Enhancing Tanglish Sentiment Analysis through Hybrid NLP: Rule-Based Lexicons and Indic-BERT Integration

IBM APPLICATION COURSE PROJECT REPORT

(216CSE4301)

Submitted by

RAJESH KANNA R – 99220041074
SAKTHI SANJAY S – 99220041079
VIJAYAKUMAR M – 99220040774
KESANI ROHITH – 99220040574

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

COMPUTER SCIENCE AND ENGINEERING

KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION

KRISHNANKOIL 626 126

DECEMBER 2025

DECLARATION

We affirm that the project work titled **"Enhancing Tanglish Sentiment Analysis through Hybrid NLP: Rule-Based Lexicons and Indic-BERT Integration"** being submitted in partial fulfillment for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** is the original work carried out by us. It has not formed part of any other project work submitted for the award of any degree or diploma, either in this or any other University.

RAJESH KANNA R
99220041074

SAKTHI SANJAY S
99220041079

VIJAYAKUMAR M
99220040774

KESANI ROHIT
99220040574

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Signature of supervisor

Mrs. GURUSIGAAMANI A M

Associate/Assistant Professor

Department of Computer Science and Engineering

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade

BONAFIDE CERTIFICATE

Certified that this project report **"Enhancing Tanglish Sentiment Analysis through Hybrid NLP: Rule-Based Lexicons and Indic-BERT Integration"** is the bonafide work of "**Rajesh Kanna R (99220041074), Sakthi Sanjay S (99220041079), Vijayakumar M (99220040774), and Kesani Rohith (99220040574)"** who carried out the project work under my supervision.

Mrs. GURUSIGAAMANI A M
SUPERVISOR
Associate/Assistant Professor
Computer Science and Engineering
Kalasalingam Academy of Research
and Education
Krishnankoil 626126
Virudhunagar District.

Dr. R. Raja Subramanian
HEAD OF THE DEPARTMENT
Associate Professor/Head
Computer Science and Engineering
Kalasalingam Academy of Research
and Education
Krishnankoil 626126
Virudhunagar District.

Submitted for the Project final review and Viva-voce examination held on ……………

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We would like to begin by expressing our heartfelt gratitude to the Supreme Power for the immense grace that enabled us to complete this project.

We are deeply grateful to the late **"Kalvivallal" Thiru T. Kalasalingam,** Chairman of the Kalasalingam Group of Institutions, and to "**Illayavallal" Dr. K. Sridharan**, Chancellor, as well as **Dr. S. Shasi Anand**, Vice President, who has been a guiding light in all our university's endeavours.

Our sincere thanks go to our Vice Chancellor, **Dr. S. Narayanan**, for his inspiring leadership, guidance, and for instilling in us the strength and enthusiasm to work towards our goals.

We would like to express our sincere appreciation to **Dr. P. Deepa Lakshmi**, Dean of the School of Computing and Director of International Ranking, for her valuable guidance. Our heartfelt gratitude also goes to our esteemed Head of Department, **Dr. R. Raja Subramanian**, whose unwavering support has been crucial to the successful advancement of our project.

We are especially thankful to our Project Supervisor, **Mrs. Gurusigaamani A M**, for his patience, motivation, enthusiasm, and vast knowledge, which greatly supported us throughout this work.

Our sincere gratitude also goes to **Dr.Vijay M** and **Dr.N.Sundareswaran** Overall Project Coordinators, for their constant encouragement and support in completing this IBM Project.

Finally, we would like to thank our parents, faculty, non-teaching staff, and friends for their unwavering moral support throughout this journey.

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade

SCHOOL OF COMPUTING

COMPUTER SCIENCE AND ENGINEERING

PROJECT SUMMARY

Project Title	Enhancing Tanglish Sentiment Analysis through Hybrid NLP: Rule-Based Lexicons and Indic-BERT Integration	
Project Team Members (Name with Register No)	**Rajesh Kanna R** – 99220041074, **Sakthi Sanjay S** – 99220041079, **Vijayakumar M** – 99220040774, **Kesani Rohith** – 99220040574	
Guide Name/Designation	Mrs. Gurusigaamani A M	
Program Concentration Area	Artificial Intelligence & Machine Learning, Natural Language Processing (NLP)	
Technical Requirements		
Engineering standards and realistic constraints in these areas		
Area	**Codes & Standards / Realistic Constraints**	**Tick ✓**
Economic		
Environmental		
Social		
Ethical		
Health and Safety		
Manufacturability		
Sustainability		

REALISTIC CONSTRAINTS:

Environmental:

Sustainability:

Engineering Standards:

ABSTRACT

The rapid growth of social media has led to widespread use of code-mixed languages such as Tanglish, a hybrid form of Tamil and English commonly used in informal digital communication. Traditional sentiment analysis models, which are designed for monolingual and well-structured text, perform poorly on Tanglish due to its inconsistent spellings, informal grammar, and multilingual characteristics. This project addresses these challenges by developing and evaluating a range of machine learning and deep learning models for Tanglish sentiment classification.

A curated dataset of Tamil–English code-mixed text was preprocessed and used to train six models: Support Vector Machines (SVM), Naive Bayes, Logistic Regression, 1D Convolutional Neural Networks (1D-CNN), Long Short-Term Memory (LSTM), and the transformer-based BERT architecture. The models were assessed using standard performance metrics, including accuracy, precision, recall, and F1-score. Comparative analysis revealed that deep learning approaches, particularly BERT, demonstrated superior performance in capturing contextual cues and handling non-standard linguistic patterns.

To translate these findings into a practical application, the best-performing model was deployed in a user-friendly web application capable of providing real-time sentiment predictions for Tanglish input. This project contributes to the under-explored field of Tanglish NLP by presenting baseline performance benchmarks, highlighting the strengths and limitations of different modeling techniques, and offering a functional tool that supports sentiment analysis for code-mixed text.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACADEMIC REFERENCE COURSES

S. NO.	COURSE CODE	COURSE NAME

CHAPTER – I

INTRODUCTION

Sentiment analysis plays an essential role in understanding public opinions expressed across digital platforms. While significant progress has been made in high-resource languages such as English, analyzing sentiments in low-resource and code-mixed languages remains a challenging research area. One such language variety is Tanglish, a commonly used blend of Tamil and English, characterized by informal writing styles, non-standard spellings, and frequent code-switching. These characteristics make traditional sentiment analysis methods less effective and demand specialized approaches.

This project focuses on developing a robust sentiment classification system designed specifically for Tanglish text. The study evaluates and compares the performance of several machine learning and deep learning models—including Support Vector Machines (SVM), Naive Bayes, Logistic Regression, 1D-CNN, LSTM, and BERT—to classify Tanglish sentences into positive, negative, and neutral sentiment categories. By examining both traditional and modern architectures, the project aims to identify models capable of handling the linguistic irregularities present in code-mixed text.

A key outcome of this work is the development of a real-time sentiment analysis web application. The application allows users to input Tanglish text and instantly receive sentiment predictions, demonstrating the practical deployment potential of the trained models. The dataset used for this project is curated specifically for Tamil–English code-mixed content across diverse domains, ensuring reliable training and evaluation.

Overall, this project contributes to the growing field of multilingual and code-mixed NLP by offering practical insights, comparative model performance, and an accessible sentiment analysis tool tailored for Tanglish users. This work aligns with the broader goal of making NLP technologies more inclusive and effective for underrepresented language communities.

1.1 Background and Motivation

The rapid expansion of social media and digital communication has given rise to widespread linguistic blending, commonly known as *code-mixing*. Among Tamil-speaking communities, Tanglish—a hybrid mixes of Tamil and English—has become a dominant mode of informal online expression. This form of communication is marked by phonetic Tamil words written in Roman script, irregular spellings, and the seamless incorporation of English vocabulary into Tamil sentence structures. While natural and intuitive for speakers, these characteristics introduce considerable complexity for computational models.

Sentiment analysis, a core task within Natural Language Processing (NLP), plays an important role in understanding public opinion, evaluating customer feedback, and monitoring social trends. However, conventional NLP models are predominantly trained on monolingual, grammatically consistent text. These models struggle to interpret code-mixed languages like Tanglish, resulting in poor sentiment classification performance. This gap highlights a growing need for specialized tools capable of handling the linguistic unpredictability of code-mixed data and extracting meaningful insights for researchers, businesses, and social platforms.

1.2 Problem Statement

Traditional sentiment analysis systems exhibit low accuracy when applied to Tanglish due to the language's informal and highly variable nature. Tanglish text lacks standardized spelling frequently switches between Tamil and English and blends two distinct linguistic systems into a single expression. These properties challenge standard NLP pipelines, leading to misclassification, confusion between sentiment categories, and unreliable analytical outputs.

Therefore, the core problem addressed in this project is the absence of effective sentiment analysis models tailored for Tanglish. A reliable system must account for inconsistencies in transliteration, grammatical fluidity, and semantic ambiguity unique to code-mixed text. This project aims to design and evaluate such models to improve the accuracy and reliability of Tanglish sentiment classification.

1.3 Objectives of the Project

To bridge the performance gap in Tanglish sentiment analysis, the project focuses on the following objectives:

1. Data Curation

   Compile, preprocess, and label a comprehensive dataset of Tanglish text suitable for training and evaluating machine learning and deep learning models.

2. Model Implementation

   Develop a variety of sentiment analysis models, ranging from classical machine learning algorithms (SVM, Naive Bayes, Logistic Regression) to advanced deep learning architectures (1D-CNN, LSTM, BERT).

3. Performance Comparison
   Assess and compare the performance of all implemented models using standard evaluation metrics such as accuracy, precision, recall, and F1-score to identify the most effective model for Tanglish sentiment classification.
4. Application Development
   Build a user-friendly web application that integrates the best-performing model, providing real-time sentiment analysis for user-input Tanglish text.

