From Black Box to User Insight: Explainability and Usability in Toxicity Detection

Oliver Lohaj
Department of Cybernetics and Artificial Intelligence
Faculty of Electrical Engineering and Informatics
Technical university of Košice, Letná 9, 040 01, Košice, Slovakia
oliver.lohaj@tuke.sk

Anastasiia Radishevska
Department of Cybernetics and Artificial Intelligence
Faculty of Electrical Engineering and Informatics
Technical university of Košice, Letná 9, 040 01, Košice, Slovakia
anastasiia.radishevska@student.tuke.sk

Abstract*. This study investigates the detection of toxic content in text data by integrating deep learning models with explainable artificial intelligence (XAI) techniques, with a particular focus on model transparency and usability. We evaluate three widely used neural architectures—CNN, LSTM, and BERT—on a labeled Twitter dataset, comparing their classification performance and the interpretability of their outputs. To enhance model explainability, we apply SHAP and Layer-wise Relevance Propagation (LRP) methods, visualizing word-level contributions to each prediction. The usability of these models is assessed through the clarity and reliability of their explanations. Our results show that while LSTM achieved the best overall classification performance, the combination of SHAP with LSTM provided the most interpretable and actionable insights. This work highlights the trade-offs between accuracy, explainability, and usability in toxicity detection, offering practical guidance for deploying trustworthy AI systems in content moderation.*

Keywords. BERT, classification, CNN, detection of toxicity, explainable AI, LRP, LSTM, SHAP

1 Introduction

This work addresses the detection of toxic content in text data, combining deep learning with explainability techniques. It begins with a theoretical overview and analysis of existing approaches. The main goal is to apply and evaluate selected models and explainability methods to better understand the decision-making of neural networks. Finally, all methods are assessed and compared using standard evaluation metrics.

This work also builds on previous research presented at CECIIS 2024, where we explored the usability challenges of integrating multiple data sources for toxic behavior detection in social media (Lohaj et al., 2024). While that study focused on the broader context of combining heterogeneous data inputs to support detection models, the present work narrows the focus to the explainability and usability of the models themselves. Specifically, we aim to make the decision-making processes of individual deep learning models more transparent and actionable for end users. This progression reflects a shift from system-level integration concerns to model-level interpretability, aligning with the growing need for trustworthy and user-centered AI systems in content moderation. While our dataset originates from sentiment analysis, we frame negative sentiment as indicative of toxic content for the purposes of model training and evaluation. This framing enables us to assess toxicity detection methods using sentiment-labelled social media data, while maintaining focus on the broader societal relevance of mitigating harmful online interactions.

Toxicity detection plays a critical role in protecting individuals from online harassment, hate speech, and other harmful behaviors. As online communication increasingly shapes public discourse, detecting and mitigating toxic content is essential for maintaining respectful dialogue, preventing psychological harm, and supporting the work of moderators on social media platforms. By focusing on explainability and usability, this study addresses not only the accuracy of detection but also the trust and accountability required for societal adoption of AI moderation tools.

2 Related Works

2.1 Text classification

Several studies have addressed toxic content classification in online environments. One notable work (Grine, 2021) analyzed various methods, including SVM (Support Vector Machine), CNN (Convolutional Neural Network), and LSTM (Long-Short-Term Memory Network), to identify toxic comments and evaluate their performance. The main

challenges tackled in the study were the lack of multi-label support and significant class imbalance.

The models were trained and tested on the Conversation AI "Wikipedia Talk" dataset, containing nearly 160,000 comments labelled across six categories: toxic, severe toxic, obscene, threat, insult, and identity hate. The study revealed a substantial imbalance in label distribution, which negatively influenced the models' performance on minority classes. A correlation matrix also showed strong co-occurrence between classes such as toxic and insult or obscene (e.g., r = 0.74 for insult–obscene).

