iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.61699

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Tone Tracker

Mrs. E. Sharmila[1], K. Dhivya[2], C. Durga Devi[3], S. Guru Priya[4]
Idhaya College of Arts and Science for Women, India

Abstract: ***Tone Tracker's integration of BERT and LSTM technologies allows it to predict and flag offensive language in both Tamil and English social media comments. BERT, a transformer-based model, enables the tool to understand the semantic context of text in multiple languages, ensuring accurate detection of inappropriate content regardless of language. The incorporation of LSTM further enhances this capability by capturing nuanced contextual information, refining the tool's proficiency in content moderation for both Tamil and English content. With its user-friendly features, Tone Tracker becomes accessible to a diverse user base, empowering them to swiftly remove offensive content in both languages and contribute to fostering a secure digital environment. This groundbreaking innovation not only boosts content moderation efficiency but also ensures scalability across various digital platforms, making it adaptable to the linguistic diversity of online communities. Ultimately, Tone Tracker, powered by LSTM and BERT, plays a pivotal role in cultivating positive online spaces where users can engage confidently and respectfully in both Tamil and English.***

Keywords: LSTM, BERT, Tone Tracker.

I. INTRODUCTION

The evolution of Information and Communications Technology (ICT) has undeniably facilitated global communication and accessibility among online communities. While this progress has connected people across the globe, it has also given rise to challenges, particularly concerning the prevalence of false identities and the cloak of anonymity on online platforms. This freedom often allows individuals to express their thoughts and comments without constraints, leading to the widespread dissemination of aggressive behavior and hate speech. Major Social Media Platforms (SMPs) like Facebook, Twitter, and Internet forums have become breeding grounds for cyber threats and vulnerabilities, adversely affecting users' mental health. The anonymity associated with online interactions creates an environment where online abusive behavior and hate speech can flourish, potentially leading to severe consequences, including criminal activities and, in extreme cases, suicide.

In addressing the escalating concerns surrounding online behavior, it is crucial to recognize the prevalence and impact of profanity in contemporary conversations, both in informal settings and on social media platforms. Profanity, including cursing and swearing, has become commonplace, contributing to an atmosphere of offensive, aggressive, and hateful language. Distinguishing between hate speech and offensive speech is essential, as highlighted in a referenced study [3]. Hate speech is characterized by language expressing hatred towards a specific person or group based on attributes such as religion, gender, race, sexual orientation, or disability. It aims to humiliate or insult the target, while offensive speech is described as language with the intent to hurt the recipient's feelings but lacks the specific focus on key characteristics. Understanding these distinctions is critical in developing strategies to combat the negative impact of such language on individuals and society. To effectively mitigate the adverse effects of online abusive behavior and hate speech, there is a pressing need for comprehensive cybersecurity measures and content moderation strategies on SMPs. Additionally, fostering digital literacy and promoting responsible online behavior can contribute to creating a safer and more positive digital environment. The collaborative efforts of technology companies, policymakers, and users are essential to addressing these challenges and building a more respectful and secure online community.

II. RELATED WORK

1) DETECTING HATE TWEETS — TWITTER SENTIMENT ANALYSIS Hate speech detection has substantially increased interest among researchers in the domain of natural language processing (NLP) and text mining. The number of studies on this topic has been growing dramatically. Thus, the purpose of this analysis is to develop a resource that consists of an outline of the approaches, methods, and techniques employed to address the issue of Twitter hate speech. This study can be used to aid researchers in the development of a more effective model for future studies. This review focused on studies published over the past eight years, i.e., from 2015 to 2022. This systematic search was carried out in December 2020 and updated in July 2022. Ninety-one articles published within the mentioned period met the set criteria and were selected for this review. From the evaluation of these works, it is clear that a perfect solution has yet to be found. To conclude, this paper focused on presenting an in-depth understanding of current perspectives and highlighted research opportunities to boost the quality of hate speech

detection systems. In turn, this helps social networking services that seek to detect hate messages generated by users before they are posted, thus reducing the risk of targeted harassment.

