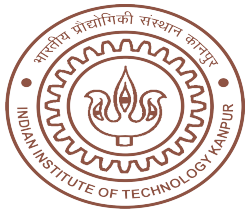
Center for Artificial Intelligence and Robotics (CAIR)
Defense Research and Development Organization (DRDO), Bengaluru

A Summer Internship Project Report on:

"Comparative Analysis of Out-Of-Distribution Detection Methods: A Literature Review and Experimental Evaluation"

Under the Supervision of:
Smt. Faheema AGJ, Sc 'F',
Ms. Lakshmi Annamalai, Sc 'E'
ISRD, CAIR, DRDO
(15/5/23 *to* 14/7/23)



Department of Mathematics and Statistics
Indian Institute of Technology Kanpur

Submitted by: Kaushik Thakkar
Roll Number: 221329
Branch: MSc in Statistics (2022-2024)
Student ID(CAIR): 1122

Acknowledgement

I would like to express my sincere and profound gratitude to the Director, CAIR, DRDO, Bangalore for providing an opportunity to undergo project internship of 8 weeks. In regards to the internship, I bear immense pleasure in expressing my special gratitude to Ms. Lakshmi Annamalai, scientist "E" and Smt. Faheema AGJ, Sc 'F' of ISRD Deparment as my project guide and thanks to all the employees of CAIR for their support and guidence whenever required.

This training wasn't possible if Dr. Mohua Banerjee, HOD , Department of Mathematics and Statistics , IIT Kanpur wouldn't have allowed me in the first place, so thanks to her as well.

Contents

1 Abstract **3**

2 Introduction **4**

3 Models and Datasets used in the study **5**

4 Methodology Used **6**

5 Methods Applied **8**
5.1 Mahalanobis Method. 8
5.2 Energy based Method. 9
5.3 Entropy Based Method. 10
5.4 Max-Logit 11
5.5 KL-Matching 12
5.6 Virtual Logit Matching 13

6 Evaluation Metrics used **15**
6.1 Precision 15
6.2 Recall 15
6.3 F1 Score 16

7 Results **17**
7.1 MNIST 17
7.2 SVHN 17
7.3 CIFAR-10 18

8 Conclusion **19**

1 Abstract

Deep neural networks have achieved high accuracy on many classification tasks, e.g. speech recognition, object detection and image classification. However one challenge is to measure the predictive uncertainty of these models. The predictive uncertainty of DNNs is closely related to the problem of detecting abnormal samples that are drawn far away from in-distribution (i.e., distribution of training samples) statistically or adversarially.

Deep neural networks are often trained with closed-world assumptions, i.e., the test data distribution is assumed to be similar to the training data distribution. However, when employed in real-world tasks, this assumption doesn't hold true, leading to a significant drop in their performance. An ideal AI system should generalize to Out-of-Distribution (OOD) examples whenever possible and flag the ones that are beyond its capability to seek human intervention. Deployment of AI solutions has increased manifold during the last few years, even in safety-critical domains such as medicine, finance, military and many more. Knowing whether an AI model is safe to use in a deployment context is crucial to AI users.

In this study several methods for OOD detection are studied and applied on various state of the art datasets. Finally they are compared with the method provided by CAIR. This report gives an idea about the competence of different OOD Detection methods in identifying OOD samples or non-believable samples as in this case.

2 Introduction

What's Out-Of-Distribution?

The term "distribution" has slightly different meanings for Language and Vision tasks. If we Consider a dog breed image classification task, here the images of dogs would be in-distribution while images like bike, ball, etc. would be out-of-distribution. Another example would be for a Question-Answering model trained on Maths questions, a question from History is out-of-distribution.

What's Out-Of-Distribution Detection?

Out-of-Distribution (OOD) detection separates ID (In-Distribution) data and OOD data from input data through a model. This problem has attracted huge attention in the area of AI. The method based on deep learning is the most studied in OOD detection. There are several pre existing state of the art methods for detecting out of distribution samples viz. Maximum Softmax (MSP), Maximum Logit, Method based on Mahalanobis Distance, Energy Based Method, Entropy Based Method, Method based on KL Divergence, and Virtual Logit Matchign(Vim). The methods named above have been studied and applied in this Experiment and was compared with the Method provided by CAIR, But this was done in the context of Believable and Non-Believable samples which is stated afterwards.

