Reliable AI through Out-of-Distribution Detection: A Benchmark Study

Eliott Barbot
CentraleSupélec
barbot.eliott@gmail.com

José Lucas de Melo Costa
CentraleSupélec
jmcosta@usp.br

Abstract

This article delves into the issue of detecting out-of-distribution (OOD) examples in machine learning models, with a focus on natural language processing (NLP) applications. It is crucial to identify OOD data to build reliable AI, as our models are not trained to handle such examples and may perform poorly on them. To tackle this problem, we implement and compare various OOD detection methods from literature, and evaluate their effectiveness across different datasets, similarity scores between data points, expected distribution, and computational constraints. Furthermore, this article offers open-source code for the implementation of these methods. [1]

1 Introduction

Machine learning models have demonstrated their effectiveness when trained and tested on datasets that are closely related, resulting in high performance in specific tasks [11; 27; 15; 22; 29; 18; 13; 35; 17; 32; 21; 28]. However, these models pose a significant risk of making erroneous predictions when faced with inputs that differ from the training data, such as out-of-distribution (OOD) examples [33; 31; 19; 44; 36; 45; 42; 46; 40; 30]. In some cases, these models may even assign high confidence scores to incorrect predictions, which can have catastrophic outcomes. This has raised concerns about the societal impact of machine learning across various domains [41; 39; 25; 34], including safety-critical applications [16] and privacy and data protection [37].

To address this issue, it is crucial to detect and flag OOD examples, so the human user can recognize when further expertise is needed. The detection of OOD examples is especially important in real-world applications [24], where the data distribution can change over time, leading to drift in the test distribution. Traditionally, evaluation methods have assumed that the train and test datasets are independent and identically distributed, which may not be effective in detecting OOD examples [43; 9; 26; 1; 38; 12; 47; 6] . The limited characterization of evaluation datasets and the drift of the test distribution over time can lead to train-test mismatches, making the detection of OOD examples even more challenging.

Recent research has shown that new evaluation techniques, such as the softmax confidence score[3], can help create OOD detection methods. These methods rely on different evaluation techniques and aim to create reliable AI by detecting OOD examples more accurately. Therefore, it is essential to investigate this issue further, particularly for large black-box models such as BERT [14; 23] or DistilBERT [20], and to develop new tools for Natural Language Processing (NLP) applications that can handle OOD examples reliably.

This study aims to investigate the impact of various parameters on the detection of out-of-distribution instances, specifically in relation to different datasets, similarity scores between data points and the expected distribution, as well as computational constraints. Multiple methods are evaluated and compiled into a benchmark, with the code made publicly available.

1.1 Goals

Different out-of-distribution methods have been developed in the last few years, and in this article some of these methods are compared. As so, this work has the following goals:

1. **Create a benchmark** using the trending methods in out-of-distribution detection
2. **Release open-source code** and data to ease new experiments

[1] https://github.com/jose-melo/nlp-ood-detection

2 Related Works

In recent years, several approaches have been proposed to detect Out-of-Distribution (OOD) samples in neural network classifiers. In particular, the work in [8] proposed a method for OOD detection based on Mahalanobis distance, which uses the covariance matrix of in-domain data to determine the likelihood of an input belonging to the in-domain or out-of-domain category. Their method achieved state-of-the-art results on several benchmark datasets, demonstrating the effectiveness of Mahalanobis distance in OOD detection for NLP tasks. However, these methods often require additional labeled data for calibration and may not perform well in high-dimensional input spaces. In this context, the work proposed by [4], offers a promising alternative to Mahalanobis-based scores. It is an unsupervised OOD detector that leverages information from all hidden layers of a transformer-based neural network to compute a similarity score based on data depth concepts. Unlike Mahalanobis-based scores, their method does not require any additional labeled data and can operate efficiently in high-dimensional input spaces. The experimental results reported by [4] show that this kind of approach consistently outperforms existing OOD detectors, including those based on Mahalanobis-based scores.

