine learning classifier to generate misclassified results. However, human still can correctly classify the adversarial inputs. Fig. 1 demonstrates an excellent example [7,8]. As shown, both images is the same in human eyes: each of them is identified by our visual system as stop sign and left image is an ordinary stop sign image. According to [7], the image on the right-hand side is the ones with a precise perturbation, which can make a particular DNN to misclassify it as a yield sign via hard training process. This can be used as an attack which needs to modify the image to cause dangerous and confusing behavior of self-driving car [9].

* Corresponding author at: 18 Malone Road, Belfast BT9 5AF, Northern Ireland, UK.
E-mail addresses: j.shen@qub.ac.uk (J. Shen), n.robertson@qub.ac.uk (N. Robertson).

https://doi.org/10.1016/j.ins.2020.11.026
0020-0255/

Fig. 1. A good example to demonstrate potential attack by adding precise perturbation.

Generally, the attacks can be categorized two independent classes: white-box and black-box. The white-box attack is assumed to have explicit knowledge about target model or/and its related training data. However, for many real applications or cases, it may not be easy or feasible to acquire that information or/and knowledge (e.g., key parameters about target models). As such, black-box attacks become popular and practical approach for adversaries in real world. And the adversary explored in this study is assumed to have no,

- preknowledge and information about the DNN settings (e.g., structure or parameters), and
- access to large training data or related training algorithms.In this case, the target model is just a black-box to attacker and extracting details about DNN learning model's configuration (e.g., structure or parameters or training data) becomes very hard and make effective adversarial example development extremely difficult. However, it has been shown that strong transferability of the adversarial examples exists between different learning models [10–13]. This suggests the black-box attacks can be carried out with the property. The query to target system is issued by the attacker, who develops substitute model according to the query results. Then, the adversarial examples for the substitute model can be generated and could be transferred to target system to make it behavior disordered or abnormal. Based on [14], basic idea for most of the existing black-box attack strategies is to train a single substitute model to generate adversarial examples with a weak transfer capability. This suggests lower defense difficulty with existing scheme. Recent studies show that when application task scale increases, chance of failed defense will be significantly [15]. For example, it has been proven unsuccessful if adversarial training scales up to ImageNet-scale tasks [16].

Motivated by the key observation above, this research addresses the problem of adversarial training with ensemble-based approach [17–19]. A novel boosting based black-box attack scheme called BBAS is proposed to facilitate high diverse adversarial example generation. BBAS not only separates example generation from the settings of the trained model but also enhance the diversity of perturbation over class distribution during training. Further, BBAS seamlessly integrate stratified sampling [20,21] and ensemble adversarial training to achieve reliable and comprehensive training example selection. To validate the superiority of the proposed BBAS, we have applied this framework to two different large datasets (MNIST dataset and GTSRB dataset) and carried out a comprehensive experimental study over different performance evaluation metrics with various DNN learning settings. The comparative analysis of various methods for this study indicates that our approach achieves promising performance over different perspectives.

Table 1
Summary of symbols and definitions.

Symbols	Definitions
x	Legitimate example
x^*	Adversarial example
α	Perturbation magnitude
δ	Minimal perturbation
P	Population
N	Population size
L	Number of layer in BASS
C	Class number in dataset
M	Number of temporary model
Ω	Projection function used in PGD
$II()$	Indication function
TM	Target model
SM	Substitute model
SR	Success ratio
$Loss$	Loss function
$Var(\hat{\mu})$	Variance for estimator $\hat{\mu}$
w_h	Weight for stratum h
N_h	Size of population in stratum h

The rest of the paper is organized as follows. Section 2 gives a brief literature review in the related areas. In Section 3, we review our proposed BBAS scheme, giving the detailed structure of its component modules and its learning algorithms. Section 4 reports our experimental configuration and Section 5 introduces and analyzes experimental results. Finally, in Section 6, we conclude the article with key results and findings discussion and directions of future work (Table 1).

2. Related work

Our research draws from multiple streams of the related work. In Section 2.1, we introduce existing work in black-box attack especially with focus on the work related to adversarial training. Section 2.2 gives a review on stratified sampling.

2.1. Adversarial training

Basic idea of adversarial training is to add adversarial examples into the training dataset during DNN learning. It has been proven a successful method to construct robust and reliable model against attacks [22–25]. While the approach demonstrates promising performance when applied to white-box attacks, black-box context often reduces its robustness significantly. In fact, they become extremely vulnerable to adversarial attacks and this can go worse when application scale goes up significantly.