1.4 Scope of the Project

This project is specifically focused on the domain of Tanglish sentiment analysis and is limited to the following boundaries:

- Language: Tanglish (Tamil–English code-mixed text) is the sole language under analysis.
- Task: Sentiment classification into *positive*, *negative*, and *neutral* categories.
- Models: Evaluation is restricted to selected machine learning and deep learning models including SVM, Naive Bayes, Logistic Regression, 1D-CNN, LSTM, and BERT.
- Output: The final deliverable is a functional sentiment analysis web application capable of real-time prediction.

The project does not extend to other code-mixed languages or additional NLP tasks such as machine translation, named entity recognition, or text summarization.

1.5 Methodology Overview

The methodology adopted for this project follows a systematic and structured workflow:

1. Data Collection and Preprocessing
   Tanglish datasets from multiple online sources were compiled. The data underwent preprocessing steps such as noise removal, normalization, and tokenization to address spelling inconsistencies and prepare text for modeling.
2. Feature Extraction
   Traditional machine learning models used TF-IDF-based vectorization, while

deep learning architectures utilized word embeddings and contextual embeddings (e.g., BERT).

3. Model Training
   All models were trained on the prepared dataset to learn sentiment patterns, linguistic cues, and code-mixed structures.
4. Evaluation
   Model performance was assessed on a test dataset using standard metrics to determine generalizability and robustness.
5. Deployment
   The best-performing model was integrated into a Streamlit/Flask-based web application to enable interactive, real-time sentiment prediction.

1.6 Organization of the Report

This report is structured to provide a clear and comprehensive understanding of the project.

- The Introduction outlines the relevance of the study.
- The Literature Review examines existing research on code-mixed sentiment analysis.
- The Methodology section explains data preprocessing, feature engineering, and model architectures.
- The Experimental Results section presents model performance and comparative analysis.
- The Web Application Development chapter describes deployment and user interface design.
- The report concludes with Findings, Limitations, and Future Work, offering insights for further research.

CHAPTER II

LITERATURE REVIEW

2.1 Overview of Related Work

Sentiment analysis in multilingual and code-mixed environments has gained significant attention due to the widespread use of mixed-language communication on social media. Traditional sentiment analysis models perform well on high-resource languages such as English but struggle with code-mixed structures like Tanglish (Tamil written in Roman script), where inconsistent transliteration, phonetic variations, and contextual ambiguity create additional complexity. Existing studies predominantly depend on machine learning and transformer-based models, which require large, annotated datasets and substantial computational resources. These limitations directly motivate the need for lightweight and hybrid approaches capable of handling low-resource linguistic settings.

2.2 Review of Similar Projects or Research Papers

Several recent studies have contributed to multilingual and code-mixed sentiment analysis:

- Gupta et al. (2025) proposed an unsupervised self-training sentiment classifier, improving adaptability to unlabeled datasets. However, the model was highly sensitive to noise, making it less reliable for highly informal code-mixed text.
- Dharini et al. (2025) introduced a multilingual ensemble-based sentiment model for YouTube comments. Although it improved accuracy, the system demanded high computational power, limiting real-time applications.
- KT et al. (2025) compared transformer models such as mBERT and IndicBERT for Indic code-mixed sentiments. While these models achieved strong performance, their heavy computational requirements prevented deployment in low-resource settings.
- Chakraborty et al. (2025) developed LINGUABRIDGE, an AI-powered translation and sentiment engine. Despite its multilingual capability, the system struggled with contextual accuracy in informal social media text.
- VP et al. (2025) focused on sentiment analysis of Tamil–Tulu code-mixed data using machine learning models, showing competitive results but difficulty in detecting sarcasm or implicit emotions.

- Goje and Patil (2025) evaluated word embeddings for Marathi political tweets. Their approach worked well within the domain but lacked generalization for multilingual datasets.
- Sherif and Sabty (2024) examined Arabic–English code-switched sentiment using traditional and transformer architectures, concluding that modern transformers outperform older neural models but require large, annotated corpora for reliability.

Overall, these studies highlight the ongoing challenges in sentiment classification for low-resource and code-mixed languages.

2.3 Summary and Gap Identification

From the reviewed literature, it is evident that:

- Most existing systems rely heavily on large, labeled datasets, which are not available for code-mixed languages like Tanglish.
- Transformer-based systems provide strong context understanding but have high resource consumption, making them unsuitable for lightweight or real-time applications.
- Many models focus on specific platforms (YouTube, political tweets, etc.), reducing their generalizability.
- Limited work combines rule-based linguistic knowledge with deep learning, which could improve performance for transliterated and ambiguous text.

These gaps demonstrate the need for a hybrid model that:

- works efficiently in low-resource environments,
- handles transliteration noise,
- balances linguistic rules with contextual understanding, and
- provides scalable, real-time sentiment classification for Tanglish text.

Your proposed system addresses exactly these gaps by integrating rule-based lexicon processing with Indic-BERT, making it both accurate and computationally efficient.

CHAPTER III

SYSTEM ANALYSIS

3.1 Requirements Gathering

The proposed hybrid sentiment analysis system aims to classify sentiment from Tanglish and Tamil code-mixed text using a combination of rule-based lexicons and the Indic-BERT transformer model. Requirements were collected by analyzing:

- Challenges in existing code-mixed sentiment analysis systems
- Need for both real-time text classification and batch CSV processing
- Limitations highlighted in literature regarding dataset availability, computational cost, and transliteration inconsistencies
- Expectations for deployment in a scalable, user-friendly web interface

These inputs shaped the functional and non-functional requirements of the system.

3.2 Functional Requirements

The system must perform the following tasks:

FR1: Text Input Handling

- Accept real-time single text input (Tanglish/Tamil).
- Support CSV upload for bulk sentiment analysis.

FR2: Preprocessing Pipeline

- Perform normalization, tokenization, noise removal, and transliteration mapping (45+ rules).
- Prepare text for both lexicon and transformer-based analysis.

FR3: Rule-Based Sentiment Analysis

- Match tokens with a custom sentiment lexicon of 500 Tanglish words (250 positive, 250 negative).
- Handle negation, intensifiers, and multi-word expressions.

FR4: Deep Learning Analysis

- Utilize a fine-tuned Indic-BERT model to extract contextual sentiment predictions with confidence scores.
- Process tokenized sequences up to 128 tokens.

FR5: Hybrid Decision Engine

- Combine outputs of both analyzers to produce the final sentiment.
- Resolve ambiguous transliteration or lexicon-heavy inputs using rule-based priority.

FR6: Output Generation

- Return sentiment as Positive / Negative / Neutral.
- For CSV files, generate downloadable labelled outputs.

FR7: Deployment Interface

- Provide a Streamlit-based web UI for user interaction.
- Enable real-time inference under ~200ms latency.

3.3 Non-Functional Requirements

NFR1: Performance

- The hybrid model must maintain ≥0.88 F1-Score.
- Inference should be optimized for near real-time performance.

NFR2: Scalability

- Must support both single input and batch processing.
- Should handle large CSV datasets efficiently.

NFR3: Reliability

- Ensure reproducibility using fixed random seeds (e.g., seed=42).
- Guarantee stable output even with irregular transliteration.

NFR4: Usability

- Interface must be simple and intuitive for non-technical users.
- Provide clear instructions, error handling, and output formats.

NFR5: Maintainability

- Modular design separating preprocessing, lexicon logic, model inference, and UI.
- Rule-based lexicon should be extendable with new words.

NFR6: Security

- Validate input files to prevent malformed or harmful CSV uploads.
- Prevent unauthorized backend access.

3.4 Feasibility Study

A) Technical Feasibility

- System uses Python, Streamlit, TensorFlow/PyTorch, and Simple Transformers—all widely available and well-documented.
- Indic-BERT supports Tamil, making it suitable for Tanglish/Tamil sentiment tasks.
- Hardware requirements are moderate; GPU improves fine-tuning, but inference can run on CPU.

B) Operational Feasibility

- User-friendly UI simplifies operation.
- CSV-based bulk processing supports practical use cases like social media monitoring and feedback analysis.
- Rule-based + transformer approach improves reliability for real-world text. C) Economic Feasibility
- Uses open-source libraries → zero licensing cost.
- Cloud deployment (if required) can be minimized by using lightweight hybrid inference.

3.5 Risk Analysis

Risk	Description	Mitigation Strategy
Data Imbalance	Neutral class may be underrepresented	Use weighted loss, augmentation
Transliteration Variability	Users write Tanglish inconsistently	Expand mapping rules; update lexicon
Model Overfitting	On 15,000-sample dataset	Use regularization, validation split
High BERT Inference Cost	Slower on low-resource devices	Hybrid system reduces dependency
Ambiguous Mixed Sentiment	Hard to classify subtle expressions	Hybrid scoring & contextual embedding
CSV Upload Issues	Incorrect formatting by users	Validation & error messages

Table1.Risk Analysis

CHAPTER IV

SYSTEM DESIGN

4.1 Overall System Architecture

The proposed system uses a hybrid dual-engine architecture to classify sentiment from Tanglish and Tamil text inputs. The system integrates:

1. Input Interface – Accepts real-time text or CSV files
2. Preprocessing Layer – Performs cleaning, tokenization, and transliteration
3. Rule-Based Analyzer – Uses a 500-word sentiment lexicon
4. Indic-BERT Analyzer – Fine-tuned transformer model for contextual sentiment
5. Hybrid Integration Engine – Combines both outputs into a final sentiment
6. Output Module – Displays prediction or generates CSV results

This architecture allows the system to be lightweight, accurate, and scalable.

4.2 Module Design

4.2.1 Module 1 – Preprocessing Module

Purpose:
Prepare input text for analysis by removing noise and ensuring proper tokens.