Evaluation based on Precision, Recall, and F1-score metrics demonstrated that CNN outperformed others in most categories, especially for the minority class threat. SVM achieved the best precision, while LSTM showed slightly better recall. All models performed better on majority classes, with F1-scores above 80%, and worse on minority ones (47–73%).

Learning curve analysis further showed that CNN reached strong performance with only 2% of training data, outperforming both SVM and LSTM early in training. LSTM required more data (up to 25%) to surpass SVM, with diminishing returns observed beyond 80% of the dataset.

Another study by authors (Anand & Enswari, 2019) focused on CNN and LSTM models, comparing their training behavior using two key metrics: training accuracy and training loss. The models were trained on a dataset from Wikipedia's talk page edits. In this dataset, there are almost 160.000 comments and labelled with different categories some of the comments belong to more than one category. The visualized results showed that CNN achieved rapid accuracy gains during training, consistent with previously reported trends for convolutional architectures in text classification. CNN achieved a rapid increase in accuracy, reaching 97.8%, with a low training loss of 5.42%. In contrast, LSTM demonstrated lower accuracy and higher loss, indicating slower convergence and potentially less effective learning dynamics.

In another publication by (Maslej-Krešňáková et al., 2020), the authors examined and compared traditional deep learning models (FFNN (Feedforward Neural Network), CNN, GRU (Gated Recurrent Unit), BiGRU (Bidirectional Gated Recurrent Unit), BiLSTM-CNN (Bidirectional Long Short-Term Memory)) and transformer-based language models (BERT, DistilBERT, XLNet) under various preprocessing and text representation techniques. The experiments were conducted on the Kaggle Toxic Comment Classification dataset, which training data contains almost 160.000 Wikipedia comments labelled across six toxicity categories. The dataset exhibits a strong imbalance toward non-toxic classes, making it a challenging benchmark for classification models. The experiments revealed that the combined BiLSTM-CNN model achieved the best results among the traditional architectures, with an F1 score of approximately 0.67. When TF-IDF (Term Frequency-Inverse Document Frequency) text representation was used along with standard preprocessing techniques—tokenization, lowercasing, punctuation removal, and stop word elimination—performance significantly improved. The F1 score increased from 0.09 to 0.35. Furthermore, in the comparison of transformer-based models, the BERT-base (uncased) variant achieved the highest performance, with an F1 score of 0.69 and an AUC score of 0.984, outperforming both other transformer variants and traditional models. Study by (Ansar et al., 2024) reviews new strategies for optimizing transformer-based NLP models for faster inference and lower resource use, including pruning, quantization, and low-rank adaptation and supports the transformer advancement and efficiency. Recent work by (Wu et al., 2025) highlights how transformer-based architectures, such as BERT and GPT, have redefined state-of-the-art performance in text understanding by effectively handling long-range dependencies and complex contextual relationships. Their methodological framework and insights into efficiency optimization could inform the integration of advanced NLP techniques into similar AI-driven applications discussed in the present study.

2.2 Using explainable artificial intelligence in text classification

Authors (Nguyen et al., 2024) compared three popular XAI techniques: LIME (Local Interpretable Model Agnostic Explanations), SHAP (SHapley Additive exPlanations), and CAM (Class Activation Mapping), and proposed enhancements for their evaluation and application. Their stability analysis showed that LIME and SHAP produce consistent results with 1,000 samples, but become less reliable with 500 or 200 samples, indicating that a higher sample count improves result stability and accuracy.

Each method had distinct advantages: CAM offered the fastest computation, while SHAP uniquely identified both positive and negative feature contributions. However, LIME and CAM are limited to classification tasks, and CAM is image-specific. In contrast, SHAP demonstrated broader applicability across data types and tasks.

In another paper by (Gholizadeh & Zhou, 2021) the authors applied LRP (Layer-wise Relevance Propagation) to explain SVM-based review classification. They used a dataset of over 229,000 customer reviews, reformulated as binary classification. To analyze the SVM predictions, a CNN model was trained to replicate SVM outputs, using pre-trained word embeddings and multiple convolutional filter sizes. After five epochs (batch size 30), the CNN achieved high agreement with SVM predictions (F1 score of 0.93).