2) Improving Hate Speech Detection of Urdu Tweets using Sentiment Analysis Sentiment Analysis is a technique that is being used abundantly nowadays for customer reviews analysis, popularity analysis of electoral candidates, hate speech detection and similar applications. Sentiment analysis on tweets encounters challenges such as highly skewed classes, high dimensional feature vectors and highly sparse data. In this study, we have analyzed the improvement achieved by successively addressing these problems in order to determine their severity for sentiment analysis of tweets. Firstly, we prepared a comprehensive data set consisting of Urdu Tweets for sentiment analysis-based hate speech detection. To improve the performance of the sentiment classifier, we employed dynamic stop words filtering, Variable Global Feature Selection Scheme (VGFSS) and Synthetic Minority Optimization Technique (SMOTE) to handle the sparsity, dimensionality and class imbalance problems respectively. We used two machine learning algorithms i.e., Support Vector Machines (SVM) and Multinomial Naïve Bayes' (MNB) for investigating performance in our experiments. Our results show that addressing class skew along with alleviating the high dimensionality problem brings about the maximum improvement in the overall performance of the sentiment analysis-based hate speech detection.

3) IMPROVING HATE SPEECH DETECTION OF URDU TWEETS USING SENTIMENT ANALYSIS Sentiment Analysis is a technique that is being used abundantly nowadays for customer reviews analysis, popularity analysis of electoral candidates, hate speech detection and similar applications. Sentiment analysis on tweets encounters challenges such as highly skewed classes, high dimensional feature vectors and highly sparse data. In this study, we have analyzed the improvement achieved by successively addressing these problems in order to determine their severity for sentiment analysis of tweets. Firstly, we prepared a comprehensive data set consisting of Urdu Tweets for sentiment analysis-based hate speech detection. To improve the performance of the sentiment classifier, we employed dynamic stop words filtering, Variable Global Feature Selection Scheme (VGFSS) and Synthetic Minority Optimization Technique (SMOTE) to handle the sparsity, dimensionality and class imbalance problems respectively. We used two machine learning algorithms i.e., Support Vector Machines (SVM) and Multinomial Naïve Bayes' (MNB) for investigating performance in our experiments. Our results show that addressing class skew along with alleviating the high dimensionality problem brings about the maximum improvement in the overall performance of the sentiment analysis-based hate speech detection.

4) Detecting twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques The COVID-19 pandemic has impacted every nation, and social isolation is the major protective method for the coronavirus. People express themselves via Facebook and Twitter. People disseminate disinformation and hate speech on Twitter. This research seeks to detect hate speech using machine learning and ensemble learning techniques during COVID-19. Twitter data was extracted from using its API with the help of trending hashtags during the COVID-19 pandemic. Tweets were manually annotated into two categories based on different factors. Features are extracted using TF/IDF, Bag of Words and Tweet Length. The study found the Decision Tree classifier to be effective. Compared to other typical ML classifiers, it has 98% precision, 97% recall, 97% F1-Score, and 97% accuracy. The Stochastic Gradient Boosting classifier outperforms all others with 99 percent precision, 97 percent recall, 98 percent F1-Score, and 98.04 percent accuracy.

5) Demographical Based Sentiment Analysis for Detection of Hate Speech Tweets for Low Resource Language Advancement in IT and communication technology provides the opportunity for social media users to communicate their ideas and thoughts across the globe within no time as well big data promulgated in a result of the communication process itself has immense challenges. Recently, the provision of freedom of speech has witnessed immense promulgation of ofensive and hate speech content on the internet aimed the basic human rights violation. The detection of abusive content on social media for rich resource language has become a hot area for researchers in the recent past. However, low-resource languages are underprivileged due to the non-availability of large corpus and its complexity to understand. The proposed methodology mainly has two parts. One is to detect abusive content and the other is to have a demographical analysis of the Indigenously developed dataset. The process starts with the development of a unique unlabeled Urdu dataset of 0.2 M from Twitter through a web scrapper tool named snscraper. The dataset is collected against the 36 districts of Punjab from Pakistan and from the duration 2018- Apr 2022. The dataset is labeled into three target classes Neutral, Ofensive, and Hate Speech. After data cleaning, the feature extraction process is achieved with the help of traditional techniques such as Bow and tf-idf with the combination of word and char n-gram and word embedding word2Vec. The dataset is trained on both machine learning algorithms SVM and Logistic regression and deep learning techniques Long Short Term Memory (LSTM) and Convolutional Neural Networks (CNN). The best F score achieved through LSTM on this dataset is 64 and accuracy is 93 th/rough CNN.