Earliest work on adversarial training can be traced back to [26]. In the work, learning models were restructured via generating adversarial examples and integrating them into training data. There is strong correlation between robustness gained by adversarial training and strength and number of the adversarial examples used. As one of the most popular methods, gradient based adversarial attack is largely based on a simple idea, which originates from back-propagation learning. FGSM [26] is one of the most well-known and popular gradient based adversarial example generation methods. Its key advantage is superior efficiency and high simplicity. For a substitute model *SM* with loss function *Loss*, the adversarial example

$$x^* = x + \delta$$

can be generated for a legitimate example x by computing the following perturbation,

$$x^* = x + \delta = x + \epsilon.sign\left(\nabla^{Loss,t}(x)\right), \tag{1}$$

F is an object function and t is a target class. After each iteration, the gradient is updated with δ based on the x. ϵ is set to be sufficiently small to enable example changes imperceptible to human eyes. The second method we consider is I + FGSM (Iterative + FGSM), which is a FGSM extension. Comparing to FGSM, I-FGSM achieves better effectiveness and efficiency. For each iteration, it only updates δ with smaller amount α, which can be changed based on the different assumption. One popular setting for α is

$$\alpha = \epsilon/steps$$

. Thus, it is eventually clipped with function $cl()$ and the same ϵ value,

$$x_{i^*} = x_{i-1^*} - cl_\epsilon\left(\alpha.sign\left(\nabla^{Loss_{F,t}\left(x_{x-1^*}\right)}\right)\right) \tag{2}$$

In order to gain comprehensive capability, R + FGSM (Random + FGSM) applies small random Gaussian noise to input images before generating adversarial sample using FGSM method. According to [27], this can significantly improve the transferability in black-box scenario.

$$x^\blacklozenge = x + \alpha.sign\left(\mathcal{N}\left(0^d, I^d\right)\right) \tag{3}$$

$$x^* = x^\blacklozenge + (\epsilon - \alpha).sign\left(\nabla_{x^\blacklozenge} Loss\left(f\left(x^\blacklozenge;\theta\right), y_{true}\right)\right) \tag{4}$$

PGD - Projected gradient descent is an iterative based algorithm to support adversarial sample generation. Different from the previous similar approaches, its perturbation is initialized with a random point with the range of L_p ball. Then re-projection is applied based on I-FGSM. The basic idea of the algorithm can be described as follows,

$$x^*_{i+1} = \Omega\left(x^*_i + \frac{\epsilon}{\alpha}.sign\left(\nabla_{x^\blacklozenge} lf\left(f\left(x^\blacklozenge;\theta\right), y_{true}\right)\right)\right) \tag{5}$$

x^*_0 ($i = 0$) is initial input x. The function *sign* is similar to the ones used in FGSM, I + FGSM and R + FGSM. $\Omega()$ is a projection function to guarantee the input's each dimension can be projected into a valid range and α is tunable parameter for adjusting perturbation magnitude.

2.2. Stratified sampling

As a classical survey sampling technique, stratified sampling is widely applied to estimate population parameters efficiently when substantial diversity and difference exists among sub-populations. In this section, we give a brief introduction

to stratified sampling and how to apply it to estimate an expected value (e.g., mean of the population). The basic procedure of stratified sampling estimation can be divided into two steps,

- At the beginning, a limited population P having N elements is partitioned into L mutually exclusive sub-populations - strata. The sizes of sub-populations are $N_1, N_2, \ldots$, and N_L.
- Once the division is completed, certain parameters (e.g., the population mean) can be obtained based on samples drawn from each stratum h with size N_h independently. Using a weighted average of the stratum estimates, parameters for the whole population can be estimated based on a weighted average of stratum and weight for stratum h is calculated with $w_h = \frac{N_h}{N}$, where $\sum_{h=1}^{L} w_h = 1$.

To estimate sample mean $\bar{x}$, an estimator $\bar{x}_{st}$ based on stratified sampling can be expressed as:

$$\bar{x}_{st} = \frac{1}{N}\sum_{h=1}^{L} N_h \bar{x}_h = \sum_{h=1}^{L} w_h \bar{x}_h \tag{6}$$

where $\bar{x}_h$ is the mean derived from stratum h with $\bar{x}_h = \frac{1}{n_h}\sum_{i=1}^{n_h} x_{(h,i)}$. As applied to estimation, there are two basic preconditions to ensure successful stratified sampling: 1) independence - all strata must be independent but don't need to follow the same distribution; and 2) homogeneity: sample variance within strata is significantly less than the variance between different strata.

