ROBUSTNESS TO ADVERSARIAL EXAMPLES THROUGH AN ENSEMBLE OF SPECIALISTS

Mahdieh Abbasi & Christian Gagné
Computer Vision and Systems Laboratory, Electrical and Computer Engineering Department
Université Laval, Québec (Québec), Canada
mahdieh.abbasi.1@ulaval.ca, christian.gagne@gel.ulaval.ca

1 INTRODUCTION

Due to the recent breakthroughs achieved by Convolutional Neural Networks (CNNs) for various computer vision tasks (He et al., 2015; Taigman et al., 2014; Karpathy et al., 2014), CNNs are highly regarded technology for inclusion into real-life vision applications. However, CNNs have a high risk of failing due to adversarial examples, which fool them consistently with the addition of small perturbations to natural images, undetectable by the human eyes.

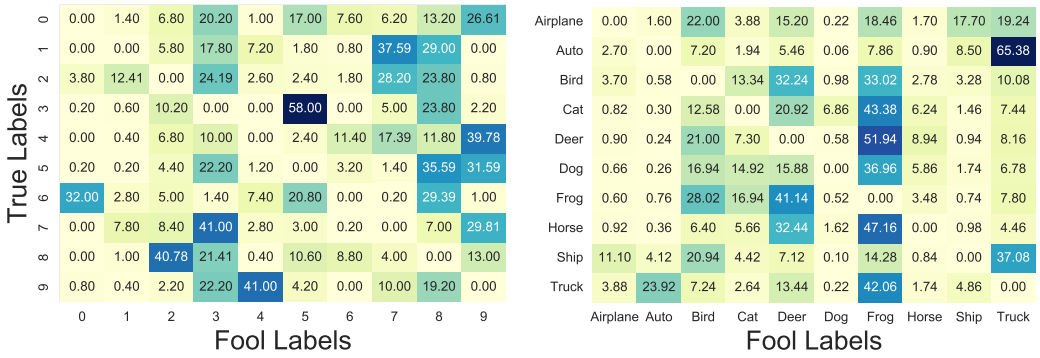
To mitigate this risk, it has been proposed to train CNNs on both clean training samples and corresponding adversarial examples, generated by some existing algorithms (Goodfellow et al., 2014; Szegedy et al., 2013; Moosavi-Dezfooli et al., 2016; Sabour et al., 2015). Although such trained CNNs are robust to specific types of adversaries, they are not necessarily protected from all possible types. To increase the robustness of CNNs, it has been proposed to train them on a *diverse* set of adversaries, generating adversarial examples for any single images with various algorithms (Rozsa et al., 2016). However, it is still possible to produce other types of adversaries, uncovered by the current set, impacting significantly the reliability of CNNs. Moreover, training on some type of adversaries has been demonstrating to harm the performance on clean test samples (Jin et al., 2015; Moosavi-Dezfooli et al., 2016).

We are rather considering recognition of adversarial examples as an open set recognition problem, where *unknown* samples should be detected and rejected by the underlying models. Bendale & Boult (2016) have adapted CNNs by adding an extra layer designed to recognize the unknown samples, which can be either from unknown classes or fooling adversarial instances from Nguyen et al. (2015). However, as mentioned by the authors, the method fails to detect hard adversaries where the target class and the true class of an adversary are close together, like those generated by Fast Gradient Sign (FGS) (Goodfellow et al., 2014) and DeepFool (DF) (Moosavi-Dezfooli et al., 2016).

We are proposing to use an ensemble of diverse specialists, where speciality is defined according to the confusion matrix. Indeed, we observed that for adversarial instances originating from a given class, labeling tend to be done into a small subset of (incorrect) classes. Therefore, we argue that an ensemble of specialists should be better able to identify and reject fooling instances, with a high entropy (i.e., disagreement) over the decisions in the presence of adversaries. Experimental results obtained confirm this interpretation that a rejection mechanism can provide a means of rendering the system more robust to adversarial examples, rather than trying to classify them properly at any cost.