Why Out-Of-Distribution Detection?

When AI models are deployed in real life application, there may be cases when it would encounter such data which is unknown to the model and it is not trained accordingly to make predictions on such data. So we need to prevent the models from making unreliable and erroneous predictions on unfamiliar data. Making wrong predictions with high confidence on unfamiliar data can lead to mishaps especially when AI models are deployed in safety-critical applications such as defence, security and medical fields .
We also know that usually data drifts over time, i.e. there is change in the statistical properties or characteristics of the input data that the model was trained on. Also there is Concept drift that refers to changes in the relationship between the input features and the target variable that the model is trying to predict. Hence, OOD detection is important to prevent AI systems from making prediction errors.

3 Models and Datasets used in the study

The experiments were performed using deep CNN such as custom networks, ResNet-18, ResNet-50 on three state-of-the-art datasets viz. MNIST, CIFAR-10 and SVHN.

1. MNIST Dataset:

The MNIST dataset is a collection of 70,000 handwritten digits from 0 to 9, presented as 28x28-pixel grayscale images. It is widely used as a benchmark dataset for training and evaluating machine learning models in the field of computer vision. The dataset consists of 60,000 images for training and 10,000 images for testing, with each image associated with a label indicating the digit it represents.

2. CIFAR-10:

The CIFAR-10 dataset is a widely-used benchmark dataset in the field of computer vision. It consists of 60,000 color images, each with a resolution of 32x32 pixels. The dataset is divided into 10 classes, with approximately 6,000 images per class. The classes include common objects such as airplanes, cars, birds, cats, dogs, and more. The CIFAR-10 dataset is often used for tasks like image classification, object recognition, and machine learning algorithm evaluation.

3. SVHN

The SVHN (Street View House Numbers) dataset is a widely-used dataset in the field of computer vision and machine learning. It consists of real-world images of house numbers captured from Google Street View. The dataset contains over 600,000 images divided into three sets: a training set with around 73,000 images, a validation set with around 26,000 images, and a test set with around 130,000 images. The house numbers in the dataset range from 0 to 9 and are often displayed in varying sizes and orientations. The SVHN dataset is commonly used for tasks like digit recognition, multi-digit recognition, and object detection.

4 Methodology Used

This section gives the complete methodology applied in the experiment and the way of comparison of the method given by CAIR with other state of the art methods.

Firstly we classify the test set samples as believable and non-believable samples. Believable samples refers to those test samples where the True Test label matches the Prediction made by the model that is trained on that particular data set. If there are n test samples then we create a vector $yBlv$ where:

$$yBlv_i = \begin{cases} 1, & \text{if } True\,Label\,of\,i^{th}\ \ sample = Predicted\,Label \\ 0, & \text{if } Otherwise \end{cases}$$

Also created a vector $yNonBlv$ where:

$$yNonBlv_i = \begin{cases} 0, & \text{if } True\,Label\,of\,i^{th}\,sample = Predicted\,Label \\ 1, & \text{if } Otherwise \end{cases}$$

Thus vector $yBlv$ has value 1 for all believable samples and $yNonBlv$ has value 1 for all the non-believable samples.

Now defined a threshold vector ranging from 0 to 1 with step 0.05

All the methods discussed below calculates an anomaly score to each of the test samples and we assign the sample to in-distribution or the believable space if the anomaly score is lower than the threshold otherwise we call it a out-of-distribution sample.

A vector $yPredBlv$ is created where :

$$yPredBlv_i = \begin{cases} 1, & \text{if } anamoly\,score < threshold \\ 0, & \text{if } Otherwise \end{cases}$$

Also created a vector $yPredNonBlv$ where:

$$yPredNonBlv_i = \begin{cases} 1, & \text{if } anamoly\, score > threshold \\ 0, & \text{if } Otherwise \end{cases}$$

Now this same exercise is repeated for all the 20 values of threshold.

