Textual OOD Detection for Intent Classification in the Banking Industry

Pierre-Emmanuel Diot*
ENSAE
pierre-emmanuel.diot@ensae.fr

Paul O'Mahony*
ENSAE
paul.omahony@ensae.fr

Abstract

With the growing number of online banks or digitalized services of traditional banks, the need for human contact in customer support drops drastically. Deep learning and natural language processing methods allow today to answer efficiently and precisely to customers' questions via chatbots and to get closer to the way a human would answer. However, it is important that these methods do not give wrong information and do not answer questions they are not capable of answering. In this report we evaluate several out-of-distribution detection methods to overcome these problems, and focus on the NLP task of intent classification in the banking domain. The code is released to the community[1].

1 Introduction

Large Language Models (LLMs) have recently made significant breakthroughs in various Natural Language Processing (NLP) tasks, such as text generation, question-answering, and language translation. However, as the use of LLMs becomes more widespread, concerns about their safety and ethical implications have arisen. To ensure the widespread adoptability of LLMs, it is crucial to address issues such as fairness [2, 3, 4], out-of-distribution (OOD) detection [5, 6, 7], and adversarial attack detection [8, 9]. These issues are critical to ensure that LLMs operate within ethical and legal boundaries, do not propagate harmful biases or stereotypes, and can detect and defend against malicious attacks. In this context, research efforts have focused on developing methods to detect and mitigate these issues, as well as building more transparent and interpretable models that can be better understood and audited by humans.

[1]The code is available at https://github.com/PeDiot/OOD-Detection-Intent-Classification and is based on [1]

In this work, we choose to focus on OOD detection as it is a crucial aspect of ensuring the safety and reliability of large language models. OOD detection refers to the ability of a model to identify input data that falls outside of the distribution of data it was trained on, and flag it as potentially unsafe or unreliable. This is important because large language models are often deployed in real-world applications where they may encounter input data that is significantly different from what they were trained on. If the model is not able to detect such OOD inputs, it may produce unreliable or even harmful outputs, putting users at risk. Therefore, developing effective OOD detection techniques is critical for ensuring the safety and trustworthiness of large language models.

1.1 Problem Framing

Mathematical formulation. We take the formulation of the problem made in [10]. For a multiclass classification problem, we have $\mathcal{X}$ the textual input space, $\mathcal{Y} = \{1, \cdots, C\}$ with $C \geq 2$ the target space. We denote p_{XY} the probability distribution of the training dataset $\mathcal{D}_N = \{(\mathbf{x}_i, y_i),\ \mathbf{x}_i, y_i \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^N$ supposed made of $N \geq 1$ i.i.d samples. Moreover, the classifier trained using $\mathcal{D}_N$ is denoted by $f_N : \mathcal{X} \to \mathcal{Y}$.

It is very likely that the classifier f_N will have to deal with samples which are out of the training distribution, such that $(\mathrm{x}, y) \not\sim p_{XY}$, once put into production. To label such samples we define the indicator variable $z = \mathbb{1}\{(\mathrm{x}, y) \not\sim p_{XY}\}$.

Finally, the decision function for OOD detection is defined by:

$$g(\mathbf{x}, \gamma) = \begin{cases} 1 & \text{if } s(\mathbf{x}) > \gamma \\ 0 & \text{if } s(\mathbf{x}) \leq \gamma \end{cases} \tag{1}$$

with $s : \mathcal{X} \to \mathbb{R}$ a dissimilarity function which computes a score of an element of $\mathcal{X}$ w.r.t the training in-distribution. $\gamma \in \mathbb{R}$ is a threshold whose value will be discussed later.

Performance evaluation To measure the performance of OOD detectors we use several metrics that we find significant. First, recall that the **False Alarm Rate** (FAR) is the proportion of IN samples that are misclassified as OUT while the **True Detection Rate** (TDR) corresponds to the proportion of OUT samples that are correctly classified.

The **Area Under the Receiver Operating Characteristic curve** (AUROC) [11] is the area under the ROC curve which plots the true detection rate against the false alarm rate for different thresholds. It corresponds to the probability that an in-distribution example is given a lower score than an OOD example, according to 1.

The **Area Under the Precision-Recall curve** (AUPR-IN, AUPR-OUT) [12] is the area under the curve which plots the true detection rate (recall) against the proportion of true OOD samples amongst all the samples classified as OOD by the detectors (precision), for different thresolds.

2 Experiments Protocol

We discuss in this section the chosen benchmark datasets, fine-tuned models and baseline methods we use and then evaluate.

2.1 Previous work

Dissimilarity functions for OOD detection have been introduced in previous works. Specifically, we use the *Maximum Soft-max Probability*, *Energy*, *Mahalanobis* and finally *Cosine projection* functions respectively proposed by [13, 14, 15, 16]. Previous works have compared the performance of these functions in OOD detection on several datasets, several NLP tasks and several models [10, 17]. These works present very good results but use datasets that do not correspond to our use case. The work presented by [15] use a banking dataset but the results and comparisons of the methods rely on a single transformer model which is RoBERTa [18]. Moreover this paper is not focused on the banking domain but rather on intent classification in several domains and only uses the Mahalanobis function as hidden-states-based scorer.