LRP highlighted the most relevant tokens influencing classifications and visualized them for interpretability. This approach demonstrated that LRP + CNN can

effectively interpret machine learning models and offer actionable insights for improving business decisions based on customer feedback.

3 Deep Learning Methods

Deep learning has become a dominant approach in many natural language processing (NLP) tasks due to its ability to automatically extract complex features from large datasets. Among the wide variety of deep learning architectures, certain models have demonstrated exceptional performance in text classification and interpretability research. This section provides a concise overview of three models: CNN, LSTM, and BERT.

3.1 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) (O'Shea, 2015), originally designed for image processing, have been adapted to text classification to capture local n-gram patterns using convolutional filters. A typical CNN for text includes convolutional and activation layers (e.g., ReLU) to detect patterns, pooling layers to reduce dimensionality, and fully connected layers for classification. CNNs are valued for their computational efficiency and ability to capture short-range dependencies in text.

3.2 Long Short-Term Memory Network (LSTM)

LSTM networks (DiPietro, 2020) are designed to capture long-term dependencies in sequential data by using a memory cell and three gates (forget, input, output) to regulate information flow. This architecture overcomes the vanishing gradient problem in traditional RNNs and is particularly effective for tasks requiring contextual understanding, such as sentiment analysis and speech recognition.

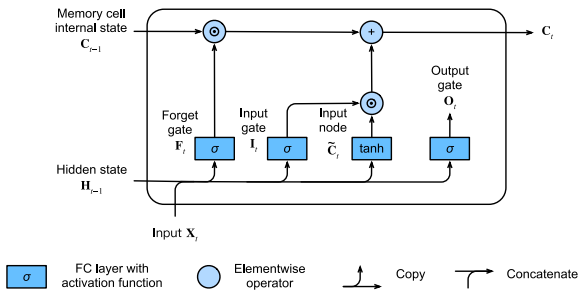

Figure 1. LSTM architecture.[1]

3.3 Bidirectional Encoder Representations from Transformers (BERT)

BERT is a transformer-based model that utilizes a bidirectional attention mechanism to understand the context of a word based on its surrounding words, regardless of their position (Sun et al., 2019). Unlike traditional left-to-right models, BERT processes text in both directions simultaneously.

The training process includes:

- Masked Language Modeling (MLM): predicting masked words in a sentence.
- Next Sentence Prediction (NSP): determining whether two sentences follow each other.

After pre-training, BERT is fine-tuned on specific NLP tasks using smaller, domain-specific datasets. Its strong contextual understanding has made it state-of-the-art in various classification and interpretation tasks.

4 Explainable AI Methods

Modern machine learning models, especially deep learning architectures, are often referred to as black box (Awati & Yasar, 2024) due to their complex internal structures that make it difficult to understand how input data leads to a particular output. This lack of transparency poses significant challenges in domains where trust, accountability, or legal compliance are crucial. To address this issue, XAI (Ryo, 2022) methods have been developed to provide insights into model behavior and decision-making. We can count the following methods as frequently used:

- SHapley Additive exPlanations (SHAP)

This approach is based on game theory and uses Shapley values to evaluate the contribution of individual factors to the model's outcome (Atesli, 2023). Since computing exact Shapley values is computationally expensive, the method approximates them to provide an understanding of how each factor contributes to the prediction.

- Layer-wise Relevance Propagation (LRP)

LRP is a model-specific method that provides detailed explanations of predictions, especially for deep neural networks (Praveen, 2021). This process analyzes which parts of the input data—such as image pixels or words in text—contributed the most to a given prediction. Relevance is propagated backward through the layers of the neural network, helping to identify the inputs that had the greatest impact on the model's result.

