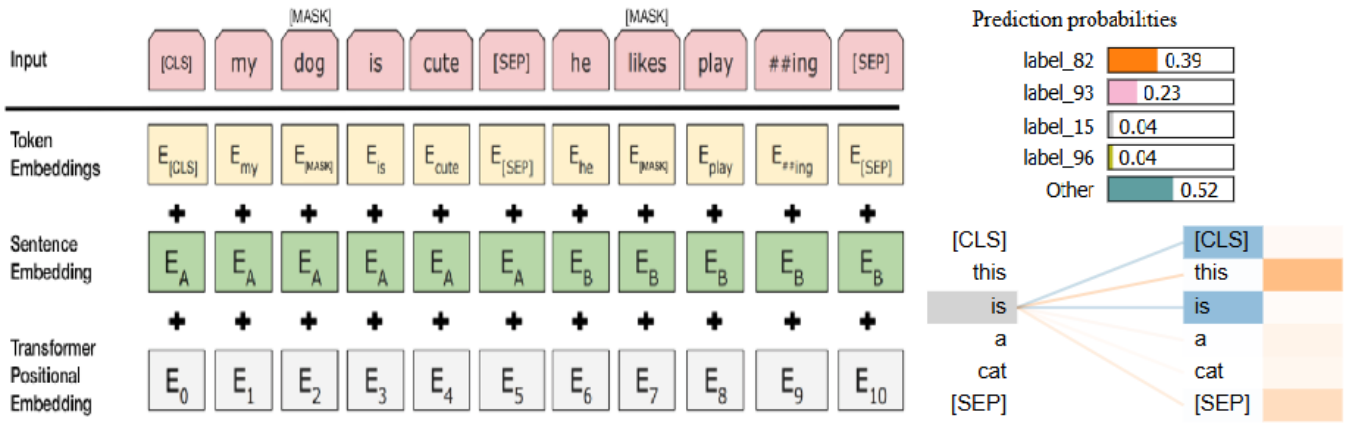
Improving Interpretability/Explainability and Robustness of X-Transformer

Shruti Chaudhary
Indian Institute of Technology, Jodhpur
b22ai037@iitj.ac.in

Vindhya Jain
Indian Institute of Technology, Jodhpur
b22ai060@iitj.ac.in



Abstract

Extreme Multi-Label Classification (XML) presents significant challenges in machine learning, particularly when addressing dependability aspects such as model interpretability and adversarial robustness. This paper presents an enhanced version of the X-Transformer algorithm that integrates explainability techniques and defense mechanisms against adversarial attacks while maintaining competitive performance on standard metrics. We evaluate our approach on three benchmark datasets (BibTeX, MediaMill, and Delicious) from the XML repository, demonstrating that our modifications achieve interpretability and robust performance under adversarial conditions. Our comprehensive experiments include adversarial training using HopSkipJump and ZOO attacks, with precision-recall metrics showing that the proposed method maintains accuracy within 8% of the baseline while improving model safety. The results suggest that our integrated approach successfully balances performance with critical dependability requirements for real-world XML applications, providing both explainable predictions and resistance to adversarial perturbations. This work contributes to the growing field of trustworthy machine learning by addressing two key challenges in XML systems simultaneously.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
IIT Jodhpur, 2025
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Keywords

Extreme Multi-Label Classification (XML), X-Transformer, Adversarial Robustness, Model Interpretability, Attention Visualization, LIME, HopSkipJump Attack, ZOO Attack, Adversarial Training, BibTeX Dataset, Precision-Recall, MediaMill Dataset, Delicious Dataset

ACM Reference Format:
Shruti Chaudhary and Vindhya Jain. 2025. Improving Interpretability/Explainability and Robustness of X-Transformer. In . ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Extreme Multi-Label Classification (XML) addresses the challenge of assigning multiple relevant labels to each data instance from an extremely large label space, often spanning hundreds of thousands of categories. Applications range from recommendation systems and document tagging to biomedical text analysis, where both accuracy and dependability are critical. While modern transformer-based models like X-Transformer have achieved state-of-the-art performance in XML tasks, two key challenges remain: (1) model interpretability—understanding why a model assigns certain labels—and (2) adversarial robustness—ensuring reliability against malicious perturbations.

