NLP-ADBench: NLP Anomaly Detection Benchmark

Yuangang Li[1,*], Jiaqi Li[1,*], Zhuo Xiao[1,*], Tiankai Yang[1], Yi Nian[1], Xiyang Hu[2†], Yue Zhao[1†]
[1]University of Southern California [2]Arizona State University
{yuangang, jli77629, zhuoxiao, tiankaiy, yinian, yzhao010}@usc.edu,
xiyanghu@asu.edu
[†]Corresponding authors

Abstract

Anomaly detection (AD) is an important machine learning task with applications in fraud detection, content moderation, and user behavior analysis. However, AD is relatively understudied in a natural language processing (NLP) context, limiting its effectiveness in detecting harmful content, phishing attempts, and spam reviews. We introduce NLP-ADBench, the most comprehensive NLP anomaly detection (NLP-AD) benchmark to date, which includes eight curated datasets and 19 state-of-the-art algorithms. These span 3 end-to-end methods and 16 two-step approaches that adapt classical, non-AD methods to language embeddings from BERT and OpenAI. Our empirical results show that no single model dominates across all datasets, indicating a need for automated model selection. Moreover, two-step methods with transformer-based embeddings consistently outperform specialized end-to-end approaches, with OpenAI embeddings outperforming those of BERT. We release NLP-ADBench at https://github.com/USC-FORTIS/NLP-ADBench , providing a unified framework for NLP-AD and supporting future investigations.

1 Introduction

Anomaly detection (AD) is a fundamental area in machine learning with diverse applications in web systems, such as fraud detection, content moderation, and user behavior analysis (Chandola et al., 2009; Ahmed et al., 2016). Substantial progress has been achieved in AD for structured data such as tabular, graph, and time series (Chalapathy and Chawla, 2019; Han et al., 2022; Lai et al., 2021; Liu et al., 2022), but its extension to natural language processing (NLP) remains relatively underexplored (Ruff et al., 2021; Yang et al., 2024). This gap limits our ability to identify harmful content, phishing attempts, and spam reviews.

For instance, detecting abusive or threatening language is crucial for ensuring that social media platforms and online forums remain safe environments for users (Fortuna and Nunes, 2018). Likewise, detecting anomalous product reviews or descriptions in e-commerce is important for preserving user trust and platform credibility (Chino et al., 2017). However, many standard AD methods are designed for *numeric* or *categorical* data and are not easily adapted to unstructured text (Zhao et al., 2019; Chen et al., 2024). Existing studies on NLP-specific AD are limited in both dataset variety and algorithmic range (Han et al., 2022; Liu et al., 2022; Yang et al., 2024), leaving open questions about which approaches work best under different conditions. These gaps lead to a central research question: *How can we systematically evaluate and compare diverse AD methods across real-world text datasets, and what insights can be gained to guide future development in NLP-based AD?*

Our Proposal and Key Contributions. We introduce NLP-ADBench, the most comprehensive benchmark for NLP-AD tasks. NLP-ADBench offers four major benefits compared to prior work (Bejan et al., 2023): (***i***) eight real-world datasets covering a wide range of web use cases; (***ii***) 19 advanced methods that apply standard AD algorithms to language embeddings or use end-to-end neural architectures; (***iii***) detailed empirical findings that highlight new directions for NLP-AD; and (***iv***) fully open-source resources, including datasets, algorithm implementations, and more, aligns with the *Resources and Evaluation* track.

Key Insights/Takeaways (see details in §3). Our comprehensive experiments reveal: (***i***) No single model dominates across all datasets, showing the need for model selection; (***ii***) Transformer-based embeddings substantially boost two-step AD methods (e.g., LUNAR (Goodge et al., 2022) and LOF (Breunig et al., 2000)) relative to end-to-end approaches; (***iii***) High-dimensional embeddings (e.g., from OpenAI) improve detection performance, but

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 2464–2474
November 4-9, 2025

also raise computational overhead; and (***iv***) Dataset-specific biases and human-centered anomaly definitions remain challenging for building robust and widely applicable NLP-AD systems.

2 NLP-ADBench: AD Benchmark for NLP Tasks

2.1 Preliminaries and Problem Definition

Anomaly Detection in Natural Language Processing (**NLP-AD**) focuses on identifying text instances that deviate significantly from expected or typical patterns. Unlike structured data, text data is inherently unstructured, high-dimensional, and deeply influenced by the nuances of human language, including syntax, semantics, and context (Aggarwal, 2017; Yang et al., 2024). These unique properties introduce significant challenges, making the development of robust and accurate AD methods for NLP a complex and demanding task.

Formally, let $\mathcal{D} = x_1, x_2, \ldots, x_N$ denote a corpus where each x_i is a text instance. The goal of NLP-AD is to learn an anomaly scoring function $f : \mathcal{X} \to \mathbb{R}$ that assigns a real-valued anomaly score to each text instance. Higher scores denote greater deviations from normal patterns, indicating a higher likelihood of an anomalous instance.

2.2 Curated Benchmark Datasets

The limited availability of purpose-built datasets constrains the development and evaluation of effective methods in NLP-AD. To address this gap, we **curated** and **transformed** 8 existing classification datasets from various NLP domains into specialized datasets tailored for NLP-AD tasks, ensuring that all data are presented in a standard format. These datasets, collectively called the NLPAD datasets, provide a foundational resource for advancing research.