And for each value of this threshold, recall and precision is calculated between the pair of vectors $yBlv, yPredBlv$ and $yNonBlv, yPredNonBlv$.

Having calculated this values for all the thresholds, they are plotted and the resulting plots are shown in the section 7.

5 Methods Applied

The following 6 methods were studied and implemented for the comparative study in this Experiment.

5.1 Mahalanobis Method.

[2]This is a method for detecting any abnormal samples, which is applicable to any pre-trained softmax neural classifier. In this method the class conditional Gaussian distributions with respect to features of the deep models under Gaussian discriminant analysis are obtained, which result in a confidence score based on the Mahalanobis distance.

Let $x \in \mathscr{X}$ be an input and $y \in \mathscr{Y} = \{1, 2, ..., C\}$ be its label. Suppose that a pre-trained softmax neural classifier is given. Then, without any modification on the pre-trained softmax neural classifier, obtained a generative classifier assuming that a class-conditional distribution follows the multivariate Gaussian distribution. Specifically, we define C class-conditional Gaussian distributions with a tied covariance $\sum$ and assume that $P(f(x)|y = c) = N(f(x)|\mu_c, \sum)$, where μ_c is the mean of multivariate Gaussian distribution of class $c \in \{1, 2, ..., C\}$. Here, our approach is based on a simple theoretical connection between GDA and the softmax classifier: the posterior distribution defined by the generative classifier under GDA with tied covariance assumption is equivalent to the softmax classifier . Therefore, the pre-trained features of the softmax neural classifier $f(x)$ might also follow the class-conditional Gaussian distribution.

To estimate the parameters of the generative classifier from the pre-trained softmax neural classifier, we compute the empirical class mean and covariance of training samples $\{(x_1, y_1), ..., (x_N, y_N)\}$:

$$\hat{\mu_c} = \frac{1}{N_c} \sum_{i:y_i=c} f(x_i)$$

$$\hat{\sum} = \frac{1}{N} \sum_{c} \sum_{i:y_i=c} (f(x_i) - \hat{\mu_c})(f(x_i) - \hat{\mu_c})^T$$

where N_c is the number of training examples with the class label c. This is equivalent to fitting the class conditional Gaussian distributions with a tied covariance to training samples under the maximum likelihood estimator.

Using the above induced class-conditional Gaussian distributions, we define the confidence score $M(x)$ using the Mahalanobis distance between test sample x and the closest class-conditional Gaussian distribution, i.e.,

$$M(x) = max_c \; -(f(x) - \hat{\mu}_c)^T \hat{\sum}^{-1} (f(x) - \hat{\mu}_c)$$

Each test sample is assigned the class to which it has the least mahalanobis distance. First the Test data is split into believable and non-believable sample points. Now the distance matrices are calculated separately for these believable and non-believable samples. For the believable set, first we identify the class predictions for each of the samples. A sample is assigned to a class c if its distance to class c is minimum. Further for a given threshold if the minimum distance exceeds the threshold limit we assign the sample to class -1. Same thing is also done for the set of non-believable samples. Now Precision and Recall is calculated by considering the true labels of the test set and the predictions based on the Mahalanobis-distance.

Since this method utilizes empirical class mean and covariance of training samples, there is a caveat such that it can be affected by the properties of training data.

5.2 Energy based Method.

[3] This OOD Detection method uses energy scores. Energy Scores better distinguish in- and out-of-distribution samples than the traditional approach using the softmax scores. Unlike softmax confidence scores, energy scores are theoretically aligned with the probability density of the inputs and are less susceptible to the overconfidence issue. Hence samples with higher energies can be interpreted as data with a lower likelihood of occurrence.

For an given input (x, y) we can express the free energy function $E(x; f)\, over\, x \in R^D$ in terms of the denominator of the softmax activation:

$$E(x; f) = -T \log \sum_i k e^{\frac{f_i(x)}{T}}$$

And finally we decide whether the sample is an OOD/non-believable or not based on the negative energy score of the sample. Suppose τ is the threshold value then

$$g(x) = \begin{cases} 1, & \text{if } -E(x; f) > \tau \\ 0, & \text{if } Otherwise \end{cases}$$

where if $g(x) = 1$ implies x is classified as in distribution otherwise it is classified as an OOD sample or a non-believable sample.