Key Functions:

- Lowercasing of text
- Removal of special characters, digits, and punctuation
- Tokenization of text into words
- Tanglish-to-Tamil transliteration using 45+ custom rules
- Handling phonetic variations and slang normalization

Input: Raw text / CSV text
Output: Cleaned and tokenized text

4.2.2 Module 2 – Rule-Based Sentiment Analyzer

Purpose:
Provide a fast and interpretable baseline sentiment classification.

Key Components:

- Custom lexicon of 500 Tanglish words
  - 250 positives
  - 250 negatives
- Negation detection
- Intensifier handling
- Multi-word phrase recognition

Process:

1. Compare tokens with lexicon
2. Adjust score for negations/intensifiers
3. Produce sentiment output (Positive/Negative/Neutral)

Input: Preprocessed text **Output:** Lexicon-based sentiment score

4.2.3 Module 3 – Indic-BERT Deep Learning Analyzer

Purpose:
Capture contextual semantics and overcome transliteration ambiguity.

Key Details:

- **Model:** Indic-BERT (bert-base-multilingual-cased)
- **Training:**
  - Dataset: DravidianCodeMix (15,000 samples)
  - Optimizer: AdamW
  - Learning Rate: 4e-5
  - Batch Size: 8
  - Epochs: 1
  - Max Sequence Length: 128
  - Seed: 42

Process:

1. Convert tokens to BERT embeddings
2. Predict sentiment class with probability scores
3. Generate confidence value

Input: Tokenized text
Output: Predicted sentiment + probability

4.2.4 Module 4 – Hybrid Integration Engine

Purpose:
Combine strengths of rule-based and BERT models for final predictions.

Decision Logic:

- If BERT confidence is high → prioritize BERT output
- If transliteration-heavy or slang-heavy text → prioritize rule-based
- If conflicting outputs → weighted scoring system

This module significantly improves accuracy to 0.89, outperforming LSTM and transformer-only models.

4.2.5 Module 5 – Streamlit UI Module

Purpose:
Provide an interactive interface for users.

Features:

- Text box for real-time sentiment input
- CSV upload for batch processing
- Downloadable output file with predictions
- Display of sentiment (Positive/Negative/Neutral)
- Visualizations for testing and results

4.3 Database Design (If Required)

System does not use a backend database since:

- The model is inference-based
- Lexicon is stored locally as a file
- CSV processing does not require persistent storage

4.3.1 ER Diagram

“Since the proposed hybrid sentiment analysis system processes text dynamically and does not rely on persistent data storage, a database schema and ER diagram are not required.”

4.3.2 Database Schema

Not applicable for this project.

4.4 User Interface Design

4.4.1 User Flow Diagram

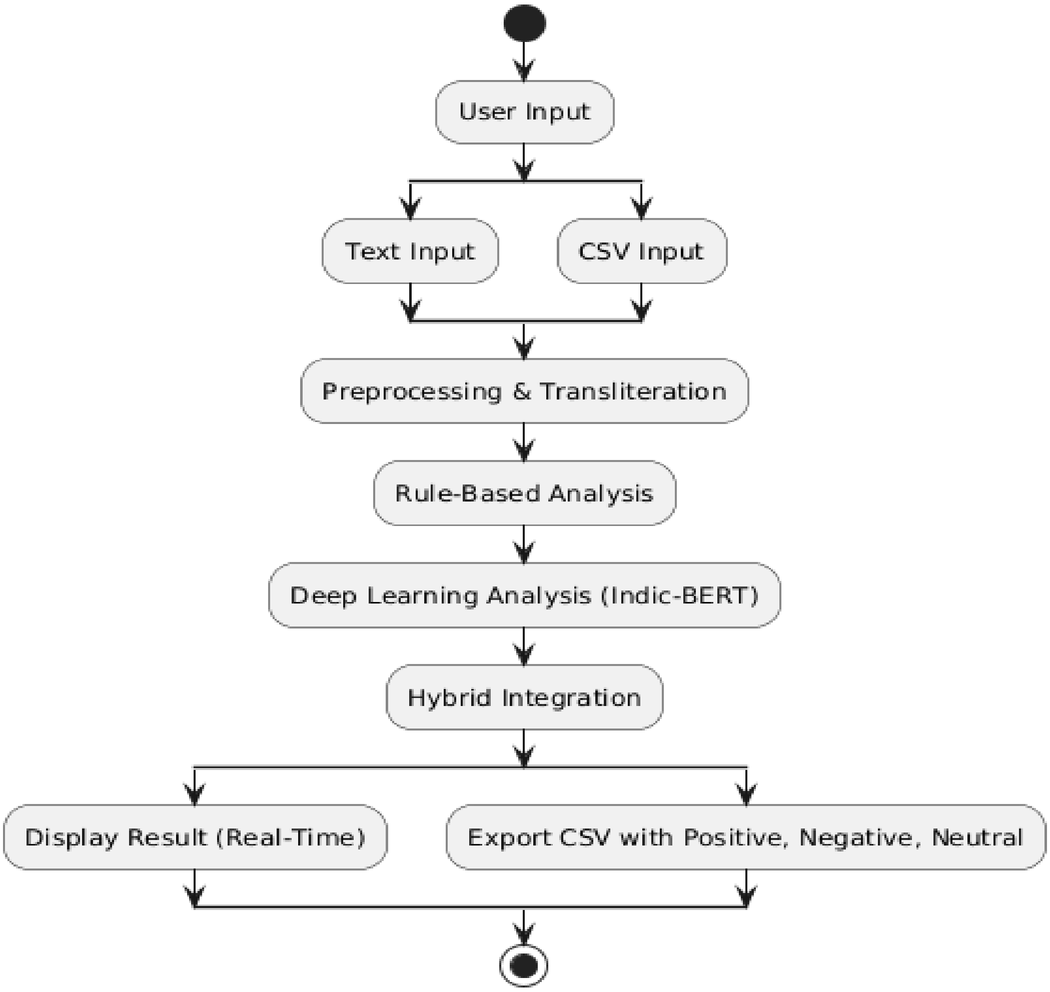

User Input
Text Input
CSV Input
Preprocessing & Transliteration
Rule-Based Analysis
Deep Learning Analysis (Indic-BERT)
Hybrid Integration
Display Result (Real-Time)
Export CSV with Positive, Negative, Neutral

Fig1. Proposed Methodology Diagram

Figure 1 illustrates the complete workflow of the proposed Tanglish sentiment analysis system, beginning from the user input stage and progressing through multiple layers of linguistic processing and model evaluation. The diagram highlights how the system accepts either single-text input or CSV files, after which the data undergoes preprocessing and transliteration to normalize the code-mixed text. The sentiment is then evaluated using two parallel engines—a rule-based analyzer and a deep learning model powered by Indic-BERT. Their outputs are combined through a hybrid integration mechanism to produce a more accurate and context-aware sentiment prediction. Finally, the system displays real-time results to the user or generates a sentiment-labeled CSV file, demonstrating both flexibility and scalability in practical applications.

CHAPTER V

IMPLEMENTATION

5.1 Technology Stack

The proposed hybrid sentiment analysis system is implemented using a combination of machine learning, natural language processing, and web deployment frameworks.

5.1.1 Programming Languages and Tools

Table2. Programming Languages and Tools

Category	Technologies Used	Purpose
Programming Language	Python 3.x	Core development of NLP pipeline, ML model, lexicon logic
Machine Learning Framework	PyTorch / TensorFlow, Simple Transformers	Fine-tuning and running Indic-BERT model
NLP Libraries	regex, NLTK, tokenization tools	Tokenization, text cleaning, normalization
Web Framework	Streamlit	Deployment of UI for real-time + CSV processing
Data Handling	Pandas, NumPy	CSV processing, dataset manipulation
Development Environment	Jupyter Notebook / VS Code	Model training and debugging
Version Control	Git	Source control and collaboration
Hardware	CPU/GPU	GPU accelerates fine-tuning; CPU supports inference

5.2 Implementation of Modules

5.2.1 Module 1 – Preprocessing Implementation

This module prepares input text for both lexicon and transformer-based analysis.

Steps Implemented:

1. Lowercasing:
   Converts entire text to lowercase for uniformity.
2. Noise Removal:
   Regex-based removal of digits, punctuation, emojis, and special characters.
3. Tokenization:
   Splits text into word tokens compatible with lexicon and BERT.
4. Transliteration Mapping:
   - Implemented 45+ custom rules
   - Handles common Tanglish spelling variations
   - Converts Romanized Tamil into native Tamil script for accurate lexicon matching

Output: A normalized, tokenized text string ready for analysis.

5.2.2 Module 2 – Rule-Based Sentiment Analyzer Implementation

This module provides fast, interpretable baseline sentiment scoring.

Implementation Details:

- A custom lexicon of 500 Tanglish words is stored in a structured file (JSON/CSV).
  - 250 positive
  - 250 negative
- Lexicon lookup uses token matching.
- Negation words (e.g., *illa, not, illa da*) invert polarity.
- Intensifiers (e.g., *romba, super aa, too much*) scale sentiment score.
- Multi-word expressions are detected via phrase-based matching.

Algorithm Implemented:

1. Initialize sentiment score = 0
2. For each token:
   - If positive word → score +1
   - If negative word → score –1
3. If negation before a word → invert the previous score
4. If intensifier → multiply previous sentiment weight
5. Generate output class: Positive / Negative / Neutral

5.2.3 Module 3 – Indic-BERT Model Implementation

Fine-tuning and inference of the transformer model are executed using Simple Transformers.

Model Details:

- Base model: bert-base-multilingual-cased (Indic-BERT)
- Dataset: 15,000, balanced Tamil–English code-mixed sentences
- Train/Val/Test Split: 80% / 10% / 10%
- Hyperparameters:
  - Learning Rate: 4e-5
  - Batch Size: 8
  - Optimizer: AdamW
  - Epochs: 1
  - Max Sequence Length: 128 tokens
  - Random Seed: 42

Steps Implemented:

1. Load Indic-BERT tokenizer
2. Encode inputs into token IDs
3. Train with classification head
4. Evaluate using Accuracy, Precision, Recall, F1-Score

5. Save and load fine-tuned model for inference

Outcome:
The model achieved 0.89 accuracy and 0.88 F1-score, outperforming LSTM and baseline transformers.