[1] https://d2l.ai/chapter_recurrent-modern/lstm.html

5 Data Understanding and Preparation

5.1 Data Description

This section provides an overview of the dataset used in this study, focusing on its structure, quality, and potential issues that may impact model accuracy. Understanding the data is crucial for optimizing preprocessing and designing effective models.

The dataset consists of tweets from Twitter\Hugging Face in English, divided into three subsets:

- Training set: 12.9k tweets
- Test set: 3.7k tweets
- Validation set: 1.85k tweets

The dataset has two attributes:

- Text – the tweet content
- Sentiment – numeric attribute (1 for Positive, 0 for Negative) representing the sentiment of the tweet.

Each row represents a tweet with its assigned sentiment. The sentiment labels in this dataset were provided by the original dataset creators on the Hugging Face platform. Annotation was performed through a combination of manual review and automated classification heuristics, with 0 representing a negative sentiment and 1 representing a positive sentiment. The labeling guidelines ensured that tweets containing expressions of dissatisfaction, criticism, or negative tone were marked as negative, while those with expressions of approval, gratitude, or positive tone were marked as positive. Data quality checks revealed no missing values or duplicates, and the sentiment attribute contains only unique values (0, 1). In this study, tweets labeled with negative sentiment polarity (0) are treated as toxic content, while those with positive sentiment polarity (1) are considered non-toxic. This mapping allows us to apply toxicity detection techniques to a sentiment-labeled dataset, reflecting the assumption that negative sentiment often corresponds to toxic or harmful language in online discourse. Initial data analysis was performed through visualizations to better understand the dataset's structure. The training set contains approximately 7,000 positive and 6,000 negative tweets, showing a slight imbalance (see Fig. 1).

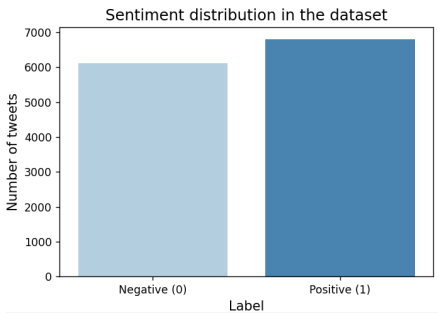

Figure 2. Visualizing sentiment distribution in the Twitter dataset

The analysis of the tweet lengths revealed that the average tweet contains 70.17 characters (see Fig. 2). Interestingly, there was little difference in length between positive and negative tweets, with the average lengths being 70.13 characters for positive tweets and 70.21 characters for negative ones. This suggests that there is no significant correlation between tweet length and sentiment in the dataset.

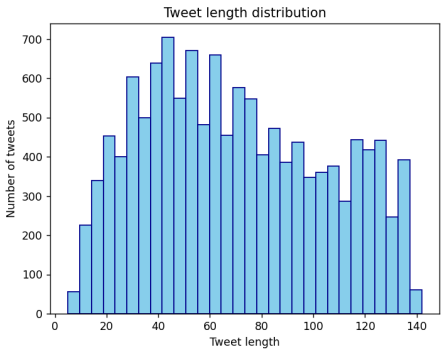

Figure 3. Tweet length distribution in the Twitter dataset

Additionally, the average word count per tweet was 13.24 (see Fig. 3), with slightly higher counts for negative tweets (13.43 words) compared to positive ones (13.07 words). This indicates that tweet content does not appear to vary significantly in terms of word count between the two sentiment classes.

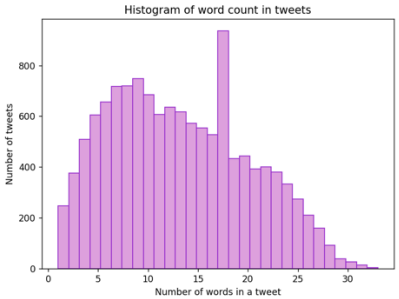

Figure 4. Number of words in tweets in the Twitter dataset

One particularly insightful visualization was the word cloud in Fig. 4, which was used to display the most frequent words in the tweets. This graphical representation highlighted words like "love", "thank", and "good", which were used frequently in the dataset. In addition to these common words, the word cloud also revealed recurring fragments such as "S", "m", "ll", "im", and "t", which likely represent truncated words or abbreviations. These patterns provide valuable insights for the next steps in preprocessing, where handling these truncated forms and abbreviations could improve model performance and ensure better text representation.