5.3 Entropy Based Method.

Entropy refers to a measure used to assess the uncertainty or confidence of a model when making predictions on unseen or out-of-distribution samples. It helps identify samples that are significantly different from the training data and potentially belong to a different distribution.
This method implements Entropy Based OOD Detection.
In this we calculate the Entropy based on the logits of a classifier. Higher entropy means more uniformly distributed posteriors, indicating larger uncertainty. Entropy is calculated as :

$$H(x) = -\sum_{i=1}^{C} \sigma_i(f(x)) \log(\sigma_i(f(x))) \tag{1}$$

where σ_i indicates the i^{th} softmax value and C is the number of classes.

In the context of OOD detection, a threshold can be set on the entropy value. Samples with entropy below the threshold are considered in-distribution, as they resemble the training data and the model is confident in its predictions. Conversely, samples with entropy above the threshold are considered out-of-distribution, as they exhibit characteristics that are significantly different from the training data, leading to higher uncertainty in the model's predictions.

Suppose τ is the threshold value then:

$$g(x) = \begin{cases} 1, & \text{if } H(x) < \tau \\ 0, & \text{if } Otherwise \end{cases}$$

where if $g(x) = 1$ implies x is classified as in distribution otherwise it is classified as an OOD sample or a non-believable sample.

5.4 Max-Logit

[1] In this method it is proposed that instead of using the softmax probabilities(However, it is shown that the MSP is problematic for realistic in-distribution datasets with many classes, such as ImageNet and Places365) using the negative of the maximum unnormalized logit for an anomaly score $-\max_k f(x)_k$,, which we call MaxLogit would perform well.

Here $f(x)_k$ is the k^{th} logit value predicted by f Since the logits are unnormalized, they are not affected by the number of classes and can serve as a better baseline for large-scale out-of-distribution detection.It is based on the observation that OOD samples often result in lower confidence or less well-calibrated predictions compared to in-distribution samples.
Logit values represent the output of the model before applying the softmax function to obtain class probabilities. By comparing the maximum logit value of a given input with a predefined threshold, we can determine whether the input is likely to be in-distribution or OOD. If the maximum logit value is below the threshold, the sample is classified as OOD; otherwise, it is considered in-distribution. i.e.

$$g(x) = \begin{cases} 1, & \text{if } ML(x) > \tau \\ 0, & \text{if } Otherwise \end{cases}$$

where τ is the threshold value and $ML(x) = -\max_k f(x)_k$ if $g(x) = 1$ implies x is classified as in distribution otherwise it is classified as an OOD sample or a non-believable sample.

The max Logit method is a straightforward technique that does not require additional training or complex modifications to the model. It leverages the observation that OOD samples tend to have lower maximum logit values compared to in-distribution samples. By setting an appropriate threshold, the method can effectively identify potential OOD inputs.

It is important to note that the max Logit method is a heuristic approach and may not be suitable for all scenarios or datasets. It is recommended to evaluate and validate its performance on a separate OOD dataset or using appropriate evaluation metrics to ensure its effectiveness in OOD detection for a specific problem domain.

5.5 KL-Matching

Kullback-Leibler (KL) divergence: It is a measure of how one probability distribution differs from another. It is commonly used in information theory, statistics, and machine learning. KL divergence quantifies the amount of information lost when one distribution is used to approximate another.