Based on [28], stratified sampling offers the following advantages on estimation over simple random sampling (SRS): 1) accurate estimated results, 2) better coverage of the population, 3) more efficient to obtain estimates. Point 1) implies that for evaluation, a significantly increased precision of performance assessment) is obtained. Point 2) suggest that small size populations will not be ignored, as they might under SRS; to overcome this under SRS requires the use of larger samples. Existing adversarial training methods apply random sampling, which might not cover all subgroups or subcategories. This can lead to less effective training. BBSA gains comprehensive capability to achieve good quality training example via stratified sampling. Thus, its performance can be enhanced greatly.

3. Boosting based black-box attack scheme

In this section, we introduce details about proposed boosting based black-box attack scheme (BBAS).

3.1. Preliminary

In real applications, the black-box attack strategy consists of two main components as follows,

- Step 1 – Training of Substitute Model: Attacker issues queries over target model *TM* with synthetic inputs selected by a Jacobian-based heuristic to build a model *M* approximating the target model *TM*.
- Step 2 – Adversarial Example Generation: Attacker uses substitute model *SM* to produce adversarial samples, which are misclassified by target model *TM* due to the transferability of adversarial samples. In this study, we apply gradient-based algorithm to generate adversarial example for a substitute model. Details about the algorithms are presented in the following section.

Generally, core objective of adversary is to generate altered version of input x with minimal changes, which is called adversarial example x^*. Target model *TM* misclassifies the example and this misclassification is achieved with adding a minimal perturbation δ.

3.2. Ensemble based attack model – a boosting approach

BBAS is designed based on boosting structure to achieve effective ensemble learning. Fig. 2 illustrates its basic structure, which consists of L layers. In each layer, detail architecture of substitute model selection is shown in Fig. 3. In order to achieve good transferability – ability of an attack against a machine-learning model to be effective against a different model [1], M heterogeneous substitute models based on stratified data samples (training examples) built using input–output pairs from target system and their augmentation with Jacobian-based technique. The approach based on stratified sampling can largely improve performance and is developed based on key observation that images can be generally clustered into various categories using certain similarity criteria. This can lead to superior feasibility of enhancing effectiveness and robustness of the training process with stratification since ignorant clusters with small number of objects might lead to less comprehensiveness or mismatching. Then one of four adversarial example generation algorithms mentioned in Section 2.1 is applied to generate adversarial examples.

[1] Potentially it can be unknown

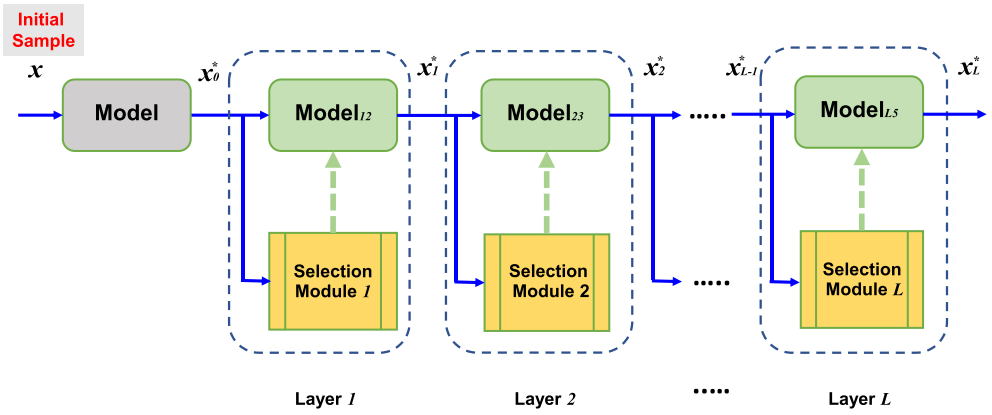

Fig. 2. BBAS – boosting based black-box attack scheme.

