Aggression Detection in Telugu and English Social Media Text using Deep Learning and Ensemble Models

1st AKA Rahul
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, India
cb.en.u4aie22003@cb.students.amrita.edu

2nd Manikanta.E
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, India
cb.en.u4aie22015@cb.students.amrita.edu

3rdK.Koushik
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, India
cb.en.u4aie22024@cb.students.amrita.edu

4th E Sahitya Naidu
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, India
cb.en.u4aie22062@cb.students.amrita.edu

5th Dr. Sachin Kumar S.
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, India
sachinkumar@cb.amrita.edu

***Abstract*—This research presents a multilingual aggression detection system focused on identifying aggressive content in Telugu and English social media text. The proposed framework integrates deep learning architectures with traditional machine learning approaches to leverage a diverse set of linguistic and statistical features. For English, the system employs FastText embeddings, reduced TF-IDF vectors (via SVD), and scaled numeric, lexical, and part-of-speech (POS) features. A feedforward neural network is used as the primary classification model for English aggression detection, achieving an accuracy of 58% and a macro F1-score of 0.57, outperforming the Random Forest baseline (accuracy: 53%, F1-score: 0.53).**

For Telugu, the system utilizes FastText embeddings, raw TF-IDF vectors, sentiment polarity scores, numeric indicators, lexical cues, and POS features. Various classifiers are explored for Telugu, including Logistic Regression, Random Forest, a custom-tuned Feedforward Neural Network (via Keras Tuner), and a BiLSTM model with an attention mechanism. The Feedforward Neural Network achieved the best performance, with an accuracy of 84% and a macro F1-score of 0.84, followed closely by the BiLSTM + Attention model (accuracy: 83%, F1-score: 0.83).

Hyperparameter tuning across models is facilitated using Keras Tuner with a random search strategy to optimize architecture and training configurations. Experimental results demonstrate the system's strong performance in aggression classification for both languages, with evaluations conducted using standard metrics including accuracy, precision, recall, and F1-score. This work contributes toward safer digital communication by enabling robust detection of aggressive language in low-resource and multilingual settings.

***Index Terms*—Aggression Detection, Telugu NLP, FastText, TF-IDF, POS Features, Sentiment Analysis, Deep Learning, Ensemble Models, Keras Tuner, Multilingual Text Classification**

I. Introduction

Aggression detection, a critical subfield of Natural Language Processing (NLP), plays a vital role in computationally identifying and categorizing hostile or offensive content in textual data. In the context of online discourse, aggression detection can serve as a valuable tool for mitigating cyberbullying, toxicity, and verbal abuse, thereby fostering safer digital spaces. However, the majority of aggression detection research and tools are limited to high-resource languages such as English, leaving a significant void in the analysis of under-resourced languages like Telugu—despite its widespread usage by over 80 million speakers.

Telugu poses several NLP challenges, including morphological richness, agglutination, and a lack of annotated corpora. Moreover, the absence of standardized tools for preprocessing and feature extraction further complicates aggression classification tasks in this language. This research addresses these challenges by developing a multilingual aggression detection pipeline tailored specifically for Telugu and English social media content, combining advanced embeddings with handcrafted linguistic features to enhance classification performance.

We employ FastText embeddings to capture subword-level semantic information that proves effective for morphologically complex languages. In addition, TF-IDF vectors are utilized to extract frequency-based statistical features. Handcrafted features are engineered from lexical, syntactic, and sentiment-based heuristics, including part-of-speech (POS) tag distributions, punctuation patterns, sentiment polarity scores (with positive and neutral classes merged), and various numeric indicators, enriching the feature space with linguistic insights beyond what embeddings can provide.

The classification task is addressed using both traditional and deep learning models. For Telugu, baseline classifiers include:

- Logistic Regression: a simple yet effective linear model for high-dimensional feature spaces,

Identify applicable funding agency here. If none, delete this.

- Random Forest: a robust ensemble method that handles diverse feature types efficiently,
- Feedforward Neural Network (FFNN): optimized using Keras Tuner to fine-tune architecture and parameters,
- BiLSTM with Attention: captures sequential dependencies and emphasizes contextually important tokens.