KL divergence measures the "distance" or dissimilarity between two probability distributions, denoted as P and Q. The KL divergence from P to Q, denoted as $DKL(P||Q)$, is defined as:

$$DKL(P||Q) = \sum P(x) \log \left(\frac{P(x)}{Q(x)} \right)$$

where x represents the possible outcomes of the distributions

The KL (Kullback-Leibler) matching method is an approach for Out-of-Distribution (OOD) detection based on minimizing the KL divergence between the predicted class probabilities of an in-distribution model and a reference distribution.
Since it is seen that some classes tend to be predicted with low confidence and others high confidence. The shape of predicted posterior distributions is often class dependent. We capture the typical shape of each class's posterior distribution and form posterior distribution templates for each class. During test time, the network's softmax posterior distribution is compared to these templates and an anomaly score is generated. More concretely, we compute k different distributions d_k, one for each class. We write

$$d_k = E_{x' \sim X_{val}}[p(y|x')]$$

where $k = argmax_k\, p(y = k|x')$ Then for a new test input x, we calculate the anomaly score as $KL(x) = min_k KL[p(y|x)||d_k]$ rather than the MSP baseline $-\max_k\, p(y = k|x)$.

If the calculated KL divergence exceeds the threshold, the sample is classified as OOD; otherwise, it is considered in-distribution. i.e.

$$g(x) = \begin{cases} 1, & \text{if } KL(x) < \tau \\ 0, & \text{if } Otherwise \end{cases}$$

where τ is the threshold value and $ML(x) = -\max_k\, f(x)_k$ if $g(x) = 1$ implies x is classified as in distribution otherwise it is classified as an OOD sample or a non-believable sample.

5.6 Virtual Logit Matching

[4] Most of the existing OOD Detection algorithms solely depend a single input source: they depend either on the feature, the logits or the softmax probabilities. But the space of OOD samples is very huge, there may be OOD samples which are easier to detect in the feature space but difficult to distinguish in the logit space and vice versa. "Virtual-Logit-Matching:ViM" is an OOD scoring method which combines the class agnostic score from the feature space and the In-Distribution(ID) class-dependent logits. Specifically, an additional logit representing the virtual OOD class is generated from the residual of the feature against the principal space, and then matched with the original logits by a constant scaling. The probability of this virtual logit after softmax is the indicator of OOD-ness.

The pipeline for the above method is as follows: For feature x,
(1) extract the residual $x^{P^\perp}$ of x against the principal subspace P;
(2) Converted the norm $\| x^{P^\perp} \|$ to a virtual logit by rescaling; and
(3) output the softmax probability of the virtual logit as the ViM score.
Notations: C is the number of classes, N is the feature dimension, and W and b are the classification weight and bias, respectively. and D is the dimensionality of the principal subspace.

Complete details:

1. Firstly we offset the feature space by a vector $o = -(W^T)^+ b$ so that it is bias free in the computation of logits.

2. The principal subspace P is defined by the training set X, where rows are features in the new coordinate system with origin o. Suppose the eigen decomposition on the matrix $X^T X$ is $X^T X = Q \Lambda Q^{-1}$ where eigenvalues in Λ are sorted decreasingly, then the span of the first D columns is the D-dimensional principal subspace P.

3. The residual $x^{P^\perp}$ is the projection of x onto $P^\perp$, Let the $(D+1)^{th}$ column to the last column of Q be a new matrix $R \in \mathscr{R}^{N*(N-D)}$, then $x^{P^\perp} = RR^T x$. The residual $x^{P^\perp}$ is sent to the next step.

4. The Virtual Logit is given by: $l_0 := \alpha \sqrt{x^T R R^T x}$ is the norm of the residual rescaled by a per-model constant α.

5. Where

$$\alpha = \frac{\sum_{i=1}^{k} \max_{j=1,2,..C} \left\{ l_j^i \right\}}{\sum_{i=1}^{k} \parallel x^{P^{\perp}} \parallel}$$

where $x_1, x_2, ..., x_K$ are uniformly sampled K training examples, and l_j^i is the $j-th\, logit$ of x_i. In this way, on average, the scale of the virtual logit is the same as the maximum of the original logits.

6.The probability corresponding to the virtual logit is defined as ViM. Mathematically, let the i-th logit of x be l_i , and then the score is

$$ViM(x) = \frac{e^{\alpha\sqrt{x^T R R^T x}}}{\sum_{i=1}^{C} e^{l_i} + e^{\alpha\sqrt{x^T R R^T x}}}$$

This equation reveals that two factors affect the ViM score: if its original logits are larger, then it is less of an OOD example; while if the norm of residual is larger, it is more likely to be OOD. The computational overhead is comparable to the last fully-connected layer (mapping from feature to logit) in the classification network, which is small.

