Towards a "Safe" MNIST Classifier

Exploring Tradeoffs Between Safety Metrics

Introduction

In recent years machine learning (ML) models have become both more capable and more ubiquitous. A consequence of this is that the impacts of machine learning are now much greater than they once were, forcing us to question the extent to which the ML models we design and deploy can be considered “safe”. In this report, I consider this issue of “ML safety” on the toy example of MNIST classification (Deng 2012), where a model is trained to classify handwritten digits. As such, this report is aimed at understanding the extent to which a classifier for a toy problem such as MNIST can be made “safe” along various dimensions without incurring undesirable tradeoffs.

However, “ML safety” safety is a somewhat loaded phrase, with it connoting things ranging from the extreme, theoretical issue of aligning ML models with human values (Bostrom 2014) (Russell 2019) to more grounded issues such as model interpretability (Lipton 2016) and robustness to irregular inputs (Hendrycks et al. 2021). A useful typology of the issues encompassed within ML safety is presented by Hendrycks et al (Hendrycks et al. 2022), who distinguish ML safety into four subproblems:

1. Robustness (i.e. a model’s ability to cope with “hard” unseen inputs)
2. Monitoring (i.e. our ability as humans to monitor and understand models)
3. Alignment (i.e. the extent to which a model does what we want it to do)
4. Systemic risk (i.e. broader societal and technical factors)

In this report, I focus on the first two of these subproblems (robustness and monitoring). Specifically, I examine the degree to which even the stereotypically easy MNIST classification task remains imperfectly solved with respect to these problems, and the extent to which potential solutions exist. In what follows, I first outline the problems of robustness and monitoring (and metrics we can use to understand them). I then experimentally test the degree to which MNIST classifiers of various forms are, and can be made, resilient to these problems. Finally, I conclude by summarising my findings and presenting directions along which this project could be extended.

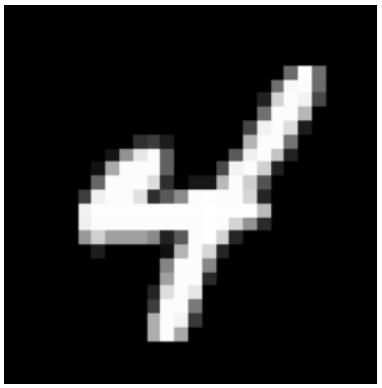
Background

Robustness

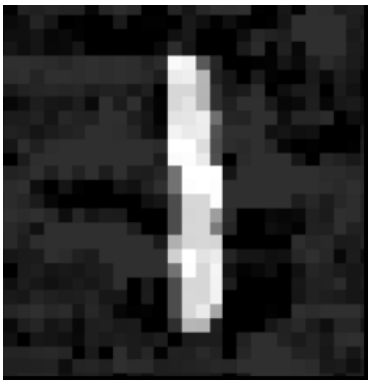
The two most easily tractable problems within ML safety are model robustness and model monitoring. Model robustness refers to the ability of a model to retain high accuracy at whatever task it is designed for in the face of inputs that are, in some fashion, “hard”. One way in which inputs can be “hard” is if they are out-of-distribution (OOD), whereby the model struggles to perform well using them due to some distribution shift compared to the data the model was trained on (Hendrycks et al. 2021). For example, in our MNIST case, models are

trained to recognise white digits on blac backgrounds such that a distribution shift might be to blur images in some way.

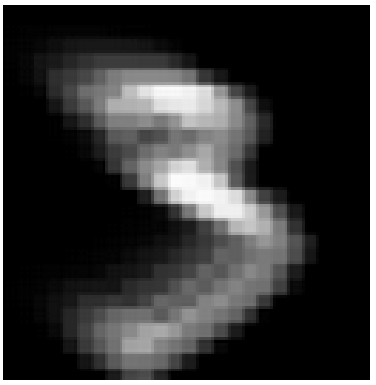
Another way in which inputs can pose a threat to model robustness is if they are adversarial (e.g. adversarial attacks) (Madry et al. 2017). Such adversarial images are the product of a process by which an "adversary" seeks to alter a standard image in such a way to maximise loss[1], subject to the constraint that the norm of the distortion to the image remains below some attack budget ε. In this report, I focus on the two most common attacks in the literature, which are the L-infinity projected gradient descent attack (PGD) (Madry et al. 2017) and the fast gradient sign method (FGSM) (Goodfellow, Schlens, and Szegedy 2014). While the technical details of these attacks are out of scope, the general idea is that both attacks calculate the gradient of the model with respect to an input, and then "step" in gradient-space in the appropriate direction to cause the model to misclassify, with PGD doing so in a more intelligent fashion. As such, PGD is (generally) the more powerful attack. Examples of these two types of inputs (e.g. OOD and adversarially distorted) are provided alongside a standard "in-distribution" (ID) example for the MNIST dataset in figure 1.



Standard Digit From ID MNIST Dataset (Deng 2012)



MNIST Digit Adversarially Distorted using L-infinity PGD (Madry et al. 2017)



Digit From OOD MNIST-C Dataset (Mu and Gilmer 2019)

Model robustness is still very much an unsolved issue, with both adversarially distorted inputs and OOD inputs leading to substantial drops in model accuracy (Hendrycks et al. 2021) (Madry et al. 2017). Indeed, even on the toy problem of MNIST classification, models remain notably non-robust to both adversarially distorted (Schott et al. 2018) and OOD inputs (Mu and Gilmer 2019).

Monitoring

Model monitoring is a broader issue than model robustness and itself covers many sub-issues. One branch of monitoring is transparency, which involves developing techniques

[1] This is usually the case as most common attacks are "untargeted" in the sense that they merely seek to maximise loss on that input subject to keeping the norm of the difference between the original and distorted image below some attack budget. However, we also sometimes can think about "targeted" attacks, when an adversary distorts an image in some way in order to minimise the loss a model achieves when it classifies an input into a (wrong) category that the attacker has specified. Roughly, we can think of the former type of attack as a blunt-force attack aiming to maximise loss for a given input with no care for what exactly the input is then classified as, with the latter type of attack being an attack that aims to cause the model to specifically misclassify the input as some wrong target label.

to understand the internal functioning of models (Borowski et al. 2021) (Lipton 2016) (Olah, Mordvintsev, and Schubert 2017). While this is an interesting topic, it is out of scope of this report and I will instead focus on two monitoring issues that are more tractable and empirical.

One such interesting issue within monitoring is the ability of models to detect when inputs are OOD (Hendrycks, Mazeika, and Dietterreich 2018). We can refer to this as the issue of anomaly detection. The idea here is that, as discussed above, models often struggle to perform as desired on OOD (e.g. “anomalous”) inputs and that we hence want models to realise when an input is anomalous in this way and report it to some special overseer before deciding what to do next. The thought is that we therefore want to teach models to be able to perform anomaly detection in some fashion.

The most common kind of technique for endowing models with the ability to detect anomalies involves anomaly scoring. Anomaly scoring techniques extract from models an “anomaly score” for each input, classifying that input as an anomaly if the score crosses some threshold. Common scoring techniques involve the use of maximum softmax or logit values for an input (Hendrycks and Gimpel 2017) or the creation of a virtual logit (Wang et al. 2022)

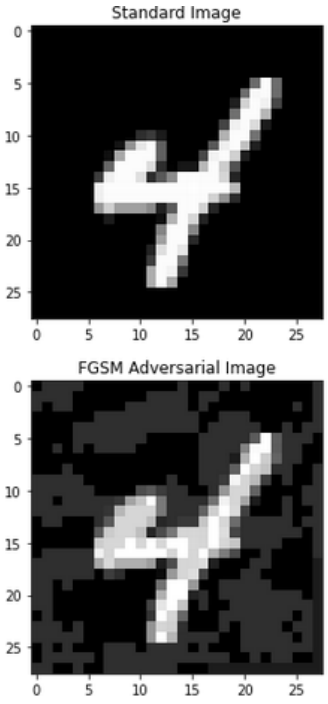
A second interesting issue within model monitoring is model calibration (Ovadia et al. 2019). The idea here is that we want models to be able to quantify how certain / uncertain they are in their outputs since there is a lot of difference between an output that a model is 50.1% confident in as opposed to one it is 99.9% confident in. Not only is this useful from the perspective of decision makers who seek to integrate ML model outputs into their decision-making process, but it is also useful from the perspective of integrating multiple models, since if they are not both well-calibrated the models may struggle to communicate as effectively as possible. As such, we want to train our models not just to be accurate (e.g. make the correct decisions / classifications), but also to be well calibrated such that, for instance, when it makes a prediction with 60% confidence, it will be correct in that prediction 60% of the time

Now, all four of these issues (adversarial robustness, OOD robustness, anomaly detection and calibration) seem important from the perspective of making ML models `safe’. However, while some work is starting to be conducted on each of the issues separately, almost zero work has been conducted on them together. This is to say that while we are making progress in making ML models “safe” along each of the four dimensions individually, we know next to nothing about how to make ML models “safe” along all four dimensions simultaneously. As such, in this report I consider the extent to which we can improve safety along each dimension without actively harming safety along other dimensions.