Another used metric was proposed in [3]. This work presents a straightforward and effective method based on softmax probabilities. The proposed baseline is tested on various computer vision, natural language processing, and automatic speech recognition tasks and demonstrates its robustness across different architectures and datasets. Furthermore, the authors show that their method can be improved by incorporating an abnormality module to detect more subtle errors and out-of-distribution examples.

3 Background

3.1 Energy-based methods

As presented by [5],a class of energy-based models can be used asa scoring function to detect out-of-distribution data points. The core idea is to find a map $E(x) : \mathbb{R}^d \to \mathbb{R}$ of each point x to a scalar (non-probabilistic). This scalar is called energy, and the mapping is the energy function.It is possible to transform the energy function to a probability density $p(x)$, using:

$$p(y|x) = \frac{e^{-E(x,y)/T}}{Z} \tag{1}$$

where $Z = \int_{y'} e^{-E(x,y')/T}$ is called the partition function and T is a temperature factor. As so, in a classification problem, with our discriminative model $f(x) : \mathbb{R}^D \to \mathbb{R}^K$ produces K logits, then the categorical distribution would be given by the softmax function. In this case, the energy function, that would be further considered as an OOD score, is written as:

$$E(x; f) = -Tlog \sum_{i}^{K} e^{f_i(x)/T} \tag{2}$$

3.2 Affine invariante integrated rank-weighted

In order to address the problem of defining a data depth measure for multivariate data, some methods have been proposed, as the seminal work of [10], based on concepts of half-spaces. In this context, the Affine-Invariant Integrated Rank-Weighted (AI-IRW) statistical depth was proposed as an extension of the original integrated rank-weighted statistic, introduced in [2]. The AI-IRW depth is modified to satisfy the property of affine-invariance.

In discrete setups, the IRW depth can be seen as a weighted average over a finite set of univariate ranks. More formally, for a point $x \in \mathbb{R}^d$ following P_X on $\mathbb{R}^d$, the IRW depth is given by:

$$d_{irw}(x, P_X) = \int_{\mathbb{S}^{d-1}} d_t(u, F_u)du \tag{3}$$

where, $d_t(u, F_u) = min\{F_u(\langle u, x\rangle, 1 - F_u(\langle u, x\rangle)\}$. The efficiency of this approach is that it can be calculated through the expectation by means of Monte-Carlo.

3.3 Mahalanobis distance

The Mahalanobis distance is a metric used to measure the distance between two points in a multivariate space. It takes into account the covariance between the variables, which makes it a useful measure for datasets with correlated variables.

Given two vectors $\mathbf{x}$ and $\mathbf{y}$, the Mahalanobis distance $d_M(\mathbf{x}, \mathbf{y})$ is defined as:

$$d_M(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{y})}, \tag{4}$$

where $\mathbf{S}$ is the covariance matrix of the dataset.

Dataset	Train	Test	Task
imdb	22500	25000	sentiment analysis
sst2	67 349	1821	sentiment analysis
trec	4907	500	multi-class classification
news-summary	1000	20400	summarization
race	87900	49390	multiple choice
yelp_review_full	650000	50000	sentiment analysis
paws	49400	8000	similarity classification

Table 1: Datasets considered in the benchmark

3.4 Autoencoders

Autoencoders (AE) are composed of an encoder and a decoder. Their architecture is designed to reconstruct the input data x from a hidden representation z that will be learned by the network. The encoder is a function that converts the input into the latent representation $z = \mathcal{E}(x)$ while the decoder will be responsible for converting the latent representation into the decoded output of the network $\tilde{x} = \mathcal{D}(z)$. It is the latent space, e.i. the embedded representation of the time series, that will be the focus of this project. The vast majority of work uses the reconstruction error of each subsequence to calculate anomaly scores, on the premise that AE can perform well at reconstructing normal data while failing to reconstruct data with unseen anomalies.

4 Experimental settings

4.1 Data processing

Multiple datasets were evaluated for OOD detection across various NLP tasks (see Table 1). Corpus distribution varies with different NLP tasks, hence the need for multiple tasks. However, limited computational resources resulted in partial use of the datasets.