If the ViM socre is larger than a threshold the sample is considered as OOD and if it is less than the threshold it is considered an in-distribution sample.i.e.

$$g(x) = \begin{cases} 1, & \text{if } ViM(x) < \tau \\ 0, & \text{if } Otherwise \end{cases}$$

where τ is the threshold value and if $g(x) = 1$ implies x is classified as in distribution otherwise it is classified as an OOD sample or a non-believable sample.

6 Evaluation Metrics used

Evaluation Metrics used are: Precision, Recall and F1 Score
Precision and Recall are performance metrics used to evaluate the accuracy of a classification model.

6.1 Precision

It measures the proportion of correctly predicted positive instances out of all instances predicted as positive by the model. In other words, precision quantifies the ability of a model to accurately identify the positive class.

Precision is calculated as:

$$Precision = \frac{TruePositives}{(TruePositives + FalsePositives)} \tag{2}$$

Interpreting Precision Score: A high precision value indicates that the model has a low rate of false positives, meaning it is effective at identifying positive instances correctly and minimizing the risk of falsely labeling negative instances as positive. On the other hand, a low precision value suggests a higher rate of false positives, indicating that the model may be incorrectly classifying negative instances as positive.
Limitation: This emphasizes that, while precision is beneficial, it does not tell the entire tale. It makes no mention of how many actual positive class examples were incorrectly classified as negative, resulting in so-called false negatives.

6.2 Recall

Recall, also known as sensitivity or true positive rate measures the proportion of actual positive instances that are correctly identified by the model. In simple terms, recall quantifies the model's ability to find all positive instances.

Recall is calculated as:

$$Recall = \frac{TruePositives}{(TruePositives + FalseNegatives)} \tag{3}$$

Interpretation of Recall: A high recall value indicates that the model is adept at identifying positive instances, minimizing the number of instances that are incorrectly labeled as negative. It signifies that the model is effective at avoiding false negatives, which can be critical in scenarios where the cost of missing positive instances is high, such as in medical diagnosis or detecting fraudulent transactions.

Conversely, a low recall value suggests that the model may miss a significant number of positive instances, leading to a higher rate of false negatives. It implies that the model may not be sufficiently sensitive in capturing positive instances.

While precision focuses on the accuracy of positive predictions, recall provides insights into the model's ability to find all positive instances.

6.3 F1 Score

The F1 score is a metric that combines precision and recall to provide a balanced evaluation of a classification model's performance. It is particularly useful in situations where there is an imbalance between the positive and negative classes. The F1 score is calculated as the harmonic mean of precision and recall, using the following formula:

$$F1Score = \frac{2 * (precision * recall)}{(precision + recall)} \tag{4}$$

The F1 score ranges from 0 to 1, where 1 represents the best possible score (perfect precision and recall) and 0 represents the worst score (either zero precision or recall). Overall, the F1 score offers a comprehensive evaluation of a model's performance by considering both precision and recall, making it a valuable metric for assessing the effectiveness of a classification model.

In this experiment we calculate the Precision and Recall and F1 Score of all the methods at different thresholds ranging from 0 to 1.