2.2 Dataset selection

The benchmark we build is composed of in-distribution samples from the training set of the BANKING77 [19] dataset. The out-distribution instances are derived from the ATIS [20] (Airline Travel Information Systems), the banking examples of CLINC150 [21], BITEXT[2] datasets, and the BANKING77 testing set. We indeed consider that all the samples that have not been used to train the classifiers are OOD and therefore are not used to fit the detectors. Both CLINC150 and BANKING77 contain intents in the banking domain, ATIS contains commercial support data from airline companies and BITEXT from 20 different domains such as retail, utilities, shipping, and so forth.

2.3 Methods and fine-tuned models

Scoring functions. We consider the four following scoring functions:

1. *Maximum Soft-max Probability* (MSP) [13]:

$$s_{\text{MSP}}(\mathbf{x}) = 1 - \max_{y \in \mathcal{Y}} p_{XY}(y|\mathbf{x}) \quad (2)$$

   where p_{XY} is the soft-probability predicted by the classifier after $\mathbf{x}$ has been observed.

2. *Energy-based* (E) [14]:

$$s_E(\mathbf{x}) = -T \times \log \left[\sum_{y \in \mathcal{Y}} \exp \left(\frac{g_y(\mathbf{x})}{T} \right) \right] \quad (3)$$

   where $g_y(\mathbf{x})$ represents the logit corresponding to the class label y and T is the temperature parameter.

3. *Mahalanobis* (M) [15]: Given a probability distribution P on $\mathbb{R}^d$ with mean vector μ and positive-definite covariance matrix S:

$$s_{\text{M}}(\mathbf{x}) = d_{\text{M}} \left(F(\mathbf{x}), F\left(\mathcal{S}^{\text{train}}_{n,\hat{y}} \right) \right) \quad (4)$$

   where d_M is the Mahalanobis distance and is defined by:

$$d_M(\mathbf{x}, P) = \sqrt{(\mathbf{x} - \mu)^T S^{-1} (\mathbf{x} - \mu)}. \quad (5)$$

[2] You can download the full dataset at https://www.bitext.com/chatbot-training-data/

and F is an aggregation function which we will define below.

4. *Cosine projection* (C) [17]:

$$s_C = -\max_{\mathbf{x}_i \in S^{\text{train}}_{n,\hat{y}}} \cos(\mathbf{x}, \mathbf{x}_i) \quad (6)$$

where $(\mathbf{x}, \hat{y})$ is an unseen sample, $S_{n,\hat{y}}$ is the set of reference embeddings with $\hat{y}$ as predicted class and cos is the cosine similarity function:

$$\cos(\mathrm{x}_1, \mathrm{x}_2) = \frac{\mathrm{x}_1 \cdot \mathrm{x}_2}{\max\left(\|\mathrm{x}_1\|_2 \cdot \|\mathrm{x}_2\|_2, \epsilon\right)} \quad (7)$$

where ϵ is a small value to avoid division by zero.

Aggregation procedures. To compute d_M, we want to extract relevant feature representations of the data from the neural networks. We take up the mathematical formulation made in [10]: ϕ_l is the function corresponding to the l-th layer of the encoder, with $1 \leq l \leq L$ and L the total number of layers in the encoder. Thus, for any given input $\mathbf{x}$, $\phi_l(\mathbf{x}) \in \mathbb{R}^d$ is the embedding of $\mathbf{x}$ in the l-th layer, where d is the dimension of the corresponding embedding space. Note that the outputs of each hidden layer are elements of $\mathbb{R}^d$. We consider three aggregation functions:

1. *Logits layer selection*:

$$F_{\text{Logits}} \equiv F(\phi_1(\mathbf{x}), \ldots, \phi_L(\mathbf{x})) = \phi_{L+1}(\mathbf{x}) \quad (8)$$

2. *Last layer selection*:

$$F_L \equiv F(\phi_1(\mathbf{x}), \ldots, \phi_L(\mathbf{x})) = \phi_L(\mathbf{x}) \quad (9)$$

3. *Power Mean aggregation*:

$$F_{\text{PM}}(\mathbf{x}) \equiv F(\phi_1(\mathbf{x}), \ldots, \phi_L(\mathbf{x})) = \frac{1}{L}\sum_{l=1}^{L} \phi_l(\mathbf{x}) := \overline{\mathbf{x}}. \quad (10)$$

Fine-tuned encoders. We used pretrained encoders that have been fine-tuned on the BANKING77 [19] dataset for intent classification purposes [3]. We test the methods described earlier on BERT [22] and DistilBERT(DIS.) [23] which have respectively reached an accuracy of 0.93 and 0.92 on the validation set.

[3] You can find the fine-tuned models on the HuggingFace hub following the links: DistilBERT and BERT

3 Results

This section aims at highlighting the empirical results obtained with the aforementioned detectors for the two fine-tuned encoders introduced in 2.3.

3.1 Methods comparison

In order to assess the performance of the different methods, we compare the classification metrics presented in 1.1 for several scorers combined with different aggregation procedures, for both the BERT and DistilBERT-based classifiers.