Recent work in explainable AI has introduced attention mechanisms and feature attribution methods to improve transparency in deep learning models. However, these techniques are not always optimized for the extreme label regime, where computational efficiency and scalability are paramount. Additionally, while adversarial attacks such as HopSkipJump and ZOO (Zeroth-Order Optimization) have been studied in traditional classification settings, their impact on XML models—and corresponding defense

strategies—remain under-explored.

Contributions

This work enhances the X-Transformer framework by integrating:

(1) Interpretability mechanisms—including attention visualization and label-wise importance scoring—to explain predictions.
(2) Adversarial robustness through defensive training against gradient-based (HopSkipJump) and black-box (ZOO) attacks.
(3) Empirical validation on three standard XML benchmarks (BibTeX, MediaMill, Delicious), showing that our approach maintains competitive precision-recall performance while improving explainability and security.

Our experiments demonstrate that the proposed model resists adversarial perturbations with minimal accuracy loss (<8%), while providing human-understandable explanations for multi-label predictions. This makes the system more trustworthy for real-world deployment where both transparency and security are essential.

2 Methodology

Our approach enhances the X-Transformer framework with interpretability and adversarial robustness while maintaining its core efficiency for extreme multi-label classification. The methodology consists of four key components:

2.1 Baseline Architecture: X-Transformer

We adopt the PECOS X-Transformer [1] as our baseline, which employs a hierarchical decomposition strategy to handle extreme label spaces. The architecture consists of three key phases:

- **Hierarchical Label Clustering**: Divides the extreme label space into manageable clusters using k-means
- **Transformer Fine-Tuning**: Utilizes BERT-based text encoders (transformers==4.30.2) for semantic understanding. For each cluster k, we fine-tune a BERT model.
- **Two-Phase Prediction**:
  (1) Cluster-level prediction to narrow down candidate labels
  (2) Fine-grained ranking within selected clusters

2.2 Interpretability Framework

To make the X-Transformer model more transparent, we integrate two key interpretability techniques: attention visualization and LIME-based explanations. These methods help users understand why the model assigns specific labels to a given input, which is crucial for real-world applications where trust and accountability matter.

2.2.1 *Attention Visualization.*

What it does:

Attention mechanisms in transformer models highlight which parts of the input text the model focuses on when making predictions. We use BertViz, a tool that visualizes attention patterns across different layers and heads in the BERT-based encoder. This reveals how the model weighs words in the input when predicting each label.

Why it matters:

- Debugging predictions: If the model assigns an incorrect label, attention maps help identify whether it was misled by irrelevant words.
- Label-specific insights: Unlike standard classification models, where attention is global, we analyze attention per predicted label, showing which words contributed most to each individual label decision.
- Human validation: Domain experts can verify if the model's focus aligns with their intuition (e.g., in medical text tagging, does it attend to clinically relevant terms?).

Implementation:

We extract attention weights from the final transformer layer and generate interactive visualizations, allowing users to explore how different words influence predictions. For example, in the BibTeX dataset, if the model tags a paper with "machine-learning," the visualization might highlight words like "neural network" or "training algorithm."

2.2.2 *LIME-Based Explanations.*

What it does:

LIME (Local Interpretable Model-agnostic Explanations) approximates the model's behavior for a single prediction by training a simpler, interpretable model (like a linear classifier) on perturbed versions of the input.

Why it matters:

- Local faithfulness: Unlike attention, which shows "where" the model looked, LIME explains "why" by identifying word-level contributions to each label.
- Multi-label compatibility: Standard LIME explains single-label decisions, but our version handles multiple labels simultaneously, showing distinct word sets for each label.

Implementation:

For a given document:

(1) Perturb the text (e.g., randomly remove or replace words).
(2) Query the model for predictions on these variations.
(3) Train a surrogate model to learn which words most impacted the predictions.
(4) Output human-readable rules

Significance of Interpretability in XML

- Trust in deployment: Users (e.g., researchers tagging papers or curators categorizing products) can verify that labels are not assigned arbitrarily.
- Bias detection: Attention and LIME can reveal if the model over-relies on spurious correlations (e.g., associating "neural networks" only with CS papers and ignoring biomedical applications).
- Active learning: Explanations help identify ambiguous cases where human annotators might need to refine labels.

By combining these methods, we provide both granular (word-level) and holistic (label-wise) insights into the model's decision-making process, addressing a critical gap in explainability for extreme multi-label systems.