4.2 Metrics

The two main metric used were the AUC-ROC and the false positive rate. The FPR is defined as $FPR = \frac{FP}{FP+TN}$, where FP is the number of false positives and TN is the number of true negatives. In other words, the FPR measures the fraction of negative instances that are incorrectly classified as positive.

The ROC curve is a plot of the TPR versus the FPR for different classification thresholds. The AUC ROC is the area under the ROC curve, which provides a measure of how well the model is able to distinguish between positive and negative instances. A model with an AUC ROC of 1.0 is perfect, while a model with an AUC ROC of 0.5 is no better than random guessing.

4.3 Models

As presented in [7], pre-trained architecture perform better when fine-tuned to the specific tasks. Given the constraints in calculation, a reduced amount of models were considered. Two main models were used along with the benchmark: the first one was the *BERT-tiny model* that was fine-tuned, and the second one was a *distilbert-uncased* loaded with the weigths of a fine-tuning in the imdb dataset.

4.4 First experience - Training a tiny BERT

To test our OOD detection methods, we used a BERT-tiny model that we finetuned on a classification task. We trained the model on a subpart of the yelp_review_full dataset, which consists of shop reviews associated with an appreciation level. However, as our in-distribution data, we wanted to compare it to a more general text dataset, which is why we used the PAWS dataset as our out-of-distribution data. This decision was made due to limited access to computational resources.

In this method, the model we want to evaluate is used as the encoder part of an encoder-decoder model, and we want to reconstruct the input sentence by using the encoded representation created by our model.

To do so, we use a encoder-decoder model using our classifier model as encoder and bert-tiny as decoder. Then we train only the decoder part of the network on an other part of the $yelp\_review\_full$ dataset (we want to keep the encoder part unchanged since we want to evaluate it).

4.5 Using an autoencoder for dimensionality reduction

In a second experiment, an approach was taken to reduce the dimensionality of the input data and simplify computations. This was done by training an autoencoder on the previously collected embedding data and using it to perform dimensionality reduction. Figure 3 presents the pipeline of the hidden space extraction, mean aggregation and then reduction.

An autoencoder is a type of neural network that is trained to encode and decode input data, such

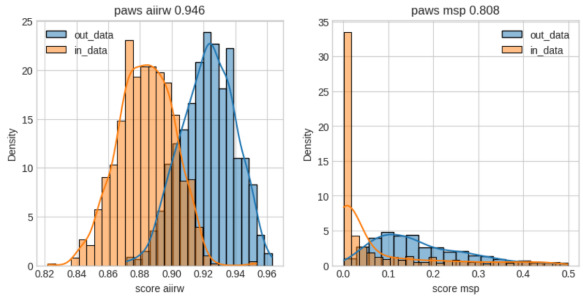

Figure 1: Benchmark: histogram comparing the AI-IRW and the MSP methods. The IN_DS in training was the sst2 and the OUT_DS was the paws dataset

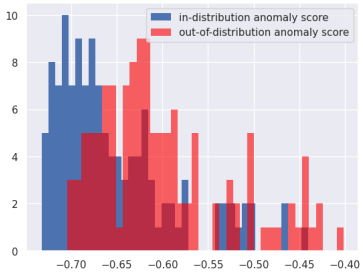

Figure 2: Fine-tuned Tiny-BERT: computed anomaly scores

that the reconstructed output is as close as possible to the original input. By training an autoencoder on the embedding data, it learned to represent the data in a more compact form while retaining important features. Using the trained autoencoder, the input data was then reduced to a lower-dimensional space, making it easier and faster to process.

5 Results

5.1 Benchmarks results

Table 2 shows the results of a benchmark study using a *distilbert* model pre-trained on the *imdb* dataset. The model was applied to four different datasets, marked as out-distribution: *trec*, *race*, *yelp*, and *paws*. The performance is presented in terms of ROC AUC. One example of the computed histogram is presented in Figure 1. The table also shows the results obtained when an autoencoder trained on the same *imdb* dataset was used to reduce the dimensionality of the data.