Model	Scorer	Ag.	AUROC	AUPR-OUT	AUPR-IN
BERT	s_{MSP}	F_{Logits}	87.26	96.83	50.40
	s_E	F_{Logits}	87.89	97.01	54.81
	s_C	F_L	99.99	100	99.91
		F_{PM}	99.99	100	99.91
	s_M	F_L	95.98	99.09	78.31
		F_{PM}	69.07	89.15	34.22
DIS.	s_{MSP}	F_{Logits}	87.88	97.04	52.59
	s_E	F_{Logits}	88.50	97.21	55.35
	s_C	F_L	99.99	100	99.86
		F_{PM}	100	100	99.94
	s_M	F_L	94.68	98.73	75.96
		F_{PM}	59.71	83.86	26.48

Table 1: OOD detection performance (in %) for different configurations

From table 1, one can notice very high AUROC and AUPR for the cosine projection scorer which remains to be qualified. Indeed this scorer outputs values close to -1 for every sample in the IN dataset while giving higher values for the OUT-DS entries, which means most of the considered thresholds manage to separate textual inputs. However, it appears more difficult to identify relevant thresolds from the distribution plot of this scorer, as shown by the related figures in the appendix. Regarding the Mahalanobis scorer computed from the encoder's last hidden layer, we obtain quite satisfactory results to separate IN from OUT instances. In other words, this scorer leverages part of the information contained in the training set to detect OOD samples.

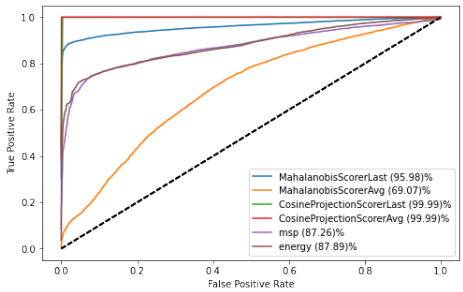

Figure 1: ROC curves for different scorers. AUROC is displayed inside brackets.

3.2 Scorer distribution

In addition to the metrics presented in the above table, it seems relevant to represent the distribution of both hidden-states-based and output-based scorers. It is decided to only report the distributions of the Mahalanobis scorer with the last-layer selection method applied on the BERT-based fine-tuned model. This choice is motivated by the noticeable split between the IN and the OUT datasets. Note the other distribution plots are depicted in the appendix.

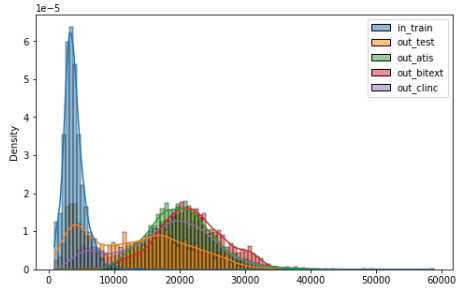

Figure 2: Density plot of s_M with F_L aggregation method

From the above plot, it can be noticed that the further the inputs from the training corpus, the higher the Mahalanobis score. Specifically, three groups can be highlighted: the training samples (in_train), banking-related text inputs (out_test and out_clinc) and not banking-related text inputs (out_atis and out_bitext). Furthermore, comparing the latter distributions to the ones derived from the s_{MSP} scorer strengthens the relevance of using hidden-states based methods to detect OOD samples. The following plot indicates more variability in the scores obtained by the OUT-DS inputs, making it more difficult to make decisions.

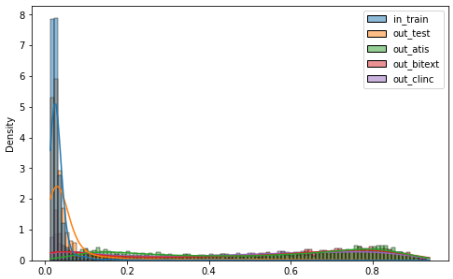

Figure 3: Density plot of s_{MSP}

Finally, combining the ROC curves from figure 1 with the distribution of the Mahalanobis scorer depicted in figures 4, 5, 6, 7, and 8, one could identify more or less flexible thresholds to detect out-of-distribution samples.

4 Discussion/Conclusion

In addition to building a benchmark for OOD detection, this work also focuses on intent classification in the banking domain. The Mahalanobis scorers, which are computed from the last layer of encoder-based classifiers, have shown to be effective in detecting OOD samples due to their ability to incorporate information from the training set. This is especially important in real-world applications where the distribution of input data may vary over time, leading to the presence of OOD samples. However, when the same scoring function is used to aggregate over all hidden states, the results obtained are not as good as those reported in the literature. This highlights the need for further investigation and development of OOD detection techniques that are robust across different types of models and scoring functions. Additionally, it is important to explore the use of other techniques such as uncertainty estimation and generative models for OOD detection in intent classification tasks. By improving OOD detection techniques, we can increase the safety and reliability of large language models for a wide range of applications, including those in the banking domain. Areas for improvement need to be emphasized. It could be interesting to use another labeling strategy as the one introduced in [17]. Another axis to be further developed could be to apply the power mean aggregation method to an arbitrary combination of hidden layers. Finally, with more resources, it would have been pertinent to leverage the potential of data depth-based scorers combined with the mean aggregation procedure which obtain very satisfying results as shown in [10].

References

[1] Maxime Darrin et al. *Todd: A tool for text OOD detection.* Version 0.0.1. Feb. 2023. URL: https://github.com/icannos/Todd.

[2] Pierre Colombo et al. "Improving Multimodal fusion via Mutual Dependency Maximisation". In: *EMNLP 2021* (2021).

[3] Georg Pichler et al. "A Differential Entropy Estimator for Training Neural Networks". In: *ICML 2022*. 2022.

[4] Pierre Colombo, Chloe Clavel, and Pablo Piantanida. "A Novel Estimator of Mutual Information for Learning to Disentangle Textual Representations". In: *ACL 2021* (2021).

[5] Eduardo Dadalto Câmara Gomes et al. "A Functional Perspective on Multi-Layer Out-of-Distribution Detection". In: ().

[6] Maxime Darrin, Pablo Piantanida, and Pierre Colombo. "Rainproof: An Umbrella To Shield Text Generators From Out-Of-Distribution Data". In: *arXiv preprint arXiv:2212.09171* (2023).