Each transformed dataset is named by adding the prefix "NLPAD-" to the original dataset's name (e.g., NLPAD-AGNews, NLPAD-BBCNews), distinguishing them from the original datasets. The NLPAD datasets are provided in a unified JSON Lines format for compatibility and ease of use. Each line is a JSON object with four fields: text (the text used for anomaly detection), label (the anomaly detection label, where 1 represents an anomaly and 0 represents normal), original_task (the task of the original dataset), and original_label (the category label from the original dataset).

To transform each dataset for NLP-AD, we established a text selection process based on the data format. For tabular data, we carefully chose appropriate columns as the text source. For document-based data, we extracted text directly from relevant documents. The anomalous class for each dataset was selected based on *semantic distinctions* within the dataset categories, ensuring that the identified anomalies represent meaningful deviations from the normal data distribution (Emmott et al., 2015; Han et al., 2022). Once identified, the anomalous class was downsampled to represent less than 10% of the total instances.

Table 1: Statistical information of the NLPAD dataset.

NLPAD Dataset	# Samples	#Normal	#Anomaly	%Anomaly
NLPAD-AGNews	98,207	94,427	3,780	3.85%
NLPAD-BBCNews	1,785	1,723	62	3.47%
NLPAD-EmailSpam	3,578	3,432	146	4.08%
NLPAD-Emotion	361,980	350,166	11,814	3.26%
NLPAD-MovieReview	26,369	24,882	1,487	5.64%
NLPAD-N24News	59,822	57,994	1,828	3.06%
NLPAD-SMSSpam	4,672	4,518	154	3.30%
NLPAD-YelpReview	316,924	298,986	17,938	5.66%

For instance, in the NLPAD-AGNews dataset (tabular data), we selected the "description" column as the text source, with the "World" category serving as the anomalous class due to its semantic divergence from other categories such as "Sports" or "Technology." This anomalous class was then downsampled accordingly. Similarly, in the NLPAD-BBCNews dataset (document-based data), text from BBC News documents was used, with the "entertainment" category identified as anomalous because its semantic content significantly differs from other categories like "Politics" or "Business." The "entertainment" category was also downsampled to maintain consistency. This semantic-driven approach to defining anomalies was consistently applied across all datasets. Further details of dataset sources and construction processes can be found in Appx. A.1.1. Table. 1 presents the statistical information of the NLPAD datasets, including the total number of samples, the number of normal and anomalous samples, and the anomaly ratio for each dataset.

2.3 The Most Comprehensive NLP-AD Algorithms with Open Implementations

Compared to the existing NLP-AD benchmark by Bejan et al. (Bejan et al., 2023), NLP-ADBench provides a broader evaluation by including 19 algorithms, categorized into two groups. The first group comprises 3 end-to-end algorithms that directly process raw text data to produce anomaly detection outcomes. The second group consists of 16 algorithms derived by applying 8 traditional anomaly detection (AD) methods to text embeddings generated

from two models: *bert-base-uncased* (Devlin et al., 2019) and OpenAI's *text-embedding-3-large* (OpenAI, 2024). These traditional AD methods do not operate on raw text directly but instead perform anomaly detection on embeddings, offering a complementary approach to the end-to-end methods. This comprehensive algorithm collection enables a robust evaluation of direct and embedding-based NLP anomaly detection techniques. Here, we provide a brief description; see details in Appx. A.2.

End-to-end NLP-AD Algorithms. We evaluate 3 end-to-end algorithms tailored for NLP-AD. (1) **Context Vector Data Description (CVDD)** (Ruff et al., 2019) leverages context vectors and pre-trained embeddings with a multi-head self-attention mechanism to project normal instances close to learned contexts, identifying anomalies based on deviations. (2) **Detecting Anomalies in Text via Self-Supervision of Transformers (DATE)** (Manolache et al., 2021) trains transformers using self-supervised tasks like replaced mask detection to capture normal text patterns and flag anomalies. (3) **Few-shot Anomaly Detection in Text with Deviation Learning (FATE)** (Das et al., 2023) uses a few labeled anomalies with deviation learning to distinguish anomalies from normal instances. We adapt it to train solely on normal data, referring to the adapted version as FATE*.

Two-step NLP-AD Algorithms. We evaluate 8 two-step algorithms that rely on embeddings generated by models such as *bert-base-uncased* (Devlin et al., 2019) and *text-embedding-3-large* (OpenAI, 2024). These algorithms are designed to work with structured numerical data and cannot directly process raw textual data, requiring text transformation into numerical embeddings. (4) **LOF** (Breunig et al., 2000) measures local density deviations, while (5) **DeepSVDD** (Ruff et al., 2018) minimizes the volume of a hypersphere enclosing normal representations. (6) **ECOD** (Li et al., 2022) uses empirical cumulative distribution functions to estimate densities and assumes anomalies lie in distribution tails. (7) **IForest** (Liu et al., 2008) recursively isolates anomalies through random splits, and (8) **SO_GAAL** (Liu et al., 2019) generates adversarial samples to identify anomalies. Reconstruction-based approaches include (9) **AE** (Aggarwal, 2017), which flags anomalies based on reconstruction errors, and (10) **VAE** (Kingma and Welling, 2013; Burgess et al., 2018), which identifies anomalies using reconstruction probabilities or latent deviations. Finally, (11) **LUNAR** (Goodge et al., 2022) enhances traditional local outlier detection with graph neural networks.

3 Experiment Results

3.1 Experiment Setting

Datasets, Train/Test Data Split, and Independent Trials. In the NLP-ADBench benchmark, the data is divided by allocating 70% of the normal data to the training set. The remaining 30% of normal data, combined with all anomalous data, forms the test set. To ensure the robustness of our findings, we repeat each experiment three times and report the average performance.