2 SPECIALISTS+1 ENSEMBLE

Ensemble construction The confusion matrices of FGS adversaries (Fig. 1) reveals that samples from each class have a high tendency of being fooled toward a limited number of classes. From these confusion matrices of training adversaries we define subsets of classes, similarly to Hinton et al. (2015) on training clean samples. Considering a classification problem of K classes ($C = \{c_1, c_2, \ldots, c_K\}$), each row of the confusion matrix is used to identify two subsets of classes: 1) the confusing target subset for class c_i (subset U_i), which is built by adding classes sequentially in decreasing c_i-related confusion values order until at least 80 % of confusions are covered, and 2) the remaining classes with lower confusion, formed as subset $U_{i+K} = C \setminus U_i$. Duplicate subsets should be ignored, although we encountered none of them for MNIST and CIFAR-10.



(a) MNIST Confusion Matrix (b) CIFAR-10 Confusion Matrix

Figure 1: Confusion matrices of adversaries for (a) MNIST and (b) CIFAR-10. These matrices have been computed from 5000 randomly selected FGS training adversaries (500 per class).

Algorithm 1 Voting Mechanism

Input: Ensemble $\mathcal{H} = \{h^1, \ldots, h^M\}$ with $h^j \in \mathbb{R}^K$, label subsets $\mathcal{U} = \{U_1, \ldots, U_M\}$, input $\mathbf{x}$
Output: Final prediction $\bar{h}(\mathbf{x}) \in \mathbb{R}^K$

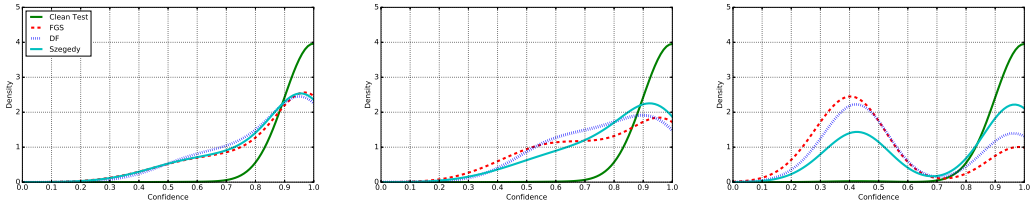
1: $v_k(\mathbf{x}) \leftarrow \sum_{j=1}^{M} \mathbb{I}[c_k = \text{argmax}_{i=1}^{K} h_i^j(\mathbf{x})],\ k = 1, \ldots, K$
2: $k^* \leftarrow \text{argmax}_{k=1}^{K} v_k(\mathbf{x})$
3: **if** $v_{k^*}(\mathbf{x}) = K + 1$
4: $\quad \mathcal{S} \leftarrow \{h^i \in \mathcal{H} \mid c_{k^*} \in U_i\}$
5: $\quad \bar{h}(\mathbf{x}) \leftarrow \frac{1}{K+1} \sum_{h^i \in \mathcal{S}} h^i(\mathbf{x})$
6: **else**
7: $\quad \bar{h}(\mathbf{x}) \leftarrow \frac{1}{M} \sum_{h^i \in \mathcal{H}} h^i(\mathbf{x})$
8: **return** $\bar{h}(\mathbf{x})$

For each of these class subsets, a specialist CNN is trained on samples from the associated classes, instances from the other classes being ignored. The ensemble also includes a generalist CNN trained on the complete labels (subset U_{2K+1}), hence the name "specialists+1 ensemble".