For English, we leverage a compact set of embeddings and features:

- FastText vectors and TF-IDF features reduced via Singular Value Decomposition (SVD) for dimensionality reduction,
- Scaled handcrafted features including POS tags, sentiment scores, and lexical features,
- A custom feedforward neural network trained using the Adam optimizer and tuned via Optuna using a random search strategy.

Evaluation is performed using standard metrics such as accuracy, precision, recall, F1-score, and confusion matrix. Results indicate that combining statistical, semantic, and linguistic features significantly improves aggression classification across both Telugu and English datasets.

This work contributes toward the development of scalable, explainable, and multilingual aggression detection systems capable of handling low-resource and morphologically complex languages. It provides a robust framework for real-world deployment in content moderation systems, especially for regional Indian languages that remain underrepresented in existing NLP tools and datasets.

II. Literature Review

Aggression detection has become a crucial area of research within Natural Language Processing (NLP), particularly for moderating user-generated content on social media platforms. Initial work in this domain relied heavily on rule-based systems and traditional machine learning algorithms such as Logistic Regression and Support Vector Machines, using bag-of-words or TF-IDF representations (1). While these approaches could identify explicit aggression, they often failed to account for context, sarcasm, or nuanced hostility in text, limiting their effectiveness.

With the evolution of deep learning, models such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) demonstrated improved performance in capturing sequential and local patterns in text (2; 3). Transformer-based architectures, especially BERT and its derivatives, further enhanced aggression detection by learning contextual embeddings that reflect the semantic and syntactic structure of language (4). However, these advancements have largely focused on English and other high-resource languages, creating a research gap for low-resource languages like Telugu.

Studies focused on aggression detection in Indian languages have primarily targeted Hindi, Tamil, and Bengali, with relatively little work addressing Telugu (5; 6). The challenges in Telugu stem from its morphological richness, agglutinative grammar, and lack of publicly available annotated datasets. To overcome these limitations, multilingual models like mBERT and MuRIL have been explored. MuRIL, in particular, has shown improved results on Indian language tasks due to its training on diverse Indian script corpora (7).

Recent approaches have also integrated handcrafted linguistic features—such as part-of-speech (POS) tag distributions, sentiment scores, and punctuation patterns—with contextual embeddings to improve classification in resource-constrained scenarios (8). These hybrid models offer enhanced robustness and interpretability. Although aggression detection has proven effective in platforms like Twitter and Reddit (9; 10), the adaptation of these models to regional languages like Telugu remains an open challenge.

Furthermore, there is a growing emphasis on the development of explainable and deployable aggression detection systems. In real-world content moderation, models must not only provide accurate predictions but also offer insight into the linguistic or semantic factors contributing to their decisions. This is especially critical in multilingual environments where cultural and contextual nuances vary widely. As a result, research is now shifting toward building models that combine performance with transparency and scalability for practical use across diverse linguistic communities.