Hyperparameter Settings. For all the algorithms in NLP-ADBench, we use their default hyperparameter (HP) settings in the original paper for a fair comparison, same as ADBench (Han et al., 2022).

Evaluation Metrics and Statistical Tests. We evaluate different NLP-AD methods by a widely used metric: AUROC (Area Under Receiver Operating Characteristic Curve) and AUPRC (Area Under Precision-Recall Curve) value.

Embeddings Definitions:

1. **BERT** refers specifically to the *bert-base-uncased model* (Devlin et al., 2019).
2. **OpenAI** refers to OpenAI's *text-embedding-3-large* model (OpenAI, 2024).
3. The term "BERT + AD algorithm" or "OpenAI + AD algorithm" means that we first generate text embeddings using BERT or OpenAI's model, respectively, and then apply the AD algorithm.

3.2 Results, Discussions, and New Directions

We analyze the AUROC results presented in Table 2 and the average rank summary in Figure 1. For completeness, AUPRC scores and their corresponding average ranks are reported in Appendix A.3.

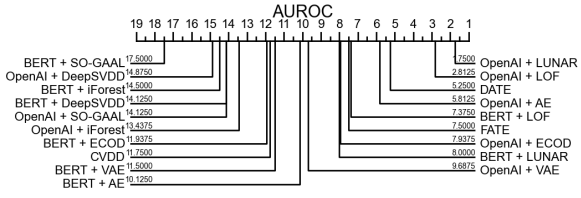

Figure 1: Average rank on AUROC of 19 NLPAD methods across 8 datasets (the lower the better).

No single model consistently excels across all datasets due to variability in dataset characteristics. AD model performance varies significantly across datasets, complicating the selection of a universally optimal model. For datasets with more categories (e.g., NLPAD-AGNews), two-step methods like OpenAI + LUNAR (**0.9226**) outperform end-to-end methods such as CVDD (**0.6046**) by

Table 2: Performance comparison of 19 Algorithms on 8 NLPAD datasets using AUROC, with **best** results highlighted in **bold and shaded**.

Methods	NLPAD-AGNews	NLPAD-BBCNews	NLPAD-EmailSpam	NLPAD-Emotion	NLPAD-MovieReview	NLPAD-N24News	NLPAD-SMSSpam	NLPAD-YelpReview
CVDD	0.6046	0.7221	0.9340	0.4867	0.4895	0.7507	0.4782	0.5345
DATE	0.8120	0.9030	**0.9697**	0.6291	0.5185	0.7493	**0.9398**	0.6092
FATE*	0.7756	0.9310	0.9061	0.5035	0.5289	0.8073	0.6262	0.5945
BERT + LOF	0.7432	0.9320	0.7482	0.5435	0.4959	0.6703	0.7190	0.6573
BERT + DeepSVDD	0.6671	0.5683	0.6937	0.5142	0.4287	0.4366	0.5859	0.5871
BERT + ECOD	0.6318	0.6912	0.7052	0.5889	0.4282	0.4969	0.5606	0.6326
BERT + iForest	0.6124	0.6847	0.6779	0.4944	0.4420	0.4724	0.5053	0.5971
BERT + SO-GAAL	0.4489	0.3099	0.4440	0.5031	0.4663	0.4135	0.3328	0.4712
BERT + AE	0.7200	0.8839	0.4739	0.5594	0.4650	0.5749	0.6918	0.6441
BERT + VAE	0.6773	0.7409	0.4737	0.5594	0.4398	0.4949	0.6082	0.6441
BERT + LUNAR	0.7694	0.9260	0.8417	0.5186	0.4687	0.6284	0.6953	0.6522
OpenAI + LOF	0.8905	0.9558	0.9263	0.7304	0.6156	0.7806	0.7862	0.8733
OpenAI + DeepSVDD	0.4680	0.5766	0.4415	0.4816	0.6563	0.6150	0.3491	0.5373
OpenAI + ECOD	0.7638	0.7224	0.9263	0.6206	**0.7366**	0.7342	0.4317	0.5984
OpenAI + iForest	0.5213	0.6064	0.6937	0.5889	0.5064	0.4944	0.3751	0.5871
OpenAI + SO-GAAL	0.5945	0.2359	0.4440	0.5031	0.6201	0.5043	0.5671	0.5082
OpenAI + AE	0.8326	0.9520	0.7651	0.7067	0.6088	0.7155	0.5511	0.8524
OpenAI + VAE	0.8144	0.7250	0.5273	0.7067	0.4515	0.7418	0.4259	0.6163
OpenAI + LUNAR	**0.9226**	**0.9732**	0.9343	**0.9328**	0.6474	**0.8320**	0.7189	**0.9452**

52.6%. Similarly, on NLPAD-BBCNews, OpenAI + LOF (**0.9558**) surpasses CVDD (**0.7221**) by **32.4%**. Conversely, on the binary-class datasets (e.g., NLPAD-SMSSpam), end-to-end methods perform better, with DATE (**0.9398**) clearly exceeding OpenAI + LUNAR (**0.7189**) by **30.7%**.

- *Future Direction 1: Automated Model Selection.* These results emphasize the importance of developing automated approaches to select the most suitable model. One feasible solution will be adapting the meta-learning framework from tabular AD settings (Zhao et al., 2021) to NLP-AD.