III. PROPOSED METHODOLOGY

Tone Tracker, is a cutting-edge tool poised to revolutionize online community management. It combines advanced AI technologies, including BERT and LSTM, to autonomously detect and flag offensive language in social media comments. BERT, renowned for its proficiency in understanding contextual nuances, enables Tone Tracker to comprehend the semantic meaning of comments in both Tamil and English. Meanwhile, LSTM enhances the system's ability to capture long-term dependencies within the text, refining its proficiency in content moderation. With unparalleled accuracy, Tone Tracker swiftly identifies inappropriate content, fostering a secure digital environment. Its user-friendly features ensure accessibility to a diverse user base, empowering them to promptly remove offensive material. Moreover, the system's scalability makes it adaptable across various digital platforms, ensuring efficient content moderation efforts. In essence, Tone Tracker represents a groundbreaking innovation that promotes responsible digital interactions, cultivating positive online spaces where users can engage confidently and respectfully.

A. Data Collection

Twitter data collected from Kaggle open source refers to publicly available datasets on the Kaggle platform that contain information extracted from Twitter. These datasets typically encompass a wide range of Twitter content, including tweets, user profiles, hashtags, and metadata. Researchers and data enthusiasts often utilize these datasets for various purposes, such as sentiment analysis, trend detection, and social network analysis. The availability of Twitter data on Kaggle enables users to access valuable insights into online conversations, behaviors, and trends within the Twitter platform. By leveraging these datasets, analysts can gain a deeper understanding of public opinions, sentiments, and interactions on Twitter, contributing to research in fields like data science, social sciences, and marketing. Additionally, the open nature of Kaggle allows for collaboration and knowledge sharing among data professionals, fostering a vibrant community of data enthusiasts working on diverse projects related to Twitter data analysis.
https://www.kaggle.com/datasets/saurabhshahane/cyberbullying-dataset

B. Pre-Processing

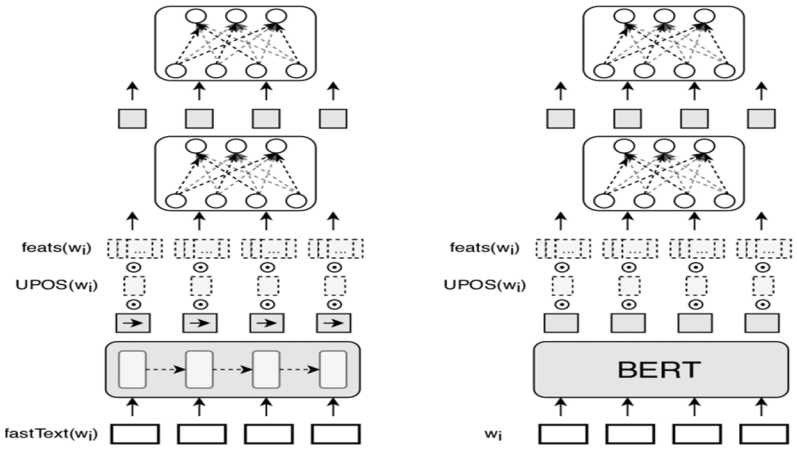
Pre-processing a Twitter dataset involves a series of steps to refine and prepare the data for analysis. Initially, duplicates, irrelevant columns, and entries with missing values are removed to ensure data consistency. Textual data undergoes several cleaning procedures, including the removal of special characters, tokenization to split text into words, and converting all text to lowercase for uniformity.
Additionally, common stopwords and variations in word forms are addressed through techniques like stemming or lemmatization. Mentions, hashtags, and URLs are extracted or eliminated, depending on their relevance to the analysis. Emoticons and emojis may be converted to text or removed to streamline the data. Text encoding techniques are then applied to represent textual data in a numerical format suitable for machine learning algorithms. Data splitting is performed to create training, validation, and testing subsets. Class balancing techniques may be applied if the dataset exhibits class imbalance. Finally, feature engineering may involve creating additional features from the text data to enhance model performance. Through these pre-processing steps, the Twitter dataset is refined and structured to facilitate accurate analysis and modeling, enabling meaningful insights to be derived from the data.