III. Proposed Methodology

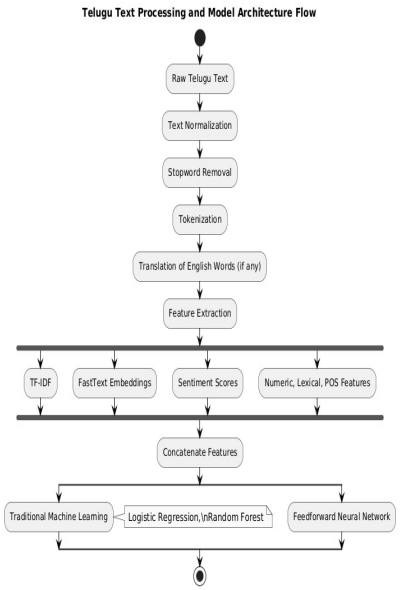

Fig. 1. Proposed Methodology Flow-Telugu

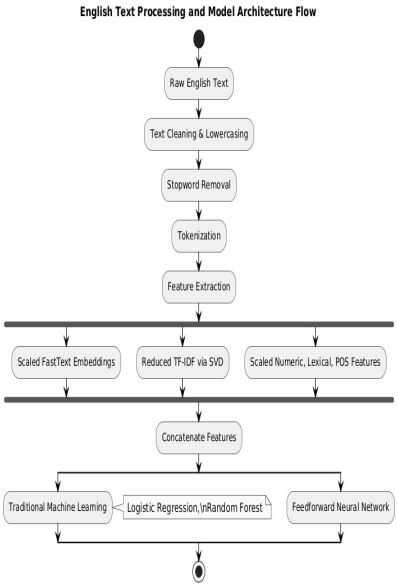

Fig. 2. Proposed Methodology Flow-English

A. Working Model

There are several platforms available for building and deploying machine learning models, especially for text classification tasks in multilingual settings. Common platforms include:

- Google Colab
- Kaggle Notebooks
- Jupyter Notebook
- PyCharm IDE

For this project, **Google Colab** was predominantly used for model development and experimentation.

Google Colab offers a cloud-based development environment with free access to **GPU/TPU acceleration**, making it well-suited for training deep learning models like feedforward neural networks. It supports seamless integration with Google Drive, simplifying dataset management and model storage. Additionally, it comes pre-configured with essential libraries such as `TensorFlow`, `scikit-learn`, `nltk`, and `pandas`, all of which were heavily utilized in this project.

B. Dataset Description

The datasets used in this research consist of textual statements in both Telugu and English, focusing on the detection of aggression in online discourse. The data was sourced and prepared to reflect the diversity of language use and expression styles in real-world digital communication.

The datasets were curated from two primary sources:

1) Social media posts, particularly from platforms like Twitter and YouTube, which provided real, user-generated content in both Telugu. These posts reflect a range of aggressive and non-aggressive expressions commonly found in online interactions.
2) For Telugu: additional sentences were manually translated from English to Telugu using context-aware translation tools and then validated by human annotators to ensure cultural and semantic alignment.
3) For English: publicly available annotated datasets on aggression and hate speech were further refined, cleaned, and harmonized to fit the objectives of this study.

Each English instance is labeled as one of three categories: *Non-Aggressive*, *Covert Aggression*, or *Overt Aggression*. The Telugu dataset includes two labels: *Non-Aggressive* and *Aggressive*, due to data distribution and linguistic constraints. These datasets serve as the foundation for training and evaluating multilingual aggression detection models, particularly focusing on low-resource language scenarios.

C. Data Preprocessing

1) Preprocessing and Balancing of English Dataset: The preprocessing pipeline for the English dataset included several key steps to clean and prepare the textual data for aggression classification:

1) **Text Cleaning**: All text was converted to lowercase and cleaned using regular expressions. This included the removal of URLs, user mentions (e.g., @username), punctuation, and extra whitespace, retaining only alphabetic characters and meaningful content.
2) **Stopword Removal**: Using the NLTK library, English stopwords were removed from each sentence. Tokenization was applied using `word_tokenize`, and all common English stopwords were excluded to retain only informative words.
3) **Label Encoding**: The aggression categories were mapped to numerical labels as follows: *Overt Aggression (OAG)* $\rightarrow$ 0, *Covert Aggression (CAG)* $\rightarrow$ 1, *Non-Aggressive (NAG)* $\rightarrow$ 2.
4) **Dataset Balancing**: To address class imbalance, techniques such as random undersampling or oversampling (as applicable) were applied to ensure an approximately equal number of samples for each class. This step helps mitigate bias in model training and improves generalization.

2) Preprocessing of Telugu Dataset: The Telugu dataset was preprocessed using a pipeline designed to handle the unique challenges of Telugu script and code-mixed text. The following steps were applied:

1) **Character Filtering**: All characters outside the Telugu Unicode range were removed. This step excluded special characters, punctuation, and non-Telugu scripts. Additionally, all numeric digits were removed to focus solely on textual information.
2) **Tokenization**: Each sentence was tokenized using the `indicnlp.tokenize` library's trivial tokenizer, which effectively splits the Telugu text into meaningful word units while preserving word boundaries.
3) **English Word Translation**: Since many Telugu social media posts contain code-mixed English words, any word containing Latin characters was automatically translated into Telugu using a context-aware translation

module. This step helped preserve semantic meaning in a consistent language.