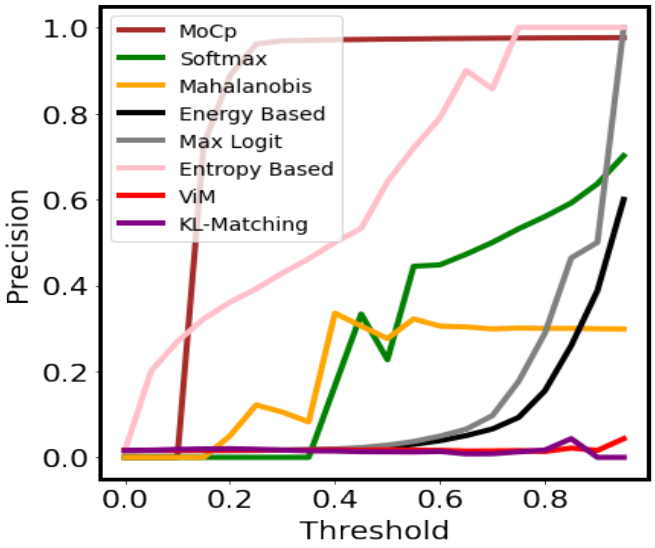
7 Results

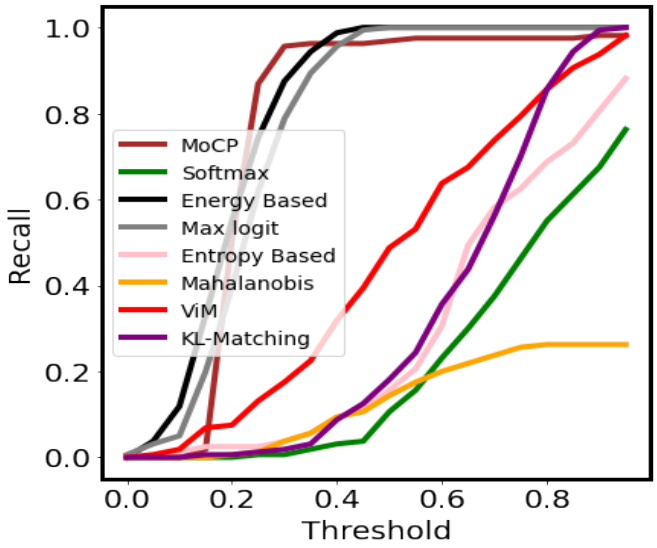
The following results were obtained in the study and we are mainly concerned with the ability of all the methods to detect the non-believable samples, so we consider the measures precision and recall only for the non-believable samples calculated over different threshold values for all the methods:

7.1 MNIST

1. **Precision of non-believable samples**: MoCP(the method provided by CAIR) is better than all the other methods.

2. **Recall of non-believable samples**: MoCP, Energy based and Max-Logit performs equally well outperforming the other two methods.

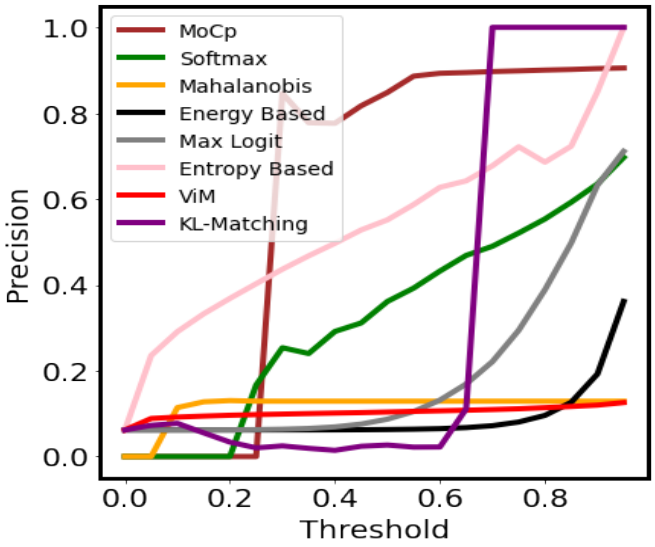


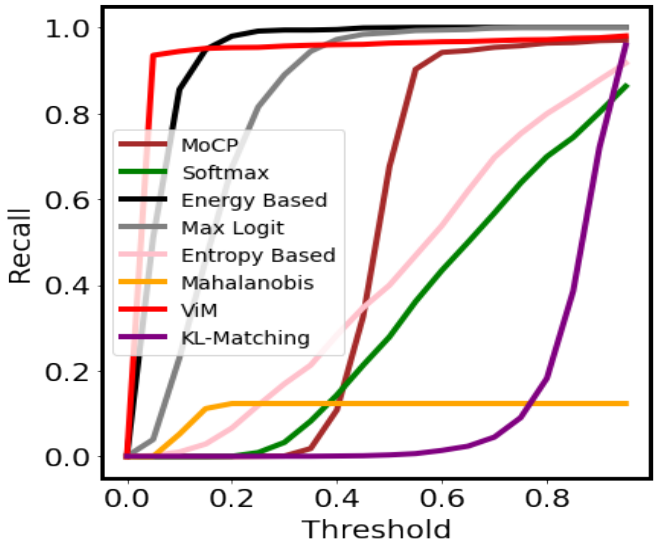


7.2 SVHN

1. **Precision of non-believable samples**: If the threshold is above 0.65 the KL-matching method beats MoCP and threshold above 0.9 Entropy method is the best. However other than this MoCP beats all the other methods.