C. Feature Extraction

Feature extraction is a pivotal process in preparing Twitter data for analysis, involving the conversion of raw text into numerical representations understandable by machine learning algorithms. In the realm of Twitter datasets, feature extraction encompasses several techniques tailored to capture the essence of tweets and their contexts. Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) methods quantify word occurrences and importance, respectively, across the entire corpus of tweets. Word embeddings, such as Word2Vec or GloVe, encode semantic relationships between words into dense vector representations, fostering contextual understanding. N-grams capture sequential information by representing adjacent word sequences, while topic modeling techniques like Latent Dirichlet Allocation (LDA) unveil latent topics within the tweet corpus. Sentiment analysis features, syntax-based attributes, and user-based metrics further enrich the feature space, offering insights into sentiment, linguistic structures, and user behaviors. Through these extraction techniques, raw Twitter text undergoes transformation into structured numerical features, enabling subsequent analysis for tasks like sentiment analysis, topic modeling, and user profiling with enhanced accuracy and depth.

D. Model Creation

Creating a model that combines LSTM (Long Short-Term Memory) and BERT (Bidirectional Encoder Representations from Transformers) involves integrating the strengths of both architectures to effectively process and understand textual data. BERT, a pre-trained model, is utilized to encode input text into contextualized representations, capturing semantic meaning and context. Its bidirectional nature ensures comprehensive understanding by considering both preceding and succeeding words. LSTM, known for capturing long-term dependencies in sequential data, further refines contextual understanding. The model architecture typically incorporates BERT's encoding layers with additional LSTM layers for sequential processing. During training, both BERT and LSTM layers are fine-tuned simultaneously to adapt to specific tasks, such as sentiment analysis or text classification.



Evaluation involves assessing the model's performance on a separate dataset, adjusting hyperparameters and architecture as needed to optimize accuracy. By combining the capabilities of LSTM and BERT, the resulting model excels in understanding and processing textual data with high accuracy and contextual comprehension, making it suitable for a wide range of natural language processing tasks.

E. Embedding Dimension (D)

The embedding dimension in the context of neural networks, including models like BERT, refers to the size of the vector space in which words or tokens are represented. It is essentially the number of dimensions in the embedding space where words are mapped. Let's denote the embedding dimension as d.

The formula for the embedding dimension can be straightforwardly expressed as:

D =Number of Dimensions in Embedding Space

This dimensionality is a hyperparameter set during the training of the model. It determines the size of the vector used to represent each word or token in the input sequence. Larger embedding dimensions may capture more nuanced semantic relationships between words, but they also come with increased computational complexity and memory requirements.

F. Number of Attention Heads (H)

The number of attention heads ***(H)*** in a model like BERT refers to the parallel attention mechanisms that operate independently but in parallel. In BERT, each attention head allows the model to focus on different parts of the input sequence, capturing different aspects of the relationships between words. Let's denote the number of attention heads as ***H***.

The formula for calculating the total number of parameters in the attention mechanism, including the number of attention heads, is:

$$\text{Total Attention Parameters} = L \times H \times d_{\text{hidden}}^2$$

Here:

- L is the number of layers in the model.
- H is the number of attention heads.
- d_{hidden} is the dimensionality of the hidden units.

In the formula, L×H represents the total number of attention heads across all layers, and d^2_{hidden} accounts for the parameters associated with the self-attention mechanism within each head.