Figure 5. Most used words in tweets

5.2 Data Pre-Processing

This stage focuses on transforming raw tweets into a clean, standardized format suitable for analysis and modeling. Tweets are often noisy — containing inconsistent casing, numbers, excessive punctuation, links, usernames, and other irrelevant elements — which can affect model performance.

To address this, a custom preprocess function was implemented. It performs the following key steps:

- Normalization – converts all text to lowercase (e.g., "HAPPY DAY" → "happy day").
- Noise removal – eliminates URLs, emails, HTML tags, usernames (e.g., _user), numbers, punctuation, and extra spaces.
- Tokenization – splits text into individual words.
- Stop-word removal – removes common, uninformative words like "is", "a", "the".
- Lemmatization - converts words to their base forms (e.g., "took" → "take").

This process uses libraries such as *re* for pattern matching, *string* for punctuation handling, and *nltk* for tokenization, stop-word filtering, and lemmatization. As a result, the text becomes cleaner and better suited for machine learning models (see Table 1).

Table 1 Comparison of input and processed tweets

The initial form of tweets	Processed tweets
Just took my IC photo! Looks good - http://tweet.sg	just take ic photo look good
I'm so mad I won't be there!!!!!!!!! UGHHH!!!	mad will ughhh
Oh yes! Level 40	yes level
thank you! Marc Jacobs thou ....love limited too	thank marc jacob thou love limit

6 Modeling

6.1 Setting up models

To evaluate the effectiveness of different deep learning architectures in toxic content classification, three representative models were implemented: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and BERT (Bidirectional Encoder Representations from Transformers). Each model was configured with tailored architectures and training strategies, considering the unique characteristics of text data and model-specific requirements.

CNN Configuration

The CNN model was designed to capture local textual patterns through convolutional operations. It began with an Embedding layer of dimension 50, which transformed input tokens into dense vector representations. Tweets were padded to a fixed length to ensure consistent input shape across the dataset. A 1D Convolutional layer with 100 filters and a kernel size of 3 was used to extract tri-gram level features. Instead of a pooling layer, a Flatten layer was employed to retain the full spatial structure of the learned features. This was followed by a Dense layer with 128 units and ReLU activation, along with a Dropout layer (rate = 0.3) to mitigate overfitting. The final layer was a single-unit output layer with sigmoid activation, suitable for binary classification. The model was compiled with binary cross-entropy loss, Adam optimizer, and trained over 2 epochs with a batch size of 32.

LSTM Configuration

The LSTM model was employed to capture the sequential dependencies and contextual flow in the text. It also began with an Embedding layer (dimension = 50). A single-layer LSTM unit with 64 memory cells processed the input sequences, capturing both short- and long-term dependencies. The output of the LSTM layer was passed through a Dropout layer (rate = 0.4) to enhance generalization. A fully connected Dense layer with a sigmoid output completed the model architecture. The model was trained using the Adam optimizer and binary cross-entropy loss, with a batch size of 32 over 2 epochs. Token sequences were padded to a uniform length, and text preprocessing ensured consistent input representation.

BERT Configuration

The transformer-based BERT model was leveraged to take advantage of its powerful contextual language understanding. Specifically, the bert-base-uncased variant from Hugging Face's Transformers library was used. Tokenization was performed using BertTokenizerFast, which ensured compatibility with the model's vocabulary and token structure. The base model outputs were passed through a custom classification head comprising three Dense layers (768 → 512 → 256 → 2), with GELU activations and Dropout layers (rate = 0.2) in between for regularization. The final output was normalized using LogSoftmax to produce log probabilities for each class.