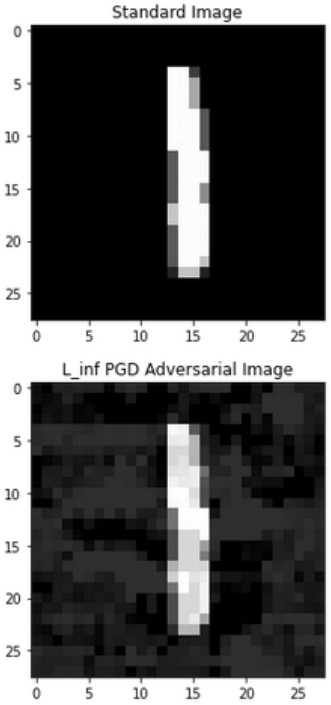
Techniques for Improving Safety

Given the scope of this report, we must hence now turn our attention towards techniques which we can use to make models safer along the dimensions I have outlined above. In this report, I focus on two commonly-proposed general techniques for improving model safety. The first of these is the use of data augmentation during the training process, while the second is architectural changes.

Many commonly-proposed solutions to our safety issues involve using specific techniques to augment the data on which a model is trained on in some way. There are, broadly, two rough categories of such techniques. The first of these is adversarial training (Madry et al. 2017), in which a certain portion of the data on which a model is trained on is adversarially distorted using some adversarial attack method. The hope here is that such adversarially distorted training data can act as a sort of regularizer, forcing the model to learn the features of inputs that we want them to learn (e.g. the features that make a hand-drawn “1” look like a 1 in the MNIST case) rather than to overfit such that it struggles with unseen distortions or becomes overconfident. This report constrains its scope by focusing on the common FGSM (Goodfellow, Schlens, and Szegedy 2014) and PGD (Madry et al. 2017) attacks, so all adversarial training in the following will be performed using these attacks.



Example of CutOut data augmentation (DeVries and Taylor 2017) applied to MNIST digit

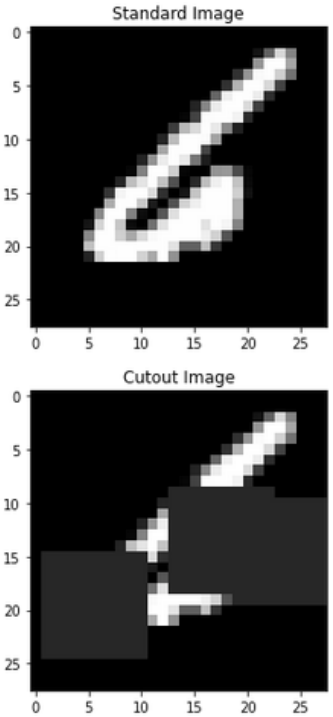


Example of MixUp data augmentation (Zhang et al. 2017) applied to MNIST digit

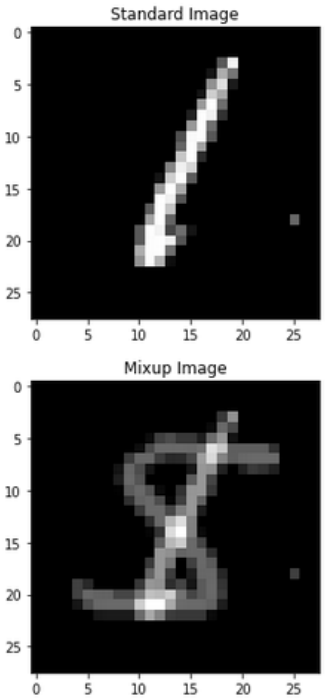
The second category of such techniques involves augmenting the training data in non-adversarial ways, but with a similar aim (e.g. to effectively impose regularisation and encourage the model to learn more reliable features that are more robust to distribution shift and better calibrated). In what follows, I focus on the data augmentation techniques of CutOut (DeVries and Taylor 2017) and MixUp (Zhang et al. 2017). These techniques are chosen due both to their popularity and due to the infeasibility / overkill of using other data augmentation techniques on the MNIST dataset.[2]

[2] For instance, a common data augmentation technique is CutMix (Yun et al. 2019) in which sections of input images are used to replace sections of other images such that the desired label for the combined image is a mix of the respective labels. However, this is unsuitable for the MNIST dataset since the majority of the space of each MNIST image is the same (i.e. black space) such that this would (and in preliminary testing does) have negative consequences for every metric we care about.

In CutOut, a specified number of squares of specified size are “cut out” of the image in order to hide certain parts of the image from the model, and encourage the learning of more robust features (DeVries and Taylor 2017). In MixUp, input images are combined in a convex way[3] such that the label which the model ought to assign to them is the respective convex combination of labels (Zhang et al. 2017). Both of the data augmentation techniques are illustrated below.



Example of CutOut data augmentation (DeVries and Taylor 2017) applied to MNIST digit



Example of MixUp data augmentation (Zhang et al. 2017) applied to MNIST digit

The second rough category of techniques I consider in this report are modifications to models that we could group under the label of “architectural”. The first of these modifications I consider is model size, since larger models are frequently found to perform better with regards to many safety metrics (Hendrycks et al. 2021). The second of these modifications I consider is model ensembling. Model ensembling involves combining the outputs of many separately trained models[4] in order to produce a final output (e.g. by averaging all softmax values to produce a final output softmax value that is understood as the output softmax of the ensemble model). Attention is paid to model ensembling due to recent findings suggesting that it is robustly beneficial to the problems such as calibration (Ovadia et al. 2019) and adversarial robustness (Strauss et al. 2018). Both of these modifications are often cited as helping safety metrics due to their ability to allow the final model to effectively learn more robust features and be less prone to stochastic undesirable features that might develop.

---

[3] Specifically, the factor λ that is used to combine the input is drawn from a beta distribution, whose parameters are the parameters of the augmentation technique.

[4] To be precise the kind of model ensemble that I am describing here (e.g. where we have an array of separately trained models of the same architecture) is only the most common implementation of model ensemble and not the only implementation.

Methodology

With all this said, I can now turn to explaining the empirical strategy used to answer my research question by outlining how I tested the extent to which safety improvements tradeoff against one another. Specifically, I will now outline the implementation of the techniques I used in order to improve model safety along different dimensions, and the metrics I used to track the safety of each model along the different dimensions.

Model Implementations

All models used in my experiments were trained over 10 epochs using Adam with a learning rate of 0.0001 (Kingma and Ba 2015), and compared against a simple baseline model. This baseline model was the modification of the standard Le-Net 5 (Cunn et al. 1998) that is frequently used to illustrate the power of convolutional networks on the MNIST digit recognition problem. As expected, this model achieves a high accuracy rate (0.993) on the standard MNIST problems.

To test the extent to which adversarially robust models achieve high safety metrics along other dimensions, I adversarially trained many models of this same basic architecture using many different hyperparameter settings for using both FGSM and PGD. Specifically, for PGD I performed a grid search over attack budgets 0.2, 0.3 and 0.4, and over number of steps 30,40,50, while for FGSM I varied the single attack budget hyperparameter across 0.2,0.3,0.4 and 0.5.

To test the ability of models that are robust to distribution shift to remain safe along other dimensions, I trained many models of the above architecture by applying the two data augmentation techniques mentioned earlier to the respective training data. For CutOut, I experimented with hyperparameters by performing a grid search over lengths 7, 10 and 14, and over the number of holes 1, 2, 3, 4 and 5. For MixUp, I experimented with varying the single distribution hyperparameters (which affects the average convex combining factor) across 0.25, 0.5, 1, 2 and 4.

Additionally, I was interested in whether these two techniques which aim at different types of robustness would, when combined, achieve good results on both types of robustness and also on other metrics. As such, I trained models by combining adversarial attacks with data augmentations during the training process. For all the hyperparameter values listed above, I experimented both with applying adversarial attacks to augmented images in the training dataset, as well as to applying data augmentations to images in the training dataset that were already adversarially distorted.

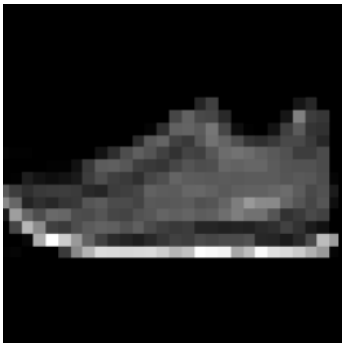
However, in addition to this category of techniques whereby the training data was adjusted in some form, I also considered techniques relating to modifying the architecture of the models. Specifically, I also tested whether a larger model would perform better than the smaller models by creating a “scaled-up” version of my base model. This model replicated the design pattern of the original model based on Le-Net 5, but scaled it up by adding an additional three convolutional layers and hence allowing it to have approximately 20 times the number of parameters as the original model. I also experimented with creating model ensembles. Specifically, I created ensemble models made up of 3, 5, 10 and 15 models respectively, where each model was trained independently of all others.