G. Number of Layers (L)

The number of layers (L) in a neural network, including models like BERT, refers to the depth of the architecture. In the context of BERT, it represents the number of times the input sequence undergoes transformations through self-attention mechanisms and feedforward neural network layers. Each layer refines the representation of the input sequence. Let's denote the number of layers as L.

The formula for calculating the total number of parameters in BERT, taking into account the number of layers, is:

$$\text{Total Parameters} = L \times (H \times d_{\text{hidden}}^2 + 4 \times d_{\text{hidden}})$$

Here:

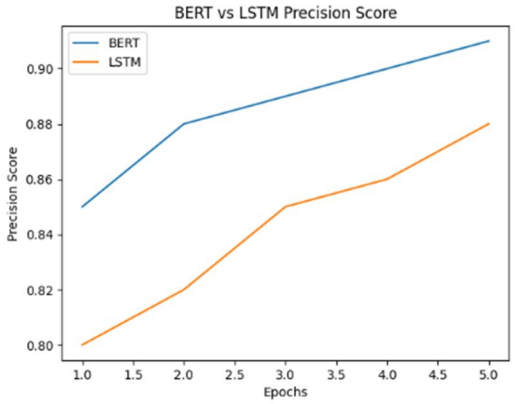
- L is the number of layers.
- H is the number of attention heads.
- d_{hidden} is the dimensionality of the hidden units.

The term H×dhidden2 accounts for the parameters associated with the self-attention mechanism within each attention head, and 4×*d*hidden represents the parameters in the feedforward neural network layers.

IV. RESULT AND DISCUSSION

A. Precision

Precision is a crucial metric in the evaluation of classification models, providing insights into the accuracy of positive predictions made by the model. It is defined as the ratio of correctly



predicted positive observations (True Positives) to the total instances predicted as positive, including both correct predictions and false positives. The precision metric is particularly valuable in scenarios where the consequences of false positives are significant. For instance, in a medical diagnosis context, precision would indicate the percentage of correctly identified positive cases among all instances predicted as positive.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

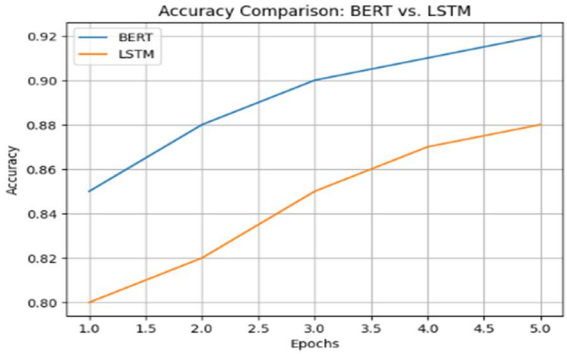
A higher precision score implies a lower rate of false positives, highlighting the model's effectiveness in accurately identifying positive instances. Precision is often considered in conjunction with other metrics, such as recall and F1 score, to provide a comprehensive evaluation of a classifier's performance.

B. Accuracy

Accuracy is a key metric used to assess the overall correctness of a classification model. It provides a straightforward measure of the model's ability to make correct predictions across all classes. The accuracy metric is calculated as the ratio of the sum of true positive and true negative predictions to the total number of observations. In other words, it gauges the proportion of correctly predicted instances—both positive and negative—out of the entire dataset.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Observations}}$$

A high accuracy score indicates that the model has successfully classified a large portion of instances correctly.



However, accuracy might not be the sole determinant of a model's performance, especially in imbalanced datasets where one class dominates.

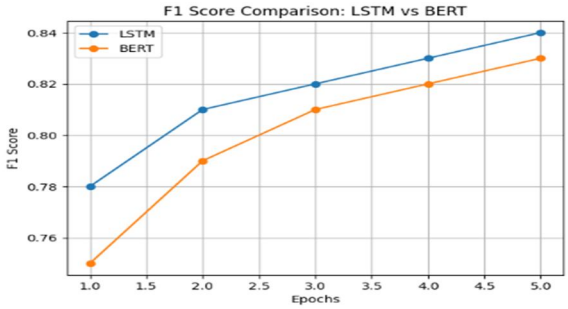
C. F1 Score

The F1 score serves as a comprehensive metric that balances the trade-off between precision and recall, offering a holistic evaluation of a classification model's performance. Particularly beneficial in scenarios where there is an imbalance between the number of positive and negative samples in the dataset, the F1 score takes into account both false positives and false negatives. By calculating the harmonic mean of precision and recall, the F1 score provides a single metric that encapsulates the model's ability to make accurate positive predictions while minimizing both types of errors.