The model was fine-tuned for 9 epochs using the AdamW optimizer with a learning rate of 2e-5, and training was stabilized using a linear learning rate scheduler with warm-up steps. The batch size was set to 32.

The number of training epochs for CNN and LSTM was limited to 2 based on preliminary experiments in which both models reached near-optimal validation performance very quickly. Extending training beyond this point led to early signs of overfitting, with increased validation loss and minimal accuracy gains. This rapid convergence is partly due to the relatively small dataset size and the extensive preprocessing, which reduced noise and simplified the learning task. The chosen setting balances performance with computational efficiency.

The selected hyperparameters for CNN and LSTM (embedding size = 50, 100 filters with kernel size = 3, dropout rates of 0.3 and 0.4) follow commonly used configurations in toxic content and sentiment classification tasks on short text datasets (e.g., Anand & Eswari, 2019; Maslej-Krešňáková et al., 2020). These settings balance model complexity and computational efficiency, ensuring stable training without overfitting. The BERT learning rate (2e-5) was chosen in line with recommendations from Sun et al. (2019) for fine-tuning transformer-based models. Preliminary tests with slightly higher embedding dimensions and lower dropout rates did not yield notable performance improvements, so the original configuration was retained.

Each model was trained and evaluated under the same data conditions to ensure a fair comparison. Evaluation metrics included Accuracy, Precision, Recall, F1 score, and AUC, allowing for both predictive and probabilistic performance analysis.

6.2 Evaluation

After completing the modeling phase, the next critical step is the evaluation of the developed models. This phase focuses on analyzing their performance using selected metrics to assess their effectiveness in toxic text classification. The main objective is to determine which of the implemented models achieves the best results and is therefore most suitable for the task.

Three previously mentioned models were evaluated and compared in this study: CNN, LSTM, and BERT — each representing a different approach to text processing. Their evaluation results are summarized in Table 2.

Table 2 Comparison of the performance metrics of CNN, LSTM and BERT

	Accuracy	Precision	Recall	F1 score	AUC score
CNN	0.8492	0.8712	0.8376	0.8541	0.92
LSTM	**0.8568**	0.8580	**0.8726**	**0.8652**	**0.93**
BERT	0.8106	0.8100	0.8106	0.8102	0.89

Based on the comparison, the LSTM model achieved the highest overall performance and is recommended as the most suitable for the given classification task. The CNN model also showed strong results and can be considered an effective alternative. Although BERT performed slightly lower in terms of accuracy, it offers stable outputs and has the potential for improvement through further training optimization.

6.3 Application of the LRP XAI method

This section presents the application of the LRP method for explaining the decision-making processes of CNN and BERT models. To enable LRP, the final sigmoid layer was removed from the CNN model, as LRP requires access to pre-activation outputs. The *innvestigate* library was used, specifically the *"lrp.alpha_2_beta_1"* rule, which balances positive and negative contributions of neurons.

The method was applied to selected test examples. For each word in the input, LRP computed relevance scores, indicating its impact on the model's prediction. These scores were normalized using min-max scaling (0 to 1) and visualized using a blue gradient — darker blue indicating higher influence, white indicating none.

For example, in a correctly classified positive tweet, CNN assigned relevance scores such as:

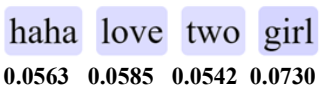

Figure 6. Relevance scores by CNN

In a negative example, LRP highlighted key influencing words with scores like:

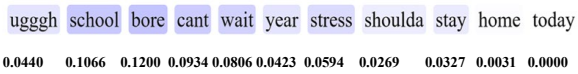

Figure 7. Negative example scores

For the BERT model, a modified version was created by removing the final LogSoftmax activation to enable analysis of raw model outputs. A custom *simple_lrp* implementation was used to compute relevance scores by backpropagating gradients through input embeddings.