[7] Maxime Darrin et al. "Unsupervised Layer-wise Score Aggregation for Textual OOD Detection". In: *arXiv preprint arXiv:2302.09852* (2023).

[8] Marine Picot et al. "Adversarial Attack Detection Under Realistic Constraints". In: (2023).

[9] Marine Picot et al. "A Simple Unsupervised Data Depth-based Method to Detect Adversarial Images". In: (2023).

[10] Pierre Colombo, Eduardo D. C. Gomes, Guillaume Staerman, Nathan Noiry, Pablo Piantanida. "Beyond Mahalanobis-Based Scores for Textual OOD Detection." In: *NeurIPS 2022* (2022).

[11] Andrew P Bradley. "The use of the area under the roc curve in the evaluation of machine learning algorithms." In: *Pattern recognition, 30(7):1145–1159* (1997).

[12] Jesse Davis and Mark Goadrich. "The relationship between precision-recall and roc curves." In: *Proceedings of the 23rd international conference on Machine learning, pages 233–240* (2006).

[13] Dan Hendrycks and Kevin Gimpel. "A baseline for detecting misclassified and out-of-distribution examples in neural networks." In: *arXiv preprint arXiv:1610.02136* (2016).

[14] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. "Energy-based out-of-distribution detection." In: *Advances in Neural Information Processing Systems* (2020).

[15] Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. "Revisiting mahalanobis distance for transformer-based out-of-domain detection." In: *arXiv preprint arXiv:2101.03778* (2021).

[16] Engkarat Techapanurak, Masanori Suganuma, Takayuki Okatani. "Hyperparameter-Free Out-of-Distribution Detection Using Softmax of Scaled Cosine Similarity." In: *arXiv preprint arXiv:1905.10628* (2021).

[17] Wenxuan Zhou, Fangyu Liu, and Muhao Chen. "Contrastive out-of-distribution detection for pretrained transformers." In: *arXiv preprint arXiv:2104.08812* (2021).

[18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." In: *arXiv:1907.11692* (2019).

[19] Iñigo Casanueva et al. "Efficient Intent Detection with Dual Sentence Encoders". In: *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*. Data available at https://github.com/PolyAI-LDN/task-specific-datasets. Mar. 2020. URL: https://arxiv.org/abs/2003.04807.

[20] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. "The ATIS Spoken Language Systems Pilot Corpus". In: *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*. 1990. URL: https://aclanthology.org/H90-1021.

[21] Stefan Larson et al. *An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction*. 2019. DOI: 10.48550/

ARXIV.1909.02027. URL: https://arxiv.org/abs/1909.02027.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of deep bidirectional transformers for language understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186* (2019).

[23] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter." In: *arXiv preprint arXiv:1910.01108* (2019).

Appendix

A BERT results

This section emphasizes the results obtained for the detectors fitted on the BERT-based intent classifier introduced in 2.3.

A.a Scorer distribution

The following distributions draw a comparison between the ability of each scorer to discriminate between IN-DS and OUT-DS inputs. The Mahalanobis hidden-states-based scorer combined with last-layer selection appears to perform better than outputs-based scorers like energy or maximum softmax probability.

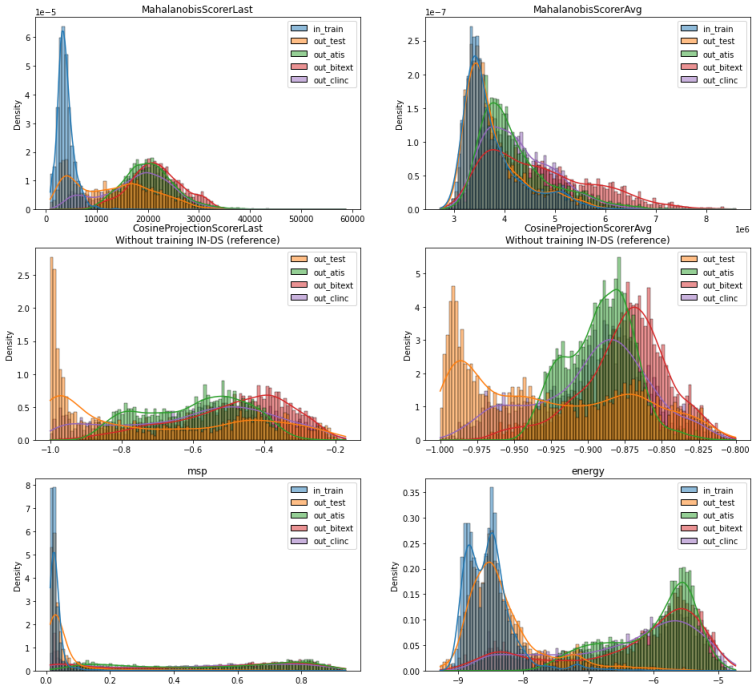

Figure 4: Distribution per scorer for different aggregation methods - All datasets. The plots related to the s_C scoring function only depicts the OUT-DS datasets since $s_C(\mathrm{x}_i) \approx -1 \; \forall(\mathrm{x}_i, y_i) \in \mathcal{D}_N$. The closer the OUT-DS score from -1, the closer to the IN-DS training instances.

