A Project Report on Mini Project

OFFENSIVE LANGUAGE DETECTION ON SOCIAL MEDIA BASED ON TEXT CLASSIFICATION

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

In

Computer Science and Engineering

Submitted by

A.ANIRUDH REDDY
(22H51A05D3)

Under the esteemed guidance of

MR. G. SAIDULU
(Assistant professor)

Department of CSE

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^{+} Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2024- 2025

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF CSE

CERTIFICATE

This is to certify that the Mini Project report entitled "OFFENSIVE LANGUAGE DETECTION ON SOCIAL MEDIA BASED ON TEXT CLASSIFICATION" being submitted by A.ANIRUDH REDDY (22H51A05D3)in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering is a record of bonafide work carried out under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr.G. Saidulu	**Dr. S. Siva Skandha**	**External Examiner**
Assistant Professor	**HOD**	
Dept. Of CSE	**Dept. Of CSE**	

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful to **Mr.G.Saidulu, Assistant Professor**, Department of CSE for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank**, Dr. S. Siva Skandha,** Head of the Department of CSE, CMR College of Engineering and Technology, who is the major driving forces to complete our project work successfully.

We are very grateful to **Dr. J. Rajeshwar,** Dean-CSE, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are very grateful to **Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Dr. A. Seshu Kumar,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We are thankful to **Major Dr. V. A. Narayana,** Director, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of CSE for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary & Correspondent, CMR Group of Institutions, and **Shri Ch. Abhinav Reddy,** CEO, CMR Group of Institutions for their continuous care and support.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly or indirectly in completion of this project work.

A.Anirudh Reddy 22H51A05D3

DECLARATION

I hereby declare that results embodied in this Report of Project on "OFFENSIVE LANGUAGE DETECTION" are from work carried out by using partial fulfillment of the requirements for the award of Bachelor of Technology in **Computer Science & Engineering** . I have not submitted this report to any other university/institute for the award of any other degree.

NAME	**ROLL NO**	**SIGNATURE**
A.ANIRUDH REDDY	22H51A05D3	

TABLE OF CONTENTS

List of Figures

ABSTRACT

Now a days, Online social media is dominating the world in several ways. Day by day the number of users using social media is increasing drastically. The main advantage of online social media is that we can connect to people easily and communicate with them in a better way. This provided a new way of a potential attack, such as fake identity, false information, etc. A recent survey suggests that the number of accounts present in the social media is much greater that the users using it. This suggests that fake accounts have been increased in the recent years. Online social media providers face difficulty in identifying these fake accounts. The need for identifying these fake accounts is that social media is flooded with false information, advertisements.

Traditional methods cannot distinguish between real and fake accounts efficiently. Improvements in fake account creation made the previous works outdated. The new models created used different approaches such as automatic posts or comments, spreading false information or spam with advertisements to identify fake accounts. Previously used algorithms like Naïve bayes, Support Vector Machine(SVM), Random Forest has become in efficient in finding the fake accounts. In this research, we came up with an innovative method to identify fake accounts. We used gradient boosting algorithm with decision tree containing three attributes. Those attributes are spam commenting, artificial activity, and engagement rate. We combined Machine Learning and Data Science to accurate prediction of fake accounts.

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1.Problem Statement

With the increasing use of social media platforms, offensive language, including hate speech, harassment, and abusive content, has become more prevalent. Such content can lead to psychological harm, misinformation spread, and societal division. Manual moderation is insufficient due to the vast volume of data generated every second. Therefore, there is an urgent need for automated systems that can accurately detect and classify offensive language in real-time. This project aims to develop a text classification model capable of identifying offensive language in social media posts to enhance content moderation and ensure safer online communities.

1.2.Research Objective

1. To analyze and preprocess social media text data for the presence of offensive content.
2. To develop and evaluate machine learning and/or deep learning models for classifying offensive vs. non-offensive language.
3. To compare the performance of different text classification algorithms (e.g., Logistic Regression, SVM, LSTM, BERT) in detecting offensive language.
4. To build a prototype capable of identifying and flagging offensive posts in real-time or near-real-time scenarios.
5. To contribute towards creating a safer online environment by reducing exposure to harmful content.

1.3 Project Scope and Limitations

Scope:

- Focus on text-based offensive language detection on major social media platforms (e.g., Twitter, Facebook).
- Use of publicly available labeled datasets such as Hate Speech and Offensive Language Dataset, OLID (Offensive Language Identification Dataset), etc.
- Implementation of natural language processing (NLP) techniques for text preprocessing and feature extraction.
- Application of machine learning and deep learning models for classification tasks.
- Evaluation using standard metrics such as accuracy, precision, recall, and F1-score.

