AI-BASED IDENTIFICATION AND CORRECTION OF INAPPROPRIATE LANGUAGE

DR. V. MUNEESWARAN
Department of ECE
Kalasalingam Academy of Research and Education
Anand Nagar, Krishnankoil-626126, Tamilnadu, India.
munees.klu@gmail.com

DR. P. NAGARAJ
Department of CSE
Kalasalingam Academy of Research and Education
Anand Nagar, Krishnankoil-626126, Tamilnadu, India.
p.nagaraj@klu.ac.in

MANORANJAN KUMAR
Nanochipskills (Venkylabs)
Bengaluru-560001,
Karnataka, India.
maanuec056@gmail.com

ALLAKA HARI SANKAR
Department of ECE
Kalasalingam Academy of Research and Education
Anand Nagar, Krishnankoil-626126, Tamilnadu, India.
9920005003@klu.ac.in

KODE VIVEK SRINIVAS
Department of ECE
Kalasalingam Academy of Research and Education
Anand Nagar, Krishnankoil-626126, Tamilnadu, India.
9920005038@klu.ac.in

SUNDU LEELA KRISHNA
Department of ECE
Kalasalingam Academy of Research and Education
Anand Nagar, Krishnankoil-626126, Tamilnadu, India.
9920005058@klu.ac.in

MD.GHOUSEE ALI SIDDIQ
Department of ECE
Kalasalingam Academy of Research and Education
Anand Nagar, Krishnankoil-626126, Tamilnadu, India.
9920005182@klu.ac.in

***Abstract*—This study investigates the effectiveness of four machine learning algorithms—Decision Trees, Random Forest, Long Short-Term Memory (LSTM), and BERT—in detecting offensive language and hate speech in digital communication. While Decision Trees and Random Forest rely on structured features, LSTM and BERT represent advanced deep learning methods for handling unstructured text data. The study evaluates these models based on accuracy, precision, recall, and F1-score, shedding light on their ability to identify inappropriate content. The research contributes to the development of AI-based content filtering systems, aiding online platforms in creating safer and more respectful digital spaces by automatically flagging offensive text, thus promoting inclusivity and responsible online discourse.**

Keywords—Random Forest, Decision Tree, LSTM, and BERT

I. INTRODUCTION

Artificial Intelligence (AI) is like a smart robot that can do cool things with words. One of its superpowers is finding bad and rude language in stuff people write. In this paper, we talk about why it's a big deal to use AI to spot naughty words in academic papers before they get published. Sometimes, people write mean or harmful stuff in their papers, and that's not good for smart discussions in science and research. AI can help us find these bad words and make academic writing better.In this paper, we'll explain how AI does this magic. We'll also talk about why AI is so helpful, like making it faster to check papers and avoiding human mistakes.

But we also need to be careful with AI because it might block good words by mistake.

So, we'll learn about how AI can make academic writing nicer, but we also need to use it wisely to not stop people from saying what they want to say. It's a bit like having a helpful robot, but we need to teach it the right manners!so we can understand how people will misuse the social media comment sections and abuse language manners

In conclusion, the application of AI-based identification of inappropriate language holds great potential in enhancing the quality and professionalism of paper publications. However, it must be approached with careful consideration of ethical and practical implications. This paper seeks to provide insights into the possibilities, challenges, and ethical guidelines associated with employing AI for content moderation in the academic publishing industry.

II. METHODOLOGY

A. LSTM: LSTM, a special type of neural network, helps solve a common problem in computers. It was created by Sepp Hochreiter and Jürgen Schmidhuber in 1997 and is very important in making computers understand things like language, speech, and time-related data.LSTMs remember and use important information in sequences. They have a "memory cell" to store information, a "current thought" state, and "gates" that control the information flow.Because of this, LSTMs are excellent at keeping track of information over long sequences, which is useful for translating languages, understanding feelings in text, and recognizing speech

B. Random Forest: A Random Forest is like a team of clever decision trees working together to make better predictions. It's great at solving problems, and each tree helps by looking at different parts of the data. This makes it more accurate and reliable than a single decision tree.

- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyperparameter tuning.
- It solves the issue of overfitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

In a real-life situation, think of it like trying to figure out if a customer will buy a phone or not based on stuff like how much it costs, how much storage it has, and its RAM. With a single decision tree, it's like making a prediction based on just one set of data. But with a Random Forest, it's like having many different decision trees, each looking at different examples, making the prediction stronger and more reliable, especially when dealing with identifying abusive language

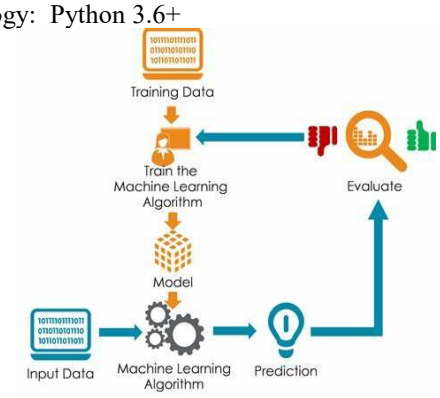
C. BERT: BERT, a groundbreaking NLP algorithm from Google AI in 2018, revolutionized the field with these key aspects:
1. Bidirectional Context: BERT grasps word context in both directions (before and after), enabling deeper understanding of language.
2. Two-Step Process: BERT first learns from a vast text dataset by predicting masked words, and then fine-tunes for specific tasks, consistently achieving top-tier performance.
3. Transformer Architecture: Built on transformers, BERT efficiently captures long-range word relationships through self-attention mechanisms.
4. Multilingual Adaptability: BERT is adaptable to various languages, making it versatile for multilingual NLP tasks.
BERT significantly enhances NLP model accuracy across applications like sentiment analysis, translation, chatbots, and search engines, thanks to its exceptional contextual comprehension.