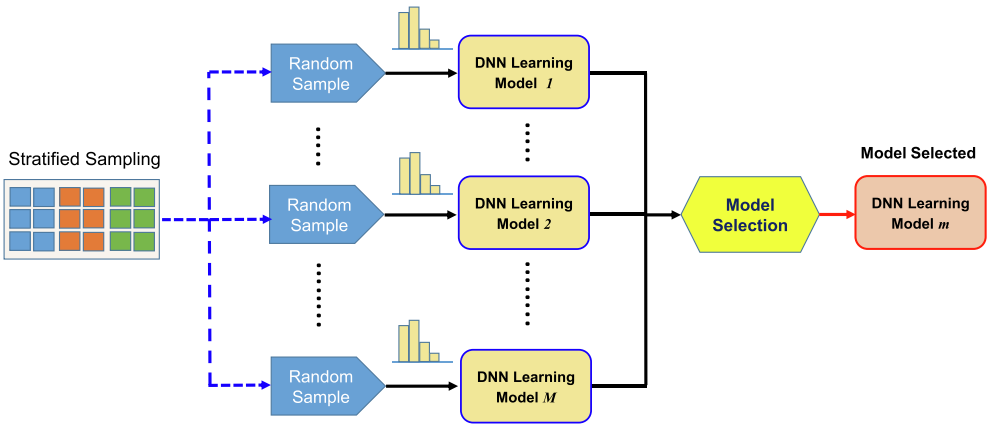

Fig. 3. Substitute model selection at each layer.

BBAS initialization starts with selection of a substitute model to output the adversarial example x_0^* and x_0^* serves as the input of BBAS's first layer. To generate effective gradient based adversarial examples, four popular algorithms applied in our study to our study include FGSM [26], I + FGSM [15], R + FGSM [27] and PGD [29]. The first layer picks a substitute model using model selection. For example, Model_{12} is selected for Layer 1 and then the substitute model will produce adversarial example x_1^* based on one of four adversarial example generation algorithms. x_1^* is input to the Layer 2. With similar structure, to facilitate substitute model selection, input to Layer l ($l \in \{1, 2, \ldots, L\}$ is the adversarial example x_{l1}^* generated by Layer $l-1$. The selection process is repeated L time and outputs x_L^* at the Layer L - final adversarial example.

BBAS applies example success ratio SR – ratio of adversarial examples unsuccessfully attacking model to choose substitute model at each Layer. For Layer l, the success ratio can be defined as,

$$SR = \sum_{i=1}^{IN} \frac{II\left(SM\left(x_{l-l,i}^*\right) = t_i, O(x_i) = t_i\right)}{IN} \tag{7}$$

$II()$ is an indication function, where $II(.) = 1$ denote adversarial examples misclassified by substitute model and target model. IN represents total inputs number to BBAS and number adversarial example generated at Layer $l-1$. The success ratio can be defined as the number of adversarial examples that can be misclassified to the target class both by the substitute model SM and the target model O divided by the total input example number. x_i is the legitimate example i and $x_{l-1,i}^*$ denotes the adversarial example for x_i generated by Layer $l1$. The model with the best success ratio which indicates the best performance is selected by substitute model. In the mean time, as shown in Fig. 3, multiple adversarial examples are used as input at Layer l and represented with x_{l-1}^*. After receiving the inputs, model m with the best success ratio among all the M models is chosen as the output of substitute model selection process. Main objective of substitute model selection is to support iterative adversarial example geneation with robust and better capability to enhance final examples' transferability and other performance.

4. Experimental configuration

We conduct a series of experiments to study the performance of the proposed BBSA scheme. In this study, Tensorflow, Cleverhans library and Python are used for system implementation [30]. To demonstrate the BBSA's promising performance, our experimental study provides systematic test and comparison on various popular gradient-based adversarial example generation algorithms with the BBSA scheme. The generation algorithms considered include FGSM, R + FGSM, I + FGSM and PGD.

4.1. Test data and test DNN structure

In our experiments, two test collections including MNIST dataset and GTSRB dataset are used for performance evaluation.

- MNIST dataset: It is one of the most well-know large database containing handwritten digit images with labels as 0–9 digits (10 classes). It has been widely used for developing and testing various image retrieval/recognition and machine learning systems. MNIST contains 60,000 training images and 10,000 testing images.
- GTSRB dataset: It is a publicly available dataset containing more than 50,000 images of German road traffic signs, belonging to 43 classes[8]. Fig. 4 shows a few good random examples. Given that raw data is RGB-encoded and varies in size, all the images is re-sized to 32*times*32 pixels for simplification purpose. In the mean time, the raw images are also recentterred and rescaled.

