EREBA: Black-box Energy Testing of Adaptive Neural Networks

Mirazul Haque*
mirazul.haque@utdallas.edu
The University of Texas at Dallas

Yaswanth Yadlapalli*
yaswanth.yadlapalli@utdallas.edu
The University of Texas at Dallas

Wei Yang
wei.yang@utdallas.edu
The University of Texas at Dallas

Cong Liu
cong@utdallas.edu
The University of Texas at Dallas

ABSTRACT

Recently, various Deep Neural Network (DNN) models have been proposed for environments like embedded systems with stringent energy constraints. The fundamental problem of determining the robustness of a DNN with respect to its energy consumption (energy robustness) is relatively unexplored compared to accuracy-based robustness. This work investigates the energy robustness of Adaptive Neural Networks (AdNNs), a type of energy-saving DNNs proposed for many energy-sensitive domains and have recently gained traction. We propose EREBA, the first black-box testing method for determining the energy robustness of an AdNN. EREBA explores and infers the relationship between inputs and the energy consumption of AdNNs to generate energy surging samples. Extensive implementation and evaluation using three state-of-the-art AdNNs demonstrate that test inputs generated by EREBA could degrade the performance of the system substantially. The test inputs generated by EREBA can increase the energy consumption of AdNNs by 2,000% compared to the original inputs. Our results also show that test inputs generated via EREBA are valuable in detecting energy surging inputs.

CCS CONCEPTS

• **Security and privacy → Software and application security**.

KEYWORDS

Green AI, AI Energy Testing, Adversarial Machine Learning

ACM Reference Format:
Mirazul Haque, Yaswanth Yadlapalli, Wei Yang, and Cong Liu. 2022. EREBA: Black-box Energy Testing of Adaptive Neural Networks. In *44th International Conference on Software Engineering (ICSE '22), May 21–29, 2022, Pittsburgh, PA, USA*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3510003.3510088

*Both authors contributed equally to this research.

ICSE '22, May 21–29, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9221-1/22/05…$15.00
https://doi.org/10.1145/3510003.3510088

1 INTRODUCTION

Recently there has been a considerable amount of research in developing energy-saving DNN models to allow state-of-art DNNs with high computational costs to be deployed in mobile and embedded architecture. *Adaptive Neural Networks (AdNNs)* [2, 7, 33] are energy-saving DNN models that determine when to switch off certain parts of the network to reduce the number of computations.

Because an AdNN model determines which parts of the neural network to run based on inputs, an adversary's ability to surge the energy consumption by carefully crafting inputs is a crucial concern in energy-critical environments. For example, AdNNs like BlockDrop [41] and SkipNet [38] can reduce the computations in ResNet significantly and an alteration on the input can nullify a large portion of the reduced computations, invalidating the models' purpose. Such behavior would lead the app or software using an AdNN model to consume energy erratically, resulting in devices' power failure and disastrous consequences. Thus, there is a strong need to provide a systematic testing method to find energy hotspots in the model and filter out potential "power-surging" inputs that will negatively impact the model's performance.

Creating testing inputs to increase the energy consumption of a DNN model is challenging because inferring the relation between energy consumption and input is a challenging task. Unlike inferring the relation between input and output, where we can find the derivatives from a series of computation functions in the model, energy consumption can only be measured by running the model. Traditional DNN testing methods [22, 30, 37, 43] and traditional adversarial attacks [4, 10, 29] on DNNs have been designed to create carefully crafted synthetic testing inputs using the gradient of generated output with respect to the input. However, for energy testing, it is unclear whether a change in the input induces an increase or decrease in the energy consumption of the model. To the best of our knowledge, ILFO [13] is the first work that seeks to formulate all types of AdNN's energy robustness (Section 2.1) problem by modeling the relation between input and intermediate output [13] (DeepSloth [15] only evaluates energy robustness of Early-termination AdNNs).

However, our investigations (Section 3) show that ILFO generated energy surging samples lack traditional transferability, *i.e.*,, the adversarial samples generated by ILFO for a target AdNN cannot be applied to a new AdNN to increase its energy consumption. Therefore, the traditional black-box accuracy testing method of DNNs using surrogate model [5, 21, 28] can not be used for energy robustness evaluation. Therefore, ILFO generated samples cannot evaluate the energy robustness of AdNNs in a black-box scenario.

This paper presents EREBA (**E**nergy **R**obustness using **E**stimator **B**ased **A**pproach) to perform energy testing on AdNNs under the black-box setting where there is no prior knowledge known about the AdNN model. To our knowledge, this is the first attempt in this direction. EREBA aims to evaluate the energy robustness of AdNN and identify inputs that will negatively impact the model's performance. Specifically, we develop two testing methods to assess any given AdNN model's energy robustness, namely Input-based testing and Universal testing. Input-based testing evaluates energy robustness where testing inputs are semantically meaningful to the AdNN (*e.g.,* meaningful images, compilable programs). On the other hand, universal testing evaluates worst-case energy robustness where each testing input maximizes the energy consumption for each target AdNN.

For generating testing inputs for AdNNs in a black-box setting, it is needed to find a relation between input and energy consumption of AdNNs. Based on the working mechanism of AdNNs, we know that different numbers of residual blocks/layers are activated during inference for different inputs. The number of activated blocks/layers during inference has a semi-linear (step-wise) relation with energy consumption, which can also be noticed in Figure 7. Through this step-wise relation between the number of activated blocks and energy consumption, we can conclude that input and energy consumption of AdNNs are related. Because of this reason, EREBA is able to learn a decent approximation of the energy consumption of an AdNN given the input. Based on such approximation, EREBA then generates input perturbations that significantly increase the energy consumption of the AdNN.

We evaluate EREBA on four criteria: effectiveness, sensitivity, quality, and robustness using the CIFAR-10 and CIFAR-100 datasets [19, 35, 36]. First, to evaluate the effectiveness of the testing inputs generated by EREBA, we calculate the energy required for AdNNs to classify these inputs while running on an Nvidia TX2 server. We then compare this value with the energy required by the inputs generated from common corruptions and perturbations techniques [14] and a surrogate model-based approach. We observe that EREBA is twice as effective. The sensitivity of EREBA is measured through the behavior of the energy consumption of testing inputs generated while limiting the magnitude of perturbation allowed, which enables a comparison between the AdNN models' energy robustness. The quality of the generated testing inputs is evaluated against the original input through Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [39, 40]. Finally, the robustness of EREBA is demonstrated by providing corrupted input images for the generation of testing inputs, which reveals the capability of the estimator model to imitate the shortcomings of the target AdNN. We further demonstrate two ways to show how EREBA generated test inputs can help to increase energy robustness: through input filtering and gradient-based detection.

Our paper makes the following contributions:

- An approach, EREBA, the first energy-oriented black-box testing methodology for AdNNs.
- A systematic empirical study on transferability of energy-based testing inputs.
- Four evaluations to demonstrate the effectiveness, sensitivity, quality, and robustness of EREBA.
- Two applications demonstrating the energy-saving capability of EREBA.

2 BACKGROUND

2.1 Energy Robustness

ILFO [13] has defined the energy robustness of a DNN as the stability of the model's energy consumption after getting a perturbed input. However, a model's energy robustness should not only depend on the inputs that belong to the training data distribution of the model. Energy robustness should also be evaluated based on the out-of-distribution inputs. Because of this reason, we define two types of energy robustness for DNNs: Input-based Energy Robustness (E_i) and Universal Energy Robustness (E_u).

E_i is defined by the maximum energy consumed by the model for an input which belongs to the training data distribution of the model. Let us assume, x is an input that is within the data distribution of a DNN f. We want to add perturbation δ to x such that energy consumption is maximum. In that scenario, E_i can be represented as,

$$E_i = -\max_{\delta \in R} \left(ENG_f(x+\delta) - ENG_f(x)\right)$$

, where R is set of admissible perturbations such that $x + \delta$ remains within distribution, and ENG_f represents the energy consumption of DNN f.