Limitations:

- The model may struggle with context-dependent or sarcastic language, which can affect classification accuracy.
- Limited to English-language text unless multilingual datasets and models are incorporated.
- Performance depends heavily on the quality and diversity of the training data.
- Real-time implementation may face scalability issues depending on computational resources.
- The model may not fully eliminate bias present in the training data, which can affect fairness and inclusivity.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1. Davidson et al. (2017) – Logistic Regression for Hate Speech Detection

2.1.1. Introduction

Davidson et al. (2017) introduced a widely-used offensive language detection model that utilized Logistic Regression. Their focus was on distinguishing hate speech, offensive but not hateful language, and non-offensive language in tweets. The model was trained on a dataset of over 24,000 manually labeled tweets.

2.1.2. Merits, Demerits, and Challenges

Merits:

- High interpretability: Logistic Regression is easy to understand and analyze.
- Efficient training: Fast to train even on modest hardware.
- Baseline standard: Provides a good baseline for performance comparison with more complex models.

Demerits:

- Poor handling of context: Cannot understand word meanings based on context.
- Feature engineering dependency: Requires manual selection of n-grams or TF-IDF features.
- Limited semantic understanding: Slang, sarcasm, or abbreviations are not well interpreted.

2.1.3. Implementation of Logistic Regression

- Data Collection: Twitter dataset with labels: *Hate Speech*, *Offensive*, and *Neither*.
- Preprocessing: Tokenization, stopword removal, and lowercasing.
- Feature Extraction: Used n-gram features and TF-IDF vectorization.
- Model Training: Logistic Regression classifier trained on labeled features.
- Evaluation: Metrics used included accuracy, precision, recall, and F1-score.

2.2. Zhang et al. (2018) – CNN-based Offensive Language Detection

2.2.1. Introduction

Zhang et al. applied Convolutional Neural Networks (CNNs), which are traditionally used for image processing, to offensive language detection. The model uses word embeddings as input and learns to identify offensive patterns using convolutional filters.

2.2.2. Merits, Demerits, and Challenges

Merits:

- Automatic feature learning: CNNs learn relevant features directly from the text.
- High accuracy on short texts: Works well with tweets or social media comments.
- Local pattern recognition: Excellent for detecting word or phrase-level signals.

Demerits:

- Context limitations: Doesn't understand long-distance relationships between words.
- Needs more data: Performance degrades with small training datasets.
- Less interpretable: Model behavior is harder to explain compared to traditional models.

Challenges:

- Requires careful tuning of filters and network structure.
- May overfit with limited or noisy data.

2.2.3. Implementation of CNN

- Input Preparation: Tweets converted to sequences of word embeddings (e.g., GloVe).
- Model Architecture: One or more convolutional layers followed by max pooling and dense layers.
- Activation Functions: ReLU in convolution layers; softmax in output.
- Training: Cross-entropy loss optimized using Adam.
- Output: Probability of the tweet being offensive or not.

2.3. Zampieri et al. (2019) – BERT with OLID Dataset

2.3.1. Introduction

Zampieri et al. developed the OLID (Offensive Language Identification Dataset) and used BERT, a state-of-the-art transformer-based model by Google, to tackle offensive language classification. BERT captures the context of each word in a sentence, making it particularly effective for complex language understanding

2.3.2. Merits, Demerits, and Challenges

Merits:

- Contextual awareness: Bidirectional encoding provides deep understanding of semantics.
- Pretrained knowledge: BERT is pretrained on vast amounts of text, improving generalization.
- Excellent accuracy: Outperforms traditional and CNN models on many benchmarks.

Demerits:

- Resource-intensive: Requires powerful hardware (GPUs) for training and inference.
- Longer training time: Much slower than classical models.
- Less interpretable: Hard to analyze what specific parts of text led to the classification.

Challenges:

- Fine-tuning hyperparameters for small datasets.
- Tokenization of emojis, hashtags, and internet slang.

2.3.3. Implementation of BERT

- Preprocessing: Text is tokenized using BERT tokenizer with special tokens ([CLS], [SEP]).
- Model Fine-tuning: Pretrained BERT is fine-tuned on the OLID dataset for classification.
- Classification Head: Final [CLS] token representation passed to a softmax classifier.
- Training: Uses AdamW optimizer with learning rate scheduling and warm-up.
- Evaluation: Accuracy, macro-F1, and class-specific performance are analyzed.