The details about two datasets are shown in Table 2. In order to achieve the effects of the real environment, the attacker is assumed to be not able to access the training/testing data and network structure details of the black-box model used. Table 3 shows general architecture of CNN used in our empirical study. The first and second column refer to the learning architecture. The first and last row specify the respectively indicate the input and output configuration of the model. Notice that both layer No. 11 and layer No. 12 are fully connected and both dimensionality is set to be either 43 or 10, depending on total class number in training examples. In addition, 6 substitute models are used for testing and Table 4 shows detail architecture.

Fig. 4. Random representatives of the 43 traffic sign classes in the GTSRB dataset, adapted from [8].

Table 2
Summary of two test collections.

Name	Training data	Testing data	Total	Class size
MNIST	50,000	10,000	60,000	10
GTSRB	39,210	12,630	51,840	43

Table 3
Summary of test DNN structure.

Layer No.	Layer type	Number of feature map	Neuron number	Convolution kernel pool size
0	Input		32 × 32	
1	Convolution	32	30 × 30	3 × 3
2	Convolution	32	28 × 28	3 × 3
3	Max pooling	32	14 × 14	2 × 2
4	Convolution	64	12 × 12	3 × 3
5	Convolution	64	10 × 10	3 × 3
6	Max Pooling	64	5 × 5	2 × 2
7	Convolution	128	3 × 3	3 × 3
8	Convolution	128	1 × 1	3 × 3
9	Max pooling	128	1 × 1	2 × 2
10	Fully connected	512		
11	Fully connected	43 (or 10)		
12	Output	43 (or 10)		

Table 4
Summary of DNN structure used in test for substitute training (Total 6 substitute models considered).

Substitute model		
Substitute Model 1	Substitute Model 2	Substitute Model 3
Dropout(0.2)	Conv(64,3,3)+Relu	Conv(64,3,3)+Relu
Conv(64,8,8)+Relu	Conv(64,3,3)+Relu	Conv(64,3,3)+Relu
Conv(l28,6,6)+Relu	Dropout(0.3)	MP(2,2)
Conv(l28,5,5)+Relu	FC(128)+Relu	Conv(128,3,3)+Relu
Dropout(0.5)	Dropout(0.25)	Dropout(0.3)
FC + Softmax	FC + Softmax	FC(128)+Relu
		FC + Softmax
Substitute model		
Substitute Model 4	Substitute Model 5	Substitute Model 6
Conv(64,3,3)+Relu	Conv(64,3,3)+Relu	FC(300)+Relu
Conv(64,3,3)+Relu	Conv(64,3,3)+Relu	Dropout(0.25)
MP(2,2)	MP(2,2)	FC(300)+Relu
Conv(128,3,3)+Relu	Conv(128,3,3)+Relu	Dropout(0.25)
Conv(128,3,3)+Relu	Conv(128,3,3)+Relu	FC(300)+Relu
Mp(2,2)	MP(2,2)	Dropout(0.25)
FC(256)+Relu	FC(200)+Relu	FC(300)+Relu
FC(256)+Relu	FC(200)+Relu	Dropout(0.25)
FC + Softmax	FC + Softmax	FC + Softmax

4.2. Evaluation metric

Evaluation metric plays an important role in experimental result comparison and analysis. In this study, we apply success rate and transferability as main evaluation metrics to quantify and compare experimental results [7]. The success rate is the proportion of adversarial samples misclassified by deep learning methods. Main goal of our test is to verify sample misclassification by target model. And transferability refers to the target model's misclassification rate of adversarial samples generated using substitute model.

5. Experimental results

This section presents an experimental study to evaluate the proposed BBAS method from two main perspectives including attack performance and transferability performance.

5.1. On attack performance

Vulnerability is an important performance indicator to measure robustness of DNN model under various attacks. In the first test, we present experimental results - success rate and transfer rate in Table 5. We can find the detail performance of Model 1–6 from row 1 to row 6 for single substitute and BBAS (M = 20) based on FGSM. BBAS (M = 20) ensembles 20 substitute models and basic model used is the substitute model 1 introduced in Table 4, to generate transferable adversarial examples.