Relevance scores were calculated for each token in a selected test example, indicating their contribution to the final classification. These scores were normalized to a [0,1] range using min-max scaling. A visualization was created using a colour scale — blue representing strong contributions, and red weak or no influence.

In a positive example, the token-level relevance scores were:

Figure 8. Token-level relevance scores

For a negative (toxic) input, the scores were:

Figure 9. Negative input scores

This visualization made it possible to identify the key tokens that influenced the classification decision. The method demonstrated that BERT relies on semantically relevant words, confirming that the model's decision-making is largely interpretable and transparent.

6.4 Application of the SHAP XAI method

Another explainability technique used in this work is SHAP, applied to interpret the predictions of the LSTM neural network model. The *KernelExplainer* from the SHAP library was used, enabling SHAP value computation for each word in the input tweet. Positive SHAP values indicated contribution toward the non-toxic class, while negative values supported the toxic class.

The explanations were visualized as horizontal bar plots, where each word was color-coded — blue for non-toxic influence and red for toxic influence. These visualizations provided clear insight into which words influenced the model's decisions.

Several examples from the test set were analyzed. In the first case, the LSTM correctly classified a toxic input. The SHAP values highlighted the words that strongly contributed to the toxic label.

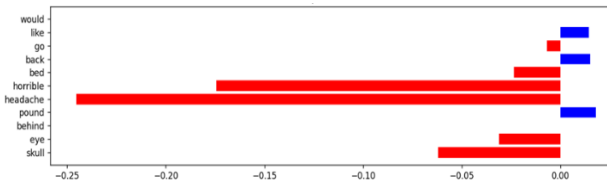

Figure 10. SHAP analysis for LSTM– 1st example

Another example showed a correctly classified non-toxic input:

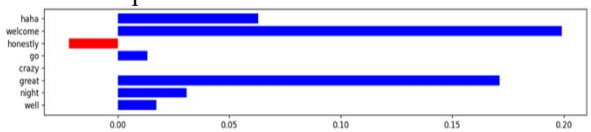

Figure 11. SHAP analysis for LSTM – 2nd example

A third example involved a misclassified input: the model predicted the toxic class with a probability of 0.2897, but the true label was non-toxic. SHAP values revealed which specific words misled the model, highlighting how even well-performing models can struggle with certain inputs due to misinterpretation of key terms.

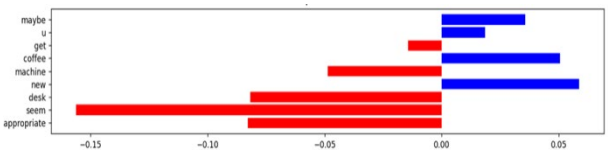

Figure 12. SHAP analysis for LSTM – 3rd example

This analysis demonstrated how SHAP helps uncover the reasoning behind both correct and incorrect classifications, enhancing the interpretability of the model.

In the context of this study, "usability" is used as a broader term that encompasses explainability, interpretability, and the clarity of the model outputs. Here, it refers to how easily a human moderator could understand and act upon the explanations provided by SHAP and LRP visualizations. While these analyses demonstrate interpretability potential, no formal user study or quantitative usability metrics were collected in this work.

7 Conclusion

This study demonstrates that integrating deep learning with explainable artificial intelligence (XAI) methods significantly enhances the transparency and usability of toxicity detection systems. By aligning model performance with transparency and usability, this work contributes to the development of AI systems that are not only effective but also trusted and responsible, supporting healthier online communities and reducing the impact of toxic discourse on individuals and society. Through comparative analysis of CNN, LSTM, and BERT architectures, we found that LSTM achieved the highest classification performance, while the application of SHAP and LRP offered valuable insights into model decision-making processes. These results underscore the necessity of balancing predictive accuracy with interpretability in applications where trust, fairness, and accountability are critical.