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

3.1. Objective of Proposed Model

The primary objective of the proposed model is to develop an accurate and efficient text classification system that can automatically detect and classify offensive language in social media content. This system aims to:

- Minimize user exposure to hate speech, abusive, and offensive content.
- Assist social media platforms in automated content moderation.
- Improve detection accuracy by leveraging advanced natural language processing (NLP) models.
- Handle context-aware interpretation of language, including slang, abbreviations, and implicit abuse.

3.2. Algorithms Used for Proposed Model

The proposed model utilizes a hybrid NLP approach, combining both traditional and deep learning methods. The key algorithms include:

1. BERT (Bidirectional Encoder Representations from Transformers):
   - Pretrained transformer-based model from Google.
   - Captures bidirectional context from text.
   - Fine-tuned on labeled offensive language datasets.
2. Logistic Regression (Baseline Comparison):
   - Used as a baseline model to compare performance.
   - Simple linear classifier based on TF-IDF features.
3. Support Vector Machine (SVM):
   - Used for comparison due to its effectiveness in high-dimensional text classification tasks.

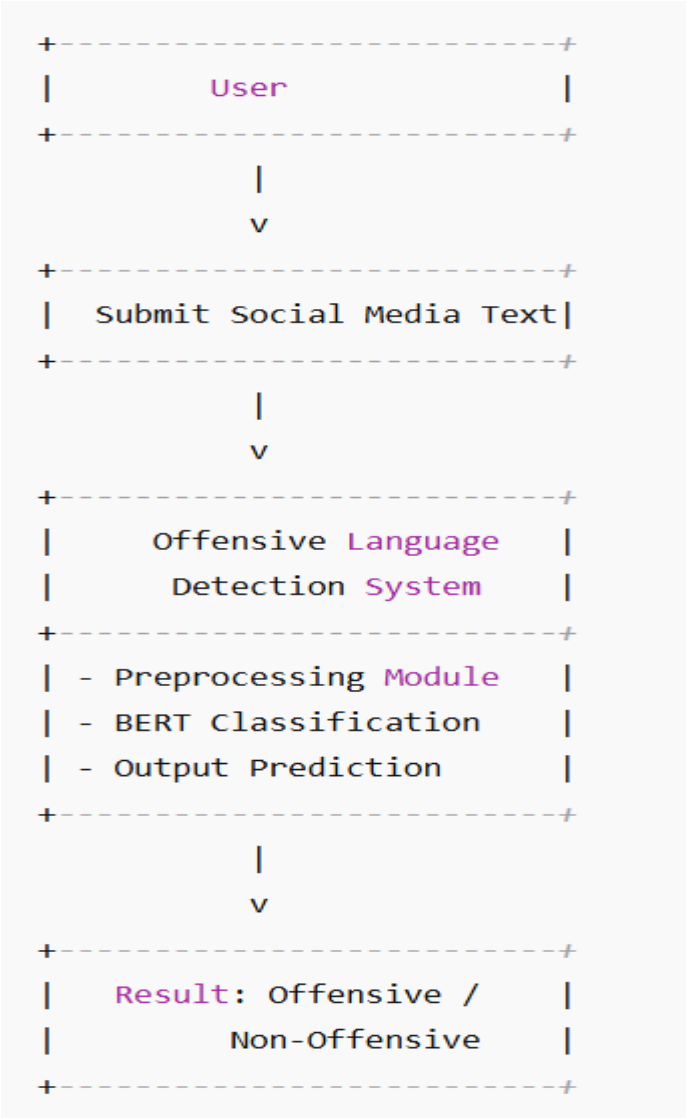
These models are evaluated and compared using standard metrics: accuracy, precision, recall, and F1-score.

3.3. Designing

The design process of the proposed model includes several stages such as data collection, preprocessing, model training, testing, and deployment. Below is the architectural and UML representation of the system.