E_u can be described as the highest possible energy consumed by a model for any input. Inputs used to measure E_u can be out-of-distribution inputs also. For a DNN f and any input x, E_u can be represented as,

$$E_u = -\max_{x} ENG_f(x)$$

, where ENG_f represents energy consumption of DNN. By increasing the value of E_i and E_u, energy robustness of a model can be increased.

2.2 AdNNs

The main objective of AdNNs is to minimize executing layers in a Neural Network while maintaining reasonable accuracy. The AdNNs can be divided mainly into two types: *Conditional-skipping AdNNs* [38, 41] and *Early-termination AdNNs* [2, 33]. Both types of AdNNs reduce computations if their intermediate output values satisfy predefined conditions. For reducing computations, Conditional-skipping AdNNs skip a few layers or residual blocks [1] (in the case of ResNet), while Early-termination AdNNs terminates the operations within a block or network early.

3 TRANSFERABILITY OF ENERGY-BASED TESTING INPUTS.

In this section, by carrying out a preliminary study, we show that traditional transferability does not exist in energy testing inputs, and existing technique like attacking surrogate model to generate accuracy-based testing inputs cannot be applied in energy testing. For traditional accuracy-based testing, transferability refers to the

[1] Residual block consists of multiple layers whose output is determined by adding the output of the last layer and input to the block.

property that adversarial examples generated for one model may also be misclassified by another model.

Motivation. In a black-box setting, existing techniques [5, 21, 28] evaluate the accuracy-robustness of DNNs based on the traditional transferability of adversarial samples [27] . Adversarial examples of DNNs are perturbed inputs close to the original correctly classified inputs but are misclassified by DNNs. Because adversarial examples are commonly used as testing inputs to measure the robustness of the neural networks [23, 43]), we will also use the term *testing inputs* to refer to the adversarial examples in this paper. Goodfellow *et al*. and Szegedy *et al.* [10, 32] have concluded that accuracy-based testing inputs on a traditional DNN model are transferable. Therefore, adversarial examples generated by attacking a surrogate DNN model can be applied to other DNNs for evaluating robustness. In this section, we investigate if traditional transferability, which is used for measuring accuracy robustness in a black-box setting, can be applicable for energy-based testing inputs.

Table 1: ITP among different architectures. RN is ResNet, BD is BlockDrop, BN is BranchyNet. BM represents Base Model, while TM is Target Model.

BM \ TM	RAN	BD (RN 110)	BN	BD (RN 32)
RAN	100.0	46.0	41.0	12.5
BD (RN 110)	64.0	100.0	68.0	72.4
BN	61.0	52.0	100.0	4.0
BD (RN 32)	5.5	45.0	75.2	100.0

Table 2: ETP among different architectures. RN is ResNet, BD is BlockDrop, BN is BranchyNet. BM represents Base Model, while TM is Target Model.

BM \ TM	RAN	BD (RN 110)	BN	BD(RN 32)
RAN	100.0	0.1	40.0	-1.8
BD (RN 110)	200.0	100.0	350.0	52.5
BN	38.0	3.0	100.0	-3.8
BD (RN 32)	-3.5	10.0	228.0	100.0

Preliminary study. We have conducted a study to investigate the traditional transferability of energy-based testing input on AdNNs. To our knowledge, this is the first effort to explore the transferability of energy-based testing input on AdNNs. We define base and target models for this study. The white-box attack is performed on the base model, and the target model classifies the testing input. We focus on two metrics to measure transferability: the percentage of the transferable adversarial inputs and the average percentage of the transferable energy consumption increase. We define two terms: *Effectiveness Transferability Percentage (ETP)* and *Input Transferability Percentage (ITP)*. *ETP* is defined based on *IncRF*, which is the fractional increase in AdNN-reduced floating-point operations (FLOPs) after feeding energy-based testing inputs. We also define P_b and P_t, the average *IncRF* on base and target models, respectively, with the same testing inputs. We define ETP=$(P_t/P_b) \times 100$. *ITP* is defined as the percentage of testing inputs for which the FLOPs count during inference increases in the target model. For an attack, if *ITP* is high, it means that most of the generated testing inputs for the base model can also increase the energy consumption in the target model. If *ETP* is high, it means that the average increase in the target model's energy consumption is comparable with the base model. Thus, if both *ETP* and *ITP* are high, then it confirms transferability in the attack.

For example, we attack the base model and perturb ten inputs. If the average $IncRF$ on base model is 0.5, *i.e.*, P_b = 0.5. If seven out of ten testing inputs increase the FLOPs on the target model, ITP will be 70 %. For target model, The average *IncRF* is 0.3. P_t would be 0.3 and *ETP*=(0.3/0.5)× 100 = 60%. We have set multiple *thresholds* of ITP and ETP to determine whether an attack is transferable. In this study, we explore how many combinations of base AdNN and target AdNN exceeds the different *ITP* and *ETP* thresholds.

We have conducted a transferability study on four AdNN models: RANet [44], BlockDrop (ResNet-38, ResNet-110) [41], and BranchyNet [33]. 1000 images sampled from the CIFAR-10 dataset were used in this study. We generate testing inputs using the ILFO attack [13] (Figure 1 Sub-figure I) on the base model. Tables 1 and 2 show the *ITP* and *ETP* among the architectures. From the tables, we can see that only four combinations (out of 12) of base and target models exceed the 50% threshold for both *ITP* and *ETP* (For other thresholds, we have added a table in the website). From these results, we can conclude that for the majority of AdNN models, the white-box attack is non-transferable.

Although traditional transferability may not be feasible for attacking AdNNs, the observation of this failure motivates us to develop an effective alternative. Specifically, our key observation on the non-transferability of energy-based attacks is that the energy-saving mechanisms of AdNN models behave differently for the same input, *i.e.,* an input causing high energy consumption on one model might consume low energy for another model. Hence, extending any white-box attack method such as ILFO through Surrogate models is not viable in the black-box scenario.

1 OACH

As we have observed that Surrogate models are not feasible to test the energy robustness of AdNNs, we develop EREBA, an Estimator model-based black-box testing system. EREBA contains two major components, namely an Estimator model and a testing input generator. Figure 1 illustrates the Estimator model's training for an AdNN using its energy consumption on an Nvidia TX2 server. The Estimator model addresses the challenge of non-transferability discovered in the previous section. Additionally, the testing input generator in EREBA has two modes of testing: Input-based, where an input image is perturbed to achieve higher energy consumption, and Universal, where a noisy testing input that can maximize energy consumption is generated. These modes enable EREBA to assess the energy robustness of each AdNN effectively. Figure 1 (Sub-figure ii) shows the generation of testing inputs using the trained Estimator model.