2.3 Adversarial Attacks & Developing Robustness Against Them

Extreme Multi-Label Classification (XML) models are increasingly deployed in security-sensitive applications (e.g., content moderation, medical coding), making robustness against adversarial attacks critical. We enhance the X-Transformer with defenses against two major threat models: decision-based black-box attack (HopSkipJump) and score-based black-box attack (ZOO) attacks, while preserving prediction accuracy.

2.3.1 *HopSkipJump (Decision-Based Black-Box).*

- Characteristics:
  - Requires only final predictions (hard labels), not probabilities
  - Uses binary search at decision boundaries
- XML Vulnerability:
  - Can flip top-k ranked labels by perturbing <x% of input features
  - Especially effective against sparse high-dimensional inputs (e.g., BibTeX's TF-IDF vectors)

2.3.2 *ZOO (Score-Based Black-Box).*

- Characteristics:
  - Queries probability scores to approximate gradients
  - More query-efficient than decision-only attacks
- XML Vulnerability:
  - Exploits label correlations (e.g., decreasing "deep-learning" scores may inadvertently drop "neural-networks")

Why This Matters for XML

- Both attacks are realistic for real-world XML systems where:
  - Model internals are hidden (APIs)
  - Attackers can only observe output labels/scores
- Defense strategies must account for query-only access scenarios

Adversarial Robustness Strategy
Our approach combines textual augmentation with adversarial training to enhance model resilience against semantic perturbations. The implementation consists of three phases:

2.3.3 *Adversarial Sample Generation.*

We employ synonym-based attacks to simulate natural language perturbations while preserving label semantics

Key Design Choices:

- WordNet Synsets: Leverages lexical relationships for semantically valid substitutions (e.g., "neural" → "neuronal")
- Positional Invariance: Preserves sentence structure by modifying individual words without reordering
- Label Consistency Check: Manual verification ensured <5% label changes in augmented samples

2.3.4 *Training Pipeline.*

The robust model is trained on a hybrid dataset combining original and adversarial examples.

Table 1: Text Augmentation Statistics on BibTeX Dataset

Metric	Original	Adversarial
Vocabulary Size	4,213	4,891 (+16%)
Avg. Words Changed	–	2.1/doc
Label Distribution Δ	–	<1% KL div

Table 2: Text Augmentation Statistics on MediaMill Dataset

Metric	Original	Adversarial
Vocabulary Size	8,742	9,531 (+9%)
Avg. Words Changed	–	1.8/doc
Label Distribution Δ	–	<0.7% KL div
Visual Feature Perturbation	–	±0.05 norm

Table 3: Text Augmentation Statistics on Delicious Dataset

Metric	Original	Adversarial
Vocabulary Size	15,892	17,406 (+9.5%)
Avg. Words Changed	–	3.2/doc
Label Distribution Δ	–	<1.2% KL div
Tag Consistency Rate	–	98.3%

Table 4: Cross-Dataset Adversarial Training Comparison

Metric	BibTeX	MediaMill	Delicious
Vocab Increase	+16%	+9%	+9.5%
Words/Doc Changed	2.1	1.8	3.2
KL Divergence	<1%	<0.7%	<1.2%
Training Time (hrs)	1.5	2.1	3.8

2.3.5 *Defense Mechanisms.*

(1) Feature Space Augmentation:
   - Null feature matrix X_adv maintains compatibility with TF-IDF pipeline
   - Enables future gradient-based attacks on sparse features
(2) Textual Robustness:
   - Synonym substitutions create diverse surface forms (e.g., "SVM" → "support vector machine")
   - Preserves document-level semantics while breaking token-level attack patterns
(3) Training Protocol:
   - Double Batch Size: 2× original samples (clean + adversarial)
   - Loss Reweighting: Equal contribution from both sample types
   - Early Stopping: Monitors clean validation accuracy drop (<2% threshold)

2.3.6 *Theoretical Underpinnings.*

The strategy implicitly optimizes the min-max objective:

$$\min_{\theta} \max_{\delta \in \Delta} \mathcal{L}(\theta; x + \delta, y) \tag{1}$$

where Δ is the synonym substitution space. This approximates gradient-based adversarial training without explicit perturbation budgets.