Voting mechanism Using the generalist to activate the related specialists is not possible as it is usually being fooled by adversaries. In Algorithm 1, we propose a voting mechanism to compute the final prediction. As each class c_i appears $K + 1$ times in M subsets of classes, *the maximum expected number of votes* to class c_i is $K + 1$. Also, we define $v_i(\mathbf{x})$ as *the actual number of votes* to class c_i for a given input image $\mathbf{x}$ (the equation in line 1 of the algorithm 1). If only one class has its actual number of votes equal to its maximum expected number of votes, i.e., $v_i(\mathbf{x}) = K + 1$, it means that all K related specialists and the generalist agree to vote to the winner class. Then only those CNNs voting for the winner class should be activated in order to compute the final prediction. Otherwise, if none of the classes obtain their maximum expected number of votes, it means that at least one of the individuals was fooled. So, some votes are incorrectly distributed between different classes. In the presence of such entropy, where there is no winner class, all of the individuals should be activated to compute the final prediction.

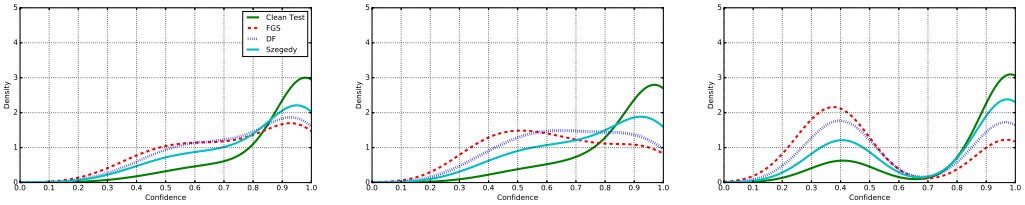
3 Empirical Evaluation

Networks and datasets Similarly to Hinton et al. (2012), we used CNNs with three convolutional layers having 32, 32, and 64 filters respectively, and a fully-connected layer followed by a softmax. The networks are trained on usual training and testing sets of MNIST and CIFAR-10, without any data augmentation. See the Appendix section for full details on the network architecture and hyper-parameters used for each dataset. Note that all CNNs presented in the experiments have an identical architecture.



(a) Naive CNN* (b) Pure ensemble (c) Specialists+1

Figure 2: Confidence densities on MNIST: (a) naive CNN*, (b) pure ensemble, and (c) specialists+1.



(a) Naive CNN* (b) Pure ensemble (c) Specialists+1

Figure 3: Confidence densities on CIFAR-10: (a) naive CNN*, (b) pure ensemble, and (c) specialists+1.

Experiments We compared our proposed specialists+1 ensemble with a pure ensemble, which consists of 5 generalist CNNs with different random initializations, and a naive CNN*, whose weights initialization is different from GA-CNN (i.e., the CNN used to generate the adversaries). Using this GA-CNN, three types of adversaries, namely FGS, DF, and Szegedy et al. (2013) adversaries, are generated for correctly classified clean test samples.

Naive CNN*, pure ensemble, and specialists+1 ensemble are compared according to their distributions of confidence on correctly classified clean test samples and their corresponding adversaries for MNIST and CIFAR-10 in Fig. 2 and 3, respectively. According to these observations for MNIST, specialists+1 successfully provides significantly lower confidence for most of the misclassified adversaries, regardless of their types, than naive CNN* and the pure ensemble. Also, as it can be seen from Fig. 2(a), 2(b), and 2(c) that the distribution of confidence on MNIST correctly classified clean test samples by these three frameworks are roughly similar. For CIFAR-10, we observed the same behavior as MNIST for adversaries (Fig. 3). But for correctly classified clean test samples, specialists+1 shifts some of these samples to lower confidence (the green curve in Fig. 3(c)).

Although the specialists+1 is not trained from any adversaries, it appears able to automatically reduce the confidence of predictions for most of the misclassified adversaries, regardless of their types, while preserving up to some point the confidence on clean samples. However, it reduces the confidence of a few clean test samples. Therefore, developing a learning model that is not confident about unknown samples but yet is confident about known samples can be a used to identify and reject adversaries. We depicted the effect of rejecting low confidence adversaries on the error rates of different types of adversaries in Fig. 5, in Appendix due to space consideration.