It should be noted that the performance of the methods varies depending on the task to which they are applied. For example, the *yelp* dataset, which are designed for semantic analysis along with the imdb dataset, yielded the worst results. Additionally, only in the *trec* dataset did the IRW fail to outperform the mahalanobis distance as the best metric.

Finally, it was observed that the use of the autoencoder did not result in a significant loss of performance. This suggests that it may be possible to improve the representation of the layer embeddings before applying the anomaly analysis, potentially increasing the efficiency of these methods.

5.2 Tiny-Bert results

As we can see in Figure 2, when we use the opposite of softmax probability as an anomaly score, the scores are clearly different between in-distribution and out-of distribution data.

And Figure 4 shows the ROC curve of detectors based on this score. The area under the ROC curve is 0.756.

When we tryed to applied this method, we realized that training the decoder is in fact very very long. So we decided to train it on a limited part of the training dataset, to see if it was enough to see the eficiency of the method.

And when we tested our encoder-decoder's ability to reconstruct the input data, for in-distribution data we got a rouge precision of 0.00279, which is not a lot but is clearly more than the 0.000625 rouge precision that we got for the out-of-distribution data.

6 Conclusion

In conclusion, this study aimed to investigate the detection of out-of-distribution (OOD) examples and their impact on machine learning models. Recent research has shown that new evaluation techniques, such as the softmax confidence score, can help create OOD detection methods. Therefore, this study aimed to create a benchmark using trending methods in OOD detection and release open-source code and data to ease new experiments. The benchmark results showed that the IRW method performed better than the others. Additionally, the model's performance dependes on the datasets, suggesting that different parameters, need to be considered according to the task.

References

[1] Marine Picot, Nathan Noiry, Pablo Piantanida, and Pierre Colombo. . Adversarial attack detection un-

	trec				race				yelp				paws			
	aiirw	msp	energy	maha	aiirw	msp	energy	maha	aiirw	msp	energy	maha	aiirw	msp	energy	maha
original	0.79	0.85	0.84	**0.87**	**0.87**	0.72	0.71	0.86	**0.69**	0.62	0.61	0.62	**0.94**	0.81	0.79	0.91
autoencoder	0.81	0.85	0.85	0.81	0.88	0.72	0.72	0.86	0.62	0.62	0.61	0.63	0.94	0.81	0.80	0.93

Table 2: Results for the benchmark. The model used was a *distilbert* pretrained on the imbd dataset. As so, the dataset considered as training in-distribution was the *sst2* and the test out-distribution are each one of the columns. The metric presented is the AUC ROC. The second line indicates the results when the data was educed by an autoencoder also trained in the imdb dataset.

der realistic constraints.

[2] Guillaume Staerman, Pavlo Mozharovskyi, and Stéphan Clémençon. Affine-invariant integrated rank-weighted depth: Definition, properties and finite sample analysis. Version: 1.

[3] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks.

[4] Pierre Colombo, Eduardo D. C. Gomes, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida. Beyond mahalanobis-based scores for textual OOD detection.

[5] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, pages 21464–21475. Curran Associates, Inc.

[6] Marine Picot, Federica Granese, Guillaume Staerman, Marco Romanelli, Francisco Messina, Pablo Piantanida, and Pierre Colombo. . A halfspace-mass depth-based method for adversarial attack detection. *Transactions on Machine Learning Research*.

[7] Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness.

[8] Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. Revisiting mahalanobis distance for transformer-based out-of-domain detection. 35(15):13675–13682. Number: 15.

[9] Marine Picot, Guillaume Staerman, Federica Granese, Nathan Noiry, Francisco Messina, Pablo Piantanida, and Pierre Colombo. . A simple unsupervised data depth-based method to detect adversarial images.

[10] TUKEY J. W. 1975. Mathematics and the picturing of data. *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, 2:523–531.