From a practical perspective, we recommend that developers and platform operators prioritize explainability when selecting models for content moderation tasks. Incorporating visual explanation tools—such as SHAP plots or LRP-based heatmaps—can support moderation teams in understanding and validating AI-driven decisions. Additionally, the early integration of user-centered evaluation into the model development process ensures that explanations are accessible and actionable for non-expert users. To maintain fairness and performance over time, toxicity detection systems should also incorporate continuous feedback mechanisms from end-users and moderators.

By aligning model performance with transparency and usability—understood here as a broader concept encompassing explainability and interpretability—this work contributes to the development of AI systems that are not only effective but also trusted and responsible. The usability evaluation in this study was based solely

on visual inspection of model explanations, without direct input from end users. As usability often depends on users' subjective perceptions and experiences, future work could benefit from incorporating structured user feedback to better assess how explanation visualizations support moderation tasks in practice.

Acknowledgments

This research was funded by the Slovak Research and Development Agency under the contract No. APVV-22-0414 and by the Scientific Grant Agency of the Ministry of Education, Research, Development and Youth of the Slovak Republic and the Slovak Academy of Sciences under grant No. 1/0259/24.

References

Anand, M., & Eswari, R. (2019). *Classification of Abusive Comments in Social Media using Deep Learning*. IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8819734

Ansar, W., Goswami, S., & Chakrabarti, A. (2024). A survey on transformers in NLP with focus on efficiency. *arXiv:2406.16893*. https://doi.org/10.48550/arXiv.2406.16893

Atesli, H. (2023). *Explainable AI With SHAP*. Medium. https://medium.com/hepsiburada-data-science/explainable-ai-with-shap-6f629dfa6eef

Awati, R., & Yasar, K. *What is black box AI?* TechTarget. https://www.techtarget.com/whatis/definition/black-box-AI

DiPietro, R., & Hager, G. D. (2020). Deep learning: RNNs and LSTM. In *Handbook of medical image computing and computer assisted intervention* (pp. 503-519). Academic Press

Grine, A. E. (2023). *Toxic Comment Classification*. Medium. https://medium.com/@alaeddine.grine/toxic-comment-classification-317628632336

Lohaj, O., Paralič, J., & Buinytska, R. (2024). Exploring usability in combining data sources for detecting toxic behavior in social media. In *Central European Conference on Information and Intelligent Systems* (pp. 1-8). Faculty of Organization and Informatics Varazdin.

Maslej-Krešňáková, V., Sarnovský, M., Butka, P., & Machová, K. (2020). *Comparison of Deep Learning Models and Various Text Pre-Processing Techniques for the Toxic Comments Classification*. MDPI. https://www.mdpi.com/2076-3417/10/23/8631#B21-applsci-10-08631

Nguyen, H. T. T., Cao, H., Nguyen, V. T. K., & Pham, D. K. N. (2021). *Evaluation of Explainable Artificial Intelligence: SHAP, LIME, and CAM*. ResearchGate. https://www.researchgate.net/publication/362165633_Evaluation_of_Explainable_Artificial_Intelligence_SHAP_LIME_and_CAM

O'shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Praveen. (2021, March 1). *Overview of Explainable AI and Layer-wise relevance propagation (LRP)*. Medium. https://praveenkumar2909.medium.com/overview-of-explainable-ai-and-layer-wise-relevance-propagation-lrp-cb2d008fec57

Ryo, M. (2022). *Explainable artificial intelligence and interpretable machine learning for agricultural data analysis*. ScienceDirect. https://www.sciencedirect.com/science/article/pii/S2589721722000216

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019, October). How to fine-tune bert for text classification?. In *China national conference on Chinese computational linguistics* (pp. 194-206). Cham: Springer International Publishing.

Wu, T., Wang, Y., & Quach, N. (2025). Advancements in natural language processing: Exploring transformer-based architectures for text understanding. In *2025 5th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA)* (pp. 1384-1388). IEEE.

Zhou, N., & Gholizadeh, S. (2021). *Model Explainability in Deep Learning Based Natural Language Processing*. arXiv. https://arxiv.org/abs/2106.07410