Safety Metrics

Having created all of these models, I then had to determine empirical metrics that I could use to test the extent to which safety metrics trade off against one another. To benchmark robustness to adversarial attacks, I use as a metric the accuracy of each model in the face of the PGD and FGSM adversarial attacks over the attack budgets 0.2, 0.3, 0.4 and 0.5. In the case of ensemble models, I test the accuracy of the models against adversarial attacks that use the average gradient across all models in the ensemble to construct attacks.[5]

To benchmark robustness To benchmark robustness to distribution shift, I measure model accuracy on the MNIST-C dataset (Mu and Gilmer 2019). This is a dataset consisting of images that we can see as “OOD MNIST digits” in that they represent a set of MNIST digits with various distortions (such as blurring, dotted lines and artificial snow) applied.

To explore the ability of models to detect OOD inputs in our MNIST case, I use the Fashion MNIST / F-MNIST (Xiao, Rasul, and Vollgraf 2017) and Kuzushiji MNIST / K-MNIST (Clanuwat et al. 2018) datasets as examples of OOD inputs that our MNIST classifier ought to recognise as OOD. These datasets respectively contain hand-drawn pieces of clothing and Japanese characters such that our models should be able to categorise them as being drawn from a different data generating process and hence as being out-of-distribution. I then test the ability of different methods to successfully detect OOD inputs, using the AUROC score as a metric for success



Example of an OOD Image From Fashion MNIST Dataset (Xiao, Rasul, and Vollgraf 2017)

Example of an OOD Image From Kuzushiji MNIST Dataset (Clanuwat et al. 2018)

To test calibration, I tracked the expected calibration error, RMS calibration error, brier score and negative log likelihood for each model on the original MNIST dataset. These metrics are all different measures of the same underlying concept, which is the ability of models to make

[5] The issue of adversarial attacks in the context of model ensembles raises an important methodological issue. Namely, since ensembles consist of many different models, and since adversarial attacks use the gradients of the models they attack in order to formulate attacks, we must choose what the attack can use as a gradient. The choice here is whether to use the gradient of a single model in the ensemble to formulate the attack or to use the average of the gradient of all models in the ensemble to create the attack. For the present work, I elected to choose the latter method (e.g. using the average gradient across all models in the ensemble) on the grounds that past work has show it to generate stronger adversarial attacks (Strauss et al. 2018)

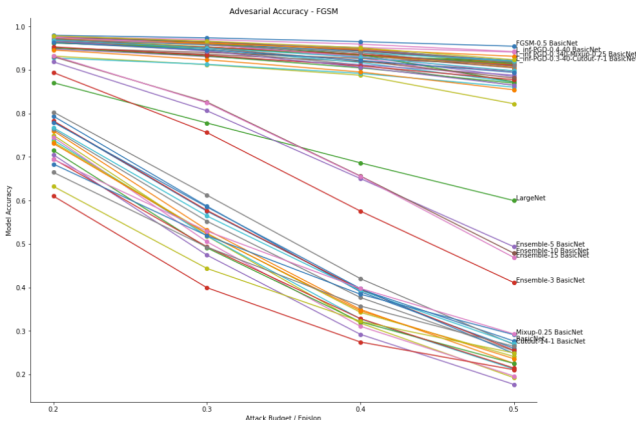
well calibrated predictions such that an event X will happen 60% of the time when the models predict X with a confidence of 0.6.

Results

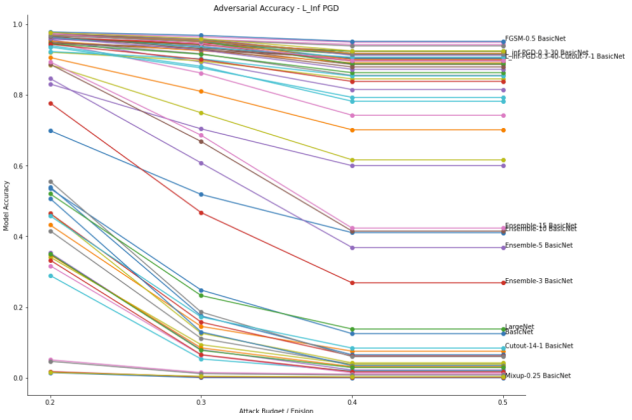
Having outlined all of the above, we can now turn attention towards the findings of my experiments. Whilst interesting, it is important to remember some caveats in all of the following. Firstly, I was limited in my access to compute during the course of this research, and hence was limited to only training each model for a limited portion of time and could only test a small number of hyperparameter settings. Whilst I don't necessarily think this detracts from my findings (e.g. preliminary testing I performed suggested there was no benefit to longer training runs, nor to radically different hyperparameter settings), it is nonetheless a thought worth keeping in mind. A second caveat that ought to be kept in mind is that MNIST classification is a very different problem to many other ML problems by nature of its simplicity, and it is not a foregone conclusion that my findings here generalise, even to similar problems such as ImageNet classification. Indeed, some evidence suggests that the ability of techniques to allow models to perform well according to safety metrics on MNIST is completely unrelated to their ability to perform this role for models working on other image classification problems (Ovadia et al. 2019). However, with all this said, we can now turn to consider my results.

Adversarial Robustness

The first area I explored was adversarial robustness, which I measured using the accuracy of models in the face of adversarial attacks of increasing strength. An interesting pattern emerges here. Across all models and all attack budgets, there appears to be three qualitatively different types of models. The figures below, which outline the adversarial accuracy for all models across all attack budgets with select models labelled, illustrates this pattern.



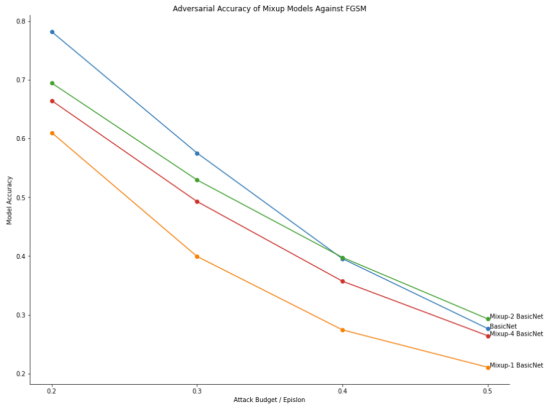
Adversarial accuracy of models with respect to FGSM adversarial attacks



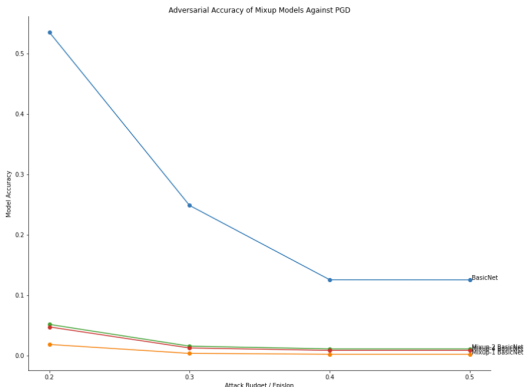
Adversarial accuracy of models with respect to PGD adversarial attacks

The pattern that is illustrated in these figures is that all the models tested fall into one of three categories. First, we have a set of models that perform consistently worse than all other models. This set consists of the baseline model, and all models trained purely using those data augmentation techniques that aim to improve OOD robustness (e.g. Cutout and Mixup). Indeed, further inspection of this set of models reveals that the models trained using

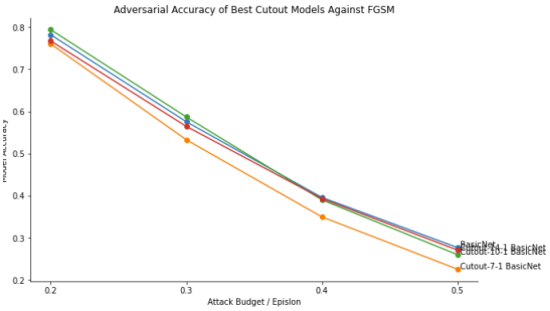
these OOD robustness enhancing techniques actually perform worse with regard to adversarial robustness than the baseline model. The figure below compares the adversarial accuracy of cutout and mixup models relative to the baseline model, and suggests an interesting tradeoff between improving adversarial robustness and improving OOD robustness. As shown, this tradeoff is especially pronounced for robustness to PGD attacks.