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

The harmonic mean inherently penalizes extreme values, making it especially useful when seeking a balanced performance in scenarios where precision and recall may have conflicting priorities.



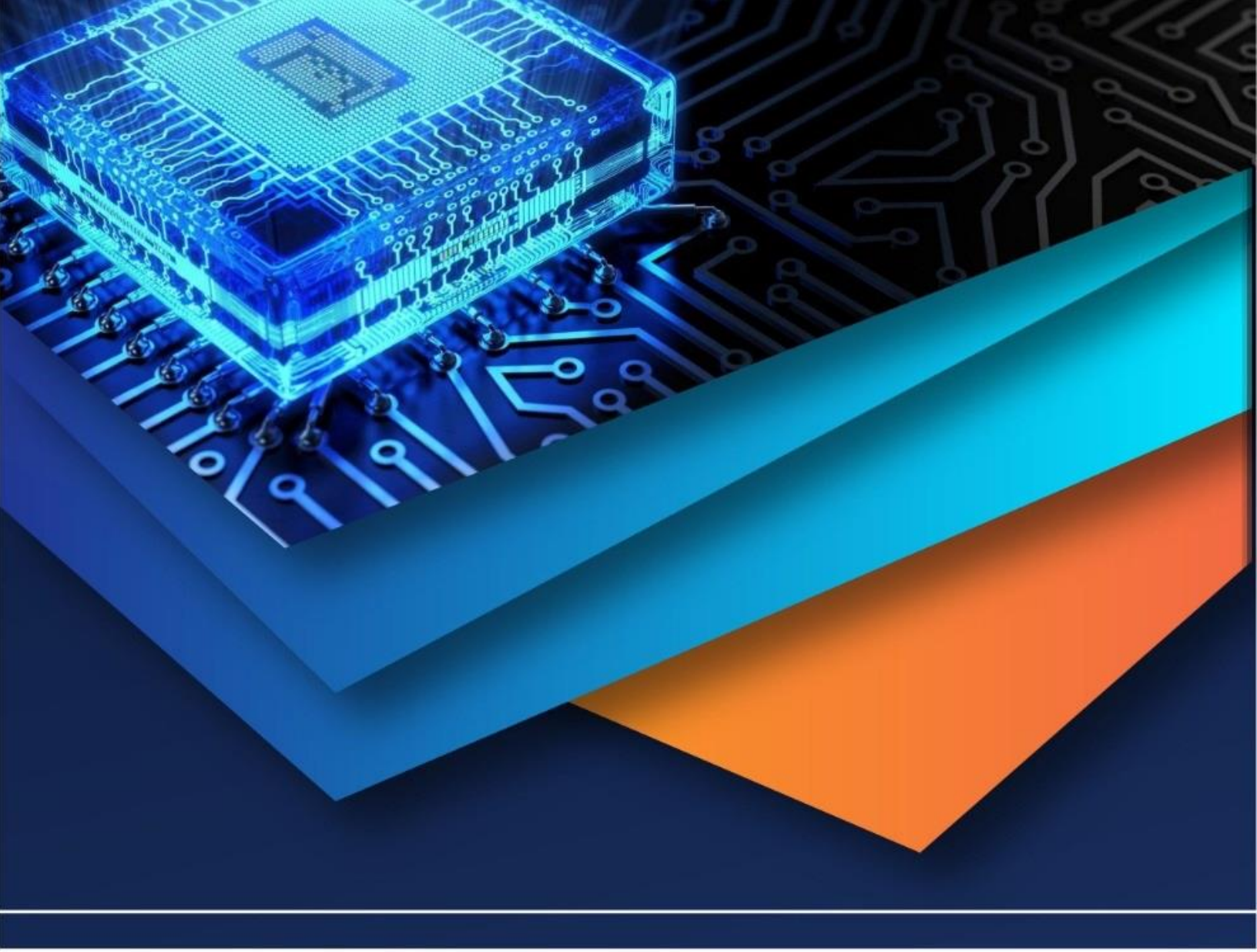
A higher F1 score signifies a better compromise between precision and recall, making it a valuable metric for assessing classification models in situations with imbalanced class distributions.