4.1 Estimator Model Design

Traditionally, misclassification black-box adversarial methods on DNNs are achieved through Surrogate models (also DNNs) trained using the output labels produced by feeding the target DNN with original images. Testing inputs generated by a white-box method against the trained Surrogate model are then used against the target

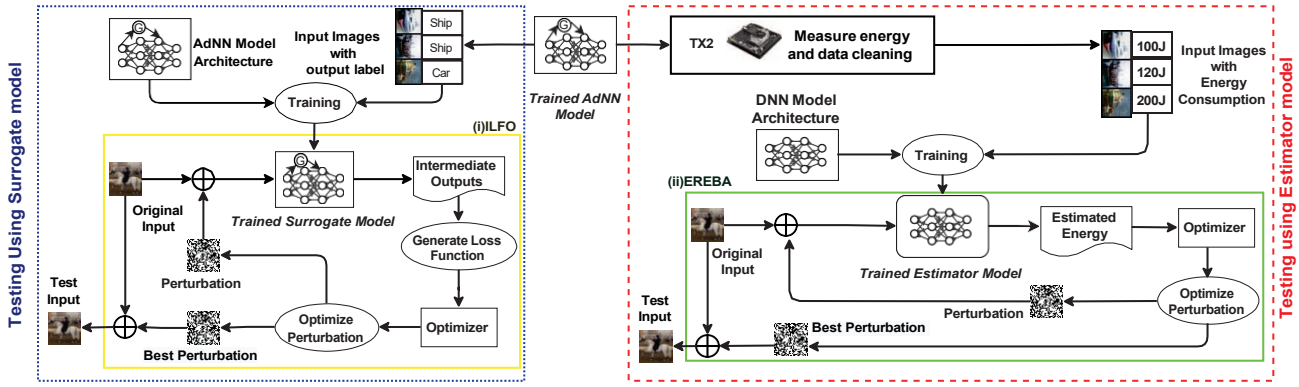


Figure 1: Difference between testing using Estimator model and Surrogate model

DNN as illustrated in Figure 1 (inside the blue-dotted box). However, this approach is not feasible for the current use-case due to a lack of traditional transferability in white-box attacks (Section 3). Because building such a Surrogate model with the target function of mapping an image to a class would not transfer similar energy characteristics to the Surrogate model. The target function should be the energy-saving mechanism in an AdNN, but the output dimensions of these change with different AdNNs. So, we need a separate Surrogate model for each AdNN; this is not viable for two reasons. First, such a model would require a new neural network architecture for each AdNN, which makes it hard to apply to future AdNN models. Second, the energy-saving mechanisms' outputs are intermediate values within an AdNN, which are not accessible in a black-box scenario.

To tackle this, we build an Estimator model to emulate the characteristics of the energy-saving mechanism in AdNNs. The feasibility of the Estimator model follows from the following two key observations. 1) We can perceive the energy-saving mechanisms' characteristics through system diagnostics such as the energy consumption for each inference, which can be observed even in a black-box setting. 2) Even though each AdNN has a different energy-saving mechanism, the resulting energy consumption is always expected to lie in a step-wise pattern [2]; see Figure 7 and Section 7.3 for more details. Thus, we seek to leverage this patterned energy consumption in our black-box approach to train an Estimator model for each target AdNN at which point the Estimator model can predict the energy consumption of each image.

However, the energy consumption of an embedded system such as Nvidia TX2 is affected by noise from the system environment, such as background processes and dynamic frequency scaling, making it challenging to create an accurate model. Further, in a black-box environment, it is hard to categorize which data is noisy. We do not have any additional information (*e.g.,* number of the executed blocks) about the inference; our approach has to tackle these challenges.

An overview of the Estimator model training is given in Figure 1 (Inside red dotted box). First, we collect energy consumption data for images used for training and, to address the noise in the data due to the system environment, we deactivate the dynamic frequency scaling of Nvidia TX2 and ensure that no other user processes are running. Additionally, we record the energy consumption by the target AdNN during inference of an input image twenty times and discard values, which are 50% higher than the median value. We define the mean of the remaining values is defined as $F(x_i)$, where x_i represents an input in the dataset. We define the Estimator's DNN loss function as:

$$estimator\ loss = \frac{1}{N}\sum_{i=1}^{N}[F(x_i) - EST(x_i)]^2$$

where EST denote the Estimator model, and N is the size of the dataset. *estimator loss* is used to train the Estimator for each target AdNN, which enables the model to give a reasonable prediction of energy consumption of the target AdNN for a given input.

4.2 Testing input generator

The objective of the testing input generator is to create testing inputs that increase the Estimator model's prediction, which in turn should increase the actual energy consumption of the AdNN. We explore two use cases of EREBA: 1) Input-based test, and 2) Universal test for measuring the Input-based and Universal energy robustness as defined in Section-2.1.

4.2.1 Input-based testing. In this use case, we modify the input image in such a way that it is imperceptible by a human, and the resulting testing input has higher energy consumption on the target AdNN. We thus add a perturbation δ_i to the input x_i. Picking the best δ_i ($\hat{\delta_i}$) can be formulated as:

$$\hat{\delta_i} = \underset{\delta_i}{\text{argmax}}\ EST(x_i + \delta_i).$$

Additionally, we have to ensure that the magnitude of perturbation is also small as higher magnitude perturbations are more susceptible to detection. We can reformulate the maximization problem to a minimization problem as follows:

$$\underset{\delta_i}{\text{minimize}}\ (||\delta_i|| - c \cdot EST(x_i + \delta_i)) \text{ where, } (x_i + \delta_i) \in [0,1]^n \quad (1)$$

where $c > 0$ is a hyperparameter chosen through grid search depending on the AdNN model. Also, c controls the magnitude of

[2] Energy consumption of processing an extra residual block or layer in an AdNN will always add similar energy consumption into the total energy consumption, making the energy consumption pattern step-wise.

generated perturbation ($||\delta_i||$), where a large c makes the loss function more dependant on the energy estimate, allowing for larger perturbations. Whereas a smaller c makes the loss function more dependant on $||\delta_i||$. Hence, c and $||\delta_i||$ are directly proportional.

This constrained optimization problem in δ_i can be converted into a non-constrained optimization problem in w_i, where the relationship between δ_i and w_i is:

$$\delta_i = \frac{\tanh(w_i) + 1}{2} - x_i$$

The $tanh$ function would ensure that the generated test input values stay between 0 and 1. The equivalent optimization problem in w_i is:

$$\underset{w_i}{\text{minimize}} \left\| \frac{\tanh(w_i) + 1}{2} - x_i \right\|_2^2 - c \cdot EST\left(\frac{\tanh(w_i) + 1}{2}\right) \quad (2)$$

4.2.2 Universal Testing. In this use case, EREBA generates a testing input only using the Estimator model. Unlike Input-based testing, which adds human imperceptible perturbation to original images, universal testing creates noisy testing inputs, which can maximize the energy consumption of the target DNN independent of the input. The intuition behind this testing is that adversaries can send noisy testing inputs exclusively to increase the system's energy consumption because human perception may not be a concern in every scenario. Hence we modify the optimization function from equation 2 to:

$$\underset{w_i}{\text{minimize}} - EST\left(\frac{\tanh(w_i) + 1}{2}\right) \quad (3)$$

Algorithm 1: Testing input generation using EREBA

Inputs : x_i : *Input Image*
Outputs: f_i : *Perturbed Image*

1 **begin**
2 Initialize(w_i)
3 $T = number\_of\_iterations$
4 $iter\_no = 0$
5 **while** $iter\_no < T$ **do**
6 $L = loss(x_i, w_i, c)$
7 $L_{new}, w_i = Optimizer(L, w_i)$
8 $iter\_no$ ++
9 **end**
10 $f_i = \frac{\tanh w_i + 1}{2}$
11 **end**

Both minimization problems can be solved through an iterative approach given by Algorithm 1, which is also illustrated in Figure 1 (sub-figure (ii) inside green box). The algorithm outputs the testing input f_i while taking the current image x_i (not required in Universal test mode) as input. w_i is initialized to a random tensor (multi-dimensional array) with a size equal to the input image dimension. For each iteration, the loss function of the current mode (given in equations 2, 3) is computed (at line 6). This loss is back-propagated, and the optimizer takes a step in the direction of the negative gradient of the loss w.r.t w_i and updates w_i with its next value. Once the iteration threshold (T) is reached, the algorithm computes and returns the testing input f_i (at Line 10). In the Universal test mode, this algorithm is repeated for N_r different random initializations of w_i, out of which w_i corresponding to the lowest loss value is used for computing f_i.

2 VALUATION

We evaluate the performance of EREBA on three popular AdNNs, **RANet** [44], **BlockDrop** [41] and **BranchyNet** [33, 34], in terms of four research questions (RQs):

RQ1: Effectiveness. How much increase in energy consumption is achievable by the testing inputs generated by EREBA?

RQ2: Sensitivity. How does the energy consumption of AdNNs react to limiting the magnitude of perturbation in EREBA?