Conclusion In brief, without training from adversaries and by leveraging diversity in ensembles by a specialization over the labels, the specialists+1 ensemble approach is able to better discriminate between legitimate samples and adversarial instances. The approach is better at rejecting adversaries while accepting clean samples based on confidence, compared to the pure ensemble and Naive CNN*. That is an important matter in order to increase robustness of CNNs to carefully crafted attacks, preferring to refuse processing suspicious instances rather than being fooled by carefully crafted attacks. As future work, we will compare our approach with CNN explicitly trained to being robust to specific types of adversaries.

ACKNOWLEDGMENTS

This work was made possible through funding from NSERC-Canada, MITACS, and E Machine Learning Inc. Computational resources were provided by Compute Canada / Calcul Québec and by a GPU grant from NVIDIA. The authors are also grateful to Annette Schwerdtfeger for proofreading this manuscript.

REFERENCES

Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1563–1572, 2016.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. Robust convolutional neural networks under adversarial noise. *arXiv preprint arXiv:1511.06306*, 2015.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436, 2015.

Andras Rozsa, Ethan M Rudd, and Terrance E Boult. Adversarial diversity and hard positive generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 25–32, 2016.

Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J Fleet. Adversarial manipulation of deep representations. *arXiv preprint arXiv:1511.05122*, 2015.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, 2014.

A APPENDIX

A.1 EXPERIMENTAL PROCEDURES

We consider a CNN with three convolutional layers and one fully connected layer, where each convolutional layer is interlaced with ReLU, local contrast normalization, and a pooling layer. For regularization, dropout is used at the last layer, i.e., fully-connected layer, with $p = 0.5$. All of the hyper parameters such as initial learning rate, training schedule, and so on are set according to Hinton et al. (2012).

MNIST This dataset contains grayscale images of size 28x28, where each image holds a handwritten digit. The training and test sets have 60,000 and 10,000 samples, respectively. All of the images are scaled to $[0, 1]$. 150 epochs for training with batch size 128 and the initial learning rate 0.1 with momentum 0.9 are exploited. The learning rate is decayed by factor 10 twice during training, at epochs 50 and 100.

CIFAR-10 This dataset consists of 50,000 RGB images of size 32x32 as training set and 10,000 32x32 RGB images as test set. Each image contains one object from one of 10 classes. All images, either from train or from test sets, are scaled to $[0, 1]$, then normalized by mean subtraction, where the mean is computed over the training set. Like MNIST, 150 epochs with batch size 128 and the initial learning rate is 0.01 with momentum of 0.9. The learning rate decays twice by factor 10 shortly before terminating training, at epochs 120 and 130.

A.2 GENERATING ADVERSARIES

Fast Gradient Sign (FGS) (Goodfellow et al., 2014), DeepFool (DF) (Moosavi-Dezfooli et al., 2016), and the algorithm proposed by (Szegedy et al., 2013) are used for adversarial example generation. The latter algorithm finds minimum required perturbations at a high computational cost, while FGS and DF generates adversaries significantly faster, i.e., less than 3 iterations.

Using the GA-CNN (the baseline CNN for generating adversaries), correctly classified clean samples are identified then used for generating adversarial examples. Therefore, 9943 and 8152 adversaries are generated from MNIST and CIFAR-10 test sets, respectively. The optimal values for hyper parameters of FGS and Szegedy et al. (2013) are obtained for each dataset such that GA-CNN misclassifies 100% of the correctly classified clean samples after adding perturbations.

In Fig. 4, the average distortions (perturbations) generated by each algorithm for MNIST and CIFAR-10 are depicted. As well, their average misclassification confidences are written in blue. Distortion is measured by $\sqrt{\frac{\sum_{i=1}^{D}(x_i - x'_i)^2}{D}}$ for each pair of clean sample ($\mathbf{x} \in \mathbb{R}^D$) and its corresponding adversary ($\mathbf{x}'$).