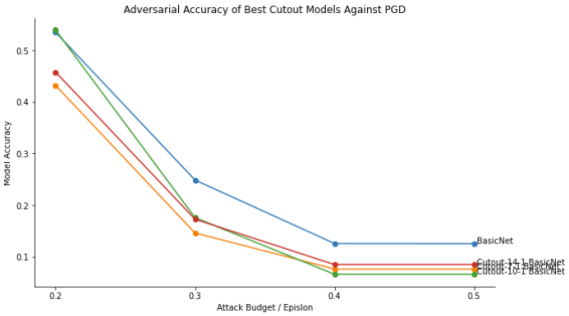
Adversarial accuracy of Mixup models with respect to FGSM adversarial attacks



Adversarial accuracy of Mixup models with respect to PGD adversarial attacks

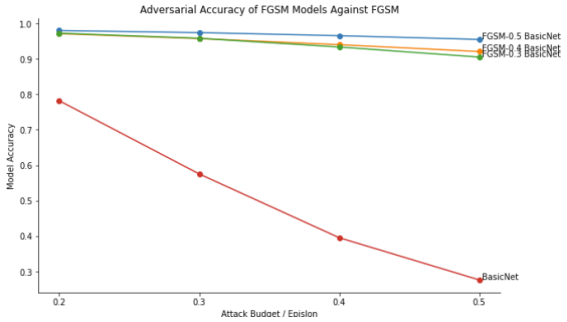


Adversarial accuracy of Cutout models with respect to FGSM adversarial attacks

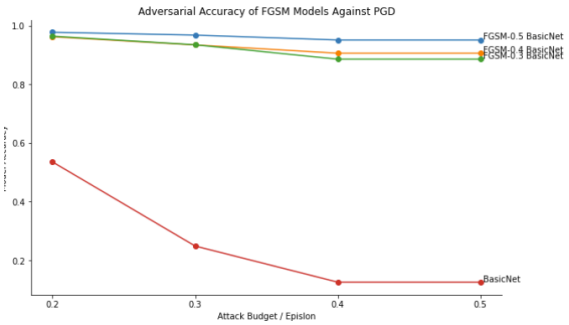


Adversarial accuracy of Cutout models with respect to PGD adversarial attacks

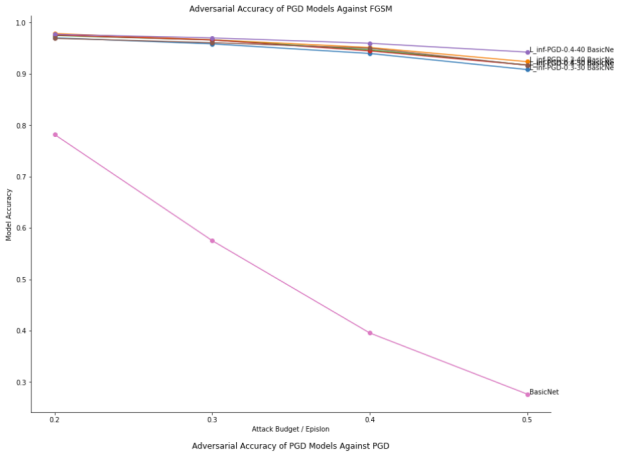
The next overall set of models is the set of models that consistently achieve high accuracy in the face of both attacks across a range of attack budgets. This set consists of three types of models. Firstly, it consists of those models that were trained solely using adversarial training, with FGSM and PGD adversarial training achieving similar results. This is illustrated below.



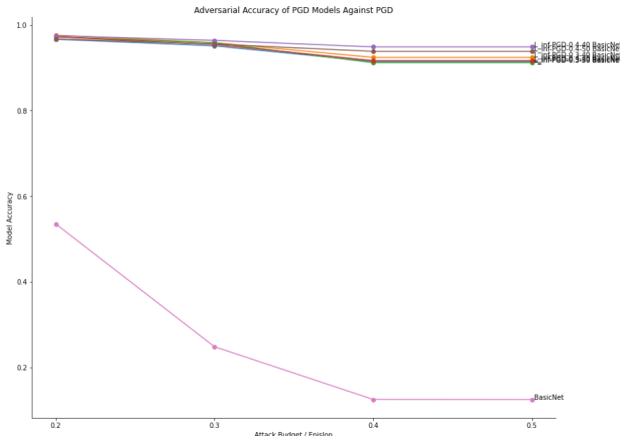
Adversarial accuracy of FGSM models with respect to FGSM adversarial attacks



Adversarial accuracy of FGSM models with respect to PGD adversarial attacks



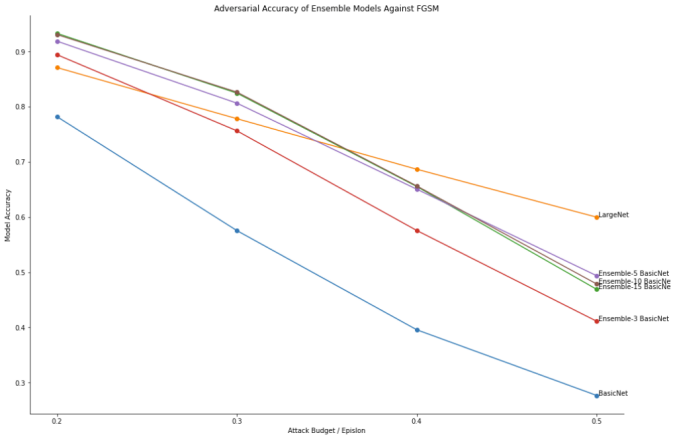
Adversarial accuracy of PGD models with respect to FGSM adversarial attacks



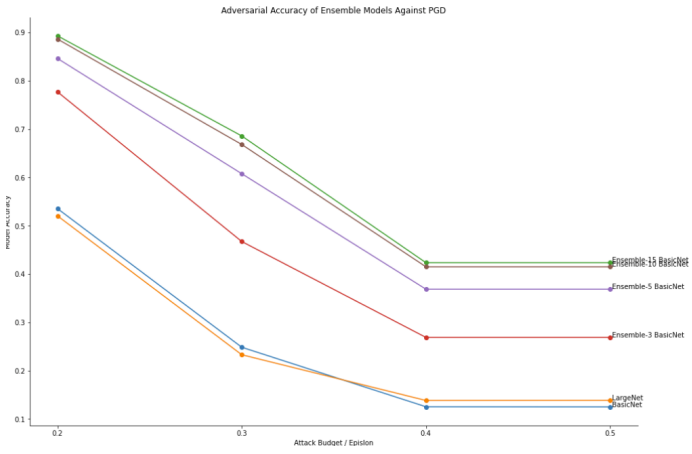
Adversarial accuracy of PGD models with respect to PGD adversarial attacks

The next group of models within this set consists of those models that are trained using images that are first adversarially distorted before having data augmentations applied. The final group of models within this set is those models that are trained using images that are first augmented and then adversarially distorted. These two groups perform similarly, both significantly outperforming the baseline model, but marginally underperforming the models trained solely using adversarial distortions[6]. One interpretation of this is to lend further credence to the idea that there is a tradeoff between achieving adversarial robustness and achieving OOD robustness.

The finally qualitatively distinct set of models is those models that outperform the baseline models and augmentation models, and yet underperform the models trained using some form of adversarial training. This set consists of ensemble models, and the large version of the baseline model, and the pattern of accuracies in the face of adversarial attacks is shown in the figure below.



Adversarial accuracy of Ensemble models and the scaled up model with respect to FGSM adversarial attacks



Adversarial accuracy of Ensemble models and the scaled up model with respect to PGDadversarial attacks

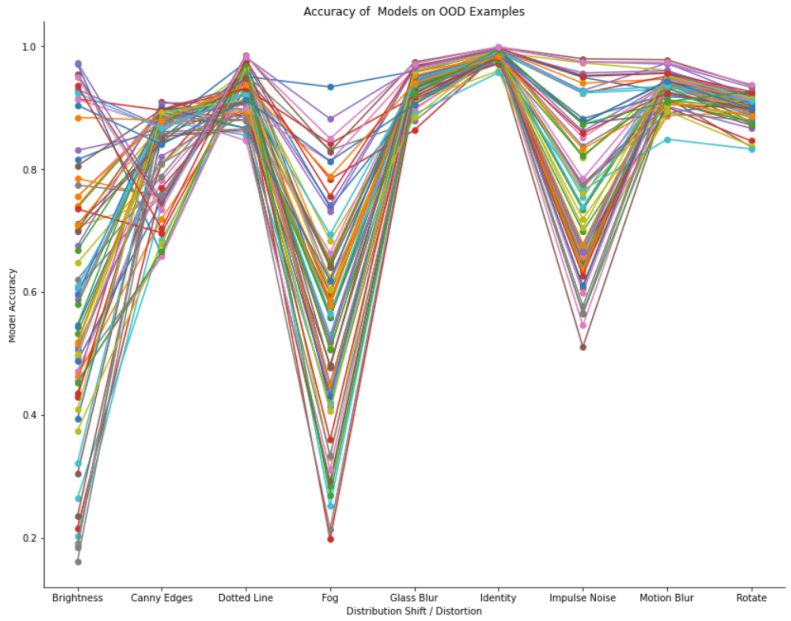
The overall takeaway from this set of models is less clear than with the other two sets. On one hand, the scaled up version of the baseline model seems to outperform all ensemble

[6] Figures detailing performance can be found in appendix

models when faced with FGSM adversarial attack. However, on the other hand, this scaled up model significantly underperforms all ensemble models when confronted with a PGD attack. Additionally, the effect of ensemble size on adversarial robustness seems unclear with respect to FGSM attacks, yet seems to be positive with regard to PGD adversarial attacks. As a broad generalisation, a takeaway seems to be that large ensembles should usually outperform the scaled up version of the baseline mode, yet this is by no means a certain thing.