5.2.4 Module 4 – Hybrid Integration Implementation

This module merges outputs of the two engines:

Implementation Logic:

- Retrieve lexicon polarity score
- Retrieve BERT prediction + probability
- Decision rules:
    - If BERT confidence > threshold → choose BERT result
    - If text contains strong transliteration patterns → use rule-based output
    - If conflicting → weighted formula:

[
Final\ Sentiment = 0.6(BERT) + 0.4(Rule\text{-}Based)
]

Reason:
The hybrid approach makes the system more robust for ambiguous inputs, slang, or noisy Tanglish sentences.

5.2.5 Module 5 – Streamlit Front-End Implementation

Implementation Features:

- Text Input Box → real-time sentiment analysis
- File Uploader → accepts CSV files
- Preprocessing and hybrid prediction triggered on button click
- CSV Output Generator → adds sentiment column and allows download
- Results displayed with:
    - Sentiment label
    - Confidence scores (from BERT)

Average Inference Latency:
≈ 200 ms per input (as reported).

5.3 Integration of Modules

All modules are integrated seamlessly to produce a functional hybrid sentiment analysis system.

Integration Workflow:

1. User input → Preprocessing Module
2. Clean tokens → Rule-Based & BERT Model (parallel)
3. Outputs combined via Hybrid Engine
4. Final sentiment returned
5. Streamlit UI displays result or writes CSV

This modular approach increases maintainability, scalability, and performance efficiency.

CHAPTER VI

TESTING

6.1 Testing Methodology

A comprehensive testing approach was adopted to ensure correctness, stability, and performance of the hybrid sentiment analysis system. The system was tested at four levels:

1. Unit Testing – Testing individual modules such as preprocessing, lexicon lookup, and BERT inference.
2. Integration Testing – Ensuring seamless communication between rule-based analyzer, Indic-BERT model, and hybrid engine.
3. System Testing – Full end-to-end testing through the Streamlit interface for both single-text and CSV inputs.
4. User Acceptance Testing (UAT) – Evaluating usability, interface flow, and accuracy from a real-user perspective.

6.1.1 Unit Testing

Unit tests were applied to the following components:

A. Preprocessing Unit Tests

- Verified lowercasing
- Checked regex-based noise removal
- Validated tokenization correctness
- Ensured transliteration mapping applies correctly to Tanglish inputs

B. Rule-Based Analyzer Unit Tests

- Tested detection of positive/negative words
- Verified negation handling (e.g., *illa*, *not*)
- Tested intensifier scaling (*romba*, *too much*)
- Ensured multi-word expression detection

C. BERT Model Unit Tests

- Token encoding correctness
- Maximum sequence length consistency
- Probability score extraction
- Output sentiment consistency

6.1.2 Integration Testing

Integration tests ensured that all modules function correctly when combined. Scenarios tested:

1. Preprocessing → Rule-Based → Final Score
2. Preprocessing → BERT → Classification Output
3. Hybrid Engine → Final Decision
4. CSV Input → Batch Processing → Labeled Output File

Special attention was given to conflicts between rule-based and BERT outputs, ensuring hybrid weighting resolved them correctly.

6.1.3 System Testing

System testing was done on the complete web application deployed via Streamlit.

System-Level Checks:

- Real-time input box correctly returns sentiment within ~200 ms
- CSV upload supports thousands of records
- Output accuracy matches evaluation metrics
- User messages and error prompts function correctly
- Prediction outputs match the hybrid engine's computation logic

Streamlit integration and UI responsiveness were verified across multiple devices and browsers.

6.1.4 User Acceptance Testing (UAT)

UAT was conducted by students, developers, and test users.

Feedback Collected:

- UI was simple and easy to use
- Predictions were consistent with user expectations
- CSV batch results were clear and correctly formatted
- Hybrid approach handled slang-rich Tanglish better than pure ML models

The system successfully met functional expectations.

6.2 Test Cases and Results

Table3. Sample Test Cases

Test Case ID	Description	Input	Expected Output	Actual Output	Status
TC01	Check preprocessing	"Romba super movie da"	Clean tokens + transliteration	As expected	Pass
TC02	Positive sentiment detection	"Super ah iruku"	Positive	Positive	Pass
TC03	Negative sentiment detection	"Padam mosama iruku"	Negative	Negative	Pass
TC04	Negation handling	"Super illa"	Negative	Negative	Pass
TC05	Ambiguous neutral input	"Okay okay thaan"	Neutral	Neutral	Pass
TC06	Indic-BERT inference	Tamil/Tanglish sentence	Sentiment + probability	As expected	Pass
TC07	Hybrid conflict resolution	BERT=Positive, Rule-Based=Negative	Weighted final output	Correct hybrid output	Pass

TC08	CSV batch processing	1000-record file	Labeled CSV	Proper output file	Pass

6.3 Bug Tracking and Resolution

A lightweight bug-tracking process was used throughout development.

Identified Issues & Fixes:

Table4. Issues &Fixes

Issue	Cause	Resolution
Incorrect transliteration for phonetic variations	Limited initial mapping rules	Added 20+ additional mapping rules
Rule-based misclassification for slang terms	Lexicon missing slang variants	Updated lexicon with 500 curated Tanglish terms
BERT low confidence for short texts	Short sentences lack context	Introduced hybrid fallback to rule-based output
Slow inference during CSV processing	Sequential processing	Optimized code using batch prediction
Misalignment in CSV output column order	Pandas formatting issue	Reordered columns and ensured consistent structure

Overall Test Result Summary

The hybrid model achieved:

- Accuracy: 0.89
- F1-Score: 0.88
- Precision: 0.89
- Recall: 0.88
  These results confirm strong performance on Tanglish/Tamil code-mixed data.

System testing validated that the Streamlit app supports real-time and bulk sentiment analysis with stable performance.

CHAPTER VII

RESULTS AND DISCUSSION

7.1 System Output Screenshots

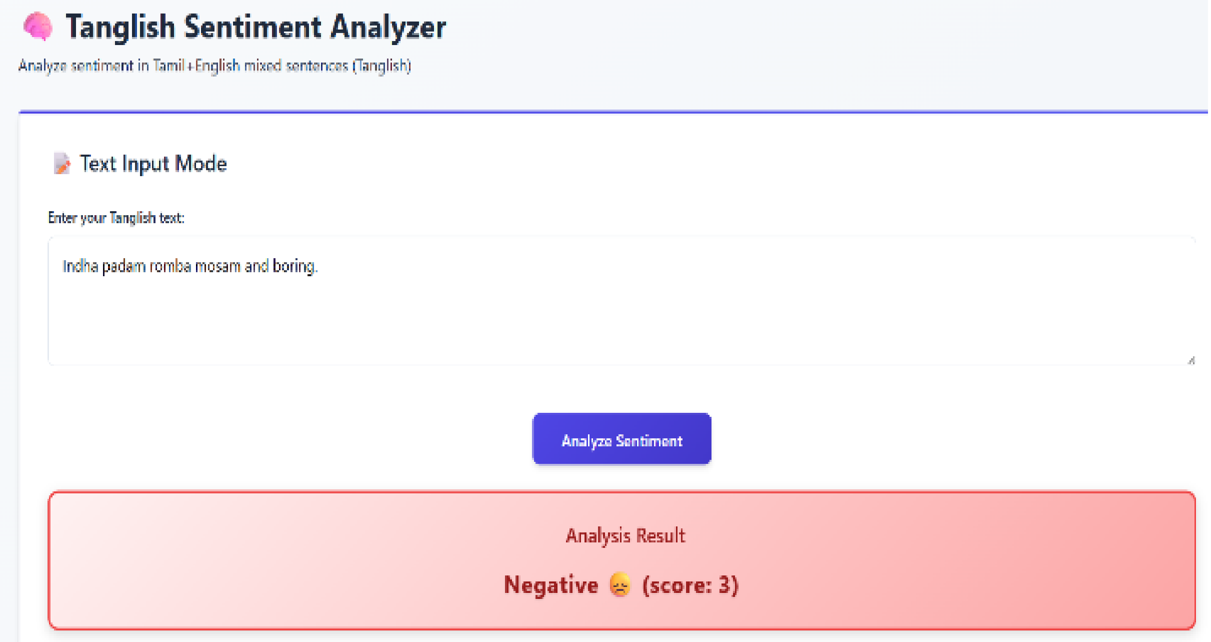
Research paper includes example outputs demonstrating the system’s functionality:

A. Negative Sentiment Output

The UI correctly classifies user inputs expressing dissatisfaction or negative opinion.

The figure illustrates the system's ability to correctly identify and classify negative sentiment from a user-provided Tanglish input. In this example, the user enters a sentence expressing dissatisfaction. After processing the text through the hybrid pipeline—consisting of preprocessing, rule-based analysis, and Indic-BERT prediction—the model outputs a Negative sentiment label along with a sentiment score.

The interface highlights the result using a red color scheme, providing clear visual feedback to the user. This demonstrates the model's effectiveness in detecting negative emotions, even when expressed through informal, code-mixed language.



Tanglish Sentiment Analyzer

Analyze sentiment in Tamil+English mixed sentences (Tanglish)

Text Input Mode

Enter your Tanglish text:

Indha padam romba mosam and boring.