[11] M.I. Jordan and T.M. Mitchell. 2015. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255 – 260. Cited by: 3393.

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

[13] Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–51, Berlin, Germany. Association for Computational Linguistics.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[15] Wojciech Witon, Pierre Colombo, Ashutosh Modi, and Mubbasir Kapadia. 2018. Disney at IEST 2018: Predicting emotions using an ensemble. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 248–253, Brussels, Belgium. Association for Computational Linguistics.

[16] José Faria. 2018. Machine learning safety: An overview. *Safety-critical Systems Symposium 2018 (SSS'18)*.

[17] Xianda Zhou and William Yang Wang. 2018. MojiTalk: Generating emotional responses at scale. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1128–1137, Melbourne, Australia. Association for Computational Linguistics.

[18] Pierre Colombo, Wojciech Witon, Ashutosh Modi, James Kennedy, and Mubbasir Kapadia. 2019. Affect-driven dialog generation. 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 3734–3743, Minneapolis, Minnesota. Association for Computational Linguistics.

[19] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

[20] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled ver-

sion of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

[21] Wei Wei, Jiayi Liu, Xianling Mao, Guibing Guo, Feida Zhu, Pan Zhou, and Yuchong Hu. 2019. Emotion-aware chat machine: Automatic emotional response generation for human-like emotional interaction. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1401–1410.

[22] Yazhou Zhang, Qiuchi Li, Dawei Song, Peng Zhang, and Panpan Wang. 2019. Quantum-inspired interactive networks for conversational sentiment analysis.

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[24] Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K Varshney, and Dawn Song. 2020. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347.

[25] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. 2020. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*, pages 1779–1788. PMLR.

[26] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475.

[27] Pierre Colombo, Emile Chapuis, Matteo Manica, Emmanuel Vignon, Giovanna Varni, and Chloé Clavel. 2020. Guiding attention in sequence-to-sequence models for dialogue act prediction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7594–7601. AAAI Press.

[28] Hamid Jalalzai, Pierre Colombo, Chloé Clavel, Eric Gaussier, Giovanna Varni, Emmanuel Vignon, and Anne Sabourin. 2020. Heavy-tailed representations, text polarity classification & data augmentation. In *Advances in Neural Information Processing Systems*, volume 33, pages 4295–4307. Curran Associates, Inc.

[29] Kai Wang, Junfeng Tian, Rui Wang, Xiaojun Quan, and Jianxing Yu. 2020. Multi-domain dialogue acts and response co-generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7125–7134, Online. Association for Computational Linguistics.

[30] Sarik Ghazarian, Ralph Weischedel, Aram Galstyan, and Nanyun Peng. 2020. Predictive engagement: An efficient metric for automatic evaluation of open-domain dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7789–7796.

[31] Pierre Colombo, Guillaume Staerman, Chloé Clavel, and Pablo Piantanida. 2021. Automatic text evaluation through the lens of Wasserstein barycenters. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10450–10466, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

[32] Pierre Colombo, Chloé Clavel, Chouchang Yack, and Giovanna Varni. 2021. Beam search with bidirectional strategies for neural response generation. In *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 139–146, Trento, Italy. Association for Computational Linguistics.

[33] Yi-Ting Yeh, Maxine Eskenazi, and Shikib Mehri. 2021. A comprehensive assessment of dialog evaluation metrics. *arXiv preprint arXiv:2106.03706*.

[34] Pierre Colombo. 2021. *Learning to represent and generate text using information measures*. Ph.D. thesis, Ph. D. thesis, Institut polytechnique de Paris.

[35] Pierre Colombo, Pablo Piantanida, and Chloé Clavel. 2021. A novel estimator of mutual information for learning to disentangle textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6539–6550, Online. Association for Computational Linguistics.

[36] Guillaume Staerman, Pavlo Mozharovskyi, Pierre Colombo, Stéphan Clémençon, and Florence d'Alché Buc. 2021. A pseudo-metric between probability distributions based on depth-trimmed regions. *arXiv preprint arXiv:2103.12711*.