Transformer-based embeddings boost the performance of two-step AD methods. Two-step AD algorithms paired with transformer-based embeddings have consistently outperformed end-to-end methods in NLP-AD tasks. For instance, OpenAI + LUNAR achieves **0.9226** on NLPAD-AGNews, surpassing CVDD by **52.6%** and FATE* by **19.0%**. Similarly, OpenAI + LOF reaches **0.9558** on NLPAD-BBCNews, exceeding CVDD by **32.4%** and FATE* by **2.7%**. This advantage arises primarily because two-step methods leverage superior contextual embeddings from modern transformer models (e.g., OpenAI), whereas end-to-end methods like CVDD rely on older embeddings (e.g., GloVe). This highlights the need for end-to-end methods to adopt more advanced embeddings to enhance performance.

- *Future Direction 2: Transformer Embedding Integration for End-to-End AD.* Future end-to-end methods should adopt transformer-based embeddings over static embeddings like GloVe. Research should focus on embedding integration optimized for end-to-end AD frameworks.

High-dimensional embeddings enhance detection but require balancing performance and efficiency. Embedding dimensionality significantly impacts both performance and computational efficiency in AD tasks. Compared to BERT-base embeddings (768 dimensions), OpenAI's *text-embedding-3-large* embeddings (3072 dimensions, a **300% increase**) consistently achieve superior results across multiple datasets in NLP-ADBench. Specifically, OpenAI + LUNAR achieves **0.9452** on NLPAD-YelpReview (outperforming BERT + LUNAR's **0.6522** by **44.9%**), **0.9226** on NLPAD-AGNews (exceeding BERT + LUNAR's **0.7694** by **19.9%**), and **0.8320** on NLPAD-N24News (surpassing BERT + LUNAR's **0.6284** by **32.4%**). These results clearly demonstrate the advantage of higher-dimensional embeddings for enhancing AD performance. However, higher dimensionality also introduces greater computational costs and potential information redundancy.

- *Future Direction 3: Optimizing Embedding Dimensionality.* Future research should explore NLP-AD-specific dimensionality reduction techniques to reduce redundancy and computational costs without compromising performance. Additionally, adaptive methods that dynamically adjust dimensionality based on dataset characteristics could enhance scalability and efficiency.

3.3 In-depth Analysis of Key Findings

3.3.1 Explaining Method-Dataset Fit

To understand why certain models outperform others on specific datasets, we conduct both quanti-

tative and qualitative analyses to identify dataset-level factors influencing model performance.

Quantitative Corpus-Level Linguistic Analysis We characterize each dataset using three linguistic indicators: (1) **Avg-Len**, the average number of BERT tokens per sample (text complexity); (2) **Lexical-Burstiness**, the proportion of top-20 anomaly-specific tokens from BERT-tokenized TF-IDF (higher values indicate stronger lexical anomaly signals); (3) **Topic-Diversity**, the Shannon entropy over the label distribution of normal-class examples (0 for binary datasets; higher values indicate broader topic coverage).

Table 3: Dataset characteristics and best method

Dataset	Avg-Len	Lexical-Burst.	Topic-Div.	Best Method
NLPAD-AGNews	39.5	0.040	1.585	OpenAI+LUNAR (two-step)
NLPAD-BBCNews	481.7	0.028	1.981	OpenAI+LUNAR (two-step)
NLPAD-EmailSpam	238.4	0.024	0.000	DATE (end-to-end)
NLPAD-Emotion	20.3	0.087	1.949	OpenAI+LUNAR (two-step)
NLPAD-MovieReview	290.3	0.026	0.000	OpenAI+ECOD (two-step)
NLPAD-N24News	1034.2	0.016	4.437	OpenAI+LUNAR (two-step)
NLPAD-SMSSpam	20.5	0.063	0.000	DATE (end-to-end)
NLPAD-YelpReview	151.6	0.023	0.000	OpenAI+LUNAR (two-step)

From Table. 3, we get two findings: (1) Datasets with short texts, high lexical-burstiness, and explicit lexical markers (e.g., NLPAD-SMSSpam) tend to favor DATE's token-level anomaly detection. (2) Datasets with moderate length, lower lexical-burstiness, and high topical diversity (e.g., NLPAD-AGNews) benefit from richer, context-sensitive embeddings (OpenAI+LUNAR).

Qualitative Analysis of Representative Cases To further investigate the model performance gap, we qualitatively examined anomaly examples from *NLPAD-SMSSpam* and *NLPAD-AGNews* (see Table 4). In *NLPAD-SMSSpam*, anomalies often contain explicit lexical irregularities, such as numeric tokens, unconventional formatting, or urgency-inducing phrases. These surface-level features align closely with DATE's self-supervised scoring mechanism, which is sensitive to token-level deviations. By contrast, anomalies in *NLPAD-AGNews* exhibit subtle semantic shifts without distinctive lexical markers. Detecting such anomalies requires a deeper understanding of contextual semantics, which exceeds DATE's capacity and favors two-step methods with embedding-based models such as OpenAI and LUNAR.

3.3.2 Performance-Efficiency Trade-offs in Embedding Dimensionality

While OpenAI-based two-step methods achieve high anomaly detection performance, their high dimensionality raises concerns about computational and financial cost in deployment scenarios. To explore whether such overhead is justified, we conduct dimensionality reduction experiments using PCA, projecting OpenAI embeddings to 768 dimensions—the same as BERT.