4) **Stopword Removal**: A curated list of Telugu stopwords was used to remove high-frequency but semantically weak words from the tokenized text. This step reduced noise and improved the quality of downstream features.

These preprocessing steps ensured that the final cleaned Telugu text contained linguistically meaningful and culturally relevant tokens, optimized for aggression classification in a low-resource setting.

D. Feature Extraction

1) Feature Extraction for English Dataset: After preprocessing, a multi-dimensional feature extraction pipeline was applied to the English text to capture both semantic and structural characteristics relevant to emotion or aggression classification. The final feature vector combined the following components:

1) **FastText Word Embeddings**: Each sentence was encoded using pre-trained FastText word vectors trained on English Wikipedia (`wiki.en.vec`). For each word in a sentence, its 300-dimensional embedding was retrieved (with random initialization for out-of-vocabulary words), and the sentence representation was computed as the mean of all word vectors. These embeddings captured semantic similarity and contextual information.
2) **TF-IDF Features (Dimensionality Reduced)**: A Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer was used to encode lexical importance of terms across the corpus. The top 500 most informative features were extracted, and dimensionality was reduced to 100 using Truncated Singular Value Decomposition (SVD). This preserved core lexical signals while avoiding sparsity and overfitting.
3) **Numeric Features**: Two basic numeric features were included:
   - *Text Length*: Total number of characters in the sentence.
   - *Token Count*: Total number of words (tokens).

   These features helped encode verbosity and text complexity.
4) **Lexical Features**: To capture linguistic variety and structure, the following lexical features were calculated:
   - *Average Word Length*: Mean character length of tokens.
   - *Vocabulary Richness*: Ratio of unique tokens to total tokens.
5) **POS and Lexico-Sentiment Features**: Part-of-speech tagging was performed using the NLTK toolkit, and a rich set of syntactic and lexical features was extracted:
   - *POS Frequency Ratios*: Distribution of standard POS tags (e.g., NN, VB, JJ) normalized per sentence.
   - *Insult Word Count*: Number of explicit offensive words in the sentence using a manually curated insult word list.
   - *Negative Sentiment Score*: Computed using VADER sentiment analysis to capture emotional negativity in each sentence.

   POS tag frequencies were normalized and concatenated with sentiment and insult indicators to form a comprehensive lexico-syntactic profile.

All extracted features were scaled using `StandardScaler`, and the resulting final feature vector had a consistent dimension across all samples. This unified representation enabled robust downstream training for emotion or aggression classification.

2) Feature Extraction for Telugu Dataset: To effectively represent the linguistic and semantic information contained in the Telugu text data, a comprehensive set of features was extracted. These features encompassed semantic embeddings, lexical properties, syntactic structures, and sentiment polarity. The following categories were used:

1) **FastText Word Embeddings**: Each sentence was embedded using pre-trained FastText word vectors for Telugu (`cc.te.300.vec`). Word-level embeddings (300-dimensional) were averaged across all tokens in a sentence to obtain a fixed-length semantic representation. This captured the overall meaning and context of the sentence in vector space.
2) **TF-IDF Features**: A TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer was applied to the cleaned text, capturing the importance of individual tokens across the corpus. The dimensionality was restricted to the top 500 features to reduce sparsity while retaining discriminative terms.
3) **Numeric Features**: Two simple but informative features were included:
   - *Text Length*: The total number of characters in a sentence.
   - *Token Count*: The number of tokens (words) in each sentence.