Out-Of-Distribution Robustness

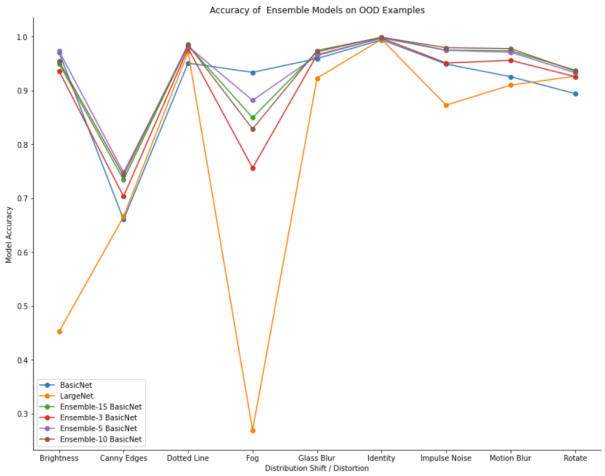
The hypothesis that there is a tradeoff between adversarial robustness and OOD robustness is, somewhat at least, supported by the experiments I performed regarding the robustness of models to distribution shift. I measured this using the accuracy of each model on each of the distortions contained within the MNIST-C dataset. The pattern for all models is illustrated in the figure below.



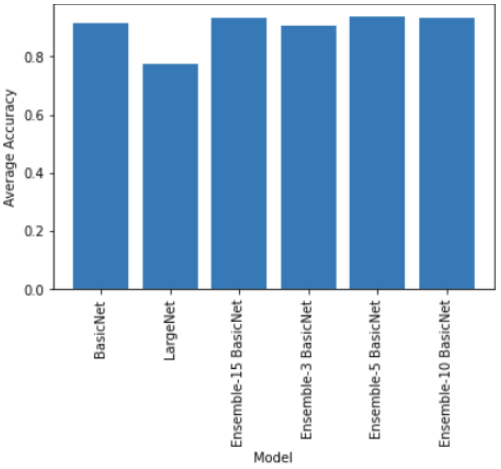
From a high-level perspective, two conclusions are sticking from this figure. Firstly, most models follow a similar pattern such that they all suffer precipitous declines in accuracy when faced with the fog and impulse noise distortions. Secondly, while most models seem to follow similar overall patterns, there is much variability in accuracy with regard to each individual distortion. Having noted these two points, I now turn attention to specific patterns relevant to the question at hand.

One interesting pattern concerns ensemble models and the effect of model size. As mentioned previously, the scaled up model of the baseline model represented either a significant or marginal improvement over the baseline model with regards to adversarial robustness. This is not true in terms of OOD robustness, with the scaled up version actually having lower average accuracy over the OOD examples contained within MNIST-C. Indeed, for certain OOD distortions such as fog, the larger model performs incomparably worse. However, the pattern with respect to ensembles remains similar to, albeit less pronounced than, with adversarial robustness. Specifically, all ensemble models slightly outperform the baseline model, with large ensembles doing better. As mentioned, this provides mixed evidence for the hypothesis that there is a tradeoff between adversarial and OOD

robustness. On one hand, larger models seem to improve adversarial robustness whilst hurting OOD robustness, while on the other large ensembles seem to improve both. These patterns are illustrated below, where both the average accuracies of all models across the MNIST-C dataset and the individual accuracies for each distortion in the dataset are shown.

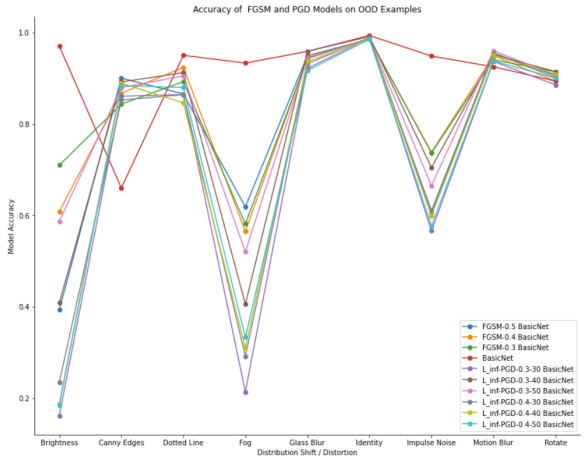


OOD accuracy of Ensemble models and the scaled up model

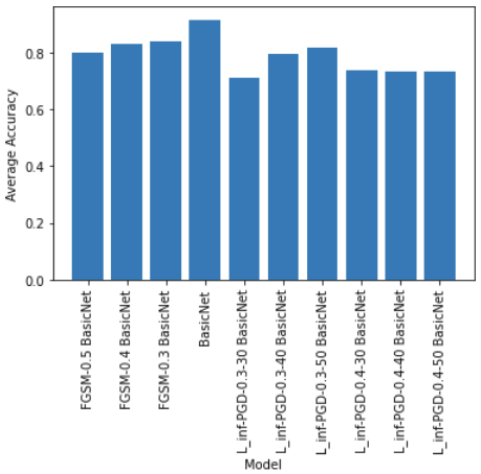


Average OOD accuracy of Ensemble models and the scaled up model

However, more evidence in favour of the tradeoff hypothesis comes from inspection of the data regarding adversarially trained models. As seen below, all models trained purely with adversarial training (which happened to be the best performing models in terms of adversarial robustness) are less accurate on average over the OOD MNIST-C dataset. Additionally, on some OOD distortions such as brightness, fog and impulse noise there experience significant declines in accuracy relative to the baseline model. A final point of interest here is that those models that were adversarially trained with a stronger attack budget (which happened to be the most adversarially robust) have lower OOD robustness than those models trained with a smaller attack budget.



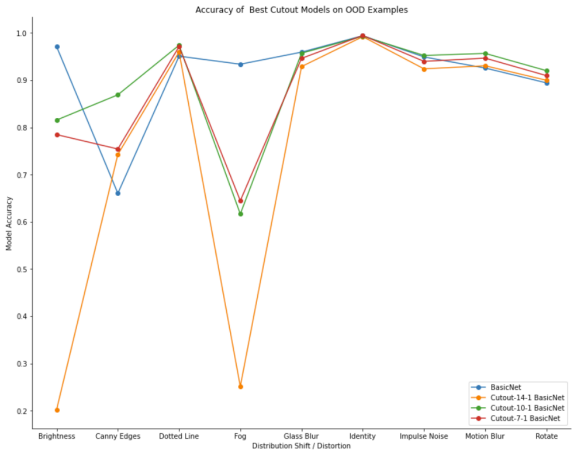
OOD accuracy of FGSM and PGD models and the scaled up model



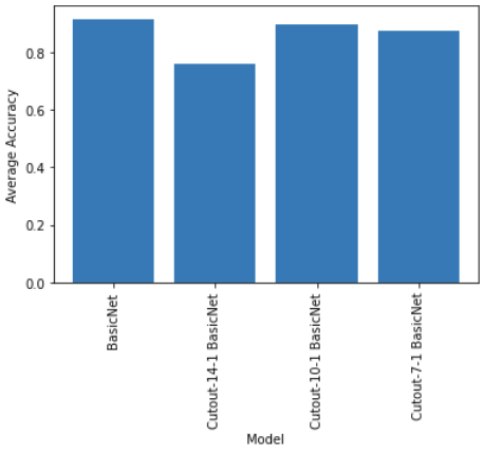
Average OOD accuracy of FGSM and PGD models and the scaled up model

Turning now to the OOD robustness of models trained using data augmentations, some interesting findings arise. Cutout and Mixup are specifically designed to increase OOD robustness, and yet I find they have, at best, limited effect while often actually harming OOD