Table 4: Representative anomaly examples from NLPAD-SMSSpam and NLPAD-AGNews

NLPAD-SMSSpam	**NLPAD-AGNews**
PRIVATE! Your 2003 Account Statement for shows 800 un redeemed S. I. M. points. Call 08715203694 Identifier Code: 40533 Expires 31/10/04	NEW YORK U.S. stocks are expected to open modestly higher Tuesday as investors use a slight let up in the rise in oil prices to add to portfolios...
FREE for 1st week! No1 Nokia tone 4 ur mob every week just txt NOKIA to 87077 Get txting and tell ur mates. zed POBox 36504 W45WQ norm150p/tone 16	
Urgent! Please call 09061213237 from landline. 5000 cash or a luxury 4 Canary Islands Holiday await collection. T Cs SAE PO Box 177. M227XY. 150ppm. 16	AFP Indian shares, Asia's second top performers last year, are poised for long term gains as foreign investors buy into the market, seeing the country as an economic "growth story," according to analysts.
XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here xxxmobile-movieclub.com?n QJKGIGHJJGCBL	

Table 5: Performance-efficiency trade-offs of dimensionality reduction on datasets

Method	Embedding Dim	Performance		Runtime (s/sample)			
		AUROC	AUPRC	Total	Embedding	PCA	Inference
NLPAD-AGNews							
OpenAI + LUNAR (orig.)	3072	**0.907**	**0.562**	0.377	0.267	0.000	0.110
OpenAI + LUNAR (PCA)	768	0.890	0.546	0.409	0.252	0.007	0.150
BERT + LUNAR	768	0.790	0.325	**0.081**	0.056	0.000	0.025
OpenAI + LOF (orig.)	3072	**0.896**	**0.575**	0.314	0.239	0.000	0.075
OpenAI + LOF (PCA)	768	0.798	0.321	0.435	0.245	0.007	0.183
BERT + LOF	768	0.771	0.304	**0.076**	0.062	0.000	0.014
NLPAD-MovieReview							
OpenAI + LUNAR (orig.)	3072	0.664	0.238	0.322	0.270	0.000	0.052
OpenAI + LUNAR (PCA)	768	**0.681**	**0.249**	0.409	0.270	0.007	0.132
BERT + LUNAR	768	0.467	0.152	**0.071**	0.060	0.000	0.010
OpenAI + LOF (orig.)	3072	**0.652**	**0.242**	0.314	0.274	0.000	0.041
OpenAI + LOF (PCA)	768	0.624	0.226	0.384	0.256	0.007	0.121
BERT + LOF	768	0.498	0.166	**0.068**	0.060	0.000	0.009

As shown in Table 5, PCA-reduced OpenAI embeddings slightly change performance (e.g., AUROC drops from 0.907 to 0.890 on NLPAD-AGNews with LUNAR) and consistently outperform BERT at the same dimension. However, PCA increases total runtime because it compresses embeddings into a denser space, which complicates decision boundaries and slows down inference.

This finding reinforces *Future Direction 3*, highlighting the need for NLP-AD-specific dimensionality reduction techniques that balance representation quality with computational efficiency.

4 Conclusion

We present NLP-ADBench, the most comprehensive benchmark for contextual NLP anomaly detection (NLP-AD), evaluating 19 state-of-the-art algorithms across 8 diverse datasets. Our findings establish the superiority of two-step methods leveraging transformer-based embeddings, such as OpenAI + LUNAR, over end-to-end approaches, demonstrating the power of hybrid strategies for handling complex NLP anomaly detection tasks. By combining advanced text embeddings with traditional anomaly detection methods, NLP-ADBench provides a robust and flexible framework that sets a new standard for evaluating NLP-AD systems. Additionally, we offer actionable insights into model performance, dataset variability, and embedding utilization, paving the way for future research.

Limitations

Despite its contributions, NLP-ADBench has certain limitations. First, the datasets included in the benchmark, while diverse, are primarily sourced from existing classification tasks and may not fully reflect emerging challenges such as anomalies in multilingual or multimodal text data. Second, our evaluations focus on static embeddings, leaving dynamic or streaming NLP-AD scenarios unexplored. Third, the reliance on predefined anomaly labels in our benchmark limits the ability to assess unsupervised or domain-adaptive approaches. Future work can expand NLP-ADBench to include more diverse datasets, such as multilingual or multimodal data, and by exploring dynamic anomaly detection in streaming text scenarios. Incorporating benchmarks for unsupervised and adaptive models can also better reflect real-world applications. These advancements will enhance NLP-ADBench's utility as a comprehensive platform for driving progress in NLP anomaly detection.

Ethics Statement

This work adheres to ethical standards emphasizing transparency, fairness, and privacy in NLP anomaly detection research. By openly sharing datasets, algorithms, and experimental results, NLP-ADBench provides a standardized foundation for advancing safer and more reliable web-based systems. All datasets are publicly available and contain no personally identifiable information, ensuring privacy compliance. Pre-trained embeddings (such as OpenAI's text-embedding-3-large) are used in accordance with their terms of service. Additionally, we used ChatGPT exclusively to improve minor grammar in the final manuscript text.

Broader Impacts

The NLP-ADBench proposed in this paper provides a comprehensive benchmark framework for anomaly detection in NLP. By standardizing datasets and algorithms, this work supports advancements in critical web-based applications, including fraud detection, spam filtering, and content moderation. The benchmark promotes transparency, reproducibility, and facilitates further innovations, ultimately contributing to safer, more reliable online environments.

Acknowledgments

This work was partially supported by the National Science Foundation under Award Nos. 2428039, 2346158, and 2449280. We also acknowledge the use of computational resources provided by the Advanced Cyberinfrastructure Coordination Ecosystem (Boerner et al., 2023): Services & Support (ACCESS) program, supported by NSF grants #2138259, #2138286, #2138307, #2137603, and #2138296. Specifically, this work used NCSA Delta GPU at the National Center for Supercomputing Applications (NCSA) through allocation CIS250073. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors also gratefully acknowledge support from the Amazon Research Awards and Capital One Research Awards.