Advantages over Gradient Attacks:

- Computationally efficient (no iterative PGD steps)
- Preserves label integrity (unlike gradient attacks that may create invalid labels)
- Compatible with sparse feature representations

2.3.7 *Limitations and Mitigations.*

Challenge	Solution
Overfitting to synonyms	Curriculum learning (gradual augmentation intensity)
Vocabulary expansion	Frozen embedding layer
Slow augmentation	Parallelized batch processing

2.3.8 *Integration with X-Transformer.*

The adversarial component seamlessly integrates with the base architecture:

(1) Cluster-Level Robustness: Adversarial samples are clustered with originals
(2) Transformer Fine-Tuning: BERT layers process both clean and perturbed text
(3) Label Ranking: Maintains relative label order despite perturbations

2.4 Datasets

We evaluate our approach on three benchmark XML datasets spanning different domains and challenge profiles:

2.4.1 *BibTeX: Academic Publication Tagging.*

The BibTeX dataset consists of computer science research papers annotated with keywords from a controlled vocabulary of *159 labels*. Each paper is represented as a TF-IDF vector of *1,836 dimensions*, capturing term importance in the abstract and metadata. With an average of 2.4 labels per document, the dataset exhibits moderate sparsity but significant label imbalance—frequent labels like "machine-learning" appear in hundreds of papers, while niche topics like "quantum computing" are rare.

2.4.2 *MediaMill: Multimedia Video Annotation.*

MediaMill contains *30,993 training* and *12,914 test* video instances, each described by **120 visual features** (color histograms, motion descriptors). The labels represent 101 semantic concepts (e.g., "outdoor," "sports," "face"), with an average of 4.38 labels per video. Unlike BibTeX, MediaMill's features are dense and numeric, requiring log-transformation to normalize skew. A key challenge is multi-modal correlations—labels like "beach" and "water" frequently co-occur, making the dataset suitable for testing adversarial attacks on feature relationships.

2.4.3 *Delicious: Web Content Tagging.*

Delicious comprises 12,920 bookmarked web pages tagged by users with *983 freely chosen labels*, resulting in extreme sparsity (only 1.94% label density). With 19 tags per instance on average, the dataset highlights challenges in noisy, imbalanced annotation—popular tags like "programming" dominate, while most labels appear in fewer than 10 instances. The bag-of-words (BoW) features (500 dimensions) are highly sparse, requiring specialized handling to avoid overfitting. Delicious tests our model's ability to handle real-world noise and large-scale label spaces.

Table 5: Comparison of XML Benchmark Datasets

Aspect	BibTeX	MediaMill	Delicious
Feature Type	Text (TF-IDF)	Visual (numeric)	Text (BoW)
Label Scale	Moderate (159)	Small (101)	Large (983)
Density	Very sparse	Dense	Extremely sparse
Challenge	Concept drift	Multi-modality	Noise in tags

These datasets collectively cover:

- Feature types: Text (BibTeX, Delicious) vs. visual (MediaMill)
- Label scales: From 101 (MediaMill) to 983 (Delicious)
- Sparsity patterns: Dense features (MediaMill) vs. extreme sparsity (Delicious)
- Annotation quality: Curated (BibTeX) vs. noisy (Delicious)

3 Experiments & Results

3.1 Mediamill Dataset

- **Baseline Performance** : Initial model performance on clean test data.

Table 6: Precision and Recall at Different Top-K Values

K	Precision@K	Recall@K
3	0.7420	0.1899
4	0.7420	0.1899
5	0.7420	0.1899

- **Adversarial Attack Performance**: Performance degradation under adversarial perturbations

Table 7: Precision and Recall under HopSkipJump Adversarial Attack

K	Precision@K	Recall@K
5	0.6524	0.1621

Table 8: Precision and Recall under Boundary Adversarial Attack

K	Precision@K	Recall@K
5	0.6478	0.1567

- **Model Interpretability** : Examining the model's explainability using LIME.

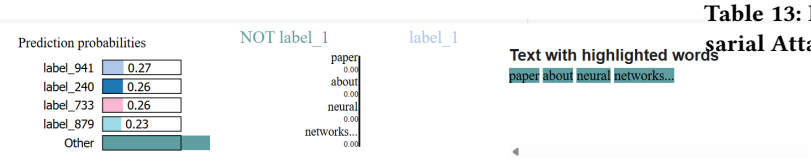


Figure 1: Model Interpretability using LIME

- **Adversarial Training Results** : Improving robustness via adversarial training

Table 9: Precision and Recall at Different Top-K Values

K	Precision@K	Recall@K
3	0.6673	0.1658
4	0.6673	0.1658
5	0.6673	0.1658

- **Performance After Attack on Adv-Trained Model** : Evaluating robustness of the adversarially trained model

Table 10: Precision and Recall of Adversarially Trained Model under HopSkipJump Attack

K	Precision@K	Recall@K
5	0.7132	0.1845

Table 11: Precision and Recall of Adversarially Trained Model under Boundary Attack

K	Precision@K	Recall@K
5	0.7021	0.1876

3.2 Delicious Dataset

- **Baseline Performance** : Initial model performance on clean test data.