RQ3: Quality. What is the difference in semantic quality between original images and testing inputs generated by EREBA?

RQ4: Robustness. Is EREBA robust against distribution shifts?

2.1 erimental Setup

Datasets. For all AdNN models, the CIFAR-10 and CIFAR-100 datasets [19, 35, 36] have been used to train the Estimator model and generate testing inputs. Both the datasets consist of 50000 training and 10000 test images, where CIFAR-10 and CIFAR-100 have 10 and 100 class labels, respectively. By using these two datasets, we show that EREBA is useful for both easier (CIFAR-10) and more complex prediction (CIFAR-100) tasks.

Baseline. As there are no existing black-box energy testing frameworks, we compare our technique with two different types of baseline techniques. First, we compare our techniques with real-world corruption and perturbation techniques (like fog, frost) [14]. The datasets generated from these techniques are commonly used [9, 26, 42] to test the robustness of neural networks. Second, we use a surrogate model technique that utilizes ILFO to generate testing inputs.

Common corruption techniques [14] contain different visual corruption, which includes practical corruptions like fog, snow, frost. We use 19 different corruption types, and for each type, five visual corruptions are created from severity level one to five, resulting in a total of 95 different visual corruptions. In the images generated by common corruption techniques, the noise present in the inputs is human perceptible.

Common perturbation techniques [14] use 14 practical perturbation types; for each original input, 30 different images are created with different amounts of perturbation. With perturbation, slightly perturbed images are generated that are difficult for humans to differentiate from the original images.

Other than using common corruptions and perturbations [14], we also use the Surrogate model-based technique as a baseline. For this approach, we create a Surrogate model for each AdNN and use the Surrogate model to generate adversarial images. As we see in Tables 1 and 2 that adversarial inputs generated using BlockDrop as the surrogate model are more effective on other AdNNs, we add a baseline that uses BlockDrop as a Surrogate model and is referred to as SURRG in the followings sections. For each AdNN, we first classify 50000 CIFAR-10 and CIFAR-100 training data on the target AdNN and based on its outputs, and we train the Surrogate model. Then we use the ILFO attack on the Surrogate model to generate test inputs. We use Surrogate model to generate both limited perturbation (Input-based SURRG) and noisy (Universal SURRG) test samples.

Models. We have selected AdNN models where the range of maximum energy consumption during inference is large (15 J-160J).

Table 3: Hyperparameters of EREBA

Mode	AdNN	c	Learning Rate	T	N_T
Input-based	BlockDrop	1	0.01	500	-
	BranchyNet	100	0.01	500	-
	RANet	10	0.01	500	-
Universal	BlockDrop	-	0.01	500	30
	BranchyNet	-	0.01	500	30
	RANet	-	0.01	500	30

We show that EREBA can be used against AdNN models with both higher and lower number of parameters. AdNN BlockDrop is built modifying ResNet-110 architecture and trained on CIFAR-10. While for training using CIFAR-100 dataset, BlockDrop is built modifying ResNet-32 architecture because ResNet-110 BlockDrop architecture trained on CIFAR-10 dataset has shown less adaptability. ResNet architecture starts with a 2D convolution layer, which is followed by residual blocks. BlockDrop selects which blocks to execute through Policy Network for BlockDrop. Both RANet and BranchyNet are multi-exit networks, where operation can be terminated in one of the earlier exits based on the confidence score of the exit. The Estimator Model is a ResNet-110, with the output fully connected ten node layer changed to a fully connected single node. The loss function is changed to the function defined in section 4.1. Hyperparameters chosen by the Estimator model (c, learning rate, number of iterations (T)) for each AdNN model are in Table 3, reasons for choosing these parameters are given in section 5.2.1. These are parameters used for the results reported in sections 5.2.1 and 5.2.3, whereas section 5.2.2 reports the behaviour of EREBA when c,T are changed.

Hardware Platform. We use the Nvidia Jetson TX2 board for our energy consumption measurements, which are used to train the Estimator Model. By default, TX2 has a dynamic power model, which scales the CPU and GPU frequencies based upon the current system load, which adds additional uncertainty to the energy consumption measurements. To combat this, we set the TX2 board to Max-N mode, which forces CPU and GPU clock to run at their maximum possible values, which are 2.0 GHz and 1.30 GHz, respectively.

Jetson TX2 module has two power monitor chips on board for measuring power consumption. One of the power monitors measures the power consumption of CPU, GPU, and SOC as in Fig. 8, 9 of the user manual of Nvidia TX2 [25]. During the inference process of the AdNNs, we measure the power consumption of GPU using a monitor program. Since the monitor program only uses the CPU, the program does not affect the energy measurement. Additionally, TX2 Internal power monitors have been validated by other studies such as S. Köhler et al. [18] (see Fig 1b.), where it is shown that the power measurements using the internal monitor chips collaborate with external measurement techniques. As stated in [18], one concern with using the internal chips is that the power consumption from the carrier board, fan, and power supply is not measured, which comes out to be around 2W. However, since our study measures the effect of various testing inputs, which cannot affect such components' behavior and both measurements (testing input, original image) ignore the consumption from these components, the conclusions drawn are valid. Further, to ensure the collected energy consumption data is correct, we run the inference twenty times and discard values outlier values that are 50% higher than the median value. The mean of the remaining values is used as the single energy consumption value reported.

(a) Original **(b) Testing Inputs**

Figure 2: Testing inputs generated by EREBA for BlockDrop in Input-based testing mode

Metrics. We evaluate the effectiveness and robustness of EREBA using the percentage increase in energy consumption for each target AdNN:

$$\frac{EnergyConsumption(f_i) - EnergyConsumption(x_i)}{EnergyConsumption(x_i)} \times 100$$

where x_i is the input image provided to EREBA and f_i is the testing input generated. For Sensitivity (RQ2) measurements, we use the increment in energy consumption in Joules (J) to better compare sensitivity between target AdNNs and use the average squared difference in pixels values between the input image and the testing input to quantify the magnitude of the perturbation. The testing inputs' quality is measured using Peak Signal to Noise Ratio (PSNR) [39], and Structural Similarity Index (SSIM) [40] because of usage of these metrics in the industry to measure image quality.

2.2 erimental Results

221 ectiveness. To evaluate testing effectiveness by EREBA, we have measured the average percentage increase in each AdNN model's energy consumption between the original images in the dataset and the corresponding testing inputs generated by EREBA. We compare the effectiveness of Universal testing against the common corruption techniques [14] because, in both approaches, noise introduced to the inputs is human perceptible. Whereas Input-based testing adds imperceptible perturbation to the input, hence we compare against the common perturbation techniques [14]. The hyperparameters chosen for each model are in Table 3. Parameters vary between AdNNs due to variations in input normalization, which is only applied in BranchyNet and RANet. We evaluate on images from the CIFAR-10 and CIFAR-100 datasets, which have a considerable reduction in the energy consumption on the target AdNNs [33, 34, 41, 44]. The Estimator models have been trained on 50000 CIFAR-10 and CIFAR-100 training images. We apply common corruption and perturbation techniques [14] to CIFAR-10 and CIFAR-100 test images. As there are numerous corruption, and perturbation techniques, we only report the best performing techniques (*i.e.,* highest $IncRF$) for each AdNN model (For the $IncRF$ values of other corruptions and perturbations, please see the website[3]); Table 4 reports the exact corruption/perturbation technique. Table 5 reports the mean percentage increase in energy consumption of the AdNN models under EREBA (Universal, Input-based testings) and the baselines on the CIFAR-10 dataset. Figure 2 illustrates some Input-based testing inputs generated by EREBA for BlockDrop. We observe that EREBA Input-based testing inputs dominate

[3] https://sites.google.com/view/ereba/home

Table 4: Corruptions and perturbations we have used for comparison for each model. *corr* represents corruptions and *per* represents perturbations.