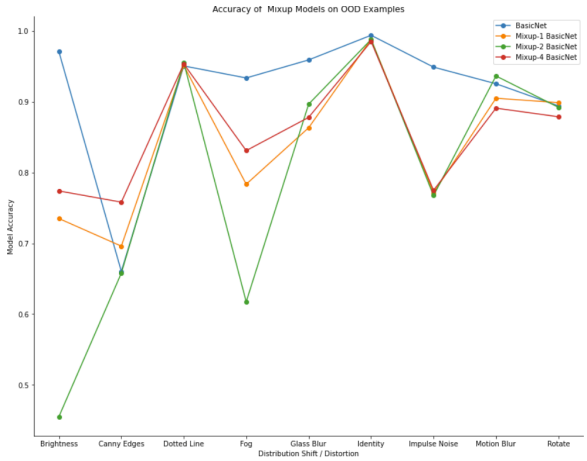
robustness of models. This is illustrated below, where the best such models marginally outperform the baseline model in terms of OOD robustness while the worst models underperform the baseline. Interestingly, this holds for both Cutout and Mixup



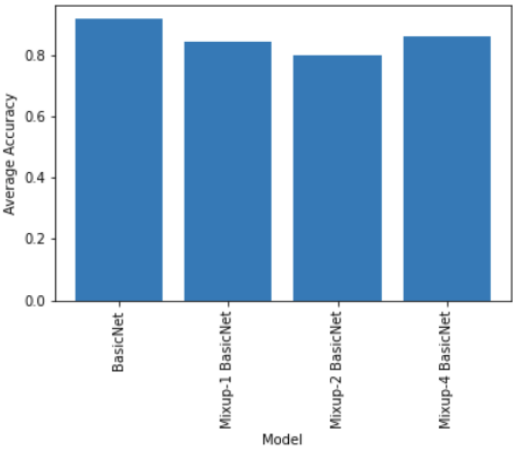
OOD accuracy of Cutout models and the scaled up model



Average OOD accuracy of Cutout models and the scaled up model



OOD accuracy of Mixup models and the scaled up model



Average OOD accuracy of Mixup models and the scaled up model

Finally, we can turn attention to models trained using both data augmentation and adversarial training. As before, I found little evidence that the order of the two processes made any salient difference in outcomes. Additionally, I found that these models performed worse in terms of OOD robustness than both the baseline model and the models trained using data augmentations alone[7]. This lends support to the idea that there seems to be some tradeoff between adversarial and OOD robustness, though, for the reasons discussed, this is not a definitive conclusion.

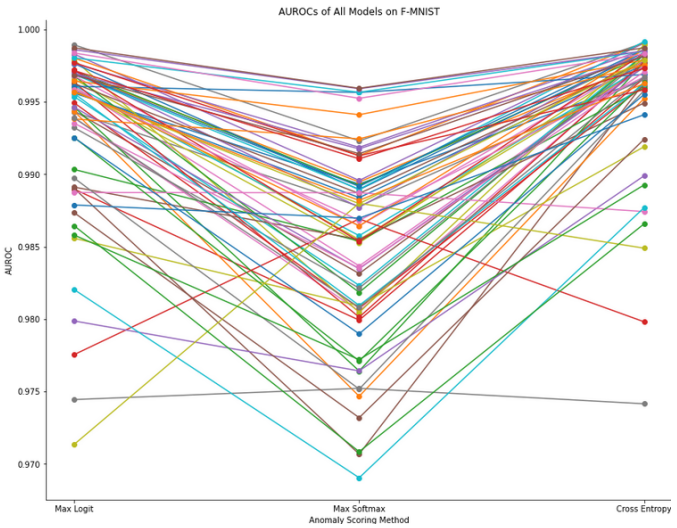
Anomaly Detection

As previously discussed, another important “safety” property that ML models can have is the ability to detect inputs that are anomalous compared to their training data. We can measure the ability of models to do this by providing them with a dataset consisting of both examples similar to those they were trained on and examples representing a distribution shift relative

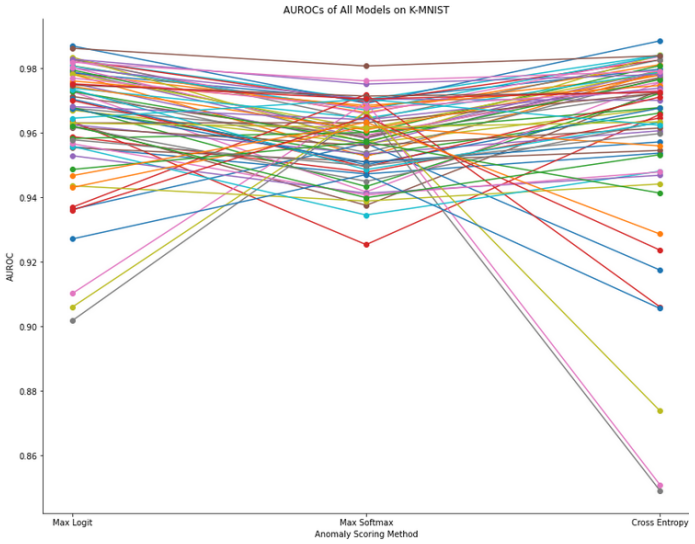
[7] See appendix for details of figures and performance.

to those they were trained on, and then tracking their ability to use whatever anomaly scoring method we decide on to successfully detect the anomalous examples. We track this using the AUROC, where an AUROC of 1 corresponds to perfect classification of anomalous inputs, and an AUROC of 0.5 represents random chance. In the following experiments, I track model AUROCs using the Fashion-MMNIST (FMNIST) and Kuzushiji MNIST (KMNIST) datasets as examples of out-of-distribution examples that a model should classify as anomalous.

An overall picture of my findings is presented below, with some notable trends being clear. Firstly, all models achieve relatively high AUROCs on both datasets, with most models achieving AUROCs of at least 0.90 across all datasets across all anomaly scoring methods. Second, there doesn't appear to be a hard and fast rule as to which anomaly scoring method is "best", with the only potential finding in this direction being that maximum softmax anomaly scores at least achieve notably less variability for KMNIST data. Finally, models seem to have a harder time classifying inputs from the KMNIST dataset as anomalous than with inputs from the FMNIST dataset since AUROCs for the former seem higher on average than for the latter.

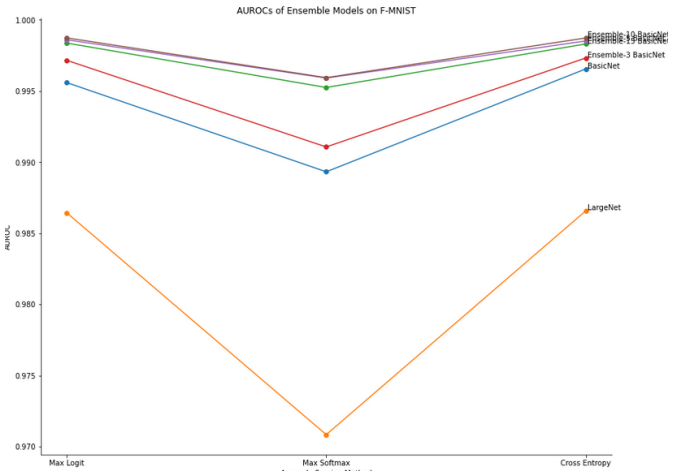


AUROCs of all models on FMNIST data

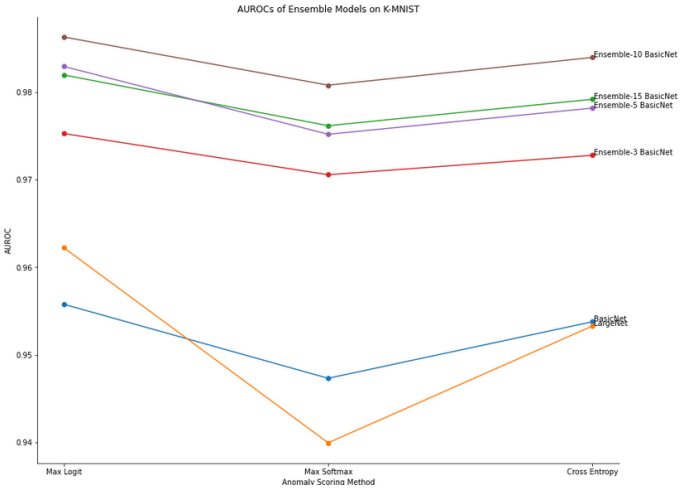


AUROCs of all models on KMNIST data

Turning our attention to specific families of models, more definitive conclusions arise. Let's first consider ensemble models and the large version of our baseline model. A notable finding is that all ensemble models achieve higher AUROCs than the baseline model, with the improvement seeming to grow as the size of the ensemble increases. This is notable since this is the same pattern that emerges when considering both adversarial robustness and OOD robustness, implying that we can increase model safety metrics without incurring tradeoffs by using model ensembles. However, this finding does not replicate for the larger version of the baseline model. Whilst this large model achieves a somewhat comparable AUROC for the KMNIST data, it significantly underperforms the baseline model for FMNIST. As such, given the improvements to adversarial robustness from this larger model, larger models seem to increase robustness at the cost of reducing their ability to detect anomalies. These patterns are all illustrated below.