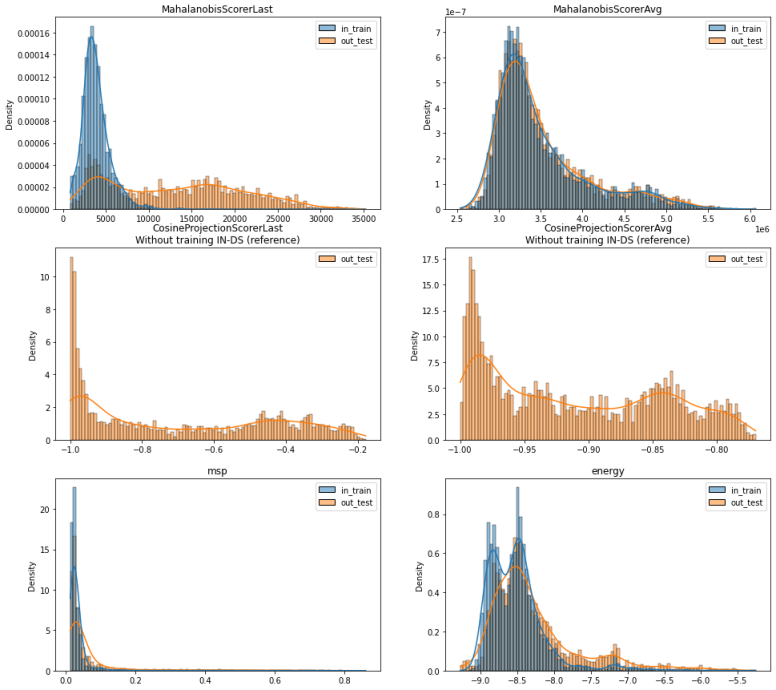

Figure 5: Distribution per scorer for different aggregation methods - BANKING77. The plots related to the s_C scoring function only depicts the OUT-DS datasets since $s_C(\mathrm{x}_i) \approx -1 \; \forall(\mathrm{x}_i, y_i) \in \mathcal{D}_N$. The closer the OUT-DS score from -1, the closer to the IN-DS training instances.

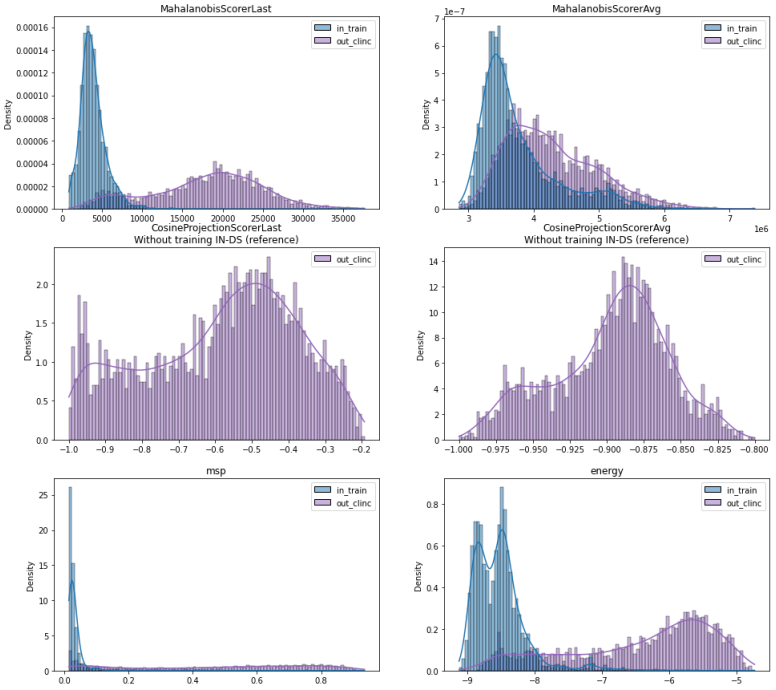

Figure 6: Distribution per scorer for different aggregation methods - CLINC150. The plots related to the s_C scoring function only depicts the OUT-DS datasets since $s_C(\mathrm{x}_i) \approx -1 \; \forall(\mathrm{x}_i, y_i) \in \mathcal{D}_N$. The closer the OUT-DS score from -1, the closer to the IN-DS training instances.

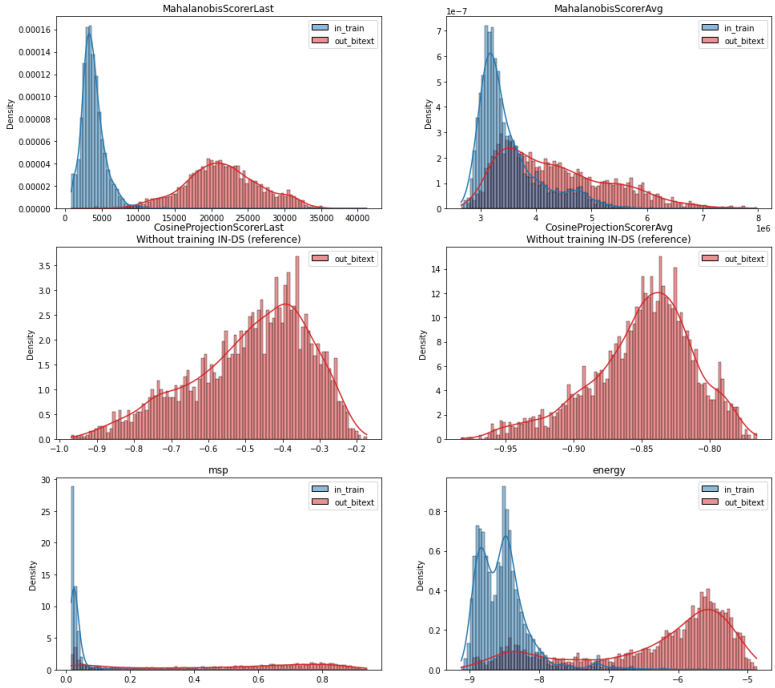

Figure 7: Distribution per scorer for different aggregation methods - BITEXT. The plots related to the s_C scoring function only depicts the OUT-DS datasets since $s_C(\mathrm{x}_i) \approx -1 \; \forall(\mathrm{x}_i, y_i) \in \mathcal{D}_N$. The closer the OUT-DS score from -1, the closer to the IN-DS training instances.