Data Type \ Models	BlockDrop	BranchyNet	RANet
Best Corr (CIFAR-10)	Contrast	Impulse Noise	Contrast
Best Per (CIFAR-10)	Gaussian Blur	Snow	Gaussian Blur
Best Corr (CIFAR-100)	Contrast	Impulse Noise	Fog
Best Corr (CIFAR-100)	Zoom Blur	Zoom Blur	Shot Noise

Table 5: Mean percentage increase in energy consumption of the models for CIFAR-10 dataset against EREBA and baseline techniques.

Perturbation Type \ Models	BlockDrop	BranchyNet	RANet
Universal Testing (EREBA)	97.54	**528.51**	**1846.18**
Best Corr	77.77	186.79	1209.00
Universal SURRG	**137.8**	26.87	302.42
Input-based Testing (EREBA)	67.92	**288.90**	885.00
Best Per	16.24	153.72	**1480.87**
Input-based SURRG	**135.5**	118.39	554.69

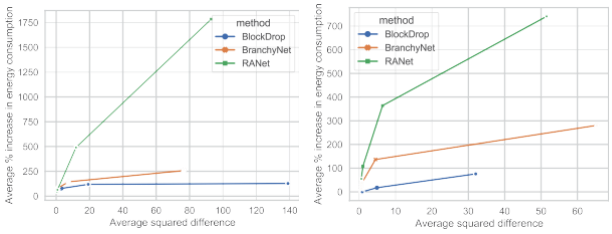
the baseline methods for mean energy increase on BranchyNet. Whereas for BlockDrop, because SURRG has BlockDrop as its architecture and ILFO is a white-box method, it is expected to outperform EREBA, a black-box method. Interestingly, for RANet, common perturbation techniques [14] induce a much higher energy increase than the corruption techniques, which is quite different from the behavior observed in the other AdNNs. While EREBA underperforms the perturbations in terms of energy consumption increase, EREBA still outperforms SURRG. Also, we observe that for all AdNNs, samples generated through Universal testing outperform the baseline techniques. Furthermore, for BranchyNet and RANet, due to fewer execution modes (two in BranchyNet and eight in RANet), the energy consumption is higher than BlockDrop (up to 2000 % more than the original data) with 2^{54} modes.

For the CIFAR-100 dataset, Table 6 shows the average percentage increase in energy consumption for EREBA and the baseline techniques. We notice that EREBA generated inputs outperform common corruption and perturbation techniques for all three AdNNs. Similar to the CIFAR-10 dataset, SURRG generated inputs consume more energy than EREBA generated inputs only for the BlockDrop model. For RANet, Universal testing inputs can not significantly increase energy consumption because RANet always predicts an input with high noise as road or shrew with high confidence; therefore, the inference is stopped at initial exits, resulting in lower energy consumption. Nevertheless, the Input-based testing inputs can increase up to 4000% energy consumption of the original inputs for the RANet model. Thus, we conclude that, on average, over all three AdNN models, EREBA performs better than any other baseline technique in terms of increasing energy consumption.

222 *y*. We define the sensitivity of EREBA in terms of the magnitude of the perturbation $|\delta_i|$. Intuitively, if an AdNN model's energy consumption spikes up with a relatively lower average perturbation magnitude, then that model is less robust.

Table 6: Mean percentage increase in energy consumption of the models for CIFAR-100 dataset against EREBA and baseline techniques.

Perturbation Type \ Models	BlockDrop	BranchyNet	RANet
Universal Testing (EREBA)	27.22	**580.50**	**479.80**
Best Corr	-29.61	5.74	-31.80
Universal SURRG	**55.40**	71.42	-29.50
Input-based Testing (EREBA)	17.27	**283.60**	**1113.28**
Best Per	-54.28	37.80	-30.73
Input-based SURRG	**41.31**	37.90	754.69



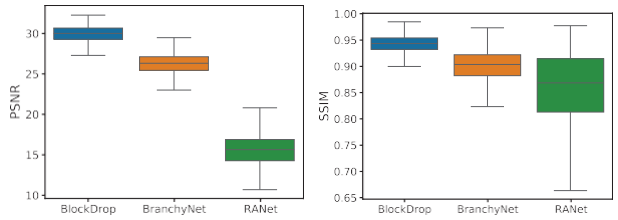
(a) CIFAR-10 **(b) CIFAR-100**

Figure 3: Average energy consumption increase of testing inputs constrained by magnitude of perturbation for BlockDrop, BranchyNet and RANet.

We know that c and $|\delta_i|$ are directly propositional, given that the number of iterations is constant. Through empirical observations, $T = 500$ is sufficient to achieve convergence in EREBA for all AdNN models. Note that sensitivity cannot be compared using the magnitude of c as only BranchyNet and RANet use normalization filters, whereas BlockDrop does not, which makes the optimal c of BranchyNet and RANet larger (see Table 3). To measure the magnitude of perturbation for a set of c, T we measure the average squared difference between the testing input and the input image, which is defined as follows:

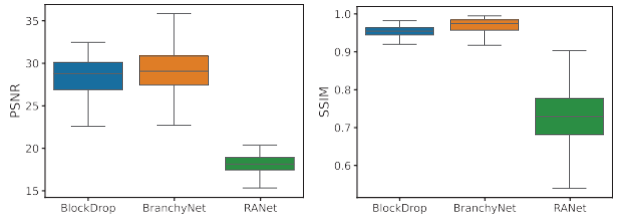
$$\text{Average squared difference} = \frac{1}{N}\sum_{i=1}^{N}(x_i - f_i)^2$$

where x_i is the input image, and f_i is its corresponding testing input. Figures 3a and 3b show the average percentage increase in energy consumption versus the average squared difference on the CIFAR-10 and CIFAR-100 datasets. We observe that for both datasets compared to BlockDrop, BranchyNet and RANet are more sensitive to the perturbation magnitude. BlockDrop's lower sensitivity is mainly due to BlockDrop's Policy Network, which provides more refined control over energy consumption; hence BlockDrop is more robust than the other two AdNNs. Additionally, we can see the potency of the Estimator model for every AdNN model. A direct proportionality between the average increase in energy consumption and the average squared difference is observed in all the AdNN models, which is evidence that the Estimator model is successful in imitating the energy consumption of each AdNN. Additionally, we observe that EREBA performs very similarly for both CIFAR-10 and CIFAR-100 datasets for BlockDrop and BranchyNet. Whereas



(a) PSNR between original and generated images **(b) SSIM between between original and generated images.**

Figure 4: Quality of the generated images for each AdNN model for CIFAR-10 dataset



(a) PSNR between original and generated images **(b) SSIM between between original and generated images.**

Figure 5: Quality of the generated images for each AdNN model for CIFAR-100 dataset

for RANet for CIFAR-10, the energy spike induced is much higher than that for CIFAR-100, indicating that the CIFAR-100 RANet is more robust than the CIFAR-10 version.

223 y. In this section, we evaluate the quality of the perturbation generated by Input-based testing of EREBA with the hyper-parameters set to the values given in Section 5.1 using Peak Signal to Noise ratio (PSNR) [39] and Structural Similarity Index (SSIM) [40]. Both of these metrics are used in the industry to measure the image quality of noisy images. SNR of an image can be represented by,

$$SNR = \frac{\mu_{image}}{\sigma_{image}}$$

where μ_{image} is the mean value of the image pixels and σ_{image} is the error value of the pixel values. For PSNR, the highest value of the image pixels is used instead of the mean value. The Structural Similarity Index (SSIM) is a perceptual metric that quantifies the image quality degradation caused by processing such as data compression or by losses in data transmission. Higher values for SSIM and PSNR indicate higher quality test inputs.