D. Decision Tree: Decision Trees are a versatile machine learning method with these key attributes:

1. Hierarchical Structure: They create a tree-like structure with nodes representing decisions and branches showing outcomes.
2. Splitting Criteria: Data is divided into subsets based on informative features, aiming for purity optimization using metrics like Gini impurity or entropy (for classification) and mean squared error (for regression).
3. Leaf Nodes: Splits continue until reaching stopping criteria, like depth or sample size, with final leaf nodes containing class predictions (classification) or numerical values (regression).
4. Interpretability: Decision Trees are easily interpretable, allowing for if-else rule-based understanding of decisions.
5. Overfitting: Care must be taken to prevent overfitting by controlling tree depth or applying pruning techniques.
6. Ensemble Methods: Decision Trees serve as foundational components for ensemble methods like Random Forest and Gradient Boosting, enhancing prediction accuracy and mitigating overfitting.
7. Applications: Their simplicity and effectiveness make Decision Trees suitable for applications in finance, healthcare, marketing, and natural language processing, handling both categorical and numerical data.

II. Requirement Analysis

Functional Requirements are user-demanded features that must be an integral part of the system. They are expressed as user inputs, operations, and expected outputs. These are directly visible in the final product.
Examples:
1. User authentication during login.
2. System shutdown in response to a cyber-attack.
3. Sending a verification email to new users during registration.
Non-Functional Requirements are quality standards dictated by project contracts. These include factors like portability, security, maintainability, and performance. Their implementation priority varies among projects.
Examples:
1. Emails sent within a 12-hour latency.
2. Processing each request within 10 seconds.
3. Site loading in 3 seconds with over 10,000 simultaneous users.
Hardware Requirements
Processor- I7/Intel Processor
Hard Disk - 160GB
Key Board - Standard Windows Keyboard
Mouse - Two or Three Button Mouse
Monitor - SVGA
RAM - 8GB
Software Requirements:
Operating System: Windows 11
Server-side Script: HTML, CSS & JS
Programming Language: Python
Libraries: Django, Pandas, Numpy
IDE/Workbench: PyCharm
Technology: Python 3.6+



III. System Analysis

Existing System:

Approach: Currently, traditional machine learning techniques like Decision Trees and Random Forest are used for content moderation.

Challenges: Struggles with understanding nuanced, context-dependent offensive language, requires manual feature engineering, and may not scale effectively for real-time content moderation.

Limitations: Inadequate for multimodal data (text with images, audio, or video) and lacks model transparency for explaining and improving moderation decisions.

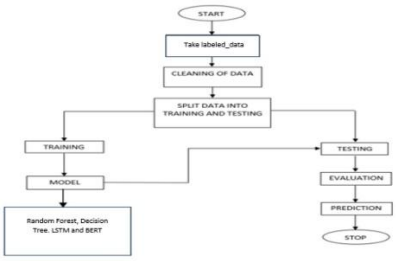
Proposed System:

Approach: The proposed system combines Decision Trees, Random Forest, and BERT, offering a comprehensive approach to identify and combat offensive content.

Benefits: Decision Trees provide transparency, Random Forest reduces overfitting, and BERT enhances contextual understanding for precise content filtering.

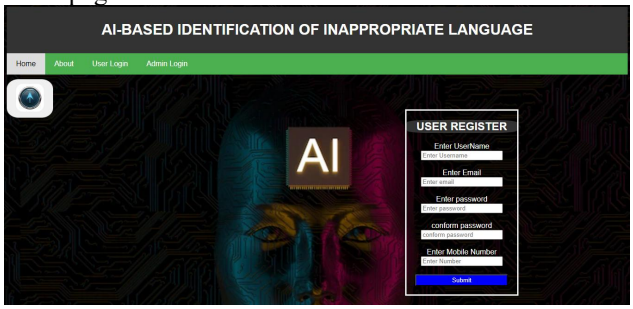
Insights: Evaluation metrics offer a holistic view of algorithm performance, contributing to ongoing improvements in content moderation.

Outcome: Aims to create safer, more inclusive digital communities by improving content moderation accuracy.