The results clearly show that higher accuracy the black-box models achieve, better susceptible to adversarial examples they are. This is more obvious for those examples generated by BBAS (M = 20). Another key observation is that crafted adversarial examples with higher success rate achieve better transfer rate. With combination of boosting and stratified sampling,

Table 5
Success rate and transfer rate comparison of adversarial examples generated by single substitute based on FGSM (perturbation magnitude $\alpha = 0.1$), basic model used for BBAS = Substitute Model 1.

Model	MNIST		GTSRB	
Configuration	Success rate	Transfer rate	Success rate	Transfer rate
Model 1	76.56%	2.09%	87.56%	16.95%
Model 2	74.91%	2.01%	90.63%	17.85%
Model 3	70.05%	1.41%	92.07%	18.20%
Model 4	77.27%	2.56%	89.15%	17.67%
Model 5	71.56%	1.71%	83.89%	12.20%
Model 6	72.67%	1.85%	85.32%	12.48%
BBAS ($M = 20$)	**84.12%**	**4.05%**	**98.97%**	**29.21%**

BBAS ($M = 20$) shows an highly effective approach to enhance success rate to produce good quality adversarial examples with strong transferability (4.05% for MNIST and 29.21% for GTSRB).

In the second test, we empirically compare the BBAS with the classic gradient-based attack algorithms under various perturbation magnitude settings (α ranging from 0 to 0.3) over two different datasets. The transferability of adversarial examples generated with different approaches are illustrated in Figs. 5 and 6. It is observed that transfer rate is increased for all the methods tested when perturbation magnitude grows. BBAS demonstrates superior attack performance among all the methods.

5.2. Transferability performance

In this section, we present experimental results and analysis for transferability performance. Transferability refers to the property where an attack developed for a particular machine learning model is also effective against the target model. The following study explores the effects of several possible factors enabling higher transfer rate of BBAS scheme. The main question addressed here is about how DNN and stratified re-sampling settings influence transferability changes. Further, four strategies (S1, S2, S3 and S4) to construct the substitute models are also developed to support diversity analysis. Details is shown as below,

- S1: Basic DNN model is the substitute model 1 introduced in Table 4 for all M substitute models trained with the same set of training data. M is set to be 20 and 40. Sampling scheme is random sampling.
- S2: Basic DNN model is the substitute model 1 introduced in Table 4 for all M substitute models trained with the M different sets of synthetic training data individually. M is set to be 20 and 40. Sampling scheme is random sampling.
- S3: All details is same as S1 except that sampling scheme is stratified sampling.
- S4: All details is same as S2 except that sampling scheme is stratified sampling.

Table 6 illustrates detail experimental results based on four strategies to construct M substitute models. The first key finding is that that greater diversity substitute models have, stronger transferability adversarial examples enjoy. It is clear that

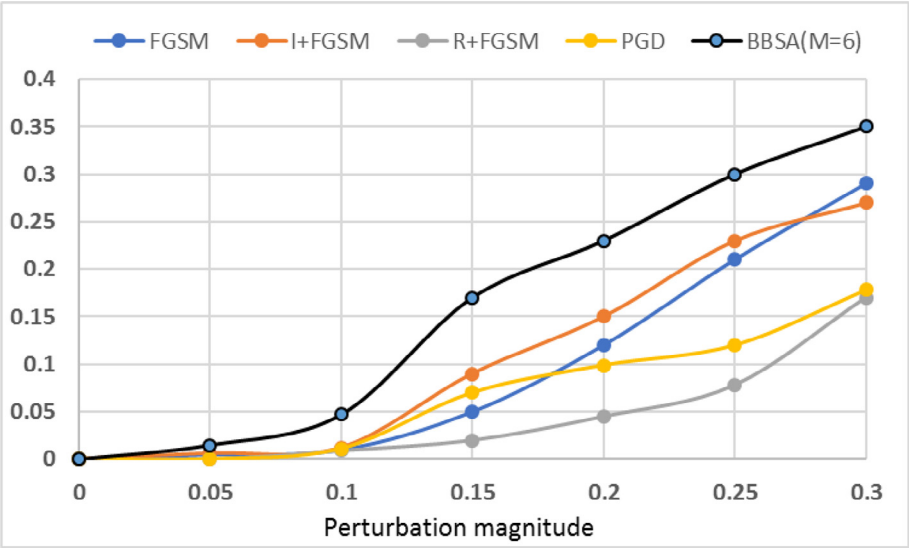

Fig. 5. Transfer rate of adversarial examples generated by BBAS and other schemes under different perturbation magnitude on MNIST.