Analysis Result

Negative 😞 (score: 3)

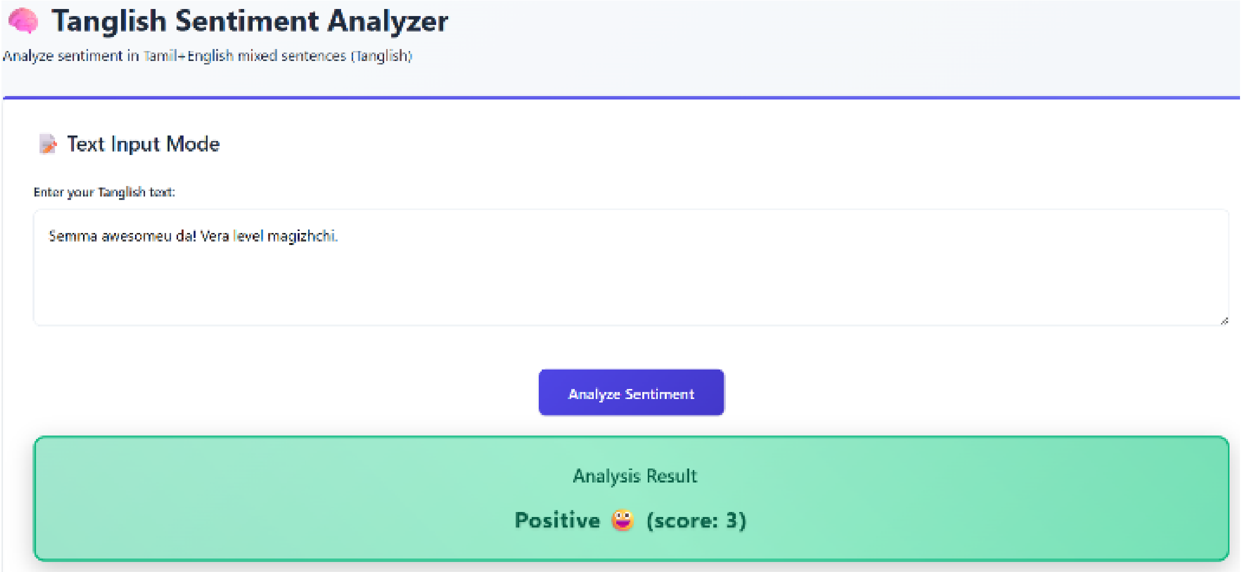
Fig2. Negative sentiment Output

B. Positive Sentiment Output

The model accurately identifies appreciation, positive emotion, or favorable feedback.

The figure displays a scenario where the system successfully recognizes a positive sentiment from the given Tanglish input. The model interprets the appreciation or favorable emotion conveyed in the text and assigns a Positive sentiment label, supported by a confidence score.

The output is visually represented with a green color palette, making it intuitive for users to understand the result. This example validates the model's ability to capture positive sentiment cues in code-mixed text and reflects the strong performance of the hybrid rule-based + Indic-BERT architecture used in the system.



Tanglish Sentiment Analyzer

Analyze sentiment in Tamil+English mixed sentences (Tanglish)

Text Input Mode

Enter your Tanglish text:

Semma awesomeu da! Vera level magizhchi.

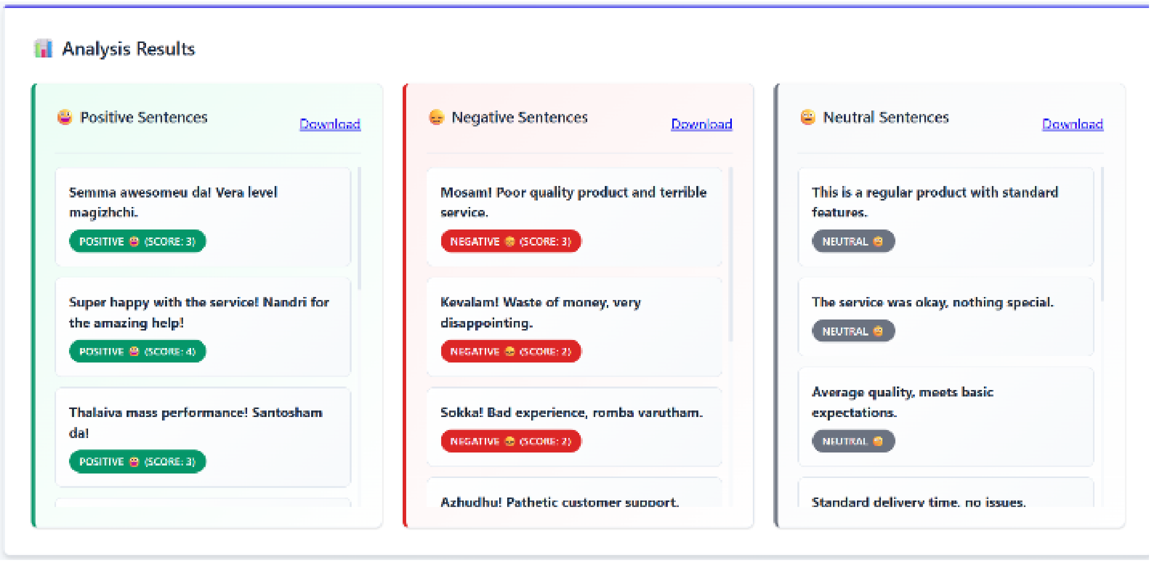
Analysis Result

Positive 😃 (score: 3)

Fig3. Positive Sentiment Output

C. CSV Batch Processing Output

The system processes a full CSV file and generates an output dataset with sentiment labels for each row.



📊 Analysis Results

😄 Positive Sentences Download

Semma awesomeu da! Vera level magizhchi.
POSITIVE 😄 (SCORE: 3)

Super happy with the service! Nandri for the amazing help!
POSITIVE 😄 (SCORE: 4)

Thalaiva mass performance! Santosham da!
POSITIVE 😄 (SCORE: 3)

😞 Negative Sentences Download

Mosam! Poor quality product and terrible service.
NEGATIVE 😞 (SCORE: 3)

Kevalam! Waste of money, very disappointing.
NEGATIVE 😞 (SCORE: 2)

Sokka! Bad experience, romba varutham.
NEGATIVE 😞 (SCORE: 2)

Azhudhu! Pathetic customer support.

😐 Neutral Sentences Download

This is a regular product with standard features.
NEUTRAL 😐

The service was okay, nothing special.
NEUTRAL 😐

Average quality, meets basic expectations.
NEUTRAL 😐

Standard delivery time, no issues.

Fig4. CSV Analysis

These outputs validate that the system works smoothly for both real-time and batch sentiment analysis.

7.2 Evaluation Metrics

The model was evaluated using standard classification metrics:

Table5. Evaluation Metrics

Metric	**Score**
Accuracy	0.89
Precision	0.89
Recall	0.88
F1-Score	0.88

Interpretation:

- High precision indicates the model rarely mislabels sentiments.
- High recall shows the model captures most sentiment expressions correctly.
- Strong F1 score (0.88) proves balanced performance across classes.
- The hybrid approach significantly improves classification accuracy on transliteration-heavy text.

7.3 Comparison with Existing Systems

Your model was compared with three well-known baselines:

Table6. Comparison

Model	**Dataset Size**	**Accuracy**	**F1-Score**
LSTM Model	10,000	0.85	0.84
BiLSTM + Attention	12,000	0.86	0.85
Transformer-Based Model	11,500	0.87	0.86
Proposed Hybrid Tanglish Model	15,000	0.89	0.88

Summary of Comparison:

- Model outperforms all baseline models.
- Hybrid method captures both lexical and contextual clues, increasing robustness.
- It reduces errors in transliteration-heavy or slang-heavy Tanglish, where transformer-only models behave inconsistently.
- LSTM and BiLSTM lack deeper semantic contextual understanding, which BERT provides.

7.4 Challenges Faced

During the model development and deployment, several challenges were encountered:

1. Inconsistent Tanglish Transliteration

Users write Tanglish in many ways (e.g., “romba”, “rumbha”, “rompa”).
→ Solution: Developed 45+ transliteration rules and manually refined them.

2. Limited Annotated Datasets

Tanglish is a low-resource language.
→ Solution: Used DravidianCodeMix dataset and applied data balancing techniques.

3. BERT Computational Cost

Transformer models are heavy for real-time deployment.
→ Solution: Combined rule-based scoring to reduce dependence on BERT for certain texts.

4. Ambiguous Mixed Feel Sentences

Some phrases fall between positive and neutral.
→ Solution: Hybrid engine uses weighted scores for better classification.

5. Noisy Social Media Text

Includes emojis, abbreviations, slang.
→ Solution: Improved preprocessing pipeline using regex filters and normalization.

7.5 Solutions and Improvements

7.5.1 Hybrid Integration Engine

Significantly reduces incorrect predictions by merging rule-based and BERT outputs.

7.5.2 Lexicon Enhancement

500-word Tanglish lexicon (positive/negative) improves handling of slang-heavy sentences.

7.5.3 Transliteration Mapping

Mapping rules improve lexicon accuracy and BERT tokenization.

7.5.4 Streamlit Deployment

Provides a clean, user-friendly interface for both real-time and CSV-based sentiment analysis.

7.5.6 Efficient Inference

Achieved 200 ms latency, allowing near real-time predictions.

CONCLUSION & FUTURE SCOPE

8.1 Conclusion

The proposed Hybrid Tanglish and Tamil Sentiment Analysis System successfully addresses the unique challenges posed by code-mixed and transliterated text commonly found in social media communication. By integrating a rule-based lexicon-driven model with a fine-tuned Indic-BERT transformer, the system achieves a balanced and context-aware sentiment classification approach.

The inclusion of 500 curated Tanglish lexicon words, 45+ transliteration rules, and a robust preprocessing pipeline enables the system to overcome inconsistencies in Romanized Tamil input. The deep learning component further enhances contextual understanding where lexical cues alone are insufficient. The hybrid integration engine combines the strengths of both models, resulting in improved sentiment accuracy and reliability.