Figure 4 and Figure 5 show the values of SSIM and PSNR between the testing inputs and the original images for CIFAR-10 and CIFAR-100 datasets. For both datasets, we see that SSIM for the generated testing inputs is similar for all the target AdNNs. Whereas PSNR for RANet is worse but still comparable to BlockDrop and BranchyNet. We conclude that most of the inputs generated through Input-based testing are of high quality; even if some test inputs might have noise, they are structurally similar to the original inputs.

Table 7: Effect of corruptions on EREBA. Average percentage increase in energy consumption for all AdNN models for various corrupted inputs from CIFAR-10 and CIFAR-100.

AdNN	Dataset	Normal	Frost	Fog	Snow
RANet	CIFAR-10	929.76	2 085.36	2 027.20	2 099.41
	CIFAR-100	340.75	6.98	13.54	25.51
BranchyNet	CIFAR-10	283.05	475.08	355.70	323.72
	CIFAR-100	278.17	197.58	374.62	360.37
BlockDrop	CIFAR-10	72.78	108.37	89.61	111.80
	CIFAR-100	17.50	24.91	21.49	36.95

5.2.4 RQ4: Robustness. EREBA estimates energy consumption based on the training data, which can be impacted by distribution shift [31]. Therefore, to evaluate the robustness of EREBA against distribution shifts, we have analyzed the behavior of EREBA against Practical corruptions. Practical corruptions (*e.g.,* fog, snow etc.) are frequently noticed, mainly when we use mobile phones or autonomous vehicles to take images. For this purpose, we have used real-world common corruption techniques [14].

Table 7 shows the average percentage increase in energy consumption for the testing inputs generated using the original and corrupted images of CIFAR-10 and CIFAR-100 by EREBA in Input-based testing. We picked the corruption classes of fog, frost, and snow due to their natural occurrence. In general, the corrupted images do not hinder the performance of EREBA except for the CIFAR-100 version of RANet. In other cases, the increase in energy consumption achieved by EREBA using corrupted images is, in fact, higher than that achieved using the original CIFAR-10 and CIFAR-100 datasets. This is mainly due to the common corruptions [14], introducing better initialization spots (white areas). For CIFAR-100 RANet, EREBA manages to increase the energy consumption slightly. However, similar to the Universal testing case, inputs with high noise are classified as road or shrew with high confidence, which leads to lower performance in comparison to other settings.

Additionally, through these results, we further notice the high stability of the Estimator model in approximating the shortcomings of the target AdNNs. EREBA can generate high energy-consuming testing inputs despite the corruption of images. While there is some variance in how EREBA behaves when provided with an image with different corruption classes; the median energy consumption increment is consistent for all target AdNNs.

3 ENERGY ROBUSTNESS

In this section, we demonstrate two ways to increase the energy robustness of AdNNs. In both ways, we use prior EREBA generated inputs to detect new EREBA generated energy-surging inputs for AdNNs. For detecting Universal test inputs, we use input filtering method based on pixel values, while for detecting input-based test inputs, we use gradient-based input detection.

Input Filtering. As we can notice that noisy samples generated from Universal testing can increase energy consumption by a significant amount, hence it is essential to adapt the AdNN mechanism against these noisy images. For traditional DNNs, highly noisy images consume the same energy as the standard images and are no threat to the robustness of the DNN (the object is not visible in those noisy images). To adapt AdNNs against high energy-consuming images, we propose to include a filter in AdNNs.

As a filter, we have created a ResNet model (binary classifier) with only six residual blocks. The classifier can classify into two categories: **normal input** and **energy-consuming noisy input**. For each AdNN, We have trained the ResNet model with 2500 normal dataset images and 2500 high energy-consuming noisy images, creating three different trained ResNet models. For testing the models, 1000 images have been used (500 from each class). Our results show that the filter can identify each test image with the correct class (100% accuracy) for all three AdNNs for both datasets. The results confirm that we can filter out high energy-consuming noisy images with a model whose energy consumption is low.

Gradient-based Input Detection. In contrast to detecting adversarial inputs similar to the samples generated by the Universal testing mode, adversarial inputs similar to the samples generated by the Input-based testing mode are harder to differentiate from benign inputs. Additionally, the energy constraint on such a detection system is a significant challenge. Therefore, the detection technique must consume significantly low energy with respect to the energy consumed by Input-based testing inputs during inference. To address these challenges, we propose a gradient-based adversarial input detection mechanism that uses partial inference from the AdNN.

In this detection mechanism, we leverage the behavior of the energy-saving mechanism within various AdNNs. These mechanisms, in general, try to ensure that the difference between an intermediate output and a predefined condition is large, which will deactivate certain parts of the AdNN. In other words, if the loss function of intermediate outputs is large for any input, the energy consumption will be low [13]. Therefore, if the gradients of the intermediate loss function with respect to the weights are large, the input is more likely to be a benign input. Hence, we only need partial inference (weight gradients of an initial layer) from the AdNNs, and a linear SVM [6] to detect the energy-surging inputs, where both are low energy consuming steps. So, the energy impact of our detection component is significantly low. If the input is predicted as energy-surging by the detector, the inference is stopped early.

To evaluate our detection component, we generate input-based testing inputs for CIFAR-10 and CIFAR-100 training and test datasets for all three AdNNs. Next, we calculate the gradients of the weights with respect to the intermediate loss function for the training section of datasets. For all three AdNNs, we consider the weights of the first layer of the AdNN. Specifically for BlockDrop, we calculate the intermediate loss function of the policy network, whereas, for RANet and BranchyNet, we use the first exit's loss function (Section 5.1). After calculating the weight gradients for the training section of datasets, we label them as either original or testing inputs and use them for training an SVM binary classification model for each AdNN. Finally, we test the SVM classifiers on the gradients generated by the original and input-based testing input for the testing section of datasets.

Table 8 shows the detection accuracy (%) and AUC score [3] of our gradient-based input detection technique. AUC score computes the area under the Receiver Operating Characteristic Curve (ROC AUC) [3] and measures the efficacy of any binary classifier, with higher AUC scores corresponding to better classifiers. The results show that in 5 out of 6 scenarios, both AUC score is higher than 0.8 and the detection accuracy is higher than 80 percent, showing our approach's efficacy. Additionally, we also report the accuracy drop of the AdNNs due to false positives from the detection system. We observe that the accuracy drop is minimal in all cases except for the CIFAR-100 BlockDrop model.

Table 8: Detection Accuracy (%), AUC score, Accuracy Drop percentage, Adversarial Energy Decrease Percentage, and Benign Energy Increase Percentage of the Gradient-based input detection incorporated in to various AdNN models

AdNN	Dataset	Detection (%)	AUC	Acc Drop(%)	Adv Eng Dec (%)	Ben Eng Inc (%)
RANet	CIFAR-10	92.50	0.924	11.50	77.10	26.80
	CIFAR-100	81.60	0.816	3.90	84.30	17.74
BranchyNet	CIFAR-10	99.90	0.999	0.01	84.40	17.90
	CIFAR-100	99.80	0.998	0.01	87.50	15.60
BlockDrop	CIFAR-10	94.39	0.943	3.30	95.00	6.25
	CIFAR-100	66.90	0.666	30.40	92.80	7.60

Furthermore, to demonstrate the benefits of our gradient-based detection system from the energy perspective, we also report the average energy decrease percentage for an adversarial input (Adv Eng Dec) and average energy increase percentage for a benign input (Ben Eng Inc). We observe that our detection system can significantly reduce energy consumption induced by adversarial inputs (up to 95%) while introducing minimal energy burden as evidenced by a low increase in consumption for benign inputs.

4

We discuss the alternative defense for AdNNs, the adaptability of AdNNs on different datasets, and the relationship between block activation and energy consumption of AdNNs. Also, we discuss correlation between measured and estimated energy consumption, and the correlation between energy consumption increased by different techniques.

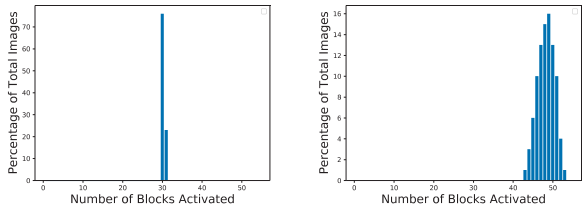
4.1 ve Defense.