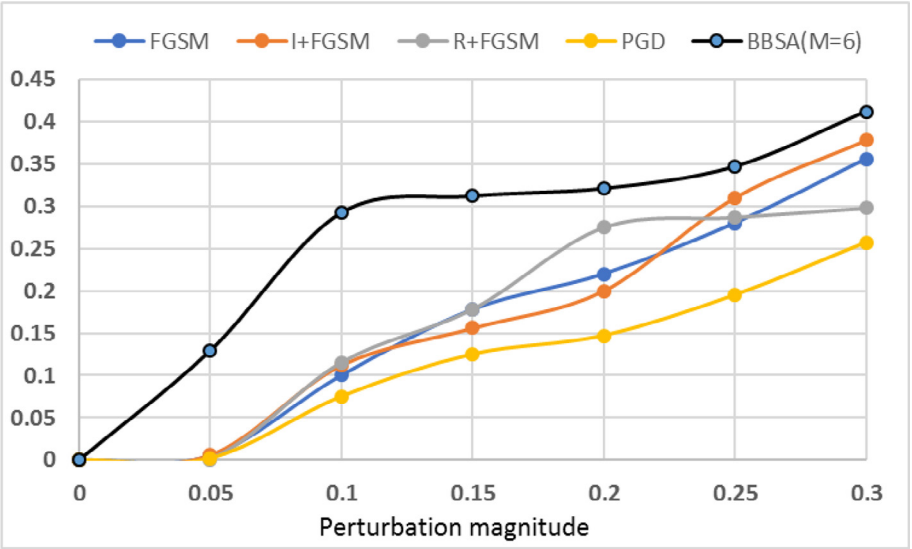

Fig. 6. Transfer rate of adversarial examples generated by BBAS and other schemes under different perturbation magnitude on GTSRB.

Table 6
BBAS's transfer rate comparison of strategies to construct substitute model based on FGSM (perturbation magnitude α = 0.1), basic model used for BBAS = Substitute Model 1.

	MNIST		GTSRB	
Strategy	M = 20	M = 40	M = 20	M = 40
S1	3.56%	4.25%	26.56%	27.95%
S2	3.82%	4.56%	27.63%	28.85%
S3	4.05%	5.01%	29.21%	30.71%
S4	6.18%	7.56%	31.15%	37.67%

the substitute model number plays an important role in transferability improvement. General observation is that bigger M value of BBAS scheme leads to higher transfer rate. For example, with same sampling strategy, BBAS (M = 40) achieves average 21.3% and 19.5% transfer rate gain over BBAS(M = 20) for MNIST and GTSRB respectively. Meanwhile, stratified re-sampling also gives an excellent lift on performance in terms of transfer rate improvement. In average, 20.5% and 23.4% increase is achieved if stratified re-sampling is applied with same other configuration. We believe that main reason behind this is that with incorporating stratification, the sample extracted enjoys better capability to reflect and cover various data characteristics which leads to promising and reliable performance.

6. Conclusion and future work

In recent years, the emergence and maturity of deep neural network technologies have enable fast development of various kinds of intelligent systems. Recent studies reveal that DNN can be very vulnerable to adversarial examples. In this study, we propose a novel boosting based black-box attack scheme called BBAS to facilitate high diverse adversarial example generation. With boosting, BBAS leverages ensemble learning to iteratively construct adversarial examples. Further, BBAS not only can effectively separate example generation from the settings of the trained model but also lift the diversity of perturbation over class distribution through seamless integration of stratified sampling and ensemble adversarial training. This leads to reliable and effective training example selection. Experimental results from a set of comprehensive tests based on two large open data-sets demonstrate the superiority of our scheme from various perspectives. This study opens up dozen of promising directions for future study. An immediate next step is to extend current experiments to include other kinds of deep learning architecture and investigate performance issues. Furthermore, another interesting direction is to adapt the current framework to handle audio and speech analysis related problems.