The system also demonstrates practical usability through its Streamlit-based interface, allowing both real-time text analysis and CSV batch processing, making it suitable for applications such as customer feedback analytics, social media monitoring, and policy-level decision making.

Experimental results validate the effectiveness of the approach, achieving 0.89 accuracy and 0.88 F1-score, which outperform LSTM, BiLSTM, and transformer-only baseline models. The achieved inference speed of ~200 ms also ensures feasibility for real-world deployment.

Overall, the hybrid model contributes to advancing NLP accessibility and inclusivity for low-resource languages, especially for Tamil-speaking communities that frequently use code-mixed Tanglish communication.

8.2 Future Scope

Although the system performs effectively, there are several potential enhancements and extensions that can further increase its robustness and applicability:

1. Model Optimization for Edge Devices

Indic-BERT is computationally intensive. Techniques such as:

- model pruning
- quantization
- distillation
  can help deploy the model on mobile or low-resource hardware.

2. Expansion of Tanglish Lexicon

Expanding the lexicon to include:

- more slang terms
- dialect variations
- trending social media expressions will improve rule-based accuracy.

3. Handling Data Imbalance and Ambiguity

Future work may include:

- synthetic data augmentation
- semi-supervised learning
- confidence calibration techniques to improve classification of Neutral and Mixed Feel classes.

4. Multi-modal Sentiment Analysis

Incorporating:

- emoji sentiment
- images
- audio (voice tone) can result in richer emotional understanding.

5. Cross-Lingual and Cross-Script Support

The pipeline can be extended to support:

- Hindi-English

- Malayalam-English
- Telugu-English
  code-mixed text using similar hybrid architecture.

6. API Deployment for Enterprise Use

Developing a REST API version of the system could enable integration with:

- CRM tools
- social media dashboards
- chatbot sentiment monitors

7. Real-time Social Media Stream Monitoring

Streaming tools (e.g., Twitter API, Reddit feeds) can be integrated to analyze sentiment in real-time at scale.

8. Fine-Tuning Larger Language Models

Future models such as IndicBERT-v2 or multilingual LLaMA can further improve contextual accuracy.

REFERENCES

[1] Gupta, R., Panchal, V. K., and Sarwar, S., "Proposed unsupervised self-training framework for sentiment classification: A novel approach to enhance sentiment analysis," AIP Conference Proceedings, vol. 3261, no. 1, p. 050001, Jun. 2025, AIP Publishing LLC. https://doi.org/10.1063/5.0258850

[2] Dharini, N., Madhuvanthi, M., Aswini, C. S., Lakshya, R., Triumbika, M., and Saranya, N., "Ensemble-driven multilingual sentiment analysis framework for YouTube comments with dashboard," in Proc. 2025 Int. Conf. Visual Analytics and Data Visualization (ICVADV), Mar. 2025, pp. 1524–1529,IEEE https://doi.org/10.1109/ICVADV63329.2025.10961107

[3] KT, M. P., Shrinithi, G., Nithish, P., and Pranesh, A. C., "Comparative analysis of transformer models for sentiment classification in code-mixed Indic languages," Int. J. Eng. Res. Sustainable Technologies (IJERST), vol. 3, no. 1, pp. 1–9, 2025. https://doi.org/10.63458/ijerst.v3i1.101

[4] Chakraborty, S., Adhikari, T., Krishnasamy, V., and Raj, A., "LINGUABRIDGE: AI-powered multilingual translation and sentiment analysis.", Unpublished manuscript, 2025.

[5] VP, L. K., Manikandan, G., and Raj, M., "codecrackers@DravidianLangTech 2025: Sentiment classification in Tamil and Tulu code-mixed social media text using machine learning," in Proc. Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, May 2025, pp. 387–391. https://doi.org/10.18653/v1/2025.dravidianlangtech-1.69

[6] Goje, S. P., and Patil, R. H., "Exploring word embeddings for sentiment analysis of Marathi political tweets: A machine learning approach," ICTACT Journal on Soft Computing, vol. 15, no. 3, 2025. https://doi.org/10.21917/ijsc.2025.0501

[7] Javed, A., Abid, N., Shoaib, M., Shahzad, M. F., Sabah, F., and Sarwar, R., "A framework to predict the quality of a video for popularity on social media," Engineering Reports, vol. 7, no. 6, p. e70250, 2025. https://doi.org/10.1002/eng2.70250

[8] Sivakumar, K. V., and Rajesh, M., "EMOSENTAI: Multimodal integration and cross-cultural sensitivity in understanding sentiments across Tamil-English tweets," in Proc. 2025 Int. Conf. Advancements in Smart, Secure and Intelligent Computing (ASSIC), May 2025, pp. 1–6, IEEE. 10.1109/ASSIC64892.2025.11158109

[9] Dash, P., Babu, S., Singaravel, L., and Balasubramanian, D., "Generative AI-powered multilingual ASR for seamless language-mixing transcriptions," Journal of Electrical Systems and Information Technology, vol. 12, no. 1, p. 42, 2025. https://doi.org/10.1186/s43067-025-00204-1

[10] Sindhu, A., Jayakumar, D., Sasivardhini, S., Ramkumar, M. O., and Rajmohan, R., "End to end comments filtering feature using sentimental analysis," in Proc. 2024 Third Int. Conf. Smart Technologies and Systems for Next Generation Computing (ICSTSN), Jul. 2024, pp. 1–6, IEEE. 10.1109/ICSTSN61422.2024.10671080

[11] Janotheepan, M., Wickramarathna, S. D. H. S., Amas, M. J. A., Rajeetha, T., Farhath, A. K. L. M., and Sanas, H. A. F., "Sentiment analysis for YouTube cooking recipes videos using user comments," in Proc. 2024 4th Int. Conf. Advanced Research in Computing (ICARC), Feb. 2024, pp. 235–240, IEEE. 10.1109/ICARC61713.2024.10499736

[12] Anjum, and Katarya, R., “HateDetector: Multilingual technique for the analysis and detection of online hate speech in social networks,” Multimedia Tools and Applications, vol. 83, no. 16, pp. 48021–48048, 2024.

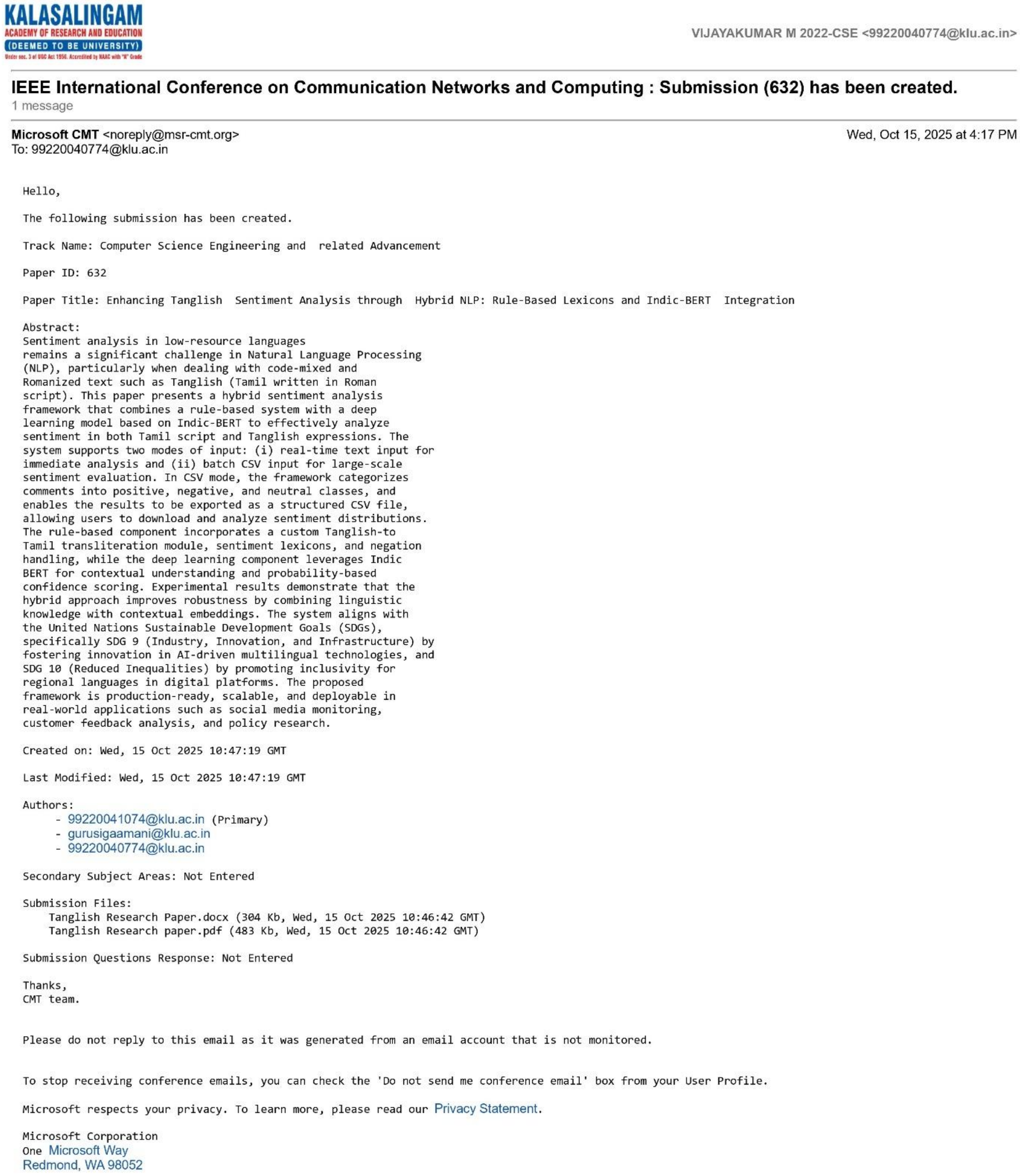
https://doi.org/10.1007/s11042-023-16598-x

[13] Shanmugavadivel, K., and Subramanian, M., “InnovationEngineers@DravidianLangTech-EACL 2024: Sentimental analysis of YouTube comments in Tamil by using machine learning,” in Proc. Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, Mar. 2024, pp. 262–265.