2. **Recall of non-believable samples**: ViM, Max-Logit and Energy-based method are better than MoCP, while MoCP beats all the other methods.

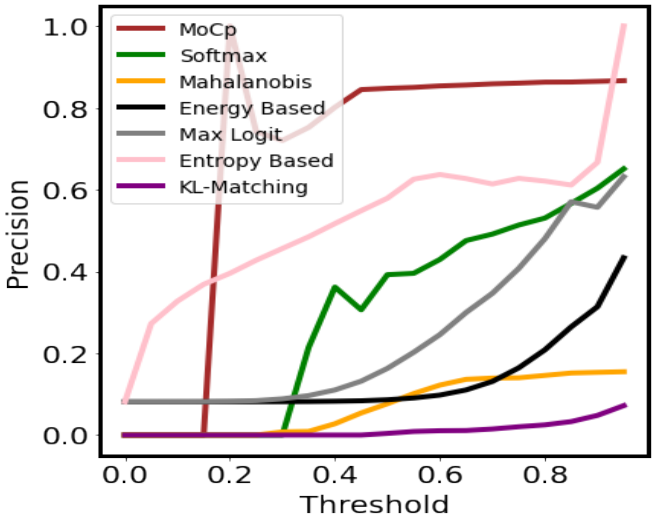


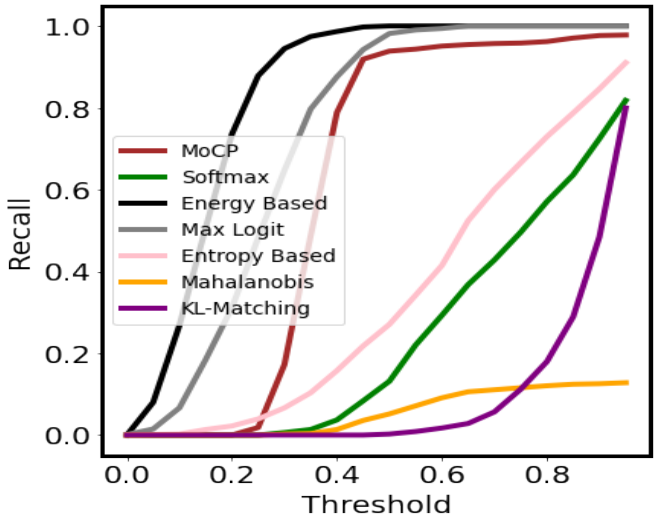


7.3 CIFAR-10

1. **Precision of non-believable samples**: MoCP outperforms all the other methods.

2. **Recall of non-believable samples**: Energy-based and Max-Logit is better than MoCP, MoCP outperforms all the other methods.





8 Conclusion

By looking at the above graphs it can be concluded that The MoCP method provided by CAIR performs better than many of the state-of-the-art methods for detecting the non-believable samples in terms of both recall and precision.

Scope for future work: Since we see that different methods performs differently on different datasets so working towards a more holistic and general approach towards OOD detection can be a key area of research in future.

References

[1] Dan Hendrycks et al. *Scaling Out-of-Distribution Detection for Real-World Settings*. 2022. arXiv: 1911.11132 [cs.CV].

[2] Kimin Lee et al. *A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks*. 2018. arXiv: 1807.03888 [stat.ML].

[3] Weitang Liu et al. *Energy-based Out-of-distribution Detection*. 2021. arXiv: 2010.03759 [cs.LG].

[4] Haoqi Wang et al. *ViM: Out-Of-Distribution with Virtual-logit Matching*. 2022. arXiv: 2203.10807 [cs.CV].