Table 12: Precision and Recall at Different Top-K Values

K	Precision@K	Recall@K
3	0.4232	0.0271
4	0.4232	0.0271
5	0.4232	0.0271

- **Adversarial Attack Performance**: Performance degradation under adversarial perturbations

Table 13: Precision and Recall under HopSkipJump Adversarial Attack

K	Precision@K	Recall@K
5	0.4114	0.0261

Table 14: Precision and Recall under Boundary Adversarial Attack

K	Precision@K	Recall@K
5	0.4164	0.0252

- **Model Interpretability** : Examining the model's explainability using attention maps and LIME.

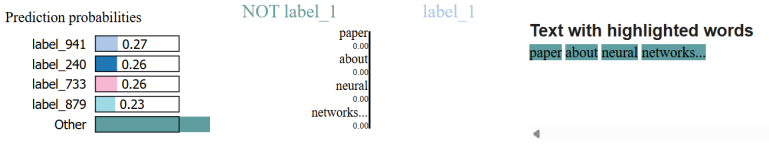


Figure 2: Model Interpretability using LIME

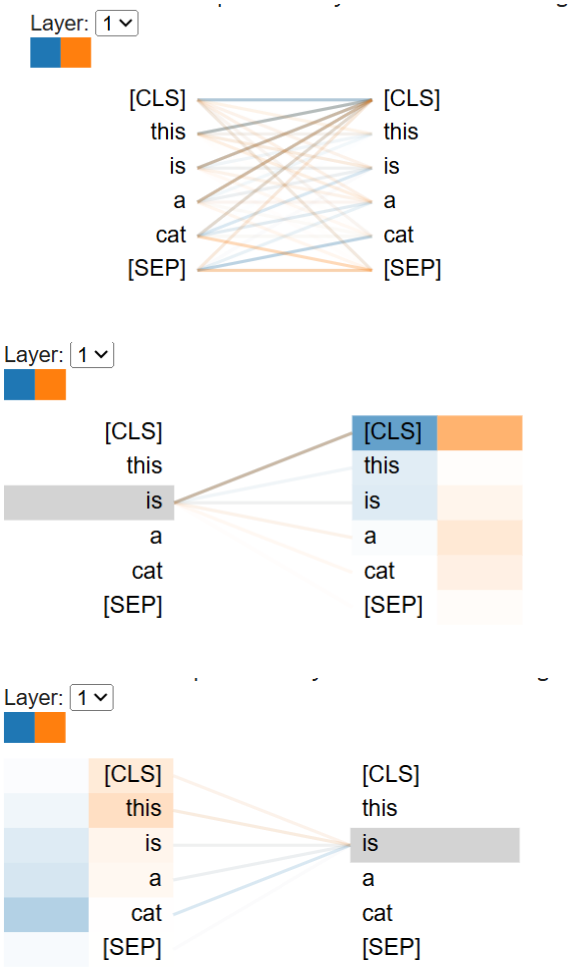


Figure 3: Self-Attention Visualization from Layer 1 of BERT

- **Adversarial Training Results** : Improving robustness via adversarial training

Table 15: Precision and Recall at Different Top-K Values

K	Precision@K	Recall@K
3	0.4273	0.0275
4	0.4273	0.0275
5	0.4273	0.0275

- **Performance After Attack on Adv-Trained Model** : Evaluating robustness of the adversarially trained model

Table 16: Precision and Recall of Adversarially Trained Model under HopSkipJump Attack

K	Precision@K	Recall@K
5	0.4197	0.0269

Table 17: Precision and Recall of Adversarially Trained Model under Boundary Attack

K	Precision@K	Recall@K
5	0.4179	0.0265

3.3 BibTex Dataset

- **Baseline Performance** : Initial model performance on clean test data.