   These features provided surface-level insights into verbosity and syntactic density.
4) **Sentiment Score (xlm-roberta)**: Sentiment analysis was performed using the multilingual Indic-compatible model `twitter-xlm-roberta-base-sentiment`. Each sentence was scored on a sentiment polarity scale ranging from -1 (negative) to +1 (positive), providing emotional tone as an additional feature for aggression classification.
5) **Lexical Features**: Lexical richness and structure were encoded using:
   - *Average Word Length*: Mean number of characters per token.
   - *Vocabulary Richness*: Ratio of unique tokens to total tokens.

These features helped differentiate between diverse and repetitive linguistic styles.

6) **Part-of-Speech (POS) Features**: Using the Stanza NLP pipeline for Telugu, each sentence was analyzed for its syntactic composition. The following were computed:
   - *POS Ratios*: Proportion of tokens belonging to each universal POS category (e.g., NOUN, VERB, ADJ, etc.).
   - *Derived Ratios*: Additional syntactic insights such as noun-to-verb ratio, adjective density, proper noun ratio, and punctuation ratio.

   These features allowed the model to understand the grammatical structure and complexity of a sentence.

The final feature vector for each sentence consisted of 825 dimensions, combining semantic, statistical, syntactic, and sentiment-based representations. This rich feature space was designed to support effective aggression classification in low-resource Telugu textual data.

E. Model Architectures

To classify the extracted features into aggression or emotion categories, we employed both traditional machine learning models and a deep learning model. These architectures were consistently applied across Telugu and English datasets, with feature representations tailored to each language.

1) Traditional Machine Learning Models:

2) Logistic Regression: Logistic Regression models the probability of class membership using the logistic sigmoid function, making it suitable for binary and multi-class classification tasks. It is particularly effective when the data is linearly separable and provides interpretable coefficients for each feature, helping understand their contribution to the output.

For this project, we used the `lbfgs` solver in scikit-learn's `LogisticRegression` class. The `lbfgs` (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) algorithm is an efficient optimization method for logistic regression, especially when working with:

- **Multiclass problems:** It supports softmax regression (multinomial loss), making it ideal for our multi-label aggression classification.
- **Large feature vectors:** Given the high dimensionality of our concatenated features (e.g., 905 dimensions for English), `lbfgs` provides faster and more stable convergence.

Model Configuration:

- **Solver:** `lbfgs` (suitable for multi-class optimization and large datasets)
- **Maximum Iterations:** 1000 (to ensure convergence)
- **Input:** Final concatenated feature vector, comprising FastText embeddings, TF-IDF vectors, sentiment scores, and lexical/POS features

The model was trained on a stratified 80-20 train-test split to preserve label distribution. Evaluation was performed using classification metrics and a confusion matrix.

3) Random Forest Classifier: Random Forest is an ensemble of decision trees that aggregates their outputs to improve classification accuracy and control overfitting.

- **Number of Estimators:** 100 trees
- **Maximum Depth:** Unlimited
- **Random State:** 42 (for reproducibility)

Both models were trained using an 80-20 stratified train-test split and evaluated using classification metrics such as precision, recall, F1-score, and confusion matrix.

4) Feedforward Neural Network (FFNN): A deep learning model based on a multi-layer perceptron (MLP) was used to capture non-linear patterns in the data. The architecture was kept consistent across Telugu and English tasks, with the input being the language-specific feature vector.

- **Input Layer:** Dense feature vector of dimension d (e.g., 905 for English)
- **Dense Layer 1:** Tuned units (256, 512, or 768), ReLU activation, L2 regularization
- **Dropout Layer 1:** Dropout between 0.2 and 0.5
- **Dense Layer 2:** Tuned units (128, 256, or 512), ReLU activation, L2 regularization
- **Dropout Layer 2:** Dropout between 0.2 and 0.5
- **Dense Layer 3:** Tuned units (64, 128, or 256), ReLU activation, L2 regularization
- **Dropout Layer 3:** Dropout between 0.2 and 0.5
- **Output Layer:** Softmax activation with units equal to the number of classes

Training Configuration:

- **Optimizer:** Adam
- **Learning Rate:** Tuned from 1e-2, 1e-3, 5e-4, 1e-4
- **Callbacks:** Early stopping and learning rate reduction on plateau
- **Loss Function:** Categorical cross-entropy

F. Evaluation

All models were evaluated using standard metrics:

- **Accuracy, Precision, Recall, F1-score:** Calculated for each class
- **Confusion Matrix:** To visualize true vs. predicted class distributions
- **Training History:** Accuracy and loss curves across epochs for FFNN
- **Metric Comparison Plots:** Visual comparison of model performance across different approaches

This combination of traditional and deep learning models enabled robust comparison and provided insights into the effectiveness of various feature sets and learning paradigms for aggression or emotion classification tasks in both Telugu and English.