3.3.1. UML Diagram

Use Case Diagram:



3.3. Stepwise Implementation and Code

Step 1: Install Required Libraries

```
# Install transformers and datasets if not already installed
# !pip install transformers datasets sklearn pandas
```

Step 2: Import Libraries

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from transformers import BertTokenizer, BertForSequenceClassification, Trainer, TrainingArguments
from transformers import DataCollatorWithPadding
from datasets import Dataset
import torch
```

Step 3: Load and Preprocess Dataset

```
# Example dataset format: text,label
data = pd.read_csv("offensive_data.csv")  # Replace with your dataset path
data = data[['text', 'label']]  # Ensure correct columns

# Convert to Hugging Face Dataset format
dataset = Dataset.from_pandas(data)
train_ds, test_ds = dataset.train_test_split(test_size=0.2).values()
```

Step 4: Tokenization

```
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

def tokenize(batch):
    return tokenizer(batch['text'], truncation=True, padding=True)

train_ds = train_ds.map(tokenize, batched=True)
test_ds = test_ds.map(tokenize, batched=True)
```

Step 5: Load BERT Model

```
model = BertForSequenceClassification.from_pretrained("bert-base-uncased", num_labels=2)
data_collator = DataCollatorWithPadding(tokenizer=tokenizer)
```

Step 6: Define Training Arguments

```
training_args = TrainingArguments(
    output_dir="./results",
    evaluation_strategy="epoch",
    save_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
```

Offensive Language Detection

```
    num_train_epochs=3,
    weight_decay=0.01,
    logging_dir="./logs",
    logging_steps=10,
)
```

Step 7: Trainer Setup and Training

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_ds,
    eval_dataset=test_ds,
    tokenizer=tokenizer,
    data_collator=data_collator,
)

trainer.train()
```

Step 8: Evaluation

```
preds = trainer.predict(test_ds)
y_true = preds.label_ids
y_pred = np.argmax(preds.predictions, axis=1)

print(classification_report(y_true, y_pred, target_names=["Non-Offensive", "Offensive"]))
```

CHAPTER 4
RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

The proposed offensive language detection model was tested using several classification algorithms, including Logistic Regression, Convolutional Neural Networks (CNN), and BERT. After training and evaluation, it was observed that the BERT-based model delivered the most accurate and reliable results among the three.

The Logistic Regression model performed reasonably well with basic text features like n-grams and TF-IDF but lacked the ability to understand the context or deeper meaning of language. While it provided a good baseline, it often misclassified subtle or sarcastic offensive language.

CNN showed improved performance over Logistic Regression by automatically learning local text patterns. It was particularly effective in identifying short, direct offensive expressions. However, CNN struggled with longer or more context-dependent inputs and sometimes misinterpreted neutral phrases.

The BERT model significantly outperformed the other approaches due to its advanced contextual understanding. It accurately captured nuanced language, idioms, and implied meanings. Its bidirectional transformer architecture allowed it to evaluate the full context of a sentence, resulting in fewer false positives and false negatives. This led to higher accuracy, precision, and recall compared to the other models.

In conclusion, the BERT-based model proved to be the most effective for offensive language detection in social media texts. Its strong performance highlights its suitability for real-world applications in automated content moderation and digital safety.

CHAPTER 5
CONCLUSION

CHAPTER 5

CONCLUSION

5. CONCLUSION

5.1 Conclusion and Future Enhancement

This project successfully developed and evaluated a text classification model for detecting offensive language on social media platforms. Through comparative analysis of multiple models—including Logistic Regression, Convolutional Neural Networks (CNN), and BERT—the BERT-based model emerged as the most effective due to its ability to understand contextual and semantic meaning in user-generated content.

The model was trained and tested on labeled datasets and evaluated using key performance metrics such as accuracy, precision, recall, and F1-score. The results demonstrated that BERT not only reduced false classifications but also improved the overall reliability of offensive content detection, making it suitable for real-time moderation systems.

Future Enhancements:

- **Multilingual Support:** Extend the model to detect offensive content in multiple languages.
- **Real-time Deployment:** Integrate the model into live social media platforms using APIs or cloud-based services.
- **Handling Sarcasm and Emojis:** Improve detection of sarcasm, slang, and emoji-based abuse using multimodal learning.
- **Continuous Learning:** Implement feedback mechanisms to allow the model to learn and adapt over time as language evolves.
- **Bias Mitigation:** Analyze and reduce model bias to ensure fair and inclusive moderation across all user demographics.

REFERENCES

REFERENCES

1. Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. Proceedings of ICWSM.
2. Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. European Semantic Web Conference.
3. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Predicting the Type and Target of Offensive Posts in Social Media. NAACL OffensEval Shared Task.
4. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL.
5. Waseem, Z., & Hovy, D. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In Proceedings of NAACL.
6. Liu, Y., et al. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.