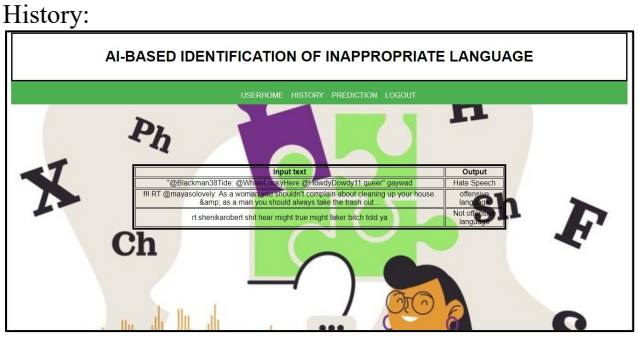


IV. OUTPUT

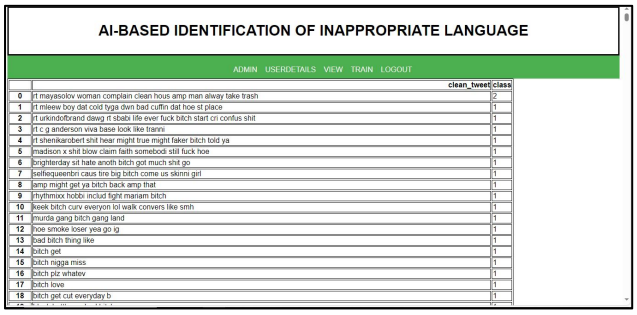
Homepage:



History:



View:



V. CONCLUSION

This study offers valuable insights into how four distinct machine learning algorithms – Decision Trees, Random Forest, LSTM, and BERT – perform in identifying offensive language in online content. Each algorithm has its strengths and weaknesses. Decision Trees and Random Forests emphasize the importance of structured feature engineering, while LSTM and BERT leverage deep learning methods for unstructured text data. Our evaluation, which includes accuracy, precision, recall, and F1-score, provides a comprehensive understanding of their prediction capabilities. These research findings are promising for the development of robust AI-powered content moderation systems, with the potential to foster more inclusive and respectful online environments, thereby promoting healthier and more responsible digital conversations for all users.

ACKNOWLEDGMENTS

We would like to extend our sincere gratitude to Leela Krishna, Hari Shankar, Vivek Srinivas, and Mohammed Siddiq, the dedicated team of volunteer annotators who generously contributed their time and expertise to assist us in creating the v1.0 dataset for detecting offensive language in English tweets across Instagram and YouTube comments. Their invaluable efforts also played a crucial role in meticulously reviewing tweets with ambivalent labels.

REFERENCES

[1] Pandian, A. Pasumpon. "Performance Evaluation and Comparison using Deep Learning Techniques in Sentiment Analysis." Journal of Soft Computing Paradigm (JSCP) 3, no. 02 (2021): 123-134.

[2] Manoharan, J. Samuel. "Study of Variants of Extreme Learning Machine (ELM) Brands and its Performance Measure on Classification Algorithm." Journal of Soft Computing Paradigm (JSCP) 3, no. 02 (2021): 83-95.

[1] [3] Ranganathan, G. "A Study to Find Facts Behind Preprocessing on Deep Learning Algorithms." Journal of Innovative Image Processing (JIIP) 3, no. 01 (2021): 66-74.

[2] [4] Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, "Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TFIDF based Approach," 2019.

[3] [5] Akhter, M. P., Jiangbin, Z., Naqvi, I. R., Abdelmajeed, M., Mehmood, A., & Sadiq, M. T. (2020). Document-level text classification using single-layer multisize filters convolutional neural network. IEEE Access, 8, 42689-42707.

[4] [6] K. J. Madukwe and X. Gao, "The Thin Line Between Hate and Profanity," in Australasian Joint Conference on Artificial Intelligence, 2019, pp. 344–356.

[5] [7] Beeravolu, A. R., Azam, S., Jonkman, M., Shanmugam, B., Kannoorpatti, K., & Anwar, A. (2021). Preprocessing of Breast Cancer Images to Create Datasets for Deep-CNN. IEEE Access, 9, 33438-33463.

[6] [8] Chen, Z., Zhou, L. J., Da Li, X., Zhang, J. N., & Huo, W. J. (2020). The Lao text classification method is based on KNN. Procedia Computer Science, 166, 523-528.

[7] [9] Diker, A., Avci, E., Tanyildizi, E., & Gedikpinar, M. (2020). A novel ECG signal classification method using DEA-ELM. Medical hypotheses, 136, 109515.

[8] [10] Heidari, M., Mirniaharikandehei, S., Khuzani, A. Z., Danala, G., Qiu, Y., & Zheng, B. (2020). Improving the performance of CNN to predict the likelihood of COVID19 using chest X-ray images with preprocessing algorithms. International journal of medical informatics, 144, 104284.

[9] [11] Poloni, K. M., de Oliveira, I. A. D., Tam, R., Ferrari, R. J., & Alzheimer's Disease Neuroimaging Initiative. (2021). Brain MR image classification for Alzheimer's disease diagnosis using structural hippocampal asymmetrical. attributes from directional 3-D logGabor filter responses. Neurocomputing, 419, 126-135.

[10] [12] Rodrigues, L. F., Naldi, M. C., & Mari, J. F. (2020). Comparing convolutional neural networks and preprocessing techniques for HEp-2 cell classification in immunofluorescence images. Computers in biology and medicine, 116, 103542.