A critique of the DEEPSEC Platform for Security Analysis of Deep Learning Models

Nicholas Carlini (*Google Brain*)

***Abstract*—A recent platform "DEEPSEC: A Uniform Platform for Security Analysis of Deep Learning Model" purports to "systematically evaluate the existing adversarial attack and defense methods". While the goals of this analysis are laudable, the actual instantiation is fundamentally flawed and highly misleading. This short paper briefly summarizes the ways in which the DEEPSEC analysis fails to in any way to meaningfully measure the power of various attacks and defenses. Specifically, the DEEPSEC framework (1) contains incorrect implementations of attacks and defenses which under-perform by a factor of two when compared to existing baselines; (2) evaluates the *average-case* robustness and not *worst-case* robustnss; (3) does not TODO; and (4) makes sweeping and incorrect conclusions as a result of the errors in data analysis.**

I. OVERVIEW OF DEEPSEC

DEEPSEC is a "uniform platform" to "measure the vulnerability of [deep learning] models" and "conduct comparative studies on attacks/defenses" [**?**]. To do this, the DEEPSEC framework implements many common attacks and defenses with a consistent interface. Ling *et al.* then use this to "systematically evaluate the existing adversarial attack and defense methods" [**?**].

The research community would be well served by such an analysis. When new defenses are proposed, authors must choose which set of attacks to apply in order to perform an evaluation. A systematic evaluation of which attacks have been most effective in the past could help inform the decision of which attacks should be tried in the future. Similarly, when designing new attacks, a comprehensive review of defenses could help researchers decide which defenses to test against.

Unfortunately, the analysis performed in the DEEPSEC report is fundamentally flawed and does not achieve any of these goals. It neither accurately measures the power of attacks not measures the efficacy of defenses. This report summarizes the many ways in which the report is misleading in its results. Performing a correct systematic evaluation of existing attacks and defenses is still an open problem which future work will need to address (perhaps even using the DEEPSEC evaluation framework).

The fact that this paper was accepted at IEEE Symposium on Security and Privacy (one of the premire venues for publishing computer security research) seriously calls into question the ability of this conference to accurately assess the quality of adversarial machine learning research.

II. CRITIQUE OF THE DEEPSEC EVALUATION

While the objective that DEEPSEC sets out to solve is laudible, the actual analysis performed is fundamentally flawed and incorrect in many ways. This section discusses a few of the aspects in which it errs in detail; we do not attempt to exhaustively enumerate all the problems.

A. DEEPSEC *Code and Framework Flaws*

It is exceptionally important when re-implementing prior work to ensure correctness of the reproduction. Unfortunately, we find significant flaws in the authors implementation of FGSM [**?**] (one of the first and simplest attack approaches) and PGD adversarial training [**?**] (a simple idea that is notoriously hard to get right in practice).

FGSM implementation is incorrect. Despite the simplicity of the Fast Gradient Sign Method [cite], it is surprisingly effective at generating adversarial examples on unsecured models. However, Table XIV reports the misclassification rate of FGSM at $\epsilon = 0.3$ on MNIST as 30.4%, significantly less effective than we expected given the results of prior work.

Fortunately, because the authors release their code[1], we are able to investigate this further. We take the authors code and run the the one-line script as described in the README to run the FGSM attack on the baseline MNIST model. Doing this yields a misclassification rate of 38.3% PLUSORMINUS TODO. [2] It is mildly concerning that this number is 25% larger than the value reported in the paper, and we are unable to account for this statistically significant deviation from what the code returns. However, this error is only of secondary concern: as prior work indicates, the success rate of FGSM should be substantially higher.

We therefore compare to the result of attacking with the CleverHans [] framework. Because DEEPSEC is implemented in PyTorch, and CleverHans only supports TensorFlow, we load the DEEPSEC pre-trained PyTorch model weights into a TensorFlow model[3] and generate adversarial examples on this model with the CleverHans [] implementation of FGSM. CleverHans obtains a 61% misclassification rate–**over double** the misclassification rate reported in the DEEPSEC paper. To confirm the results that we obtain are correct we save these adversarial examples and run the original DEEPSEC PyTorch model on them, again finding the misclassification rate is 61%.