References

Charu C. Aggarwal. 2017. *Outlier Analysis*, 2nd edition. Springer.

Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. 2016. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.*, 60:19–31.

Tiago A. Almeida, José María G. Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: new collection and results. In *ACM Symposium on Document Engineering*, page 259–262.

Matei Bejan, Andrei Manolache, and Marius Popescu. 2023. Ad-nlp: A benchmark for anomaly detection in natural language processing. In *EMNLP*, pages 10766–10778.

{Timothy J.} Boerner, Stephen Deems, {Thomas R.} Furlani, {Shelley L.} Knuth, and John Towns. 2023. Access: Advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support. In *PEARC 2023 - Computing for the common good*, PEARC 2023 - Computing for the common good: Practice and Experience in Advanced Research Computing, pages 173–176, United States. Association for Computing Machinery. Publisher Copyright: © 2023 Owner/Author.; 2023 Practice and Experience in Advanced Research Computing, PEARC 2023 ; Conference date: 23-07-2023 Through 27-07-2023.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. Lof: identifying density-based local outliers. In *SIGMOD*, pages 93–104.

Christopher P Burgess, Irina Higgins, Loic Matthey Pal, and Alexander Lerchner. 2018. Understanding disentangling in β-vae. *arXiv:1804.03599*.

Christine P. Chai. 2022. Comparison of text preprocessing methods. *Nat. Lang. Eng.*, 29:509–553.

Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv:1901.03407*.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *CSUR*, 41(3):1–58.

Sihan Chen, Zhuangzhuang Qian, Wingchun Siu, Xingcan Hu, Jiaqi Li, Shawn Li, Yuehan Qin, Tiankai Yang, Zhuo Xiao, Wanghao Ye, and others. 2024. PyOD 2: A Python Library for Outlier Detection with LLM-powered Model Selection. In *International World Wide Web Conference (TheWebConf Demo Track)*.

Daniel YT Chino, Alceu F Costa, Agma JM Traina, and Christos Faloutsos. 2017. Voltime: Unsupervised anomaly detection on users' online activity volume. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 108–116. SIAM.

Anindya Sundar Das, Aravind Ajay, Sriparna Saha, and Monowar Bhuyan. 2023. Few-shot anomaly detection in text with deviation learning. *arXiv:2308.11780*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Andrew Emmott, Shubhomoy Das, Thomas Dietterich, and 1 others. 2015. A meta-analysis of the anomaly detection problem. *arXiv:1503.01158*.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *CSUR*, 51(4):1–30.

Adam Goodge, Bryan Hooi, See-Kiong Ng, and Wee Siong Ng. 2022. Lunar: Unifying local outlier detection methods via graph neural networks. In *AAAI*, volume 36, pages 6737–6745.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *ICML*, pages 377–384.

Songqiao Han, Xiyang Hu, and 1 others. 2022. Adbench: Anomaly detection benchmark. *NeurIPS*, 35:32142–32159.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv:1312.6114*.

Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, and 1 others. 2021. Revisiting time series outlier detection: Definitions and benchmarks. In *NeurIPS*.

Zheng Li, Yue Zhao, and 1 others. 2022. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *TKDE*, 35(12):12181–12193.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *ICDM*, pages 413–422.

Kay Liu, Yingtong Dou, Yue Zhao, and 1 others. 2022. Bond: Benchmarking unsupervised outlier node detection on static attributed graphs. *NeurIPS*, 35:27021–27035.

Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He. 2019. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150.

Andrei Manolache, Florin Brad, and Elena Burceanu. 2021. Date: Detecting anomalies in text via self-supervision of transformers. *arXiv:2104.05591*.

Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. 2006. Spam filtering with naive bayes-which naive bayes? In *CEAS*, volume 17, pages 28–69.

OpenAI. 2024. New embedding models and api updates.

Ilham Fadillah Putra. 2023. Yelp review dataset. https://www.kaggle.com/datasets/ilhamfp31/yelp-review-dataset. Accessed: 2024-11-28.

Aman Anand Rai. 2023. Ag news classification dataset. https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset. Accessed: 2024-11-28.

Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Klaus-Robert Müller, and Geoff Orr. 2021. A unifying review of deep and shallow anomaly detection. *Proc. IEEE*, 109(5):756–795.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, and 1 others. 2018. Deep one-class classification. In *ICML*, pages 4393–4402. PMLR.

Lukas Ruff, Yury Zemlyanskiy, Robert Vandermeulen, and 1 others. 2019. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In *ACL*, pages 4061–4071.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *EMNLP*, pages 3687–3697.

Zhen Wang, Xu Shan, Xiangxie Zhang, and Jie Yang. 2022. N24News: A new dataset for multimodal news classification. In *LREC*, pages 6768–6775.

Tiankai Yang, Yi Nian, Shawn Li, Ruiyao Xu, Yuangang Li, Jiaqi Li, Zhuo Xiao, Xiyang Hu, Ryan Rossi, Kaize Ding, and 1 others. 2024. Ad-llm: Benchmarking large language models for anomaly detection. *arXiv preprint arXiv:2412.11142*.

Yue Zhao, Zain Nasrullah, and Zheng Li. 2019. PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research (JMLR)*, 20:1–7.

Yue Zhao, Ryan Rossi, and Leman Akoglu. 2021. Automatic unsupervised outlier model selection. *NeurIPS*, 34:4489–4502.