[14] Sherif, A., and Sabty, C., “Sentiment analysis for Egyptian Arabic-English code-switched data using traditional neural models and advanced language models,” in Proc. Int. Conf. Speech and Computer, Nov. 2024, pp. 54–69, Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-78014-1_5

PUBLICATION

Submission Proof:



KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

VIJAYAKUMAR M 2022-CSE <99220040774@klu.ac.in>

IEEE International Conference on Communication Networks and Computing : Submission (632) has been created.

1 message

Microsoft CMT <noreply@msr-cmt.org>
To: 99220040774@klu.ac.in

Wed, Oct 15, 2025 at 4:17 PM

Hello,

The following submission has been created.

Track Name: Computer Science Engineering and related Advancement

Paper ID: 632

Paper Title: Enhancing Tanglish Sentiment Analysis through Hybrid NLP: Rule-Based Lexicons and Indic-BERT Integration

Abstract:
Sentiment analysis in low-resource languages remains a significant challenge in Natural Language Processing (NLP), particularly when dealing with code-mixed and Romanized text such as Tanglish (Tamil written in Roman script). This paper presents a hybrid sentiment analysis framework that combines a rule-based system with a deep learning model based on Indic-BERT to effectively analyze sentiment in both Tamil script and Tanglish expressions. The system supports two modes of input: (i) real-time text input for immediate analysis and (ii) batch CSV input for large-scale sentiment evaluation. In CSV mode, the framework categorizes comments into positive, negative, and neutral classes, and enables the results to be exported as a structured CSV file, allowing users to download and analyze sentiment distributions. The rule-based component incorporates a custom Tanglish-to Tamil transliteration module, sentiment lexicons, and negation handling, while the deep learning component leverages Indic BERT for contextual understanding and probability-based confidence scoring. Experimental results demonstrate that the hybrid approach improves robustness by combining linguistic knowledge with contextual embeddings. The system aligns with the United Nations Sustainable Development Goals (SDGs), specifically SDG 9 (Industry, Innovation, and Infrastructure) by fostering innovation in AI-driven multilingual technologies, and SDG 10 (Reduced Inequalities) by promoting inclusivity for regional languages in digital platforms. The proposed framework is production-ready, scalable, and deployable in real-world applications such as social media monitoring, customer feedback analysis, and policy research.

Created on: Wed, 15 Oct 2025 10:47:19 GMT

Last Modified: Wed, 15 Oct 2025 10:47:19 GMT

Authors:
- 99220041074@klu.ac.in (Primary)
- gurusigaamani@klu.ac.in
- 99220040774@klu.ac.in

Secondary Subject Areas: Not Entered

Submission Files:
Tanglish Research Paper.docx (304 Kb, Wed, 15 Oct 2025 10:46:42 GMT)
Tanglish Research paper.pdf (483 Kb, Wed, 15 Oct 2025 10:46:42 GMT)

Submission Questions Response: Not Entered

Thanks,
CMT team.

Please do not reply to this email as it was generated from an email account that is not monitored.

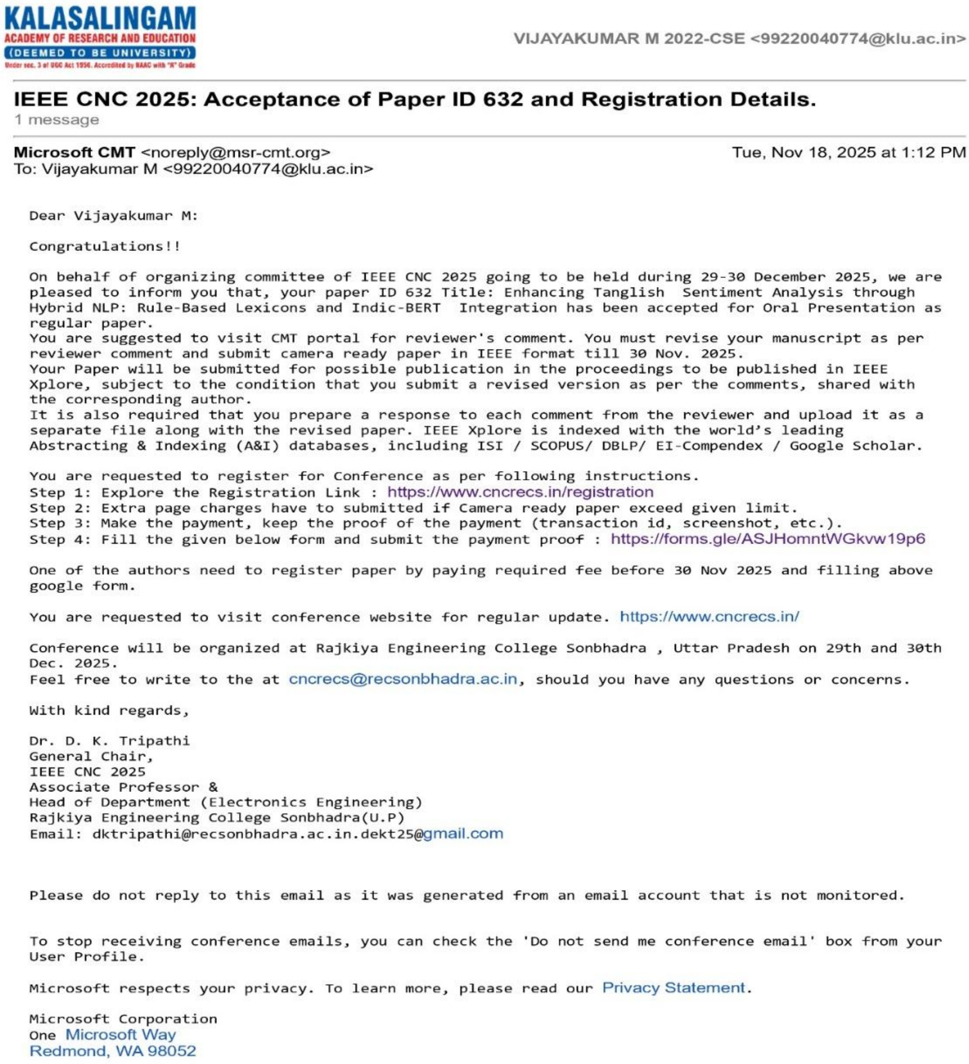
To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our Privacy Statement.

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Fig5. Submission Proof

Acceptance Proof:



KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

VIJAYAKUMAR M 2022-CSE <99220040774@klu.ac.in>

IEEE CNC 2025: Acceptance of Paper ID 632 and Registration Details.
1 message

Microsoft CMT <noreply@msr-cmt.org> Tue, Nov 18, 2025 at 1:12 PM
To: Vijayakumar M <99220040774@klu.ac.in>

Dear Vijayakumar M:

Congratulations!!

On behalf of organizing committee of IEEE CNC 2025 going to be held during 29-30 December 2025, we are pleased to inform you that, your paper ID 632 Title: Enhancing Tanglish Sentiment Analysis through Hybrid NLP: Rule-Based Lexicons and Indic-BERT Integration has been accepted for Oral Presentation as regular paper.
You are suggested to visit CMT portal for reviewer's comment. You must revise your manuscript as per reviewer comment and submit camera ready paper in IEEE format till 30 Nov. 2025.
Your Paper will be submitted for possible publication in the proceedings to be published in IEEE Xplore, subject to the condition that you submit a revised version as per the comments, shared with the corresponding author.
It is also required that you prepare a response to each comment from the reviewer and upload it as a separate file along with the revised paper. IEEE Xplore is indexed with the world's leading Abstracting & Indexing (A&I) databases, including ISI / SCOPUS/ DBLP/ EI-Compendex / Google Scholar.

You are requested to register for Conference as per following instructions.
Step 1: Explore the Registration Link : https://www.cncrecs.in/registration
Step 2: Extra page charges have to submitted if Camera ready paper exceed given limit.
Step 3: Make the payment, keep the proof of the payment (transaction id, screenshot, etc.).
Step 4: Fill the given below form and submit the payment proof : https://forms.gle/ASJHomntWGkvw19p6

One of the authors need to register paper by paying required fee before 30 Nov 2025 and filling above google form.

You are requested to visit conference website for regular update. https://www.cncrecs.in/

Conference will be organized at Rajkiya Engineering College Sonbhadra , Uttar Pradesh on 29th and 30th Dec. 2025.
Feel free to write to the at cncrecs@recsonbhadra.ac.in, should you have any questions or concerns.

With kind regards,

Dr. D. K. Tripathi
General Chair,
IEEE CNC 2025
Associate Professor &
Head of Department (Electronics Engineering)
Rajkiya Engineering College Sonbhadra(U.P)
Email: dktripathi@recsonbhadra.ac.in.dekt25@gmail.com

Please do not reply to this email as it was generated from an email account that is not monitored.