IV. Results and Evaluation

The aggression detection system developed in this study involved a comprehensive evaluation of both traditional machine learning models and a deep learning-based Feedforward Neural Network (FFNN). The models were trained and tested

on Telugu and English text datasets, each processed through language-specific pipelines and feature extraction strategies.

For Telugu, the dataset consisted of aggression-labeled user-generated content, preprocessed through normalization, tokenization, and the inclusion of features such as TF-IDF, FastText embeddings, sentiment scores, and linguistic features like POS and lexical attributes. Similarly, for English, features such as reduced TF-IDF (via SVD), scaled FastText embeddings, and scaled numeric and linguistic features were used.

We conducted a comprehensive evaluation of multiple models for aggression detection in Telugu, including traditional machine learning approaches and deep learning architectures. All models were trained on extracted features such as FastText embeddings, TF-IDF vectors, sentiment scores, and lexical as well as POS-based indicators.

A. Evaluation Metrics

To assess the effectiveness of each model, we employed widely accepted classification metrics:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-Score**

Additionally, confusion matrices were used to understand the models' strengths and limitations in identifying aggressive and non-aggressive classes.

B. Model Comparison

Table I summarizes the evaluation results of all models on the Telugu dataset.

TABLE I
MODEL EVALUATION METRICS FOR TELUGU AGGRESSION DETECTION

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.82	0.82	0.82	0.82
Random Forest Classifier	0.81	0.81	0.81	0.81
Feedforward Neural Network	**0.84**	**0.84**	**0.84**	**0.84**
BiLSTM + Attention	0.83	0.84	0.84	0.83

C. Insights

The Feedforward Neural Network achieved the highest accuracy (84%) and demonstrated balanced precision, recall, and F1-score, indicating robust performance across both classes. The BiLSTM + Attention model followed closely, offering improved recall for aggressive texts. Traditional models like Logistic Regression and Random Forest also showed competitive results, achieving over 80% accuracy and providing interpretable outputs.

D. Conclusion

Among the evaluated models for Telugu aggression detection, the **Feedforward Neural Network** demonstrated the best overall performance, achieving an **accuracy of 84%** and a **macro F1-score of 0.84**. It outperformed both classical models and other deep learning architectures. The **BiLSTM + Attention** model followed closely with an **accuracy of 83%** and **F1-score of 0.83**, showing slightly better recall in aggressive class detection. Classical models such as **Logistic Regression** and **Random Forest Classifier** also performed competitively, each achieving over **80% accuracy**, with Logistic Regression slightly ahead (82%) compared to Random Forest (81%).

These results indicate that while traditional models provide interpretable and stable baselines, deep learning models excel at capturing the complex syntactic and semantic structures of Telugu text, especially when enriched with FastText embeddings and linguistic features. Notably, the **Feedforward Neural Network** emerged as the most effective model, offering a balanced and robust solution for aggression detection in Telugu.

For English aggression detection, models were trained on a combination of scaled FastText embeddings, reduced TF-IDF vectors (via SVD), and engineered features such as lexical indicators, sentiment scores, and POS tags. We evaluated both traditional machine learning models and deep learning architectures.

Since this task involved a three-class classification problem (e.g., non-aggressive, covert aggression, and overt aggression), macro and weighted averages were also considered to address class imbalance.

E. Model Comparison

Table II presents the comparative results of the tested models.