V. CONCLUSION

Tone Tracker epitomizes a groundbreaking solution for online community management by amalgamating BERT and LSTM technologies. This innovative tool autonomously identifies and flags offensive language in social media comments, driven by BERT's contextual comprehension and LSTM's capacity to capture long-term dependencies. By seamlessly understanding both Tamil and English comments, Tone Tracker ensures an inclusive digital environment. Its precision swiftly detects inappropriate content, fostering user confidence and safety. With user-friendly features, it enables swift removal of offensive material, empowering diverse users to contribute to a respectful online space. Moreover, Tone Tracker's scalability ensures effective moderation across diverse digital platforms, amplifying its impact. In essence, it revolutionizes digital interactions, emphasizing responsibility and respect. As Tone Tracker continues to evolve, it underscores the vital role of advanced AI in nurturing positive online communities, shaping a safer and more inclusive digital landscape for all. Future work may involve enhancing multilingual support, refining model accuracy, and exploring real-time moderation capabilities.

REFERENCES

[1] Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep learning for hate speech detection in tweets. In Proceedings of the 26th International Conference on World Wide Web Companion (pp. 759–760). International World Wide Web Conferences Steering Committee.
[2] Barnaghi, P., Ghaffari, P., & Breslin, J. G. (2016). Opinion mining and sentiment polarity on twitter and correlation between events and sentiment. In 2nd IEEE International Conference on Big Data Computing Service and Applications (BigDataService) (pp. 52–57).
[3] BBC (2016). Facebook, Google and Twitter agree german hate speech deal. Website. http://www.bbc.com/news/world-europe-35105003 Accessed: on 26/11/2016.
[4] Chen, Y., Zhou, Y., Zhu, S., & Xu, H. (2012). Detecting offensive language in social media to protect adolescent online safety. In 2012 International Conference on Privacy, Security, Risk and Trust (PASSAT 2012), and 2012 International Confernece on Social Computing (SocialCom 2012)
[5] DailyMail (2016). Zuckerberg in Germany: No place for hate speech on Facebook. Website. http://www.dailymail.co.uk/wires/ap/ article-3465562/Zuckerberg-no-place-hate-speech-Facebook. html Accessed: on 26/02/2016.
[6] Davidson, T., Warmsley, D., Macy, M. W., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In Proceedings of the 11th International Conference on Web and Social Media (ICWSM 2017) (pp. 512–515).
[7] Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., & Bhamidipati, N. (2015). Hate speech detection with comment embeddings. In Proceedings of the 24th International Conference on World Wide Web Companion (pp. 29–30). ACM.
[8] Elman, J. (1990). Finding structure in time. Cognitive Science, 14 , 179– 211.
[9] Gamb¨ack, B., & Sikdar, U. K. (2017). Using convolutional neural networks to classify hate-speech. In Proceedings of the 1st Workshop on Abusive Language Online at ACL 2017 .
[10] Gandhi, I., & Pandey, M. (2015). Hybrid ensemble of classifiers using voting. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 399–404). doi:10.1109/ICGCIoT.2015.7380496.
[11] Jha, A., & Mamidi, R. (2017). When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In Proceedings of the Second Workshop on NLP and Computational Social Science (pp. 7–16). Association for Computational Linguistics.
[12] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 , .
[13] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning.
[14] Park, J. H., & Fung, P. (2017). One-step and two-step classification for abusive language detection on twitter. In Proceedings of the 1st Workshop on Abusive Language Online at ACL 2017.
[15] Saha, S., & Ekbal, A. (2013). Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. Data & Knowledge Engineering, 85, 15



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY