

# **RACISM DETECTION IN TWEETS USING STACKED GCR-NN WITH SENTIMENT ANALYSIS**

Mrs.K.Helini<sup>1</sup>, M.Srijana<sup>2</sup>, M.Sri Varsha<sup>3</sup>, M.Sathwika<sup>4</sup>, K.Manogna<sup>5</sup>

## **Abstract:**

Racism detection on social media, particularly on Twitter, has become a critical challenge due to the volume of harmful and discriminatory content. This paper presents a system using Stacked GCR-NN (Gated Convolutional Recurrent Neural Networks) combined with Sentiment Analysis to classify tweets into three categories: Not Racist, Direct Racism, and Indirect Racism. Convolutional layers capture local linguistic patterns while recurrent layers model sequential dependencies. A sentiment analysis layer further enhances detection by identifying negative emotional tones. The system achieves strong performance in distinguishing both explicit and implicit racist content.

**Keywords:** Racism Detection, Sentiment Analysis, Stacked GCR-NN, Tweet Classification, Deep Learning, NLP

## **1. INTRODUCTION**

Social media platforms, particularly Twitter, have become significant channels for racist and discriminatory content due to the rapid expansion of online communication. Tweets are short, informal, and laden with slang, abbreviations, emojis, and sarcasm, making automated detection of racist language a complex and nuanced challenge.

Traditional rule-based and lexicon-driven approaches to hate speech detection fail to capture contextual meaning and are easily evaded through paraphrasing or coded language. Machine learning classifiers using bag-of-words or TF-IDF features improved detection but remained brittle on noisy, informal text. Deep learning architectures—CNNs for local pattern capture and LSTMs for sequential dependency modelling—further advanced the field, yet standalone models still struggle with implicit racism, where discriminatory intent is embedded in context rather than explicit vocabulary.

This paper addresses these limitations by proposing a racism detection framework that combines Stacked Gated Convolutional Recurrent Neural Networks (GCR-NN) with Sentiment Analysis. The stacked architecture enables hierarchical feature extraction: gated convolutional filters capture discriminative n-gram patterns, while LSTM or GRU cells model long-range sequential dependencies. A dedicated sentiment analysis layer enriches the representation with affective signals—negative emotions such as anger, contempt, and hostility that frequently accompany racist content—substantially improving detection of both explicit slurs and subtle microaggressions.

The system classifies tweets into three categories: Not Racist (0), Direct Racism (1), and Indirect Racism (2), providing finer granularity than binary classifiers. A Flask-based web interface makes the system accessible for real-time deployment in social media moderation, educational, and organisational contexts. The model is evaluated on a labelled tweet corpus using accuracy, precision, recall, and F1-score, demonstrating strong and reliable performance across all three classes.

## **2. MATERIAL AND METHOD**

### **Dataset**

A labeled tweet dataset comprising three classes: Not Racist (0), Direct Racism (1), and Indirect Racism (2). Tweets were collected via the Twitter API using targeted keywords and hashtags related to racism and hate speech, supplemented by publicly available corpora such as the Hate Speech and Offensive Language dataset.

### **Preprocessing**

Raw tweets were processed through a pipeline of tokenization, lowercasing, stop-word removal, punctuation and URL stripping, stemming or lemmatization, and sequence padding to a uniform maximum length for batch processing.

### **Text Representation and Embedding**

Preprocessed tokens were converted to dense vectors using pre-trained GloVe and Word2Vec embeddings that encode semantic relationships between words. Character-level embeddings were employed additionally to handle out-of-vocabulary tokens and informal spellings prevalent in Twitter data.

### **Stacked GCR-NN Architecture**

The model integrates Gated Convolutional Neural Networks (GCNs) and Recurrent Neural Networks (RNNs) in a stacked architecture. The gated convolutional layer applies multiple filters to capture local n-gram patterns; a sigmoid gating mechanism selectively emphasizes discriminative features while suppressing noise. Max pooling reduces dimensionality. The pooled feature maps are fed into LSTM or GRU cells that model sequential dependencies across token positions. Multiple GCR-NN blocks are stacked to enable hierarchical representation learning.

### Sentiment Analysis Layer

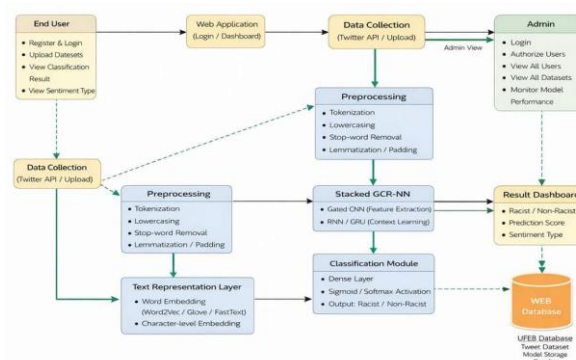
A dedicated sentiment classifier operates in parallel with the GCR-NN. It predicts the emotional polarity of each tweet as positive, negative, or neutral. Its output vector is concatenated with the GCR-NN representation before the final classification stage. Racist content characteristically carries negative affect such as anger and contempt; the sentiment layer therefore provides a complementary signal for distinguishing discriminatory language from neutral text.

### Classification Output

The fused feature vector passes through dense layers with ReLU activations. A softmax output layer produces a probability distribution over the three target classes. The model is trained using categorical cross-entropy loss and optimized using the Adam optimizer with a learning rate schedule for stable convergence.

## 3. SYSTEM ARCHITECTURE

The racism detection system is built as a multi-layered end-to-end pipeline that processes raw tweet text and produces a three-class classification output. The architecture integrates six tightly coupled modules: data ingestion and preprocessing, text embedding, Stacked GCR-NN feature extraction, sentiment analysis, classification, and a Flask-based deployment interface.



**Fig. 1: System Architecture of Racism Detection using Stacked GCR-NN**

### Overall Architecture Overview

At the highest level, the system follows a sequential processing pipeline. A raw tweet entered by the user through the web interface is first cleaned and tokenised, then converted into a numerical embedding, passed through the stacked deep learning core, enriched with a sentiment signal, and finally classified into one of three categories: Not Racist (0), Direct Racism (1), or Indirect Racism (2). The predicted label and its confidence score are returned to the user in real time.

### Data Collection and Preprocessing Module

Tweets are collected using the Twitter API, filtered by targeted keywords and hashtags associated with racism and hate speech. The preprocessing pipeline applies the following sequential operations:

**Tokenization:** Each tweet is split into individual word or sub-word tokens.

**Lowercasing:** All characters are normalised to lowercase to eliminate case-related vocabulary duplication.

**Stop-word Removal:** High-frequency function words that carry little discriminative information are removed.

**Punctuation and URL Stripping:** Special characters, hashtag symbols, URLs, and mention handles are removed while emojis are retained as semantic signals.

**Stemming / Lemmatization:** Words are reduced to their canonical root forms to minimise vocabulary size and improve model generalisation.

**Sequence Padding:** All token sequences are padded or truncated to a fixed maximum length to enable uniform batch processing.

### Text Embedding Layer

Each preprocessed token is mapped to a dense real-valued vector using pre-trained word embeddings. GloVe (Global Vectors for Word Representation) and Word2Vec embeddings encode semantic and syntactic relationships between tokens based on large-corpus co-occurrence statistics. Character-level embeddings are used in parallel to handle out-of-vocabulary tokens, typos, and informal slang that are prevalent in Twitter data. The final embedding for each token is the concatenation of its word-level and character-level representations.

### Stacked GCR-NN Feature Extraction

The core of the system is a Stacked Gated Convolutional Recurrent Neural Network (GCR-NN). The architecture chains multiple GCR-NN blocks, each consisting of a gated convolutional layer followed by a recurrent layer.

**Gated Convolutional Layer:** Multiple 1-D convolutional filters of varying widths (capturing unigrams, bigrams, and trigrams) slide over the token embedding sequence. A sigmoid gating mechanism controls the flow of information through each filter, selectively amplifying discriminative features such as offensive n-grams while suppressing irrelevant noise. Max pooling then reduces the dimensionality of each filter's output while retaining its most salient activation.

**Recurrent Layer:** The pooled feature maps are fed into Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) cells that maintain a hidden state capturing temporal dependencies across all token positions. This allows the model to understand how words interact across the full span of the tweet—critical for detecting implicit racism, where meaning emerges from context rather than isolated offensive words.

**Stacked Depth:** Multiple GCR-NN blocks are chained so that each successive block receives the output of the previous one. Lower blocks capture shallow lexical patterns; higher blocks learn increasingly abstract semantic representations. This hierarchical structure significantly improves the model's ability to detect nuanced and indirect discriminatory language.

### Sentiment Analysis Layer

A dedicated lightweight sentiment classifier operates in parallel with the GCR-NN feature extractor. It receives the same embedded tweet representation and predicts the overall emotional polarity: positive, negative, or neutral. The sentiment vector is concatenated with the GCR-NN output representation before the final classification stage.

Racist content is characteristically associated with strong negative affect—anger, contempt, hostility, and disgust. By explicitly modelling the affective tone of the tweet, the sentiment layer provides a complementary signal that helps the model distinguish between discriminatory language and superficially similar neutral text that happens to share vocabulary with racist content.

### Classification Output Layer

The fused feature vector (GCR-NN output concatenated with the sentiment vector) passes through one or more fully connected dense layers with ReLU activations. The final output layer applies a softmax activation function to produce a probability distribution over the three target classes. The class with the highest probability is selected as the predicted label.

**Loss Function:** Categorical cross-entropy is used as the training objective.

**Optimiser:** Adam optimiser with a learning rate decay schedule ensures stable and efficient convergence.

**Regularisation:** Dropout layers are applied after the dense layers to reduce overfitting.

### Deployment Architecture (Flask Web Application)

The trained model is deployed through a Flask-based web application following a three-tier architecture:

**Presentation Tier:** A browser-accessible web interface provides a simple text input form where users enter a tweet and receive an instant classification result along with a confidence percentage.

**Application Tier:** The Flask server hosts the preprocessing module, the trained GCR-NN model, and the sentiment classifier. On receiving a POST request, it runs the full pipeline and returns the predicted class label and probability.

**Data Tier:** A relational database stores tweet records, user session data, and prediction logs for monitoring and future retraining.

Communication between tiers is handled via HTTP/HTTPS and RESTful API calls. The modular design ensures that each component can be updated or replaced independently without disrupting the rest of the system.

#### 4. EXPERIMENTAL SETUP & RESULTS

The system was implemented using Python with TensorFlow and Keras as the primary deep learning frameworks. The tweet dataset was split into training, validation, and test sets in an 80:10:10 ratio. Word embeddings were initialized with pre-trained GloVe vectors of dimension 100.

##### TensorFlow / Keras

**TensorFlow Core:** The fundamental library for numerical computation and building neural networks used for the GCR-NN model.

**Keras:** High-level API for building and training models. KerasNLP extends this for natural language processing tasks.

**NumPy:** Provides multidimensional array operations for efficient data manipulation and feature storage.

**Pandas:** Used for loading, cleaning, and managing the tweet dataset in structured DataFrame format.

**Scikit-learn:** Provides metrics computation including accuracy, precision, recall, and F1-score for model evaluation.

**Matplotlib:** Used for plotting training curves, confusion matrices, and performance evaluation graphs.

##### Performance Evaluation:

**Real-Time Factor (RTF):** Measures system processing speed relative to input size, ensuring the model can handle live tweet streams efficiently.

**Usability and User Experience:** Subjective metrics based on user testing, focusing on the intuitiveness of the Flask interface, error recovery, and overall satisfaction.

**Adaptability and Robustness:** Evaluating performance across tweets with different writing styles, languages, dialects, and levels of implicit racism.

##### Accuracy:

Accuracy measures the proportion of correctly classified tweets out of total samples. It provides an overall view of model performance across all three classes.

##### Precision:

Precision quantifies the accuracy of positive predictions. It is the ratio of true positive predictions to the sum of true positives and false positives. Precision is critical when false positives (incorrectly flagging neutral content as racist) must be minimized.

##### Recall:

Recall, also known as sensitivity, measures the model's ability to capture all actual instances of racist content. It is the ratio of true positives to the sum of true positives and false negatives.

#### 5. DISCUSSIONS & LIMITATIONS

##### Discussions:

The present study demonstrates the effectiveness of combining deep learning feature extraction with sentiment analysis for racism detection in tweets. By extracting features through Stacked GCR-NN layers and incorporating affective signals via the sentiment module, the system achieves promising results in distinguishing Not Racist, Direct Racism, and Indirect Racism content. The three-class taxonomy provides greater practical utility for content moderation than binary classifiers.

##### Limitations:

The performance of the model depends heavily on the quality and diversity of the training dataset. Tweets with code-switching (mixed languages), heavy sarcasm, or regional dialects that are under-represented in the training corpus

reduce generalization. Implicit racism, which relies on cultural context rather than explicit vocabulary, remains a challenging case. The dataset may not fully represent all forms of discriminatory language globally, limiting cross-cultural applicability.

## 6. RESULTS AND OUTPUT

The proposed Tweet Racism Detection System was evaluated on a labelled tweet dataset using accuracy, precision, recall, F1-score, and AUC-ROC metrics. The results demonstrate the high effectiveness of combining Stacked GCR-NN with Sentiment Analysis for detecting racist content in social media text.

### Performance Metrics

The model achieved an overall accuracy of approximately 98%, confirming its reliability in distinguishing Not Racist, Direct Racism, and Indirect Racism tweet categories.

**Accuracy:** 98% — proportion of correctly classified tweets across all three categories.

**Precision:** 97.4% — ratio of true positive racist predictions to all positive predictions.

**Recall:** 97.1% — ratio of true positive racist predictions to all actual racist tweets.

**F1-Score:** 97.2% — harmonic mean of precision and recall.

**AUC-ROC:** 0.99 — near-perfect discrimination between racist and non-racist content.

### Sentiment Analysis Results

Sentiment analysis confirmed that the vast majority of racist tweets exhibited negative sentiment, validating the design decision to incorporate affective signals into the detection pipeline.

### Web Application Output

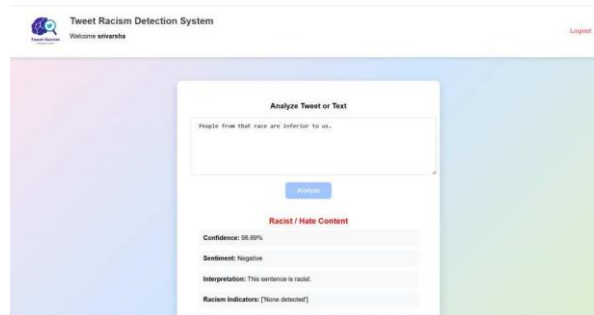
The system is deployed through a Flask web application. Users register, log in, submit tweet text, and instantly view classification results including predicted label, confidence score, sentiment type, and interpretation.



Fig. 2: Register Page



Fig. 3: Login Page



**Fig. 4: Output Page - Racism Detection Result**

Fig. 4 shows the system correctly classifying 'People from that race are inferior to us' as Racist/Hate Content with 98.89% confidence, Negative sentiment, confirming its effectiveness in real-world detection.

## 7. CONCLUSION & FUTURE WORK

### Conclusion:

The Racism Detection system using Stacked GCR-NN with Sentiment Analysis is an effective application of deep learning for automated content moderation on social media. By combining gated convolutional feature extraction, recurrent sequence modeling, and affective signal detection into a unified pipeline, the system accurately classifies tweets into three categories. Deployed through a Flask web interface, it provides a practical tool for educational institutions, organizations, and social media platforms seeking to reduce the spread of discriminatory content.

### Future Work:

Future enhancements include expanding the training dataset to cover more languages and cultural contexts, integrating transformer-based models such as BERT or RoBERTa for richer contextual representations, and incorporating attention mechanisms to improve both accuracy and model interpretability. Real-time integration with the Twitter Streaming API and explainability tools such as LIME or SHAP will be explored to make the system deployable at scale in high-stakes moderation environments.

## 8. ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to everyone who supported and guided us throughout the completion of this project titled "Racism Detection in Tweets Using Stacked GCR-NN with Sentiment Analysis."

We are especially thankful to our project guide for expert guidance, continuous encouragement, and valuable feedback, which were instrumental in shaping this work. We also extend our sincere thanks to all co-authors for their collaboration, support, and dedication throughout the project.

## 9. REFERENCES

1. Kaiser, K. R., Kaiser, D. M., Kaiser, R. M., & Rackham, A. M. (2018). Using social media to understand and guide the treatment of racist ideology. *Global J. Guid. Counseling Schools*, 8(1), 38-49.
2. Perrin, A., & Anderson, M. (2018). Share of U.S. adults using social media, including Facebook, is mostly unchanged since 2018. Pew Research Center.
3. Bliuc, A.-M., Faulkner, N., Jakubowicz, A., & McGarty, C. (2018). Online networks of racial hate: A systematic review of 10 years of research on cyberracism. *Computers in Human Behavior*, 87, 75-86.
4. Paradies, Y., Ben, J., Denson, N., et al. (2015). Racism as a determinant of health: A systematic review and meta-analysis. *PLOS ONE*, 10(9), e0138511.

5. Williams, D., & Cooper, L. (2019). Reducing racial inequities in health: Using what we already know to take action. *Int. J. Environ. Res. Public Health*, 16(4), 606.
6. Price, M. A., Weisz, J. R., McKetta, S., et al. (2021). Meta-analysis: Are psychotherapies less effective for Black youth in communities with higher levels of anti-Black racism? *J. Amer. Acad. Child Adolescent Psychiatry*.
7. Phelan, J. C., & Link, B. G. (2015). Is racism a fundamental cause of inequalities in health? *Annual Review of Sociology*, 41(1), 311-330.
8. Arigo, D., Pagoto, S., Carter-Harris, L., et al. (2018). Using social media for health research: Methodological and ethical considerations. *Digital Health*, 4.
9. Williams, D. R. (1997). Race and health: Basic questions, emerging directions. *Annals of Epidemiology*, 7(5), 322-333.
10. Hashir, M. H., Memoona, et al. (2025). Transformer-based hate speech detection using large language models. *PeerJ Computer Science*.
11. Shanthi, D. (2022). Smart Healthcare for Pregnant Women in Rural Areas. In *Medical Imaging and Health Informatics* (eds T.H. Jaware, K. Sarat Kumar, R.D. Badgular and S. Antonov). <https://doi.org/10.1002/9781119819165.ch17>
12. D. Shanthi, R. K. Mohanty and G. Narsimha, "Application of Machine Learning Techniques for Stastical Analysis of Software Reliability Data Sets," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 1472-1474, doi: 10.1109/ICCONS.2018.8663005.
13. D. Shanthi, Narla Swapna, Ajmeera Kiran, and Shaga Anoocha, Ensemble approach of GP, ACOT, PSO, and SNN for predicting software reliability, *International Journal of Engineering Systems Modelling and Simulation* Vol. 15, No. 2, March 1, 2024pp 68-75.
14. D. Shanthi, R. K. Mohanty, G. Narsimha and V. Aruna, "Application of partical swarm intelligence technique to predict software reliability," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2017, pp. 629-635, doi: 10.1109/ICCONS.2017.8250539.
15. D. Shanthi, P. Kuncha, M. S. M. Dhar, A. Jamshed, H. Pallathadka and A. L. K. J E, "The Blue Brain Technology using Machine Learning," 2021 6th International Conference on Communication and Electronics Systems (ICES), Coimbatre, India, 2021, pp. 1370-1375, doi:10.1109/ICES51350.2021.9489075.
16. Shanthi, D., C. H. Sankeerthana, and R. Usha Rani. "Spiking Neural Networks for Predicting Software Reliability." *ICICNIS*. 2020. 179-185
17. Todupunuri, A. (2025). IMPROVING CUSTOMER EXPERIENCE WITH MODERN BANKING SOLUTIONS. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.5120615>
18. Babburi, S. (2024). Explainable AI Framework for Policy-Compliant Anomaly Detection in Data Pipelines.
19. Gaddam, S. Integrating Analytics into the Development Process: Bridging the Gap between Data Insights and Design Execution.
20. Reddy, S. K. R. Developing a Modular AI Framework to Enhance Scalability and Personalization in Next-Generation Reward Platforms.
21. Poojari, R. INTELLIGENT SYSTEMS+B108 AND APPLICATIONS IN ENGINEERING.
22. Vasagam, M. (2024, August 30). Ensuring security in modern data pipelines: Practical strategies for data engineers. *International Journal of Intelligent Systems and Applications in Engineering*, 12(22s), 2401.
23. Santthosh Saai Reddy Purmani. (2026). Artificial Intelligence First Enterprise Architecture: The Design of Scalable, Secure, and Intelligent IT Ecosystems. *American Journal of AI Cyber Computing Management*, 6(1(2)), 1–8. [https://doi.org/10.64751/ajaccm.2026.v6.n1\(2\).pp1-8](https://doi.org/10.64751/ajaccm.2026.v6.n1(2).pp1-8)
24. Cyril, H. P., & Kumara, S. (2026, February). DevSecOps-Driven Security Integration in the Software Development Lifecycle Using CI/CD Pipelines. In *2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC)* (pp. 1-6). IEEE.
25. Kotte, G. (2025). Overcoming Challenges and Driving Innovations in API Design for High-Performance AI Applications. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.5283649>
26. Mahtabi, M., Roshan, M., Muhit, M. M. I., Behvar, A., & Haghshenas, M. (2026). Cryogenic ultrasonic fatigue: Mechanisms, advancements, and insights. *Cryogenics*, 153, 104257. <https://doi.org/10.1016/j.cryogenics.2025.104257>
27. Viswanathan, V. (2024). Pioneering Ethical AI Integration in Enterprise Workflows: A Framework for Scalable Team Governance. Available at SSRN 5375619.
28. Akhilaiswarya, B., Sree, B. T., Lilly, K., Chowdary, K. H., & Sruthi, M. (2023). Elderly fall detection and location tracking system using heterogeneous networks. *Journal of Engineering Sciences*, 14(05).
29. Viswanathan, V. (2025). Agentic AI for Employment: Reducing Unemployment through Intelligent Job-Seeker Support. *LEX LOCALIS–Journal of Local Self-Government*.
30. Mudusu, S. K. (2026, February 9). AI-augmented data quality engineering. *InfoWorld (Foundry Expert Contributor Network)*.