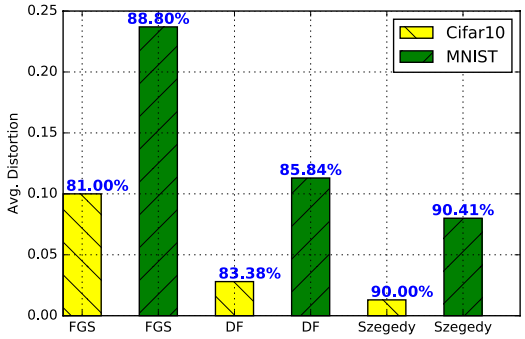


Figure 4: Average distortion to MNIST and CIFAR-10 samples by FGS, DF, and Szegedy et al. (2013). The average misclassification confidences are shown by blue text.

A.3 EXTRA EXPERIMENTAL RESULTS

Let $h(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_K(\mathbf{x})]$ be a multi-classification system (e.g., single classifier $h(\mathbf{x})$ or ensemble $\bar{h}(\mathbf{x})$) trained on clean training samples. Like Bendale & Boult (2016), we consider a threshold (τ) for rejecting instances with low confidence, assigning them to a reject class c_{K+1}. The following classical decision function is used:

$$r(\mathbf{x}) = \begin{cases} \underset{c_i \in C}{\operatorname{argmax}}\, h_i(\mathbf{x}) & \text{if } \underset{c_i \in C}{\max} h_i(\mathbf{x}) \geq \tau \\ c_{K+1} & \text{otherwise} \end{cases}. \tag{1}$$

Two types of errors should be considered: error E_D on the clean set $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, and error E_A on the adversaries set $\mathcal{A} = \{\mathbf{x}'_i, y'_i\}_{i=1}^{N'}$ (y'_i is the true label of $\mathbf{x}'_i$). Error E_D takes into account both clean samples that are misclassified and correctly classified rejected clean samples:

$$E_D = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbb{I}[r(\mathbf{x}_i) \neq y_i] + \mathbb{I}[r(\mathbf{x}_i) = c_{K+1} \wedge \underset{c_j \in C}{\operatorname{argmax}}\, h_j(\mathbf{x_i}) = y_i] \right). \tag{2}$$

Error E_A considers misclassified adversarial instances that are not rejected:

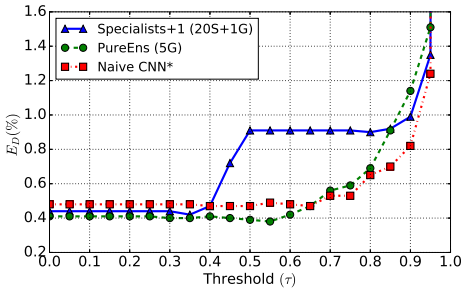
$$E_A = \frac{1}{N'} \sum_{i=1}^{N'} \mathbb{I}[r(\mathbf{x}'_i) \neq y'_i \wedge r(\mathbf{x}'_i) \neq c_{K+1}]. \tag{3}$$

Fig. 5 presents error rates E_D (Eq. 2) and E_A (Eq. 3) on the MNIST and CIFAR-10 datasets with clean samples and three types of adversaries. For naive CNN* and the pure ensemble, error rates of different types of adversaries (E_A) decrease monotonically as the threshold increases. However, the error rates of adversaries by specialists+1 ensemble are not monotonically decreased. As confidences for most of the misclassified adversaries by specialists+1 ensemble is lower than 0.5, rejection of low confidence predictions at this threshold results in a significant reduction of adversaries error E_A in comparison to naive CNN* and the pure ensemble using this threshold. Accordingly, it can be confirmed that specialists+1 can shift most of the misclassified adversaries to low confidence, thus they are being rejected.

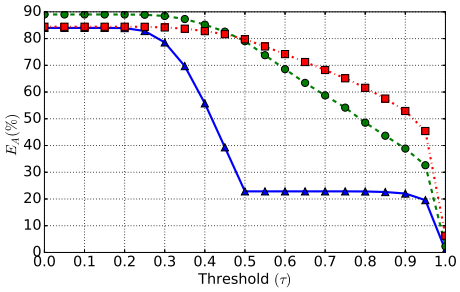
Some clean samples that can be correctly classified with high confidence by a CNN are rejected by the specialists+1 ensemble due to their low confidence, thus increasing slightly E_D at a threshold 0.5. Note that increasing the threshold to a higher value causes rejection of a vast majority of adversaries as well as more clean samples. This thus requires a trade-off between keeping rejection rate of clean test samples low vs rejecting adversaries that would otherwise fool the networks.