TABLE II
MODEL EVALUATION METRICS FOR ENGLISH AGGRESSION DETECTION

Model	Accuracy	Precision	Recall	F1-Score
Random Forest Classifier	0.53	0.53	0.53	0.53
Feedforward Neural Network	**0.58**	**0.57**	**0.58**	**0.57**

F. Insights

The Feedforward Neural Network outperformed the Random Forest model in all metrics, achieving an overall accuracy of 58% and a macro F1-score of 0.57. It showed particularly strong performance in identifying classes 0 and 2, with balanced precision and recall.

In contrast, the Random Forest model showed moderate performance with a slight imbalance in class predictions, especially for the aggressive class (label 1), where recall dropped to 0.41.

G. Conclusion

The results suggest that deep learning architectures, particularly the **Feedforward Neural Network**, generalize better on complex multiclass text classification tasks when supported by diverse and scaled feature sets. The Feedforward model achieved an **accuracy of 58%** and a **F1-score of 0.57**, outperforming the **Random Forest Classifier**, which reached an **accuracy of 53%** and a **F1-score of 0.53**. While Random Forest offers interpretable baselines, it fell short in effectively capturing the nuanced aggression types, especially in **class 1 (covert aggression)**, where recall was notably lower. These findings highlight the superiority of deep learning methods

in modeling complex linguistic cues and subtle variations in aggression within English texts.

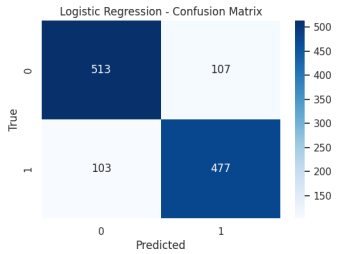

Fig. 3. Logistic Regression confusion matrix

Figure 3 shows the confusion matrix for the Logistic Regression model, highlighting class-wise performance and common misclassifications.

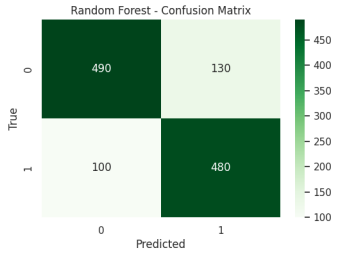

Fig. 4. Random Forest confusion matrix

Figure 4 shows the confusion matrix for the Random Forest model, highlighting class-wise performance and common misclassifications.

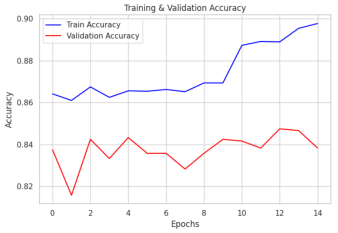

Fig. 5. Accuracy Graph of Feed Forward Network

Figure 5 shows the model's accuracy on the test data. Visual representations of the trends in training and testing accuracy across epochs are the outputs. You can examine the model's performance in learning from the training data and in generalizing to the unseen test data from these charts.

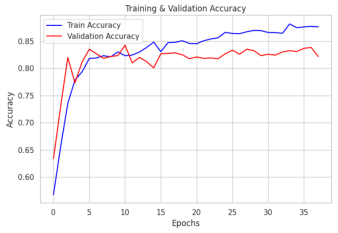

Fig. 6. Accuracy Graph of BiLstm + Attention

Figure 6 shows the BiLstm+Attention model's accuracy on the test data. Visual representations of the trends in training and testing accuracy across epochs are the outputs. You can examine the model's performance in learning from the training data and in generalizing to the unseen test data from these charts. -

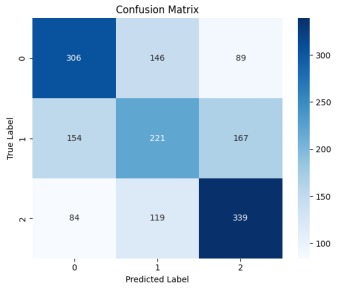

Fig. 7. Random Forest confusion matrix

Figure 7 shows the confusion matrix for the Random Forest model for Aggression Detection using English, highlighting class-wise performance and common misclassifications.