CRediT authorship contribution statement

Jialie Shen: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Neil Robertson:** Investigation, Writing - original draft, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L. Wang, X. Qian, Y. Zhang, J. Shen, X. Cao, Enhancing sketch-based image retrieval by CNN semantic re-ranking, IEEE Trans. Cybern. 50 (7) (2020) 3330–3342.
[2] J. Shen, T. Mei, Q. Qu, D. Tao, Y. Rui, Toward efficient indexingac structure for scalable content-based music retrieval, Multimedia Syst. 25 (6) (2019) 639–653.
[3] G. Wu, Z. Lin, J. Han, L. Liu, G. Ding, B. Zhang, J. Shen, Unsupervised deep hashing via binary latent factor models for large-scale cross-modal retrieval, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 2854–2860.
[4] Q. Wang, P. Jiang, Z. Guo, Y. Han, Z. Zhao, Multi-speaker video dialog with frame-level temporal localization, in: Procceding of AAAI Conference, 2020, pp. 12200–12207..
[5] Y. Shi, Y. Han, Q. Zhang, X. Kuang, Adaptive iterative attack towards explainable adversarial robustness, Pattern Recogn. 105 (2020) 107309.
[6] F. Guo, Q. Zhao, X. Li, X. Kuang, J. Zhang, Y. Han, Y. Tan, Detecting adversarial examples via prediction difference for deep neural networks, Inf. Sci. 501 (2019) 182–192.
[7] N. Papernot, P.D. McDaniel, I.J. Goodfellow, S. Jha, Z.B. Celik, A. Swami, Practical black-box attacks against machine learning, in: Proceedings of AsiaCCS, 2017, pp. 506–519..
[8] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition, Neural Networks 32 (2012) 323–332.
[9] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial examples in the physical world, CoRR abs/1607.02533, http://arxiv.org/abs/1607.02533..
[10] X. Huang, Y. Li, O. Poursaeed, J.E. Hopcroft, S.J. Belongie, Stacked generative adversarial networks, Proceedings of IEEE CVPR (2017), pp. 1866–1875.
[11] C. Xiao, B. Li, J. Zhu, W. He, M. Liu, D. Song, Generating adversarial examples with adversarial networks, in: Proceedings of IJCAI, 2018, pp. 3905–3911..
[12] Y. Li, S. Bai, Y. Zhou, C. Xie, Z. Zhang, A. Yuille, Learning transferable adversarial examples via ghost networks, Proceedings of AAAI (2020).
[13] F. Tramèr, N. Carlini, W. Brendel, A. Madry, On adaptive attacks to adversarial example defenses, Proceedings of NeurIPS (2020).
[14] N. Akhtar, A.S. Mian, Threat of adversarial attacks on deep learning in computer vision: a survey, IEEE Access 6 (2018) 14410–14430.
[15] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial machine learning at scale, Proceedings of ICLR (2017).
[16] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, ImageNet, A large-scale hierarchical image database, Proceedings of IEEE CVPR (2009).
[17] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Mach. Learn. 51 (2) (2003) 181–207.
[18] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, USA, ISBN 0471210781, 2004..
[19] W. Wei, L. Liu, Robust Deep Learning Ensemble against Deception, IEEE Transactions on Dependable and Secure Computing (Early Access)..
[20] C.E. Sarndal, B. Swensson, J. Wretma, Model Assisted Survey Sampling, Springer-Verlag, 1992.
[21] J. Shen, J. Shepherd, Efficient benchmarking of content-based image retrieval via resampling, in: Proceedings of ACM Multimedia, 2006, ACM, pp. 569–578..
[22] U. Shaham, Y. Yamada, S. Negahban, Understanding adversarial training: Increasing local stability of supervised models through robust optimization, Neurocomputing 307 (2018) 195–204.
[23] F. Tramèr, D. Boneh, Adversarial training and robustness for multiple perturbations, in: Proceedings of NeurIPS, 2019, pp. 5858–5868..
[24] P. Maini, E. Wong, J.Z. Kolter, Adversarial robustness against the union of multiple perturbation models, in: Proceedings of ICML, 2020, pp. 6640–6650..
[25] D. Stutz, M. Hein, B. Schiele, Confidence-calibrated adversarial training: generalizing to unseen attacks, in: Proceedings of ICML, 2020, pp. 9155–9166..
[26] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, Proceedings of ICLR (2015).
[27] F. Tramèr, A. Kurakin, N. Papernot, I.J. Goodfellow, D. Boneh, P.D. McDaniel, Ensemble adversarial training: attacks and defenses, Proceedings of ICLR (2018).
[28] W.G. Cochran, Sampling Techniques, John Wiley and Sons, 1977.
[29] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, Proceedings of ICLR (2018).
[30] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow: a system for large-scale machine learning, in: Proceeding of 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 265–283..