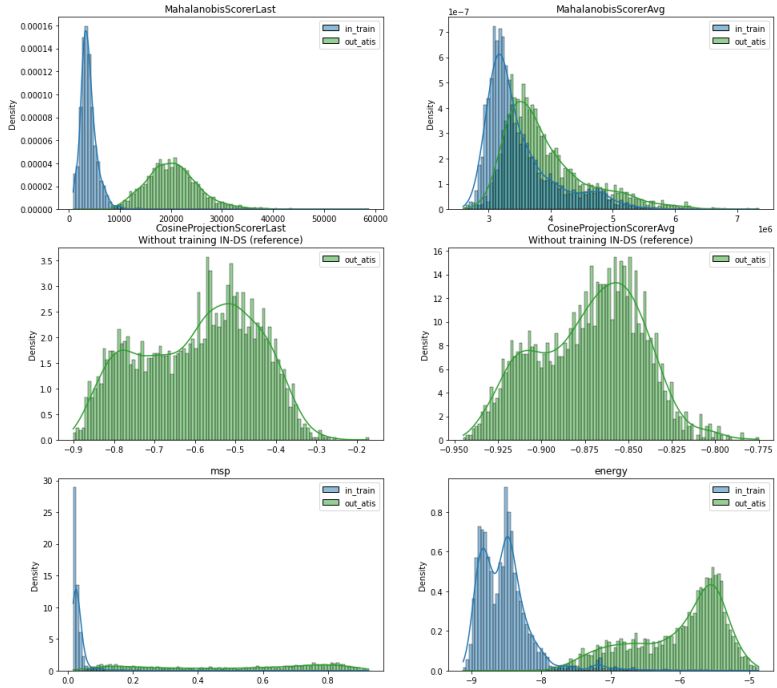

Figure 8: Distribution per scorer for different aggregation methods - ATIS. The plots related to the s_C scoring function only depicts the OUT-DS datasets since $s_C(\mathrm{x}_i) \approx -1 \; \forall(\mathrm{x}_i, y_i) \in \mathcal{D}_N$. The closer the OUT-DS score from -1, the closer to the IN-DS training instances.

A.b Classification metrics

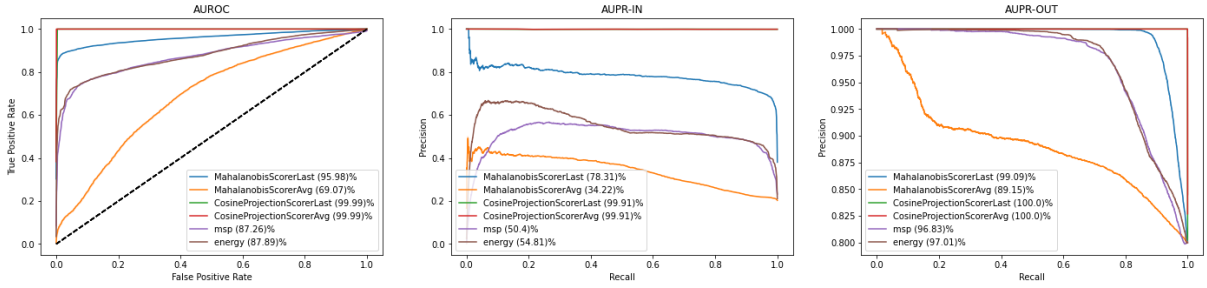

Figure 9: ROC and Precision/Recall curves per scorer. Note the amazing curves obtained for the cosine projection scorer does not necessary mean this method is more effective to detect OOD samples. The shape of the curves is due to the way the scorer is computed.

B `DistilBERT` results

This section emphasizes the results obtained for the detectors fitted on the `DistilBERT`-based intent classifier introduced in 2.3.