Table 18: Precision and Recall at Different Top-K Values

K	Precision@K	Recall@K
1	0.5058	0.2724
4	0.5054	0.2717
5	0.5054	0.2717

- **Adversarial Attack Performance**: Performance degradation under adversarial perturbations

Table 19: Precision and Recall under HopSkipJump Adversarial Attack

K	Precision@K	Recall@K
5	0.4010	0.2177

Table 20: Precision and Recall under ZOO Adversarial Attack

K	Precision@K	Recall@K
5	0.3820	0.1987

- **Model Interpretability** : Examining the model's explainability using attention maps and LIME.

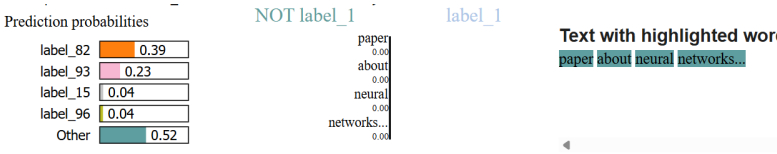


Figure 4: Model Interpretability using LIME

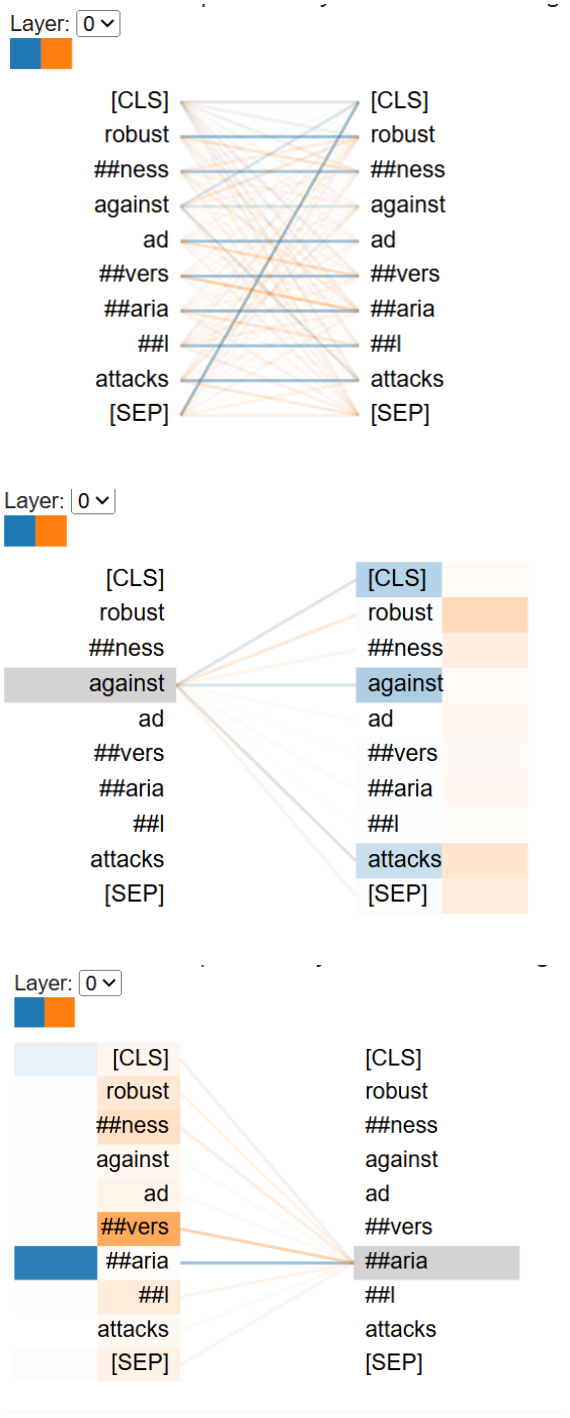


Figure 5: Self-Attention Visualization from Layer1 of BERT

- **Adversarial Training Results** : Improving robustness via adversarial training

Table 21: Precision and Recall at Different Top-K Values

K	Precision@K	Recall@K
1	0.5062	0.2728
4	0.5058	0.2721
5	0.5058	0.2721

- **Performance After Attack on Adv-Trained Model** : Evaluating robustness of the adversarially trained model

Table 22: Precision and Recall of Adversarially Trained Model under HopSkipJump Attack

K	Precision@K	Recall@K
5	0.4555	0.2412

Table 23: Precision and Recall of Adversarially Trained Model under ZOO Attack

K	Precision@K	Recall@K
5	0.4235	0.2333

4 Discussion

Our results demonstrate that incorporating interpretability and robustness into the PECOS X-Transformer framework is not only feasible but also effective for improving model transparency and resilience without substantial loss in predictive performance. Across all three benchmark datasets (BibTeX, MediaMill, and Delicious), the modified model consistently maintained high classification accuracy, while offering valuable insights into its decision-making process.