[37] Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. 2021. When machine learning meets privacy: A survey and outlook. *ACM Comput. Surv.*, 54(2).

[38] Pierre Colombo, Eduardo Dadalto, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida. 2022. Beyond mahalanobis distance for textual ood detection. In *Advances in Neural Information Processing Systems*, volume 35, pages 17744–17759. Curran Associates, Inc.

[39] Georg Pichler, Pierre Jean A. Colombo, Malik Boudiaf, Günther Koliander, and Pablo Piantanida. 2022. A differential entropy estimator for training neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume

162 of *Proceedings of Machine Learning Research*, pages 17691–17715. PMLR.

[40] Pierre Jean A. Colombo, Chloé Clavel, and Pablo Piantanida. 2022. Infolm: A new metric to evaluate summarization amp; data2text generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10554–10562.

[41] Pierre Colombo, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida. 2022. Learning disentangled textual representations via statistical measures of similarity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2614–2630, Dublin, Ireland. Association for Computational Linguistics.

[42] Cyril Chhun, Pierre Colombo, Fabian M. Suchanek, and Chloé Clavel. 2022. Of human criteria and automatic metrics: A benchmark of the evaluation of story generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5794–5836, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

[43] Maxime Darrin, Pablo Piantanida, and Pierre Colombo. 2022. Rainproof: An umbrella to shield text generators from out-of-distribution data. *arXiv preprint arXiv:2212.09171*.

[44] Pierre Colombo, Nathan Noiry, Ekhine Irurozki, and Stephan Clémençon. 2022. What are the best systems? new perspectives on nlp benchmarking. In *Advances in Neural Information Processing Systems*, volume 35, pages 26915–26932. Curran Associates, Inc.

[45] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

[46] Jiaan Wang, Yunlong Liang, Fandong Meng, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.

[47] Maxime Darrin, Guillaume Staerman, Eduardo Dadalto Câmara Gomes, Jackie CK Cheung, Pablo Piantanida, and Pierre Colombo. 2023. Unsupervised layer-wise score aggregation for textual ood detection. *arXiv preprint arXiv:2302.09852*.