Moreover, from the error rates of FGS and DF adversaries (E_A) at threshold zero (Fig. 5), it can be seen that FGS and DF adversaries can severely fool the new models since they are transferable, i.e. their cross-model generalization property, while adversaries by Szegedy et al. (2013) are less generalized across different models. So, a remarkable number of Szegedy adversaries can be correctly and confidently classified by the models that are different from GA-CNN (the adversaries generative model). Notice that E_A of Szegedy adversaries by specialists+1 does not change considerably after threshold 0.5. Since most of the high confidence predictions for this type of adversaries (shown by the cyan pick at the high confidence in Fig. 6(e) and Fig. 7(e)) are correctly and confidently classified by specialists+1 ensemble.

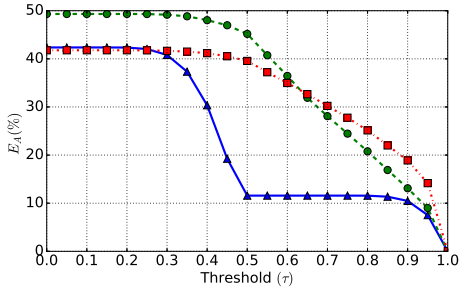
For a better insight, the rejection rate as a function of the threshold on confidence for MNIST and CIFAR-10 are shown in association with their distributions of confidence in Fig. 6 and 7, respectively. According to these observations, specialists+1 successfully provides significantly lower confidence for most of the adversaries, regardless of their types, than naive CNN* and the pure ensemble. Therefore, its rejection rate curves of adversaries are increasing at lower confidence and reach to some picks very fast at a threshold of 0.5. However, the rejection rates of adversaries by the pure ensemble are monotonically increasing by increasing the threshold, and reach to their picks at a higher threshold, when a mix of both clean and adversaries samples are being rejected. Also, as it can be seen from Fig. 6(b), 6(d), and 6(f) that rejection rate curves for MNIST correctly classified clean test samples by these three frameworks are mostly similar.