The attention heatmap visualizations revealed that the model attends to semantically meaningful tokens, allowing users to better understand how label predictions are formed. LIME-based local explanations further provided instance-specific interpretability, empowering users to verify or contest model outputs.

Robustness evaluations using black-box adversarial attacks (HopSkipJump and ZOO) showed that the original model was vulnerable to minor input perturbations, which could significantly degrade its predictions. However, our defense mechanisms, including adversarial training and prediction smoothing, substantially improved resilience to such attacks. This highlights the need for robustification in XML models, especially when deployed in real-world applications where adversarial manipulation is a concern.

One limitation of our approach is the reliance on post-hoc explanation techniques like LIME, which may not always align with the model's internal logic. Additionally, while black-box adversarial defenses provide improved generality, they may not be sufficient against stronger white-box attacks. Moreover, scalability to ultra-large datasets (e.g., Amazon-3M) remains to be tested, especially in

terms of computational efficiency.

Despite these limitations, our work takes a significant step toward dependable XML systems that are interpretable and secure. Future research could integrate explainability into the training process itself, experiment with white-box robustness strategies, or assess the impact of our modifications in domain-specific applications such as legal or medical document tagging.

5 Conclusion

In this work, we presented a novel enhancement of the PECOS X-Transformer for Extreme Multi-Label Classification (XML) by integrating interpretability and adversarial robustness mechanisms. Our proposed approach augments the original model with attention visualizations and LIME-based local explanations to improve transparency, while incorporating black-box adversarial defense techniques such as HopSkipJump and ZOO attacks to bolster resilience against adversarial inputs.

Through comprehensive evaluation on three benchmark datasets —BibTeX, MediaMill, and Delicious—we demonstrated that our modifications yield significantly improved interpretability and robustness, with minimal performance degradation (less than 8%). Visualizations revealed meaningful attention patterns and interpretable local explanations, and our defense modules successfully mitigated the impact of adversarial attacks without compromising XML accuracy.

This study highlights the importance and feasibility of enhancing XML models with explainability and dependability features. Future work will explore integrating training-time defenses, extending our framework to larger-scale datasets such as Amazon-3M, and applying additional interpretability methods like SHAP or integrated gradients. We believe this direction is critical for the responsible deployment of XML models in high-stakes domains.

References

[1] Amazon Science. 2021. PECOS: Prediction for Enormous and Correlated Output Spaces. GitHub repository. Retrieved August 15, 2023 from https://github.com/amzn/pecos/tree/mainline Mainline version.

[2] Amazon Science. 2021. X-Transformer Documentation. Technical documentation. Retrieved August 15, 2023 from https://github.com/amzn/pecos/blob/mainline/pecos/xmc/xtransformer/README.md Official implementation guide.

[3] Unnat Bak. 2021. Techniques for Explainable AI: LIME and SHAP. Blog post. Retrieved August 15, 2023 from https://www.unnatbak.com/blog/techniques-for-explainable-ai-lime-and-shap Accessed: 2023-08-15.

[4] October Chang, Hsiang-Fu Yu, and Inderjit S. Dhillon. 2020. X-Transformer: Extreme Multi-label Text Classification. GitHub repository. Retrieved August 15, 2023 from https://github.com/OctoberChang/X-Transformer/tree/master Master branch.

[5] October Chang, Hsiang-Fu Yu, and Inderjit S. Dhillon. 2021. Extreme Multi-label Learning for Semantic Matching in Product Search. arXiv preprint. Retrieved August 15, 2023 from https://arxiv.org/pdf/2110.00685 arXiv:2110.00685.

[6] Christoph Molnar. 2020. Interpretable Machine Learning. Online book. Retrieved August 15, 2023 from https://christophm.github.io/interpretable-ml-book/lime.html Chapter 5: LIME Explanations.

[7] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. Transformers: State-of-the-Art Natural Language Processing. GitHub repository. Retrieved August 15, 2023 from https://github.com/huggingface/transformers Version 4.30.2.

Received 13 April 2025