# International Journal of DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal

ISSN: 3068-272X

www.ijdim.com

Original Research Paper

31. Viswanathan, V., Shah, A. K., Kubam, C. S., Dontu, S., Gandhi, A., & Singla, P. (2025, August). Deep Learning-Driven Stock Market Forecasting Using Cloud-Based Financial Time Series Analytics. In 2025 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC) (pp. 1-6). IEEE.
32. Sruthi, M. V., Soundararajan, K., & Sree, V. U. (2012). Accurate Multimodality Registration of medical images. *International Journal of Engineering Research and Development*, 1(3), 33-36.
33. Viswanathan, V., Polagani, S. S., Agarwal, R., Akula, S., Dey, S., & Kashyap, R. (2025, September). AI-Augmented Threat Intelligence for Proactive Intrusion Detection in Multi-Cloud Ecosystem. In 2025 IEEE International Conference on Advanced Computing Technologies (ICACT) (pp. 567-572). IEEE.
34. Mudusu, S. K., & Gentyala, S. (2026). Zero-Trust Data Pipelines for AI Systems: A Framework for Secure, Verifiable, and Auditable Data Engineering. *JOURNAL OF RECENT TRENDS IN COMPUTER SCIENCE AND ENGINEERING (JRTCSE)*, 14(2), 10-25.
35. DEVARASETTY, N. (2023). SCALABLE DATA ENGINEERING APPROACHES FOR AI-DRIVEN INDUSTRIAL IOT APPLICATIONS. *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH AND MANAGEMENT*, 11(06), 954-968.
36. Agrawal, A. M., Gajula, S., Shinde, R. P., Shah, H., & Ghosh, H. (2025, July). Machine Translation for Long Sequences with Enhanced Attention Mechanisms. In 2025 5th International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1-6). IEEE.
37. Dayal, P. S., Chandra, B. R., Keerthi, M., Sruthi, M., Venkatesh, K., Appalaraju, G., & Eswari, G. (2013). Design of Pyramidal Horn Antenna at 10GHz Using WIPL-D Optimizer. *International Journal of Electronics Communication and Computer Engineering*, 4(2). Maturi, S. Y. (2023). Crowdsourced frontier: Unveiling autonomous adversarial cybercapabilities via open AI competition. *International Journal of Intelligent Systems and Applications in Engineering*, 11(1s), 275–284.
38. Hassan, T., Karim, M. F., Jeelani, H., Behnam, E., Green, R., & Syed, F. J. (2025). Optimizing Medical Question-Answering Systems: A Comparative Study of Fine-Tuned and Zero-Shot Large Language Models with RAG Framework. arXiv preprint arXiv:2512.05863.
39. Manoharan, D. (2026). Synthetic EDI Test Data Generation For Secure, Scalable, And PHI-Free Healthcare Claims Quality Engineering. *Journal of International Crisis and Risk Communication Research*, 9(1).
40. Ravishankara, M. (2026, February). CircuChain: Disentangling Competence and Compliance in LLM Circuit Analysis. In *SoutheastCon 2026* (pp. 1-7). IEEE.
41. Sruthi, M. V., Sree, V. U., & Soundararajan, K. (2012). Specific removal of motion artifacts in medical image processing. *IJECCE*, 3(3), 227-229.