We have investigated the application of adversarial training as an alternative defense mechanism against EREBA generated energy-surging inputs. To understand the effect of adversarial training on AdNNs, we use the EREBA generated testing inputs for CIFAR-10 dataset to retrain the original AdNNs. We used Input-based testing inputs generated from 1000 images of the CIFAR-10 training dataset as the training set and retrained the BranchyNet and RANet AdNNs for 150 epochs with a learning rate same parameters as the initial training. We generate the test set using a batch of 600 images from the CIFAR-10 test dataset. We found that adversarial training does not increase the energy robustness for all AdNN models. Specifically, RANet is easier to improve using adversarial training compared to BranchyNet. Due to space constraints, we have reported the results for Adversarial training in our website [4].

4.2 of AdNNs

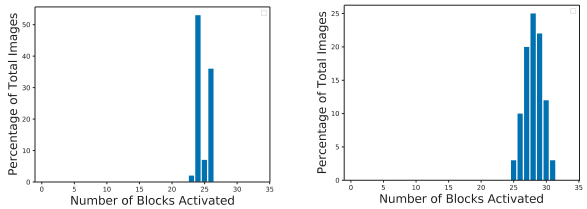
In our observations, we see that AdNNs may not be adaptive under all circumstances. Each AdNN can decrease its FLOPs count; however, this may not always result in concrete energy consumption patterns. Figures 6a, 6b, 6c, and 6d show BlockDrop (ResNet-110) and SkipNet models' adaptability on CIFAR-100 and ImageNet

[4] https://sites.google.com/view/ereba/home



(a) Adaptability of BlockDrop (RN-110) on CIFAR-100 Dataset

(b) Adaptability of SkipNet on CIFAR-100 Dataset



(c) Adaptability of BlockDrop (RN-110) on ImageNet Dataset

(d) Adaptability of SkipNet on ImageNet Dataset

Figure 6: Adaptability of AdNNs on ImageNet and CIFAR-100 dataset.

datasets. We can see that the BlockDrop's adaptability (the difference between the highest and lowest number of activated blocks) for both datasets is limited (less than 4). For SkipNet, the range of adaptability is better than BlockDrop's range.

7.3 Comparison to Misclassification Attacks

We can assume that the energy-saving mechanisms of Conditional-Skipping networks like BlockDrop would classify an image into $N-1$ categories. N is the number of blocks in the DNN, with each image in category i activating $i+1$ (the first block is always active) blocks. Whereas for the case of Early-Termination networks such as BranchyNet and RANet, they classify an image into E categories where E is the number of exits in the network. However, due to noise from the system environment, our Estimator model cannot differentiate between all the classes. Figure 7 shows the scatter plots between these image classes and their energy consumption for each AdNN model. We can see the step-wise pattern in the plot for all the AdNN models. The Estimator model is trained to learn this pattern of energy consumption of AdNNs.

7.4 Correlation between Actual Energy Consumption and Estimated Energy Consumption

To illustrate the correlation between energy consumption predicted by the estimator model and original energy consumption, we use the Pearson Correlation Coefficient (r) [1] and correlation p-value. If two sets of values are correlated, the r value would be significant, and the p-value will be low.

For CIFAR-100 dataset, the r values are 0.38, 0.17, and 0.31 for RANet, BlockDrop, and BranchtNet models, respectively, where all the p-values are less than 0.0005. These results conclude that the values are correlated.

For CIFAR-10 dataset, the value of r for RANet is 0.021; however p-value is 0.03; therefore, it is more likely that the values are correlated. For BlockDrop model, the value of r and p-value are 0.004 and 0.66, which suggests that the values are less likely to be correlated. But if we consider only the inputs whose energy consumption is higher than the 75th percentile value, the p-value becomes 0.14, suggesting a correlation. Therefore, if the estimator model can accurately predict high energy-consuming inputs (i.e., differentiate clearly between low/mid and high energy-consuming inputs), we can use the estimator model to generate energy-expensive testing inputs.

7.5 Correlation of Increase of Energy Consumption between Different Techniques

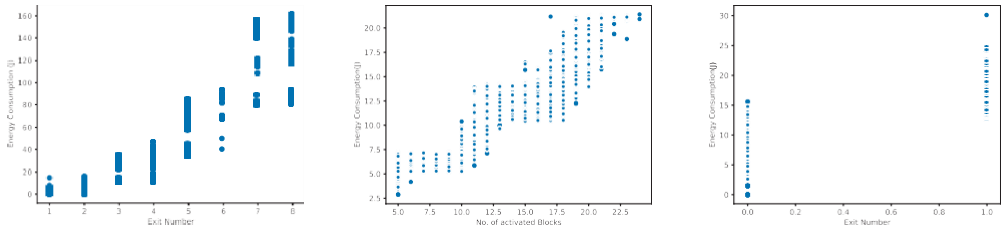
In this section, we try to explore the correlation between energy consumption modified by Input-based testing and baseline techniques. We use Pearson Correlation for that purpose. Pearson Correlation is one of the metrics that can find the strength of the relationship between two variables. For CIFAR-100 data, we show the Table 9 that represents the Pearson Correlation Coeff (r) and p-value between the percentage of energy consumption increased by Input-based testing inputs and energy consumption increased by baseline technique generated inputs. It can be noticed that for most of the cases, the energy increase percentages are less likely to be correlated. Only for BranchyNet, we can find a significant negative correlation between Input-based and Perturbation-induced energy consumption increase.

Table 9: Correlation between the percentage of energy consumption increased by Input-based testing inputs and percentage energy consumption increased by baseline technique generated inputs for different models for CIFAR-100 data

Models \ Baseline	r(Perturb)	p-value(Perturb)	r(SURRG)	p-value(SURRG)
RANet	0.142	0.328	-0.04	0.976
BlockDrop	-0.007	0.908	-0.037	0.544
BranchyNet	-0.296	0.133	-0.118	0.556

5 ATED WORKS

AdNNs. Among Conditional-skipping models, Hua *et al.* [16] and Gao *et al.* [8] explore channel gating to determine computational blind spots for channel-specific regions unessential to classification. Liu *et al.* [20] propose a new type of AdNN which utilizes reinforcement learning to achieve selective execution of neurons. SkipNet [38] uses gating techniques to skip residual blocks. On the other hand, Graves *et al.* [11], Figurnov *et al.* [7], and Teerapittayanon *et al.* [33] propose SACT and BranchyNet respectively, which are Early-termination AdNNs. SACT terminates the computation within a residual block early based on intermediate outputs, while BranchyNet uses separate exits within network for early termination. Cascading multiple DNNs with various computational



(a) (b) (c)

Figure 7: Energy consumption of (a) RANet, (b) BlockDrop, (c) BranchyNet for CIFAR-10 Training Set

costs through a single computation unit to decide which DNN to execute has been proposed. The cascading models use various techniques such as termination policy [2], reinforcement learning [12], and gating techniques [24] to achieve early termination.

Adversarial Examples. Adversarial Examples are the synthesized inputs that is able to modify the prediction of the ML model. Szegedy *et al*. [32] and Goodfellow *et al.* [10] propose white-box adversarial attacks on convolutional neural networks. Papernot *et al*. [28] have used surrogate model to attack a DNN in black-box setting. Liu *et al.* [21] use ensemble of multiple white-box models to generate adversarial examples, which can attack black-box models. Ilyas *et al.* [17] use evolutionary search strategies to estimate the gradient of a model to attack black-box models.

However, all these attacks focus on changing the prediction and do not concentrate on increasing test time. ILFO [13] is the first work to attack a DNN by increasing the energy consumption of the model. However, ILFO uses white-box setting and does not have transferability. Therefore, ILFO can not be used for black-box attack.