Supplementary Material for NLP-ADBench

A Details on NLP-ADBench

A.1 Additional Details on Benchmark Datasets

A.1.1 Datasets Sources.

1. **NLPAD-AGNews** is constructed from the AG News dataset (Rai, 2023), which was originally intended for news topic classification tasks. The AG News dataset contains 127,600 samples categorized into four classes: World, Sports, Business, and Sci/Tech. We selected the text from the "description" column as NLPAD-AGNews's text data source. The "World" category was designated as the anomaly class and was downsampled accordingly.
2. **NLPAD-BBCNews** is constructed from the BBC News dataset (Greene and Cunningham, 2006), which was originally used for document classification across various news topics. The BBC News dataset includes 2,225 articles divided into five categories: Business, Entertainment, Politics, Sport, and Tech. We selected the full text of the news articles as NLPAD-BBC News's text data source. The "Entertainment" category was designated as the anomaly class and was downsampled accordingly.
3. **NLPAD-EmailSpam** is constructed from the Spam Emails dataset (Metsis et al., 2006), originally used for email spam detection. The Spam Emails dataset contains 5,171 emails labeled as either spam or ham (not spam). We selected the text from the "text" column containing the email bodies as NLPAD-Emails Spam's text data source. The "spam" category was designated as the anomaly class and was downsampled accordingly.
4. **NLPAD-Emotion:** is constructed from the Emotion dataset (Saravia et al., 2018) , which was originally intended for emotion classification tasks in textual data. The Emotion dataset contains 416,809 text samples labeled with six emotions: anger, fear, joy, love, sadness, and surprise. We selected the text from the "text" column as NLPAD-Emotion's text data source. The "fear" category was designated as the anomaly class and was downsampled accordingly.
5. **NLPAD-MovieReview:** is constructed from the Movie Review dataset (Maas et al., 2011) , commonly used for sentiment analysis of film critiques. The Movie Review dataset includes 50,000 reviews labeled as positive or negative. We selected the full review texts as NLPAD-MovieReview's text data source. The "neg" (negative reviews) category was designated as the anomaly class and was downsampled accordingly.
6. **NLPAD-N24News** is constructed from the N24News dataset (Wang et al., 2022), originally used for topic classification of news articles. N24News contains 61,235 articles across various categories. We selected the full text of the news articles as NLPAD-N24News's text data source. The "food" category was designated as the anomaly class and was downsampled accordingly.
7. **NLPAD-SMSSpam** is constructed from the SMS Spam Collection dataset (Almeida et al., 2011), originally intended for classifying SMS messages as spam or ham (not spam). The SMS Spam Collection dataset comprises 5,574 messages labeled accordingly. We selected the text from the "message text" as NLPAD-SMS Spam's text data source. The "spam" category was designated as the anomaly class and was downsampled accordingly.
8. **NLPAD-YelpReview** is constructed from the Yelp Review Polarity dataset (Putra, 2023), originally intended for sentiment classification tasks. The Yelp Review Polarity dataset is created by considering 1-star and 2-star ratings as negative, and 3-star and 4-star ratings as positive. For each polarity, 280,000 training samples and 19,000 testing samples were randomly selected, resulting in a total of 560,000 training samples and 38,000 testing samples. Negative polarity is labeled as class 1, and positive polarity as class 2. We selected the text from the text column as NLPAD-YelpReview's text data source. The label 1 (negative reviews) was designated as the anomaly class and was downsampled accordingly.

A.1.2 NLPAD dataset's text pre-processing

On all 8 datasets, we preprocessed the raw text data to ensure consistency and usability by removing URLs and HTML tags, eliminating unnecessary special characters while retaining essential punctuation, converting line breaks and consecutive spaces into single spaces, and preserving case sensitivity and stop words to maintain linguistic integrity. After processing the text, we found that some texts

became duplicates due to the removal of certain symbols. Consequently, we removed all duplicate data to ensure the uniqueness of each text sample. These preprocessing steps follow established practices to effectively clean text data while retaining its syntactic and semantic features, providing a reliable foundation for natural language processing tasks (Chai, 2022).

A.2 Additional Details on Algorithms

A.2.1 End-to-End Algorithms

1. **Context Vector Data Description** (Ruff et al., 2019)(CVDD) is an unsupervised anomaly detection method for textual data. It utilizes pre-trained word embeddings and a multi-head self-attention mechanism to learn "context vectors" that represent normal patterns in the data. Anomalies are detected by measuring the cosine distance between sequence projections and context vectors, where larger distances indicate higher anomaly likelihoods. CVDD also penalizes overlapping contexts to enhance interpretability.
2. **Detecting Anomalies in Text via Self-Supervision of Transformers (DATE)** (Manolache et al., 2021) detects anomalies in text by training self-supervised transformers on tasks like replaced mask detection, enabling the model to learn normal language patterns and identify deviations.
3. **Few-shot Anomaly Detection in Text with Deviation Learning (FATE)** (Das et al., 2023) is a deep learning framework that uses a small number of labeled anomalies to learn anomaly scores end-to-end. By employing deviation learning, it ensures normal examples align with reference scores while anomalies deviate significantly. Utilizing multi-head self-attention and multiple instance learning, FATE achieves state-of-the-art performance on benchmark datasets. However, as our approach focuses on unsupervised anomaly detection, we adapt FATE into **FATE*** by training exclusively on normal data. This adaptation involves modifying the framework to learn reference scores and deviations without access to labeled anomalies, enabling effective detection of anomalous examples in an entirely unsupervised setting.