A Extra images

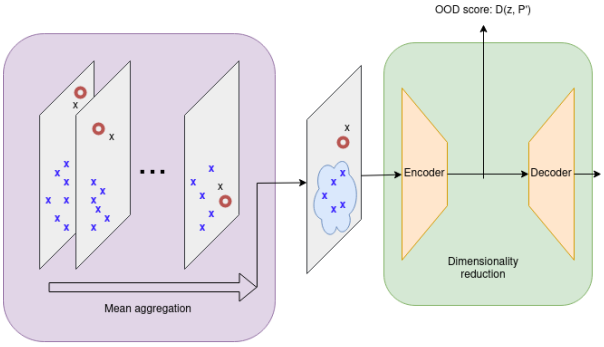

Figure 3: Pipeline with the autoencoder

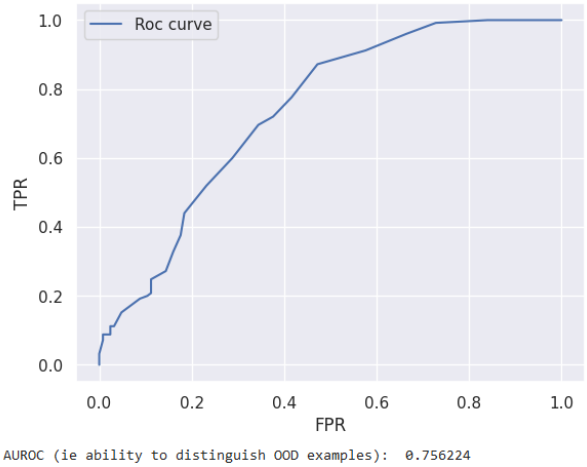

Figure 4: Fine-tuned Tiny-BERT: the ROC curve of detectors

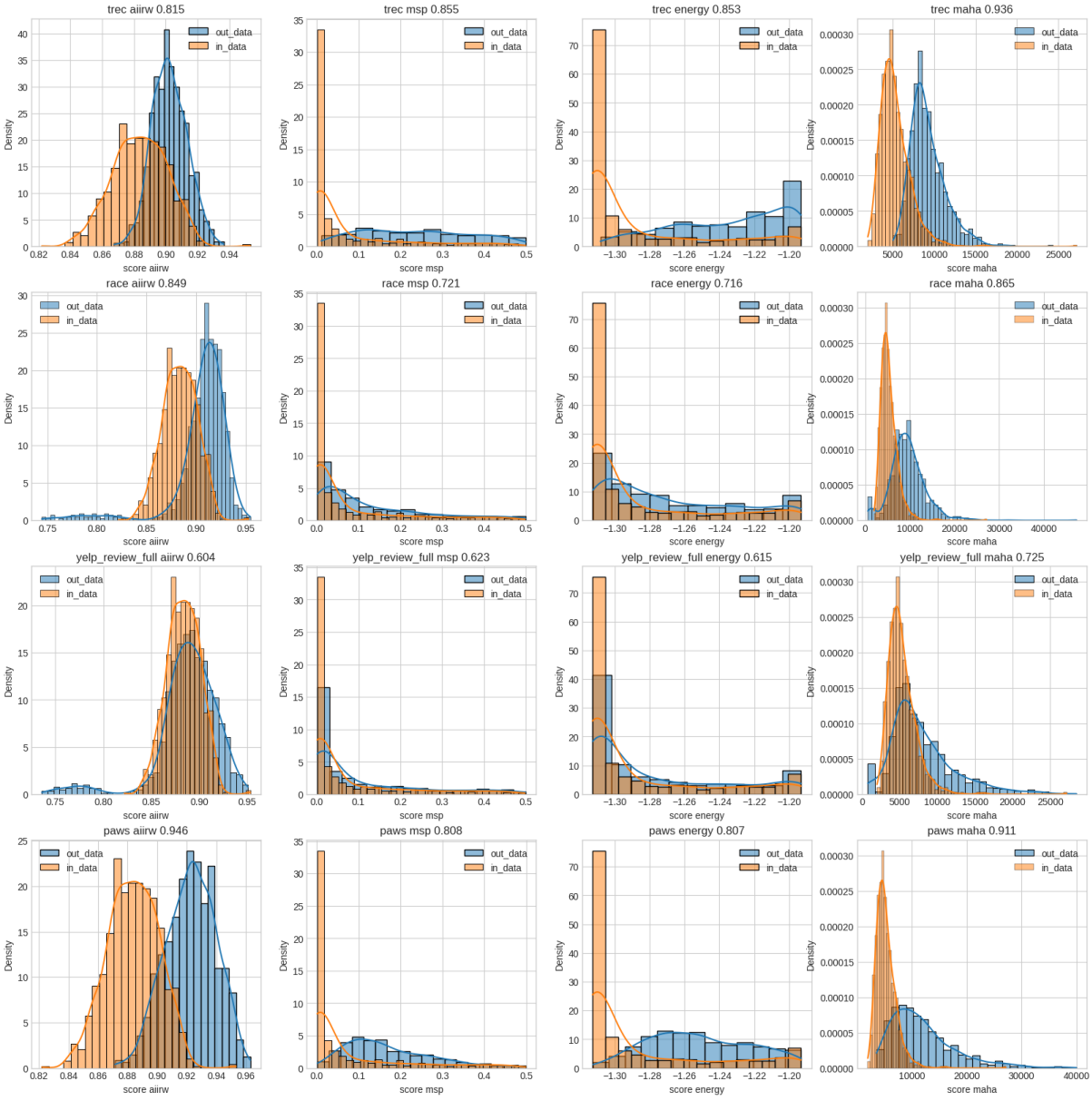

Figure 5: Benchmark: histograms of the experiments.