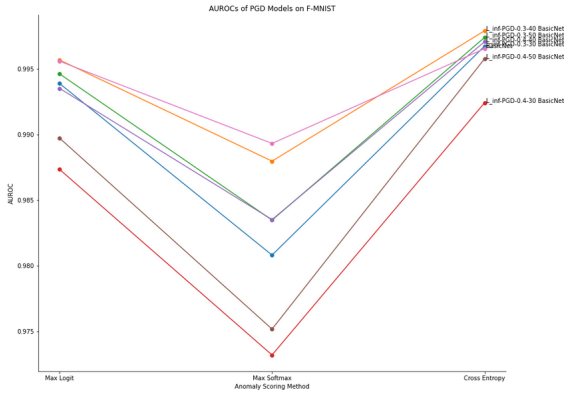


AUROCs of ensemble and scaled-up models on FMNIST data

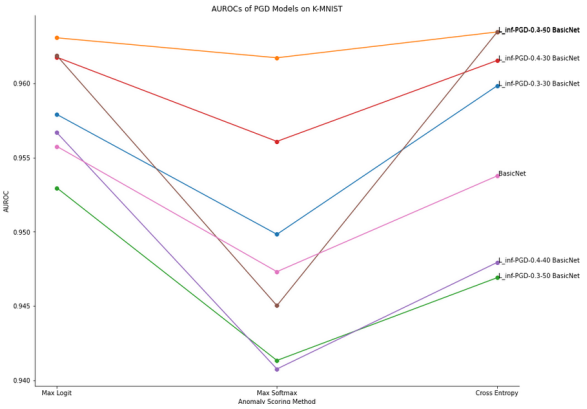


AUROCs of ensemble and scaled-up models on KMNIST data

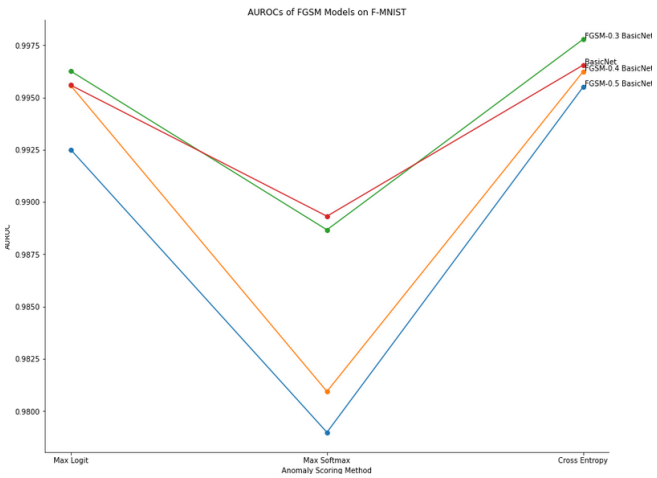
However, shifting focus to purely adversarially trained models, few conclusions seem to arise. For models adversarially trained using both PGD and using FGSM, no consistent pattern in AUROCs seems to arise. In both cases, all models seem to achieve roughly the same AUROCs as the baseline model, with no consistent ordering arising. This is illustrated below.



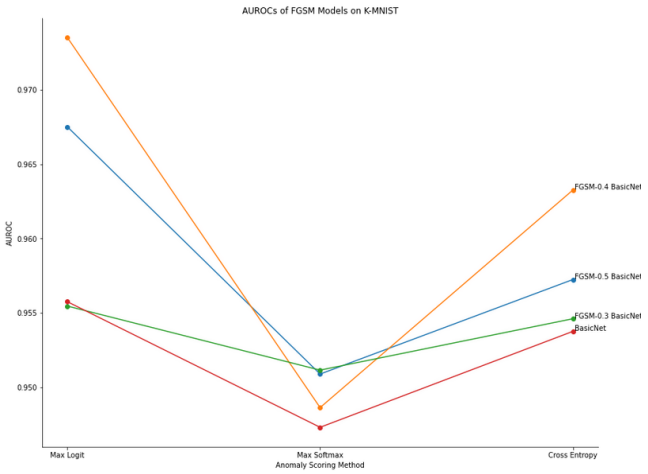
AUROCs of PGD models on FMNIST data



AUROCs of PGD models on KMNIST data

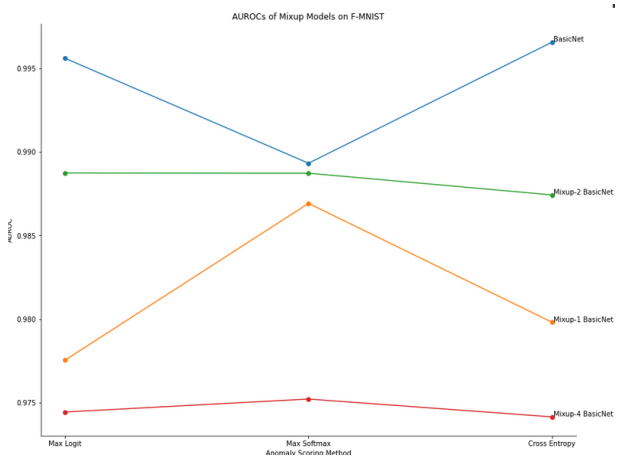


AUROCs of FGSM models on FMNIST data

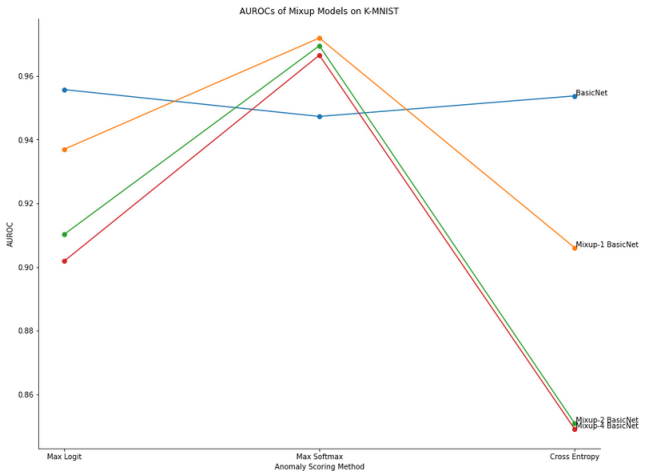


AUROCs of FGSM models on KMNIST data

More interesting findings emerge when we consider the performance of models trained using data augmentations. Specifically, I found that models trained using Mixup were notably less able to detect OOD examples than the baseline model, implying a tradeoff between OOD robustness and anomaly detection. This is illustrated below, but it is worth noting that this result is consistent, such that it is replicated for models trained using both Mixup and some form of adversarial distortions.

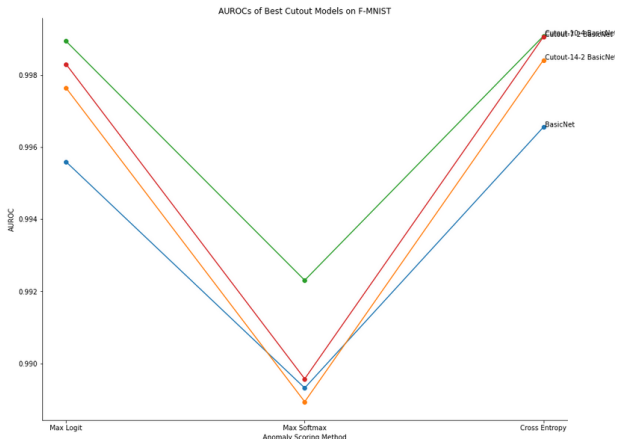


AUROCs of Mixup models on FMNIST data

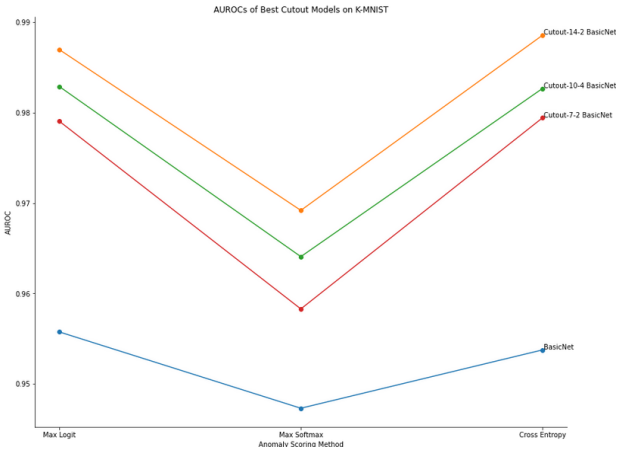


AUROCs of Mixup models on KMNIST data

However, I found that the opposite was true for all models trained using some form of cutout. Specifically, I found that models trained using Cutout (either in isolation, or combined with some form of adversarial distortion) were significantly better at detecting anomalous inputs than the baseline mode, scoring much higher AUROCs. For instance, the figure below demonstrates this for a selection of simple Cutout models, but the same pattern is replicated for models trained with Cutout and adversarial distortions.