A.2.2 Traditional Algorithms

1. **Local Outlier Factor (LOF)** (Breunig et al., 2000) calculates the local density deviation of a data point relative to its neighbors. This metric identifies points that have substantially lower density than their neighbors, marking them as outliers.
2. **Deep Support Vector Data Description (DeepSVDD)** (Ruff et al., 2018) minimizes the volume of a hypersphere enclosing the data representations learned by a neural network, capturing common patterns while identifying anomalies as points outside the hypersphere.
3. **Empirical-Cumulative-distribution-based Outlier Detection (ECOD)** (Li et al., 2022) estimates the empirical cumulative distribution function (ECDF) for each feature independently. It identifies outliers as data points that reside in the tails of these distributions. This approach is hyperparameter-free and offers straightforward interpretability.
4. **Isolation Forest (IForest)** (Liu et al., 2008) detects anomalies by isolating observations through random feature selection and splitting, with anomalies requiring fewer splits
5. **Single-Objective Generative Adversarial Active Learning (SO_GAAL)** (Liu et al., 2019) optimizes a single objective function to generate adversarial samples and effectively identify anomalies in unsupervised settings.
6. **AutoEncoder (AE)** (Aggarwal, 2017) detects anomalies by reconstructing input data, where higher reconstruction errors signify potential anomalies.
7. **Unifying Local Outlier Detection Methods via Graph Neural Networks(LUNAR)** (Goodge et al., 2022) uses graph neural networks to integrate and enhance traditional local outlier detection methods, unifying them for better anomaly detection.
8. **Variational AutoEncoder (VAE)** (Kingma and Welling, 2013; Burgess et al., 2018) uses probabilistic latent variables to model data distributions, identifying anomalies based on reconstruction probabilities or latent space deviations.

A.3 More Experiment Results

We also report AUPRC scores (Table. A1) for all 19 algorithms across the 8 NLPAD datasets, along with their average AUPRC ranks (Fig. A1), to provide a complementary evaluation perspective beyond AUROC.

Table A1: Performance comparison of 19 Algorithms on 8 NLPAD datasets using AUPRC, with **best** results highlighted in **bold and shaded**.

Methods	NLPAD-AGNews	NLPAD-BBCNews	NLPAD-EmailSpam	NLPAD-Emotion	NLPAD-MovieReview	NLPAD-N24News	NLPAD-SMSSpam	NLPAD-YelpReview
CVDD	0.1296	0.2976	0.5353	0.0955	0.1576	0.2886	0.0712	0.1711
DATE	0.3996	0.5764	**0.8885**	0.1619	0.1682	0.2794	**0.6112**	0.2149
FATE*	0.2787	0.5805	0.5529	0.1026	0.1752	0.2777	0.1257	0.2112
BERT + LOF	0.2549	0.6029	0.2370	0.1170	0.1621	0.1678	0.1837	0.2629
BERT + DeepSVDD	0.2160	0.1328	0.2117	0.0986	0.1387	0.0798	0.1178	0.2174
BERT + ECOD	0.1616	0.2037	0.2077	0.1024	0.1374	0.0928	0.1156	0.2197
BERT + iForest	0.1559	0.2131	0.1894	0.1007	0.1412	0.0872	0.0994	0.2203
BERT + SO-GAAL	0.1033	0.0849	0.1130	0.1036	0.1486	0.0837	0.0714	0.2440
BERT + AE	0.2232	0.4274	0.2937	0.1037	0.1479	0.1255	0.1914	0.2525
BERT + VAE	0.1878	0.2559	0.2247	0.1019	0.1405	0.0957	0.1360	0.2331
BERT + LUNAR	0.2717	0.5943	0.3571	0.1053	0.1497	0.1436	0.1817	0.2609
OpenAI + LOF	0.5443	0.7714	0.5967	0.2290	0.2133	0.2248	0.2450	0.5710
OpenAI + DeepSVDD	0.1062	0.1288	0.1195	0.1040	0.3278	0.1297	0.0721	0.1893
OpenAI + ECOD	0.3294	0.2424	0.5597	0.7443	**0.5165**	0.2238	0.0821	**0.8639**
OpenAI + iForest	0.1278	0.1376	0.3283	0.1311	0.1724	0.0913	0.0772	0.2527
OpenAI + SO-GAAL	0.1538	0.0665	0.1096	0.1291	0.3005	0.0963	0.1213	0.2735
OpenAI + AE	0.4022	0.7485	0.5580	**0.8355**	0.1969	0.1984	0.1030	0.7063
OpenAI + VAE	0.3659	0.2424	0.5604	0.7744	0.1486	0.2537	0.0812	0.8467
OpenAI + LUNAR	**0.6918**	**0.8653**	0.5810	0.3112	0.2193	**0.4425**	0.1640	0.4524

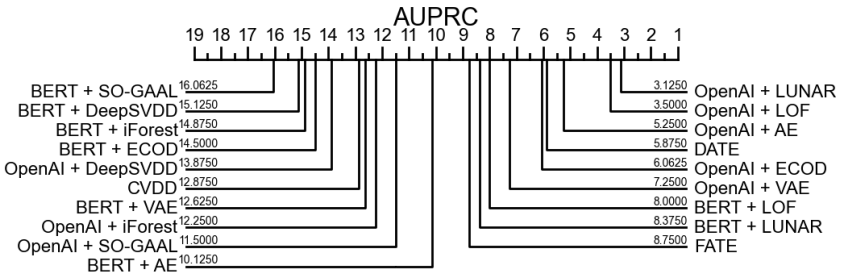

Figure A1: Average rank on AUPRC of 19 NLPAD methods across 8 datasets (the lower the better).