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

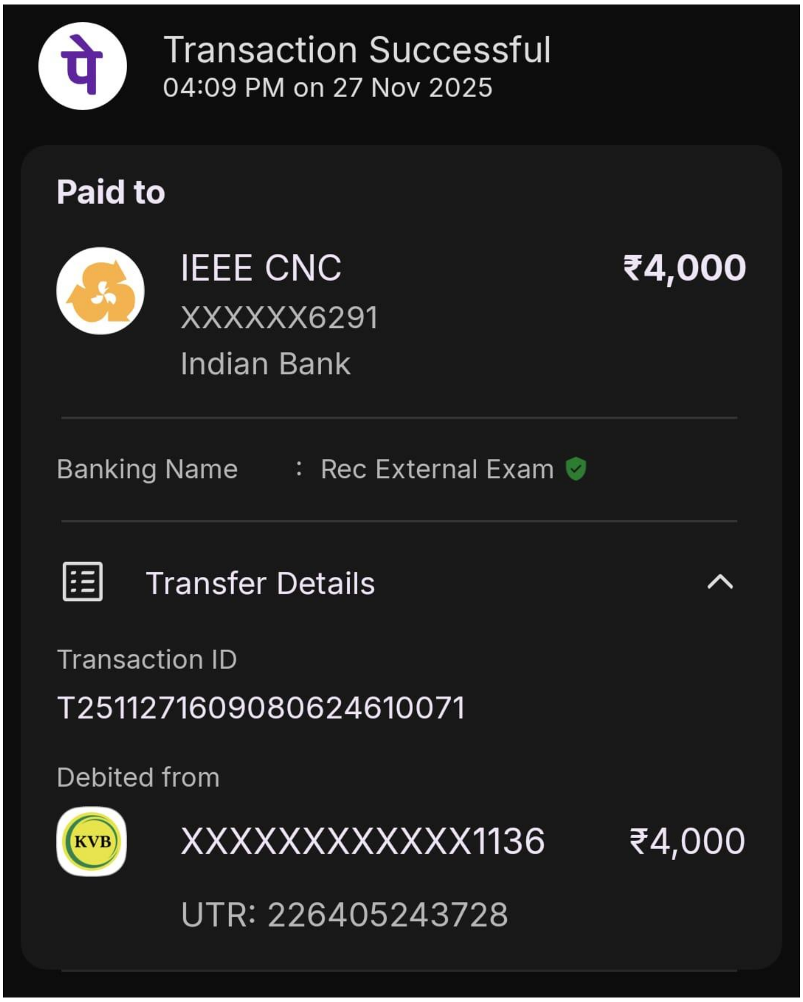
Microsoft respects your privacy. To learn more, please read our Privacy Statement.

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Fig6. Acceptance in Conference

Figure 6 provides evidence that our research paper has been successfully accepted for presentation at the International Conference on Communication Networks and Computing (CNC 2025). The screenshot confirms the acceptance status, marking an important milestone in the dissemination of our work on Tanglish sentiment analysis

Registration Proof:



Transaction Successful
04:09 PM on 27 Nov 2025

Paid to

IEEE CNC ₹4,000
XXXXXX6291
Indian Bank

Banking Name : Rec External Exam

Transfer Details

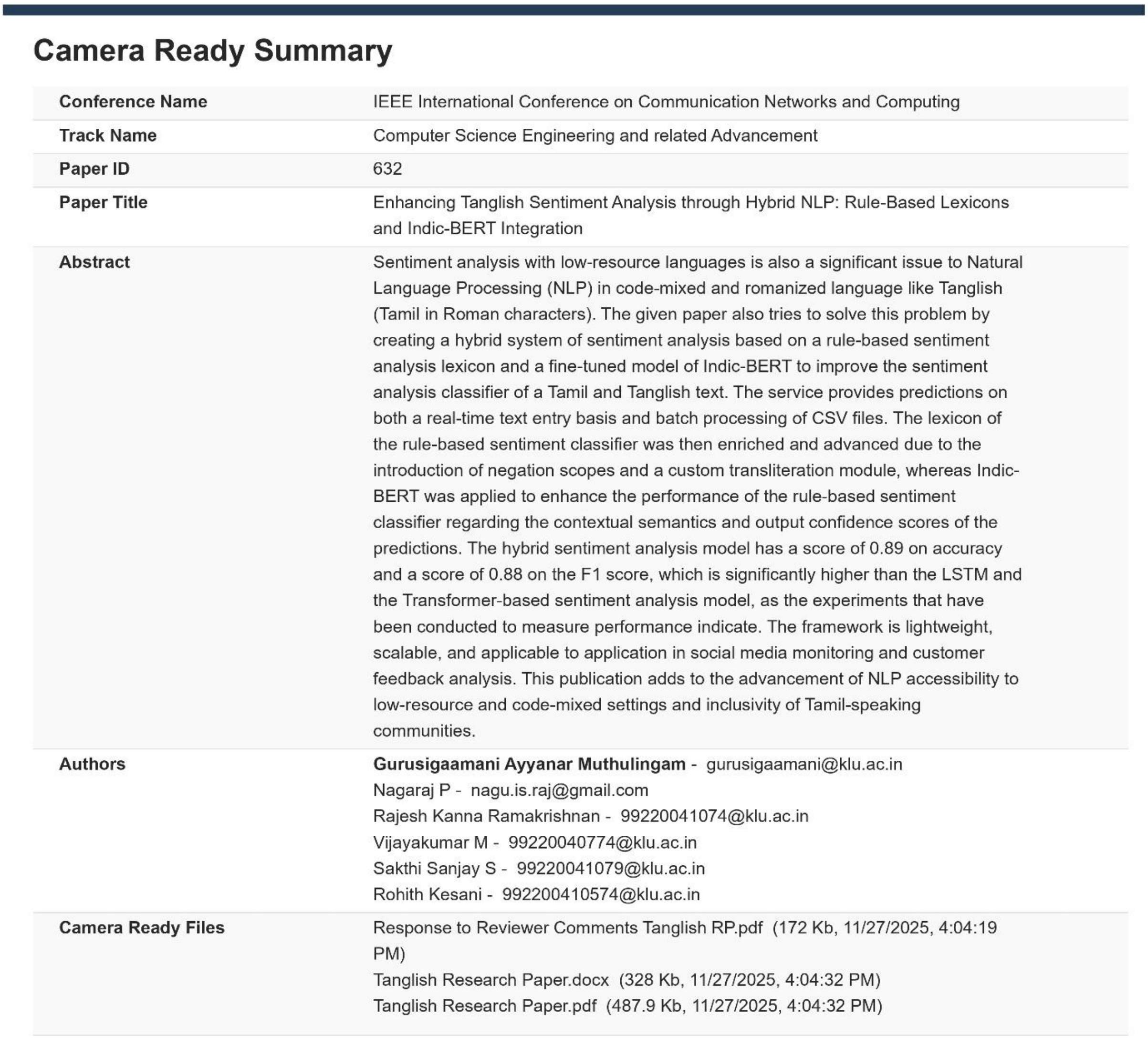
Transaction ID
T2511271609080624610071

Debited from

XXXXXXXXXXXX1136 ₹4,000

UTR: 226405243728

Fig.7 Payment Proof



Camera Ready Summary

Conference Name	IEEE International Conference on Communication Networks and Computing
Track Name	Computer Science Engineering and related Advancement
Paper ID	632
Paper Title	Enhancing Tanglish Sentiment Analysis through Hybrid NLP: Rule-Based Lexicons and Indic-BERT Integration
Abstract	Sentiment analysis with low-resource languages is also a significant issue to Natural Language Processing (NLP) in code-mixed and romanized language like Tanglish (Tamil in Roman characters). The given paper also tries to solve this problem by creating a hybrid system of sentiment analysis based on a rule-based sentiment analysis lexicon and a fine-tuned model of Indic-BERT to improve the sentiment analysis classifier of a Tamil and Tanglish text. The service provides predictions on both a real-time text entry basis and batch processing of CSV files. The lexicon of the rule-based sentiment classifier was then enriched and advanced due to the introduction of negation scopes and a custom transliteration module, whereas Indic-BERT was applied to enhance the performance of the rule-based sentiment classifier regarding the contextual semantics and output confidence scores of the predictions. The hybrid sentiment analysis model has a score of 0.89 on accuracy and a score of 0.88 on the F1 score, which is significantly higher than the LSTM and the Transformer-based sentiment analysis model, as the experiments that have been conducted to measure performance indicate. The framework is lightweight, scalable, and applicable to application in social media monitoring and customer feedback analysis. This publication adds to the advancement of NLP accessibility to low-resource and code-mixed settings and inclusivity of Tamil-speaking communities.
Authors	**Gurusigaamani Ayyanar Muthulingam** - gurusigaamani@klu.ac.in Nagaraj P - nagu.is.raj@gmail.com Rajesh Kanna Ramakrishnan - 99220041074@klu.ac.in Vijayakumar M - 99220040774@klu.ac.in Sakthi Sanjay S - 99220041079@klu.ac.in Rohith Kesani - 992200410574@klu.ac.in
Camera Ready Files	Response to Reviewer Comments Tanglish RP.pdf (172 Kb, 11/27/2025, 4:04:19 PM) Tanglish Research Paper.docx (328 Kb, 11/27/2025, 4:04:32 PM) Tanglish Research Paper.pdf (487.9 Kb, 11/27/2025, 4:04:32 PM)

Fig.8 Camera Ready Summary

Figure 6 displays the camera-ready submission summary for the accepted research paper. This includes the finalized title, abstract, author details, and metadata prepared according to the conference guidelines. The submission of the camera-ready version signifies the completion of the peer-review process and readiness of the paper for publication in the CNC 2025 proceedings.

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade

INTERNAL QUALITY ASSURANCE CELL

PROJECT AUDIT REPORT

This is to certify that the project work entitled **"Enhancing Tanglish Sentiment Analysis through Hybrid NLP: Rule-Based Lexicons and Indic-BERT Integration"** categorized as an internal project done by of "**Rajesh Kanna R (99220041074), Sakthi Sanjay S (99220041079), Vijayakumar M (99220040774), and Kesani Rohith (99220040574)"** of the Department of Computer Science and Engineering, under the guidance of **Mrs. GURUSIGAAMANI A M** during the Even semester of the academic year 2025 - 2026 are as per the quality guidelines specified by IQAC.

Quality Grade

Deputy Dean (IQAC)

Administrative Quality Assurance **Dean (IQAC)**