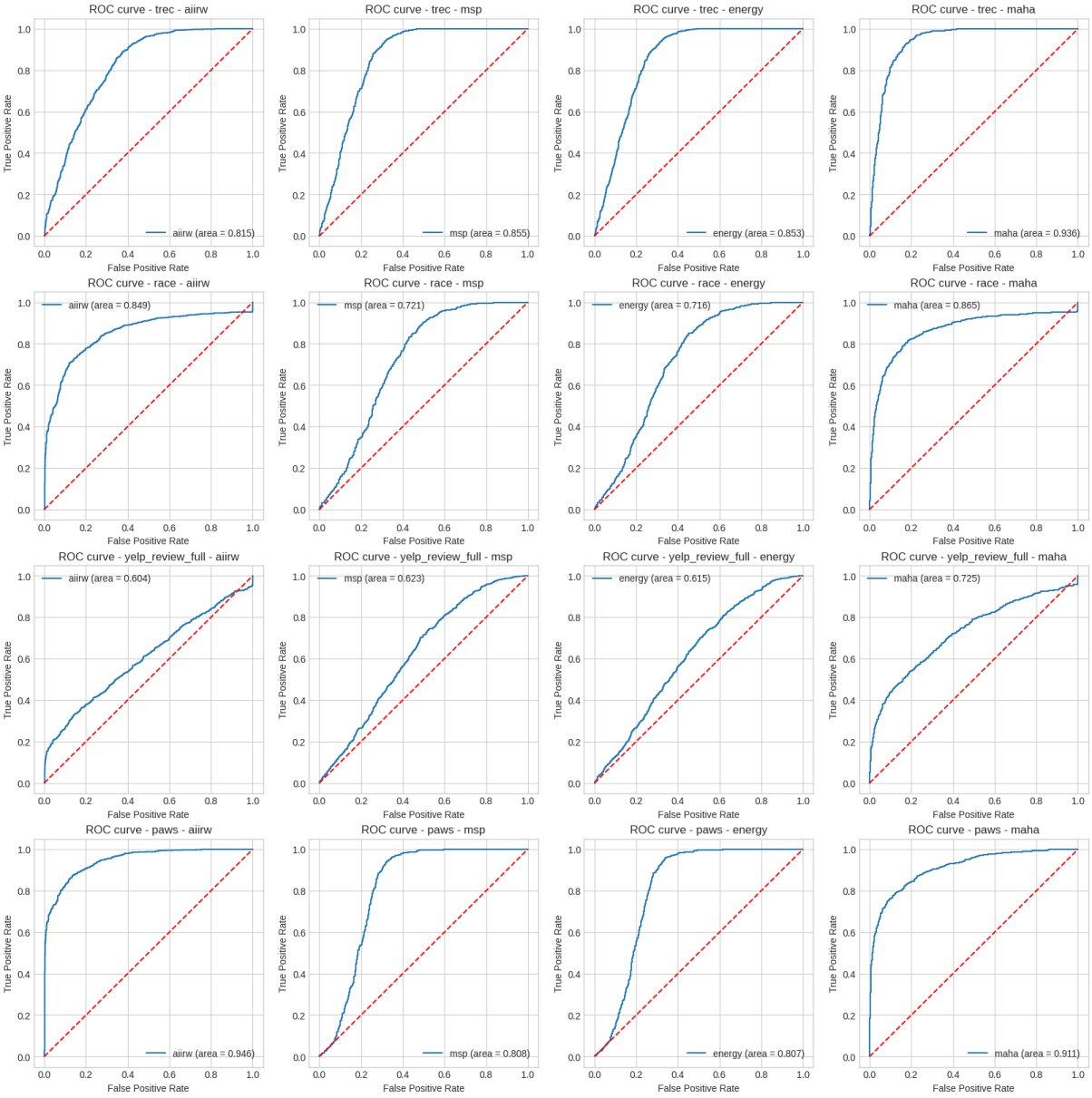

Figure 6: Benchmark: roc curves of the experiments.