B.a Scorer distribution

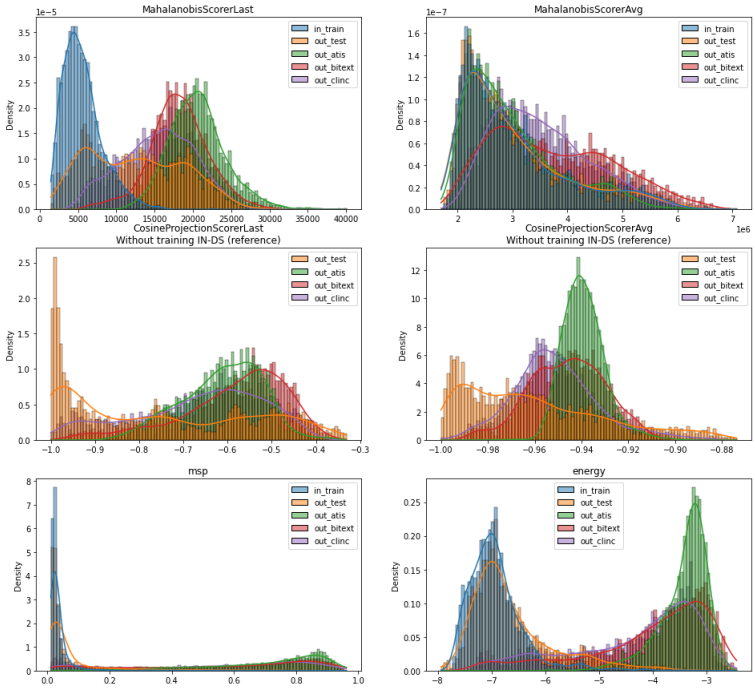

Figure 10: Distribution per scorer for different aggregation methods -All datasets. The plots related to the s_C scoring function only depicts the `OUT-DS` datasets since $s_C(\mathrm{x}_i) \approx -1 \; \forall(\mathrm{x}_i, y_i) \in \mathcal{D}_N$. The closer the `OUT-DS` score from -1, the closer to the `IN-DS` training instances.

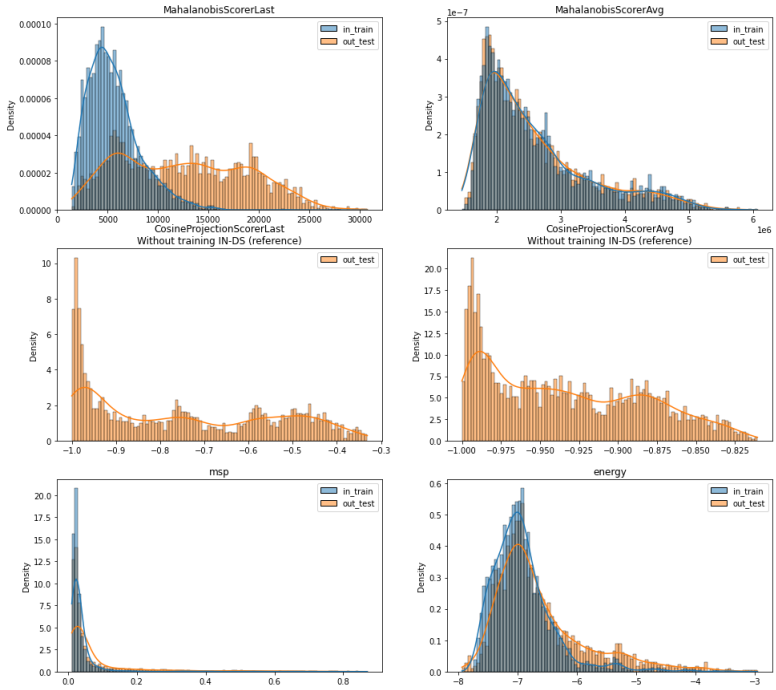

Figure 11: Distribution per scorer for different aggregation methods - BANKING77. The plots related to the s_C scoring function only depicts the OUT-DS datasets since $s_C(\mathrm{x}_i) \approx -1 \; \forall(\mathrm{x}_i, y_i) \in \mathcal{D}_N$. The closer the OUT-DS score from -1, the closer to the IN-DS training instances.

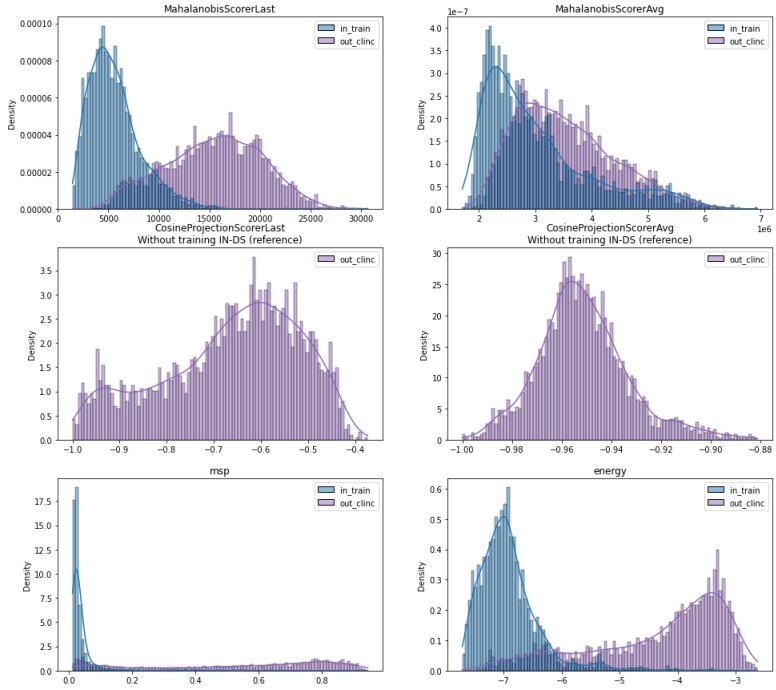

Figure 12: Distribution per scorer for different aggregation methods - CLINC150 The plots related to the s_C scoring function only depicts the OUT-DS datasets since $s_C(\mathrm{x}_i) \approx -1 \; \forall(\mathrm{x}_i, y_i) \in \mathcal{D}_N$. The closer the OUT-DS score from -1, the closer to the IN-DS training instances.