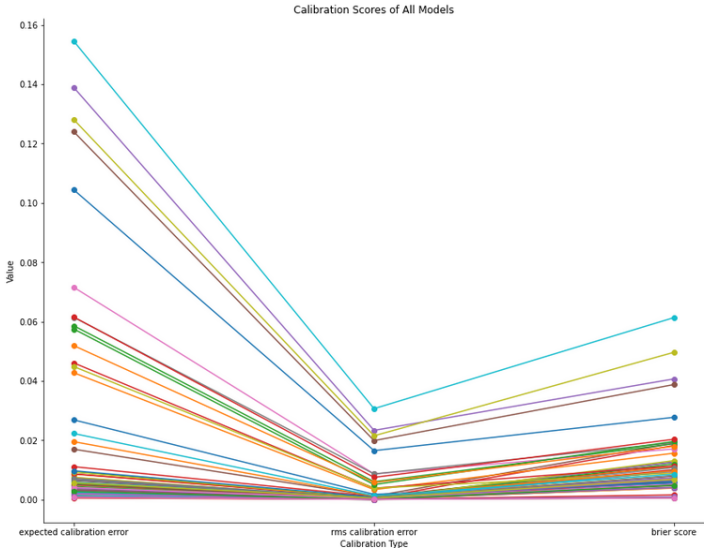
AUROCs of Cutout models on FMNIST data



AUROCs of Cutout models on KMNIST data

Calibration

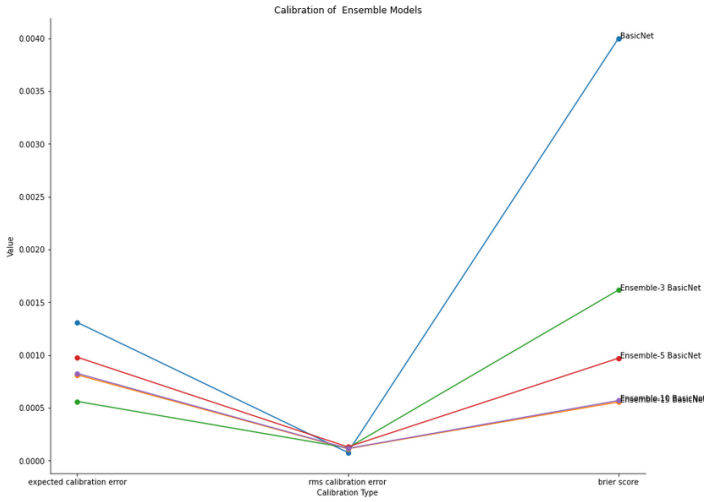
Finally, we can turn attention to the performance of the different families of models with regards to calibration. As stated before, calibration refers to the ability of models to make predictions with an appropriate degree of confidence. In the following, I report calibration using three common calibration metrics - RMS calibration error, expected calibration error and the Brier score. While the scales of the three metrics are not consistent, they are all such that a better calibrated model scores lower than a worse calibrated model. The results for all models considered are shown below.



Calibration of all models

As illustrated in this diagram, there are again three broad tiers of models. The first tier is the worst overall tier that scores notably higher than the other models (including the baseline) on all calibration metrics. Interestingly, this tier of models consists solely of those models that were trained by first applying Mixup and then applying adversarial distortions to their training data. This is a notable finding, since it is the only case (other than that of ensemble models) where models of a single type are in a category of their own. Additionally, this is interesting since it means that if we wish to achieve OOD robustness and adversarial robustness simultaneously using Mixup, training a model by first applying an adversarial and then Mixup is the strictly dominant method of doing so.

Skipping a tier, the third and best tier of models consists of ensemble models. Ensemble models achieve better results than all other models and significantly outperform the baseline model. Additionally, the larger the size of the ensemble model, the better calibrated it is. This pattern is illustrated on the figure below. Given the ability of ensemble models to perform well on all other safety metrics tested, this gives credence to the idea that we can achieve calibration without trading off against other desirable safety features.



Calibration of ensemble models

Finally, the middle tier of models consist of all other models I considered, with pretty much all of them achieving worse calibration results than the baseline. This includes all models trained using adversarial attacks, data augmentations and any mix of the two except mixup followed by adversarial distortions. Out of all these models, however, two types of models achieve marginally better calibration than the baseline. These types of models are the scaled up version of the baseline, and models trained using a weak form of cutout (i.e. a low block size and a low number of blocks). Indeed, as the form of cutout applied to the training data becomes stronger, these models actually perform much worse than most other models, suggesting that this might be something of an edge condition.

Conclusion

Having considered all of the above experiments, we can finally return to the question of whether or not there are tradeoffs between different safety metrics that it might be desirable for a model to have. I would contend that the experiments conducted here are mixed in their answer to this question. On one hand, they imply that yes, we can achieve good performance on all these safety metrics if we use ensemble models. This is because ensemble models outperformed the baseline model with regard to all four of the safety properties tested here (adversarial robustness, OOD robustness, anomaly detection and calibration).

However, this affirmative response must be qualified. This is because ensemble models do not always achieve the highest performance on any single calibration metric, and those models that do achieve this best-in-class performance on any single safety metric often end up underperforming the baseline model on another metric. For instance, the models that achieved the best adversarial robustness were those models that were trained using some

form of adversarial training. For example, many adversarially trained models outperformed all ensemble models by >50% in terms of adversarial accuracy, yet end up being incomparably worse when it comes to OOD robustness. As such, in the sense of achieving the best result possible on any single metric, it does appear that there is sometimes a tradeoff.

References

Borowski, Judy, Roland S. Zimmermann, Judith Schepers, Robert Geirhos, Thomas S. Wallis, Matthias Bethge, and Wieland Brendel. 2021. “Exemplary Natural Images Explain CNN Activations Better than State-of-the-Art Feature Visualization.”

Bostrom, Nick. 2014. *Superintelligence: Paths, Dangers, Strategies*. Oxford: Oxford University Press.

Clanuwat, Tarin, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazauki Yamamoto, and David Ha. 2018. “Deep Learning for Classical Japanese Literature.”

Cunn, Yann L., Leom Bottou, Yoshua Bengio, and Patrick Haffner. 1998. “Gradient-Based Learning Applied to Document Recognition.” *PROCEEDINGS OF THE IEEE*.

Deng, Li. 2012. “The mnist database of handwritten digit images for machine learning research.”

DeVries, Terrance, and Graham W. Taylor. 2017. “Improved Regularisation of Convolutional Neural Networks with Cutout.”

Goodfellow, Ian J., Jonathon Schlens, and Christian Szegedy. 2014. “Explaining and Harnessing Adversarial Examples.”

Hendrycks, Dan, Steve Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, et al. 2021. “The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization.”

Hendrycks, Dan, Nicolas Carlini, John Schulman, and Jacob Steinhardt. 2022. “Unsolved Problems in ML Safety.”

Hendrycks, Dan, and Kevin Gimpel. 2017. “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks.” *ICLR 2017*.

Hendrycks, Dan, Mantas Mazeika, and Thomas Dietterreich. 2018. “Deep Anomaly Detection with Outlier Exposure.” *ICLR 2019*.

Kingma, Diederik P., and Jimmy L. Ba. 2015. “Adam: A Method for Stochastic Optimization.” *ICLR 2015*.

Lipton, Zachary. 2016. “The Mythos of Model Interpretability.” *2016 ICML Workshop on Human Interpretability in Machine Learning*.

Madry, Aleksander, Aleksander Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. “Towards Deep Learning Models Resistant to Adversarial Attacks.”

Mu, Norman, and Justin Gilmer. 2019. “MNIST-C: A Robustness Benchmark for Computer Vision.”

Olah, Chris, Alexander Mordvintsev, and Ludwig Schubert. 2017. “Feature Visualization.”

Ovadia, Yaniv, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. “Can You Trust Your Model’s Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift.” *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.

Russel, Stuart. 2019. *Human Compatible: Artificial Intelligence and the Problem of Control*. N.p.: Viking.

Schott, Luks, Jonas Rauber, Mathias Bethge, and Wieland Brendel. 2018. “Towards the first adversarially robust neural network model on MNIST.”

Strauss, Thilo, Markus Hanselmann, Andrej Junginger, and Holger Ulmer. 2018. “Ensemble Methods as a Defence to Adversarial Perturbations Against Deep Neural Networks.”

Wang, Haoqi, Zhizhong Li, Litong Feng, and Wayne Zhang. 2022. “ViM: Out-Of-Distribution with Virtual-logit Matching.”

Xiao, Han, Kashif Rasul, and Roland Vollgraf. 2017. “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.”

Yun, Sangdoo, Dongyoon Han, Seong J. Ooh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. “CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features.”

Zhang, Hongyi, Moustapha Cisse, Yann Dauphin, and David Lopez-Paz. 2017. “Mixup: Beyond Empirical Risk Minimisation.”