Next, DeepSloth [15] uses modified PGD to attack against Early-termination AdNNs using the confidence scores in each exit. However, DeepSloth can not be used against Conditional-skipping AdNNs. Also, DeepSloth provides a study about the transferability of the attack. The study considers the efficacy of the Early-termination models as the transferability metric. However, we propose a more systematic transferability study by introducing of metrics like ETP and ITP.

DNN Testing. Multiple testing methods have been proposed recently to test DNNs. DeepGauge [22] is proposed based on a test criteria set that verifies the corner neuron activation values. DeepXplore [30] proposes to cover each neuron's binary activation status and use neuron coverage to test DNNs. DeepTest [37] tests autonomous driving cars by using neuron coverage. Recently, DeepHunter [43] proposes to use coverage-guided fuzz testing on DNNs. EREBA evaluates the energy-robustness of AdNNs in a black-box setting unlike the aforementioned techniques, which are focused on testing the accuracy-robustness of traditional DNNs in white-box setting

6

In this paper, we have proposed practical black-box testing methods to evaluate energy robustness of AdNNs. The core idea behind the technique is to create inputs which increase the energy consumption of AdNN to a higher level. To achieve this goal, we have presented EREBA [5], where we have proposed two types of testing: Universal testing and Input-based testing. To our knowledge, we are the first to explore black-box testing on AdNNs. Test inputs generated by EREBA can improve the energy robustness of AdNNs. Finally, this paper also analyzes the behavior of AdNNs and suggests model improvement strategies.

ACKNOWLEDGEMENT

This work was partially supported by Siemens Fellowship and NSF grant CCF-2146443.

[5] https://sites.google.com/view/ereba/home

REFERENCES

[1] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson Correlation Coefficient. In *Noise Reduction in Speech Processing*. Springer, 37–40.

[2] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. 2017. Adaptive Neural Networks for Efficient Inference. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 527–536.

[3] Andrew P. Bradley. 1997. The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recogn.* 30, 7 (July 1997), 1145–1159.

[4] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 39–57.

[5] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Improving Black-box Adversarial Attacks with a Transfer-based Prior. In *Advances in Neural Information Processing Systems*. 10932–10942.

[6] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector Networks. *Machine learning* 20, 3 (1995), 273–297.

[7] Michael Figurnov, Maxwell D Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. 2017. Spatially Adaptive Computation Time for Residual Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1039–1048.

[8] Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng-zhong Xu. 2018. Dynamic Channel Pruning: Feature Boosting and Suppression. *arXiv preprint arXiv:1810.05331* (2018).

[9] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. 2018. ImageNet-trained CNNs Are Biased Towards Texture; Increasing Shape Bias Improves Accuracy and Robustness. *arXiv preprint arXiv:1811.12231* (2018).

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572* (2014).

[11] Alex Graves. 2016. Adaptive Computation time for Recurrent Neural Networks. *arXiv preprint arXiv:1603.08983* (2016).

[12] Jiaqi Guan, Yang Liu, Qiang Liu, and Jian Peng. 2017. Energy-efficient Amortized Inference with Cascaded Deep Classifiers. *arXiv preprint arXiv:1710.03368* (2017).

[13] Mirazul Haque, Anki Chauhan, Cong Liu, and Wei Yang. 2020. ILFO: Adversarial Attack on Adaptive Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14264–14273.

[14] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *Proceedings of the International Conference on Learning Representations* (2019).

[15] Sanghyun Hong, Yiğitcan Kaya, Ionuţ-Vlad Modoranu, and Tudor Dumitraş. 2020. A Panda? No, It's a Sloth: Slowdown Attacks on Adaptive Multi-Exit Neural Network Inference. *arXiv preprint arXiv:2010.02432* (2020).

[16] Weizhe Hua, Yuan Zhou, Christopher M De Sa, Zhiru Zhang, and G Edward Suh. 2019. Channel Gating Neural Networks. In *Advances in Neural Information Processing Systems*. 1884–1894.

[17] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box Adversarial Attacks with Limited Queries and Information. *arXiv preprint arXiv:1804.08598* (2018).

[18] Sven Köhler, Benedict Herzog, Timo Hönig, Lukas Wenzel, Max Plauth, Jörg Nolte, Andreas Polze, and Wolfgang Schröder-Preikschat. 2020. Pinpoint the Joules: Unifying Runtime-Support for Energy Measurements on Heterogeneous Systems. In *2020 IEEE/ACM International Workshop on Runtime and Operating Systems for Supercomputers (ROSS)*. IEEE, 31–40.

[19] Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. (2009).

[20] Lanlan Liu and Jia Deng. 2018. Dynamic Deep Neural Networks: Optimizing Accuracy-efficiency Trade-offs by Selective Execution. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[21] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into Transferable Adversarial Examples and Black-box Attacks. *arXiv preprint arXiv:1611.02770* (2016).

[22] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, et al. 2018. Deepgauge: Multi-granularity Testing Criteria for Deep Learning Systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. 120–131.

[23] Lei Ma, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Felix Juefei-Xu, Chao Xie, Li Li, Yang Liu, Jianjun Zhao, et al. 2018. Deepmutation: Mutation Testing of Deep Learning Systems. In *2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 100–111.

[24] Feng Nan and Venkatesh Saligrama. 2017. Adaptive Classification for Prediction Under a Budget. In *Advances in Neural Information Processing Systems*. 4727–4737.

[25] Nvidia. 2017. *Nvidia TX2 User Manual*.

[26] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. In *Advances in Neural Information Processing Systems*. 13991–14002.

[27] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in Machine Learning: from Phenomena to Black-box Attacks using Adversarial Samples. *arXiv preprint arXiv:1605.07277* (2016).

[28] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical Black-box Attacks against Machine Learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 506–519.

[29] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The Limitations of Deep Learning in Adversarial Settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 372–387.

[30] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. Deepxplore: Automated Whitebox Testing of Deep Learning Systems. In *proceedings of the 26th Symposium on Operating Systems Principles*. 1–18.

[31] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2009. *Dataset shift in machine learning*. The MIT Press.

[32] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing Properties of Neural Networks. *arXiv preprint arXiv:1312.6199* (2013).

[33] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2016. Branchynet: Fast Inference via Early Exiting from Deep Neural Networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2464–2469.

[34] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2017. Distributed Deep Neural Networks over the Cloud, the Edge and End Devices. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 328–339.

[35] Tensorflow. 2009. Tensorflow Deep Learning Framework. https://www.tensorflow.org/datasets/catalog/cifar10.

[36] Tensorflow. 2009. Tensorflow Deep Learning Framework. https://www.tensorflow.org/datasets/catalog/cifar100.

[37] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. Deeptest: Automated Testing of Deep-neural-network-driven Autonomous Cars. In *Proceedings of the 40th international conference on software engineering*. 303–314.

[38] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. 2018. Skipnet: Learning Dynamic Routing in Convolutional Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 409–424.

[39] Wikipedia. [n.d.]. Peak Signal-to-Noise Ratio. https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio.

[40] Wikipedia. [n.d.]. Structural Similarity Index Measure. https://en.wikipedia.org/wiki/Structural_similarity.

[41] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. 2018. Blockdrop: Dynamic Inference Paths in Residual Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8817–8826.

[42] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with Noisy Student Improves Imagenet Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10687–10698.

[43] Xiaofei Xie, Lei Ma, Felix Juefei-Xu, Minhui Xue, Hongxu Chen, Yang Liu, Jianjun Zhao, Bo Li, Jianxiong Yin, and Simon See. 2019. DeepHunter: a Coverage-guided Fuzz Testing Framework for Deep Neural Networks. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 146–157.

[44] Le Yang, Yizeng Han, Xi Chen, Shiji Song, Jifeng Dai, and Gao Huang. 2020. Resolution Adaptive Networks for Efficient Inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2366–2375.