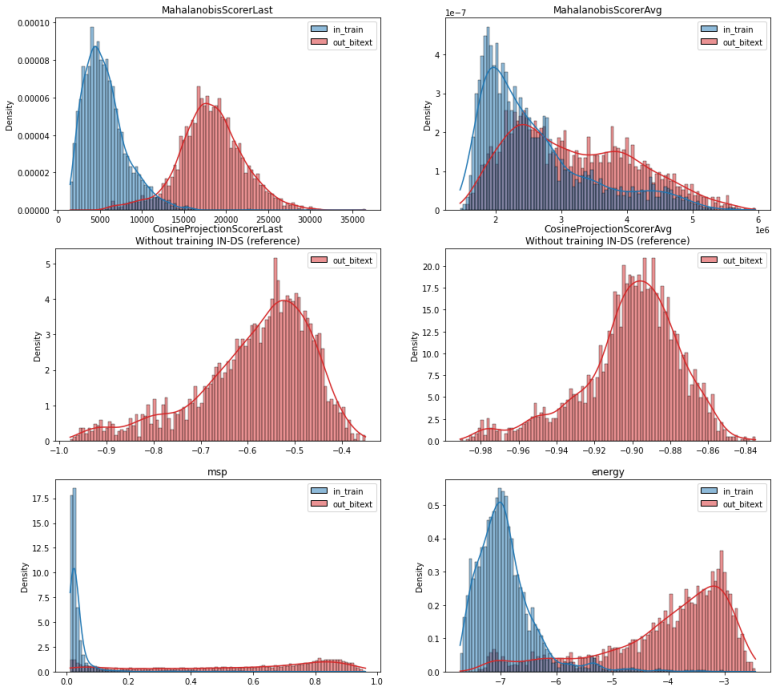

Figure 13: Distribution per scorer for different aggregation methods - BITEXT. The plots related to the s_C scoring function only depicts the OUT-DS datasets since $s_C(\mathrm{x}_i) \approx -1 \; \forall(\mathrm{x}_i, y_i) \in \mathcal{D}_N$. The closer the OUT-DS score from -1, the closer to the IN-DS training instances.

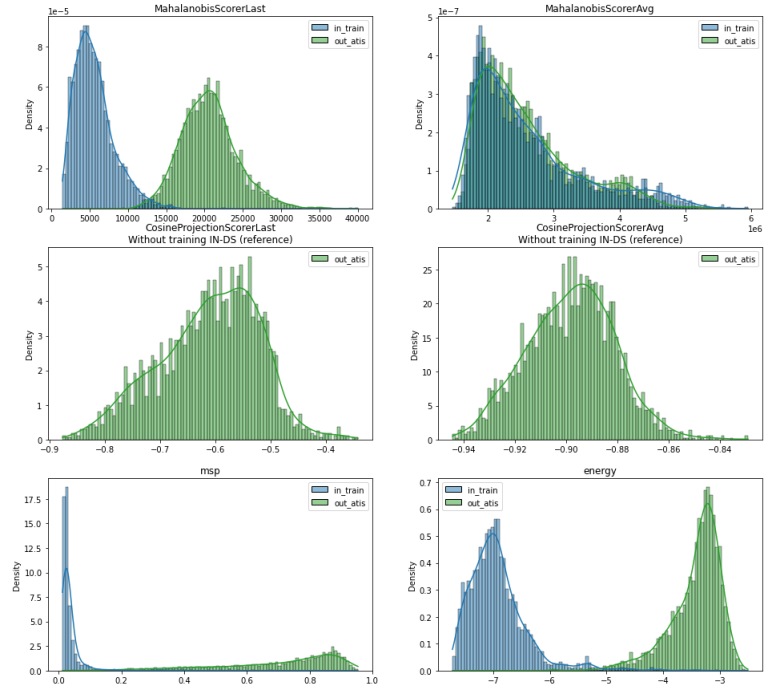

Figure 14: Distribution per scorer for different aggregation methods - ATIS. The plots related to the s_C scoring function only depicts the OUT-DS datasets since $s_C(\mathrm{x}_i) \approx -1 \; \forall(\mathrm{x}_i, y_i) \in \mathcal{D}_N$. The closer the OUT-DS score from -1, the closer to the IN-DS training instances.

B.b Classification metrics

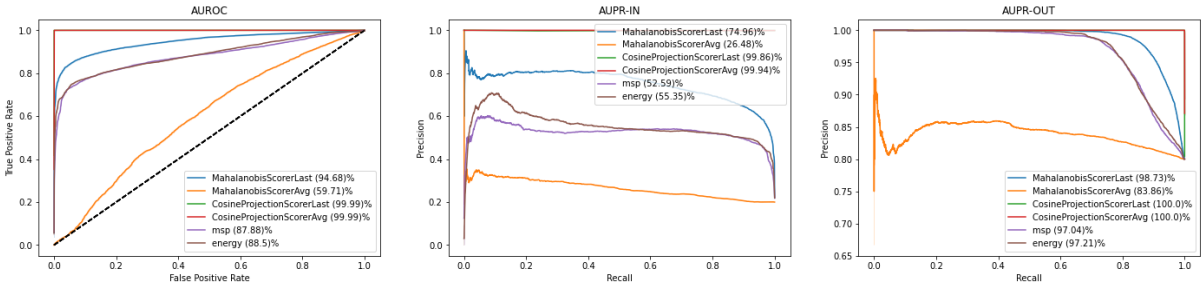

Figure 15: ROC and Precision/Recall curves per scorer. Note the amazing curves obtained for the cosine projection scorer does not necessary mean this method is more effective to detect OOD samples. The shape of the curves is due to the way the scorer is computed.