[1] TODO

[2] We compute the confidence interval at 95% by running the attack 100 times. The resulting distribution over misclassification rates is approximately normal and so it is therefore meaningful to report the confidence interval as two standard deviations.

[3] To validate that this process does not change functionality, we verify that the models agree on every example in the training and testing set not only in prediction, but in the confidence of their predictions. We observe 100% (perfect) agreement.

We are at this time unable to explain how DEEPSEC incorrectly implemented FGSM, however the fact the simplest attack is implemented incorrectly is deeply concerning.

We release our code [4] which demonstrates this error.

The remainder of this commentary on DEEPSEC therefore discusses *only* the methodology and analysis, and not any specific numbers which may or may not be trustworthy.

PGD adversarial training is implemented incorrectly. While the idea of adversarial training is straightforward—generate adversarial examples during training and train on those examples until the model learns to classify them correctly— in practice it is difficult to get right. The basic idea has been independently developed at least twice TODO cite and was the focus of several papers TODO CITE before all of the right ideas were combined by Madry *et al.* to form the strongest defense to date [**?**]. We identify at least three flaws in the re-implementation of this defense after a cursory analysis:

- **Incorrect loss function.** The loss function used in the original paper is TODO whereas this paper mixes adversarial examples and original examples to form the loss TODO. The authors do not justify this decision.
- **Incorrect model architectures.** In the original paper, the authors make three claims for the novelty of their method. One of these claims states "To reliably withstand strong adversarial attacks, networks require a significantly larger capacity than for correctly classifying benign examples only." [**?**] The code that re-implements this defense does not follow this advice and instead uses a substantially smaller model than recommended. The authors do not justify this decision.
- **Incorrect hyperparameter settings.** The original paper trains their MNIST model for 83 epochs of training; In contrast, the authors here train for only 20 epochs ($4\times$ fewer iterations). The authors do not justify this decision.

Possibly because of these implementation differences, the DEEPSEC report finds (incorrectly) that a more basic form of adversarial training performs better than PGD adversarial training.

We did not review the re-implementation of any of the other defenses; the fact that we do not report any other issues is not because there are or are not further issues

B. Methodological Flaws

The methodology of the evaluation contains significant flaws that severly limit ones ability to draw any meaningful conclusions.

Attacks are not run on defenses in an all-pairs manner. The only meaningful metric for evaluating a defense is by measuring the effictiveness of attacks which run against it.

As a point of comparison, imagine that I were designing a new computer architecture that was designed to be secure memory corruption vulnerabilities. I do this by taking a pre-existing computer architecture and instead of designing it as little-endian or big-endian, implement some new "middle-endian" where the least significant byte is put in the middle of the word. This crazy new architecture would appear to be perfectly robust against all existing malware. However it would be fundamentally incorrect to call this new computer architecture "more secure": the only thing that we have done is superficially broken existing exploits from working on our new system.

This basic flaw completely undermines the purpose of a security evaluation. Notice that this type of analysis is not useless and does tell us *something*: the analysis performed tells us something useful about the ability for these attacks to *transfer* [**?**] and for the models to defend against transferability attacks [**?**]. If the authors had made this observation and drawn the conclusions from this perspective, then at least the fundamental idea behind the table would have been correct. (None of the following errors would be resolved, still.)

However, it is, TODO.

Even worse, the DEEPSEC code itself does not support the ability to run any of the attacks on a new defense model. While the code TODO

Security analysis violates threat models of defenses. Most defenses contain a *threat model* as a statement of the conditions under which they attempt to be secure.

Attack analysis improperly measures distortions not being optimized for. Most defenses contain a *threat model* as a statement of the conditions under which they attempt to be secure.

Discrepencies between tables, text, and code. The paper contains numerous discrepencies between the code and constants given with the paper. For example, Table XIII states that on CIFAR-10 the R+FGSM attack [] was executed with $\epsilon = 0.05$ and $\alpha = 0.05$ for CIFAR-10 whereas the README in the Attack module of the open source code suggests $\epsilon = 0.1$ and $\alpha = 0.5$. Table XIII states that the "box" constraint for CWL2 is set to $-0.5, 0.5$ but in the code the (correct) values of $0.0, 1.0$ are used. Other hyperparameters are completely missing (e.g., number Table XIII does not give the number of iterations used for any of the gradient-based attacks). This is especially confusing when the default values differ from the original attack implementations; for example, this code sets the number of binary search steps for CW2 to 5 (and does not state this in the paper) whereas the original code uses the value 10; fortunately, this setting often has only a minimal impact on accuracy.

Epsilon values studied are too large to be meaningful. On at least two counts the authors chose l_∞ distortion bounds that are not well motivated.

- Throughout the paper the authors study a CIFAR-10 distortion of $\epsilon = 0.1$ and $\epsilon = 0.2$. This value is $3\times$ (or $6\times$) larger than what is typically studied in the literature. CIFAR-10 images that are perturbed with noise of distortion 0.1 are often difficult for humans to correctly classify; we are aware of no other work which studies CIFAR-10 robustness at this extremely high distortion bound.

[4] TODO

- The authors study l_∞ distortion bounds as high as $\epsilon = 0.6$ in Table VII on both MNIST and CIFAR-10, a value that is so high that any image can be converted to solid grey (and then past). The entire purpose of bounding the l_∞ norm of adversarial examples is to ensure that the actual true class has not changed. Choosing a distortion bound so large that all images can be converted to a solid grey image fundamentally misunderstands the purpose of the distortion bound.

Detection defenses set per-attack thresholds. In Table VI the authors analyze three different defense techniques. In this table, the authors report the true positive rate and false positive rate of the defenses against various attacks. In doing so, the authors vary the detection threshold on a per-attack basis:

> "we try our best to adjust the FPR values of all detection methods to the same level via fine-tuning the parameters."

When performing a security analysis between the attacker and defender it is always important to recognize that one of the players goes *first* and commits to an approach, and then the second player goes *second* and tries to defeat the other. In working with adversarial example defenses, it is the defender who commits first [**?**] and the attacker who then tries to find instance that evades the defense.

As such, it is meaningless to allow the defender to alter the detection hyperparameters depending on which attack will be encountered. If the defender knew which attack was going to be presented, they could do much better than just selecting a different hyperparameter setting for the detection threshold.

Even still, despite this claim that the authors normalize the detection rate to be "the same level", in actuality the false positive rates presented in the table vary between 1.5% and 9.0%. Comparing the true positive rate of two defenses when the corresponding false positive vary by **a factor of six** is meaningless. Worse yet, computing the *mean* TPR across a range of attacks when the FPR by a factor of six results in a completely uninterprable value.

Attack success rate *decreases* with distortion bound. It is a basic observation that when given strictly more power, the adversary should never do worse. However, in Table VII the authors report that MNIST adversarial examples with their l_∞ norm constrained to be less than 0.2 are **harder** to detect than when constrained to be within 0.5. The reason this table shows this effect is that FGSM, a single-step method, is used to generate these adversarial examples.

Reporting *success rate* of *unbounded* attacks is meaningless. Two of the attacks presented (EAD TODO and CW2 TODO BLB TODO) are *unbounded* attacks: rather than finding the "worst-case" (i.e., highest loss) example within some distortion bound, they seek to find the *closest* input subject to the constraint that it is misclassified. Unbounded attacks should always reach 100% "success" eventually, if only by actually changing an image from one class into an image from the other class; the correct and meaningful metric to report for unbounded attacks is the distortion required.

C. Analysis Flaws

The DEEPSEC report relies on *averages* for summarizing results, instead of the *minimum* or *maximum*. Perhaps the one key factor that differentiates security (and adversarial robustness) from other general forms of robustness is the worst-case mindset from which we evaluate.