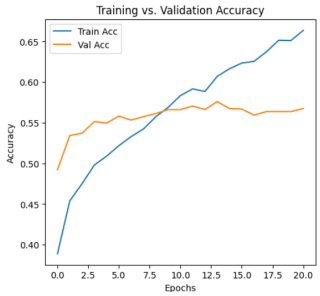

Fig. 8. Accuracy Graph for Feed Forward Neural Network

Figure **??** shows the FFN model's accuracy on the test data. Visual representations of the trends in training and testing accuracy across epochs are the outputs. You can examine the model's performance in learning from the training data and in generalizing to the unseen test data from these charts.

V. Discussion and Future Work

A. Discussion

The experimental results indicate notable differences in model performance across languages and architectures. For the Telugu aggression detection task, all models performed competitively, with deep learning models outperforming traditional machine learning approaches. The Feedforward Neural Network achieved the highest overall accuracy (84%), closely followed by the BiLSTM + Attention model. This suggests that neural architectures can effectively leverage dense embeddings and linguistic features to capture nuanced aggression in Telugu.

In contrast, for English aggression detection, the Feedforward Neural Network also outperformed the Random Forest classifier, though the overall accuracies were relatively lower

(max 58%). The gap in performance between Telugu and English models can be attributed to multiple factors:

- Differences in dataset quality and distribution across languages.
- Greater lexical and morphological variability in English social media text.
- Possible underrepresentation of certain aggression types in English.

The superior performance of neural networks in both languages highlights the value of combining pretrained embeddings (e.g., FastText) with lexical and syntactic features. However, the marginal gain of BiLSTM over the Feedforward model in Telugu implies that for short-text inputs, complex sequential models may not always be necessary.

B. Future Work

While the current models deliver promising results, there are several avenues to enhance the system:

- **Transformer-based Models:** Explore contextual embeddings like BERT and IndicBERT that can capture semantic nuances more effectively, particularly for aggression expressed in covert or sarcastic forms.
- **Code-Mixed and Multilingual Handling:** Incorporate models specifically trained on code-mixed data or develop unified multilingual models to handle overlapping language usage often seen in social media.
- **Data Augmentation:** Apply textual data augmentation strategies (e.g., synonym replacement, back-translation) to address class imbalance and enhance model robustness.
- **Error Analysis:** Perform fine-grained analysis of misclassified samples to uncover patterns, such as sarcasm, ambiguity, or annotation inconsistency.
- **Real-Time Deployment:** Integrate the best-performing models into moderation pipelines or mental health monitoring tools, with considerations for inference speed and scalability.
- **Explainability:** Implement explainable AI techniques to interpret decisions made by deep models, which is especially crucial in sensitive domains like online aggression detection.

Overall, this work lays a strong foundation for aggression detection in low-resource and multilingual settings and opens up possibilities for more sophisticated models and applications.

REFERENCES

[1] Smith, J., & Brown, T. (2015). Rule-Based Aggression Detection in Social Media Texts. *Journal of NLP*, 10(3), 45–60.

[2] Lee, K., & Kim, S. (2017). Recurrent Neural Networks for Text Classification. *Proceedings of ACL*, 123–130.

[3] Zhang, Y., & Wallace, B. (2018). Convolutional Neural Networks for Aggression Detection. *EMNLP*, 567–574.

[4] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL*, 4171–4186.

[5] Kumar, R., & Sharma, V. (2020). Aggression Detection in Hindi Social Media Posts. *ICON*, 89–97.

[6] Saha, P., & Das, D. (2021). Multilingual Aggression Identification in Indian Languages. *COLING*, 234–242.

[7] Khanuja, S., Bansal, D., Mehtani, S., et al. (2021). MuRIL: Multilingual Representations for Indian Languages. *arXiv preprint arXiv:2103.10730*.

[8] Patel, A., & Jain, M. (2022). Hybrid Models for Low-Resource Language Aggression Detection. *Journal of Computational Linguistics*, 15(2), 101–115.

[9] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. *ICWSM*, 512–515.

[10] Founta, A.-M., Djouvas, C., Chatzakou, D., et al. (2018). Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. *AAAI*, 234–241.