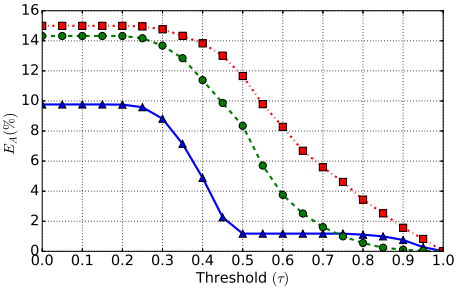
(a) $E_D(\%)$ on MNIST clean test set



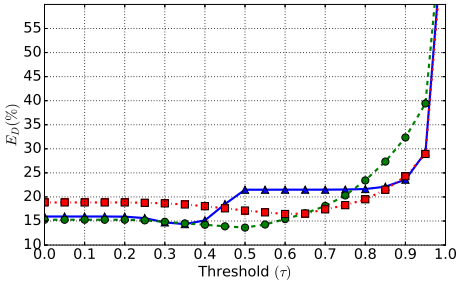
(b) $E_A(\%)$ on MNIST FGS adversaries



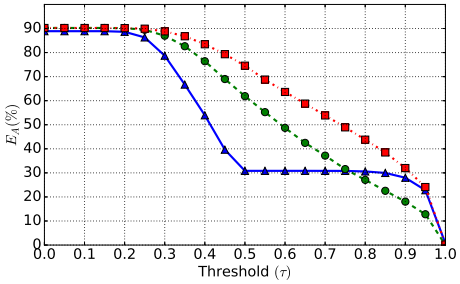
(c) $E_A(\%)$ on MNIST DF adversaries



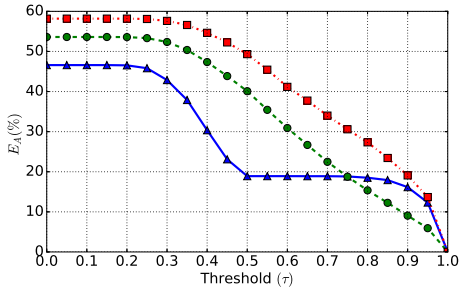
(d) $E_A(\%)$ on MNIST adversaries by Szegedy et al. (2013)



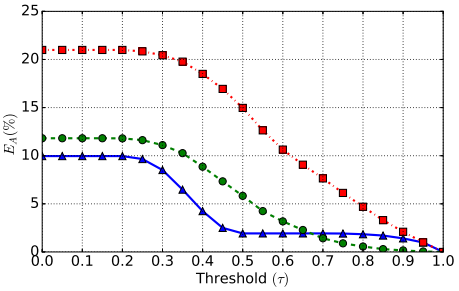
(e) $E_D(\%)$ on CIFAR-10 clean samples



(f) $E_A(\%)$ on CIFAR-10 FGS adversaries

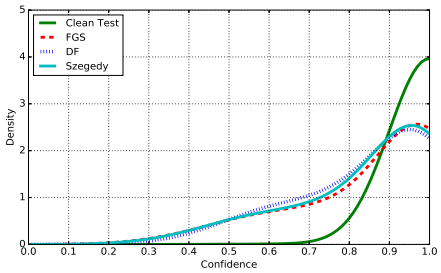


(g) $E_A(\%)$ on CIFAR-10 DF adversaries

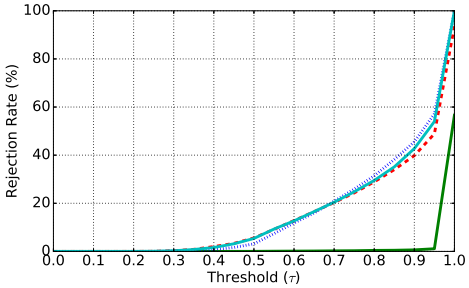


(h) $E_A(\%)$ on CIFAR-10 adversaries by Szegedy et al. (2013)

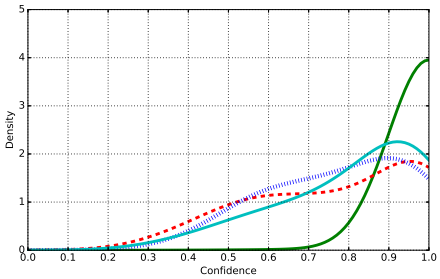
Figure 5: Error rates E_D on clean test samples, and error rates E_A on their corresponding adversaries, as a function of threshold (τ), for the MNIST and CIFAR-10 datasets.



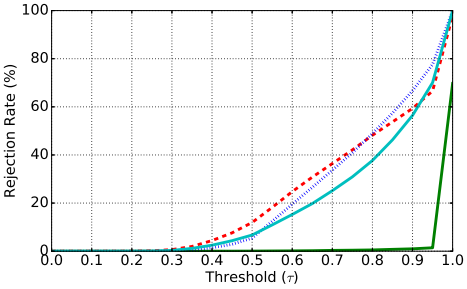
(a) Confidence densities for naive CNN*



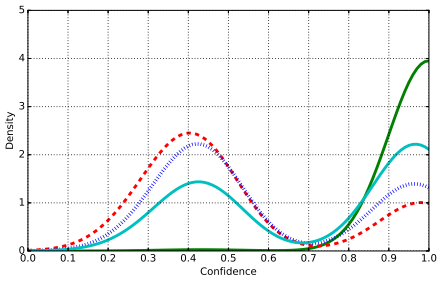
(b) Rejection rate as a function of threshold for naive CNN*



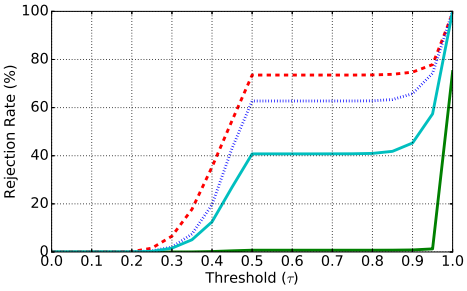
(c) Confidence densities for pure ensemble



(d) Rejection rate as a function of threshold for pure ensemble

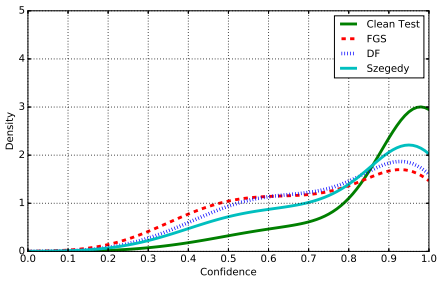


(e) Confidence densities for specialists+1

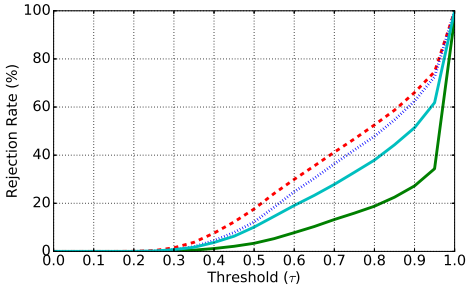


(f) Rejection rate as a function of threshold for specialists+1

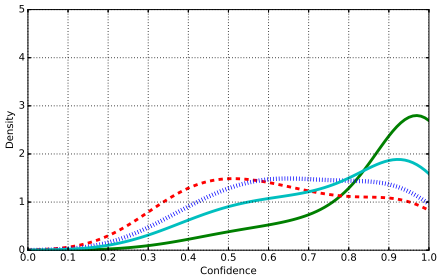
Figure 6: Confidence densities and rejection rates as a function of threshold on confidence for MNIST clean test samples and their adversaries depicted for different approaches.



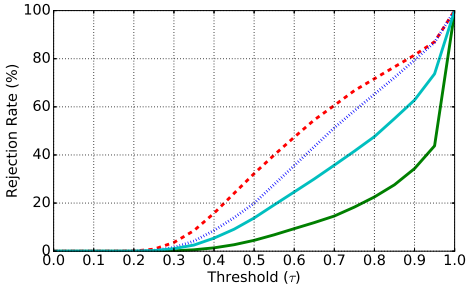
(a) Confidence densities for naive CNN*



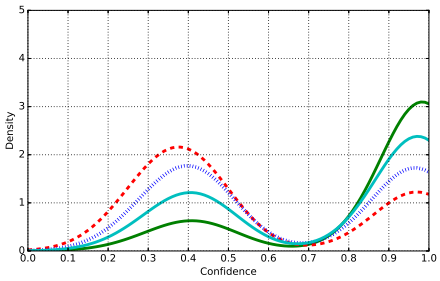
(b) Rejection rate as a function of threshold for naive CNN*



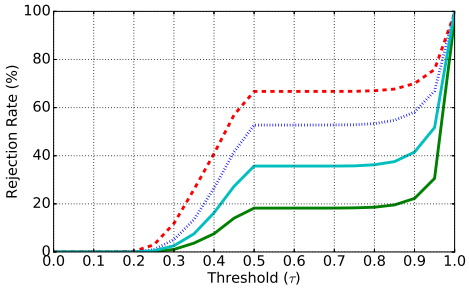
(c) Confidence densities for pure ensemble



(d) Rejection rate as a function of threshold for pure ensemble



(e) Confidence densities for specialists+1



(f) Rejection rate as a function of threshold for specialists+1

Figure 7: Confidence densities and rejection rates as a function of threshold on confidence for CIFAR-10 clean test samples and their adversaries depicted for different approaches.