Using the mean over various attacks to compute the "security" of a defense completely misunderstands what it means to perform a security evaluation in the first place. For example, the authors bold the column for the NAT defense [**?**] when evaluated on CIFAR-10 because it gives the highest "average security" against all attacks. However, this is fundamentally the incorrect evaluation to make: the only metric that matters in security is how well a defense witstands attacks *targeting that defense*. And in this setting, the alternate adversarial training approach of Madry *et al.* [cite] is strictly stronger. [5]

"According to the results, LID has the highest average TPRagainst all kinds of AEs." (IV. C. 2, p.10) WHAT IS THE AVERAGE FOR?!?

Computing the average over *different threat models* is meaningless. In essence, the authors commiting one of the most elementary flaws in mathematics and forgetting the units.

The DEEPSEC report evaluates model accuracy, not attack success rate, for targeted adversarial examples. TODO

Within one threat model, comparing attack effictiveness is done incorrectly. Using the data provided, it is not possible to compare the efficacy of different attacks across models. Imagine we would like to decide whether LLC or ILLC was the stronger attack.

Superficially, we might look at the "Average" column and see that the average model accuracy under LLC is 39.4% compared to 58.7% accuracy under ILLC. However, as discussed earlier computing averages over different defenses is meaningless. Fortunately, we can observe that on all models except one, LLC reduces the model accuracy more than ILLC does, often by over twenty percentage points.

Areasonable reader might therefore conclude (incorrectly!) that LLC is the stronger attack. Why is this conclusion incorrect? The LLC attack only succeeded 134 times out of 1000 times on the baseline CIFAR-10 model. Therefore, when the authors write that the accuracy of PGD adversarial training under LLC is 61.2% what this number means is that 38.8% of adversarial examples that are effective on the baseline model are also effective on the adversarially trained model. How the model would perform on the other 866 examples is not reported. In contrast, when the model is evaluated on the ILLC attack, because this attack succeeded on all 1000 examples for the baseline model, the 83.7 accuracy obtained by adversarial training is inherently incomparable to the the 61.2% value.

Average model accuracy under different attacks imply weak attacks are strong. Average column of Table V makes it look like FGSM is the best attack.

[5] TODO can I get alex to agree.

D. Comments on the Conclusions Drawn

"L_∞ attacks are much more transferable than others (i.e., L_2 and L_0 attacks)." Only one L_0 attack.

> most defense-enhanced models increase theirclassification accuracy against existing attacks (IV. C. 1, p.9)

"Although there have been many sophisticated defenses and strong attacks, it is still an open problem whether or to what extent the state-of-the-art defenses can defend against attacks." (IV. C., p.9)

"we suggest that all state-of-the-art defenses are more or less effective against existing attacks." (IV. C. 2, p.9)

"It is not the case that AEs with high magnitudeof perturbation are easier to be detected." (IV. C. 2, p.10)

"All detection methods show comparable discrimi-native ability against existing attacks." (IV. C. 2, p.10)

III. Conclusion

Researchers who set out to reproduce prior work must hold themselves to an exceptionally high standard. Because survey papers hold significant power impact the communities knowledge base (especially when accepted for publication at first-rate conferences), researchers reproducing prior work must ensure that the results are accurate in order to not promote misinformation. Unfortunately, the analysis of DEEPSEC [**?**] falls far below the necessary bar and makes significant and fundamental flaws across all areas of its evaluation.

While the motivation behind the DEEPSEC framework is TODO, The overall high-level approach to begin with is incorrect by design: by not actually running each attack on the corresponding defense, TODO. Worse yet, the implementations of even the simplest attacks and defenses appears incorrect. Even putting that oversight aside, by using the *average case* efficacy of attacks and defenses to draw conclusions, TODO.

Future work **should not** follow the evaluation approach taken by this paper. The analysis results of Tables V, VI, and VII should be completely disregarded except insofar as they analyze the transferability of adversarial examples. Most of the conclusions drawn from the analysis are false (e.g., while the authors claim that "all state-ofthe-art defenses are more or less effective against existing attacks" TODO) .

Improperly performed experiments are worse than experiments not performed.

References

[1] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," *AISec*, 2017.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014.

[3] G. Tao, S. Ma, Y. Liu, and X. Zhang, "Attacks meet interpretability: Attribute-steered detection of adversarial samples," in *Advances in Neural Information Processing Systems*, 2018, pp. 7728–7739.