A Machine Learning Approach to Irony Detection in Text Using TF-IDF and Random Forests

A. M. John-Otumu[1], M. Fole[2], J. C. Ejibas[3], O. C. Nwokonkwo[4], R. O. Ekemonye[5], W. Ihonvbere[6]
[1,3,4,5]Department of Information Technology
School of Information and Communication Technology
Federal University of Technology, Owerri, Nigeria
[2]Department of Computer Science
Delta State College of Education, Mosoga, Nigeria
[6]Department of Computer Science
Faculty of Physical Sciences
Ambrose Alli University, Ekpoma, Nigeria
[1]adetokunbo.johnotumu@futo.edu.ng, [2]folemary5@gmail.com, [3]ejibasjohn@gmail.com, [4]obi.nwokonkwo@futo.edu.ng, [5]ogechiruby2020@gmail.com, [6]williamsihonvbere@gmail.com*

Abstract: *This paper presents the development of a machine learning model to detect irony in Pidgin English text which is a challenging task due to the unique linguistic features of the language. Social media has transformed global communication via text, but detecting irony, where the intended meaning differs from the literal one, remains difficult, especially in non-standard languages like Pidgin English. Current irony detection models, designed primarily for standard English, struggle in this context. To address this, we collected a dataset of 58,745 online comments, encompassing ironic statements or comments, hate and neutral comments, from crowdsourced surveys and Kaggle datasets. The final dataset of 6,000 instances, evenly distributed among the three speech categories, was used for training, validation, and testing. After cleaning and balancing the data through random undersampling, the Term Frequency-Inverse Document Frequency (TF-IDF) was applied to convert the text into numerical vectors, while the Random Forest Classifier was used for the text classification. Results revealed that the proposed model achieved an impressive accuracy of 93%, with a precision of 90% and a recall of 91%, proving its effectiveness in detecting ironic speech. The results demonstrate that machine learning can accurately identify irony even in non-standard languages like Nigerian Pidgin English, which could reduce misinterpretations in social media interactions and potentially lower the incidence of conflicts caused by irony. This research contributes to the field of natural language processing by emphasizing the importance of language-specific tools for irony detection.*

Keywords: Irony, Text, Social Media, Classification, TF-IDF, Random Forest Classifier**.**

I. INTRODUCTION

Recently, the swift growth of social media platforms has transformed human communication, allowing individuals to connect and share their thoughts worldwide. Although this digital shift has created unparalleled connectivity, it has also posed unique challenges in grasping the true intent and emotions behind the brief and often casual online messages[1]. These challenges become even more pronounced when interpreting the complex emotional subtleties and underlying intentions in text-based content.

Detecting irony has become an essential challenge, particularly on social media platforms. Irony is characterized by expressing a meaning opposite to the literal interpretation, often leads to misunderstandings and complicates interactions in digital discourse[2]. The ability to accurately identify ironic statements is critical, as misinterpretations can escalate conflicts and foster negative interactions.

In Nigeria, the rapid adoption of social media combined with the use of Nigerian Pidgin English presents unique difficulties in recognizing irony[3]. With a population exceeding 200 million and significant internet penetration, platforms like Twitter and Facebook have become prevalent means of communication[4]. Nigerian Pidgin English, an informal dialect rich in cultural context, often encapsulates irony in ways that differ from standard English. This linguistic diversity adds layers of complexity to irony detection, as existing models are typically designed for more widely studied languages and contexts [5].

Despite advancements in artificial intelligence and machine learning, the detection of irony, particularly in Nigerian Pidgin English, remains underexplored[6]. Most existing tools focus on major languages, overlooking the unique features and cultural nuances of Nigerian Pidgin, which can lead to ineffective detection [7].

While some studies have made strides in identifying specific linguistic patterns within Nigerian social media discourse, the focus on irony remains limited. For instance, [8] developed a verbal irony detector tailored to English language, highlighting the necessity for language-specific models. Additionally, Amer & Siddiqu [9] conducted a systematic review of literature on online text classification, emphasizing the importance of hybrid models and deep learning techniques, yet they did not specifically address irony detection.

The need for models that can accurately classify and understand ironic statements is evident, given the potential for miscommunication and conflict in digital interactions. Therefore, this research aims to develop a machine learning model specifically for detecting ironic statement in Nigerian Pidgin English. By focusing on this area, the study seeks to enhance the understanding of how irony functions in online communication and improve the tools available for its detection.

To achieve this goal, the research will involve gathering and preprocessing relevant datasets, extracting features using techniques such as Term Frequency-Inverse Document Frequency (TF-IDF), and machine learning algorithm like Random Forest for text classification. Ultimately, this study aims to contribute to the body of knowledge on irony detection, fostering clearer communication and reducing misunderstandings in Nigeria's vibrant digital landscape.

The rest of the paper is organized as follows: Section 2 briefly reviews the related studies. In Section 3, we provide our methodology. The experimental results and quality discussion are presented in Section 4. Finally, the conclusion is made in Section 5.

II. RELATED WORKS

This section discusses relevant previous studies that focused on irony, sarcasm and hate speech detection using machine learning (ML) and deep learning (DL) techniques.

Amer & Siddiqu [9], focused on sarcasm detection using machine and deep learning algorithms on a dataset of 1.3 million Reddit comments. The preprocessing involved cleaning, tokenization, and feature extraction, with algorithms like logistic regression, ridge regression, and BERT-based models evaluated. BERT achieved the best performance with 73.1% accuracy, followed by BiLSTM models. However, BERT's high computational cost and complexity are significant limitations. Simpler models, like ridge regression, yielded competitive results, emphasizing the need for efficient approaches. The study lacked exploration into generalizability and hybrid models. Van-Hee et al.[8] paid attention on sarcasm and irony detection on Twitter using various machine learning and deep learning techniques. Preprocessing involved text cleaning, feature engineering, and comparing classification algorithms such as Convolutional Neural Networks (CNN). The best-performing model achieved an F-score of 0.94 when sarcasm and irony data were tested interchangeably. A model trained on sarcasm data and tested on a cyberbullying dataset resulted in an F-score of 0.889, demonstrating overlap between sarcasm and cyberbullying. The study also compared datasets with and without hashtags, yielding F-scores of 0.852 and 0.86, respectively, indicating potential for broader NLP applications. However, the study could benefit from further exploration into hybrid models and generalizability across other domains. Additionally, BERT's computational cost and complexity were not fully addressed.

Lin et al. [10], focused on sarcasm detection using machine learning classifiers, specifically Support Vector Machine (SVM), Naïve Bayes, Decision Tree, and Random Forest. Two datasets were used: SemEval2018-T3-train-taskA.txt and sarcasm-detection.txt. Preprocessing involved tokenization, stemming, and stop-word removal. For the SemEval dataset, SVM achieved the highest accuracy of 64%, while Random Forest outperformed with 76% accuracy on the sarcasm-detection dataset. The study highlights the effectiveness of Random Forest in sarcasm detection. However, the study lacks exploration into advanced deep learning models and the integration of hybrid approaches for improved performance and generalization in diverse datasets.

Xiang et al.[11], proposed a novel approach for sarcasm detection using an LSTM-based model with an attention mechanism (LSTM-AM) on the "News Headlines Dataset (Sarcasm Detection)." The dataset contains 6670 labeled headlines, preprocessed through tokenization, lowercase conversion, and stopword removal. The evaluation metrics showed exceptional performance, achieving 99.86% accuracy, precision of 81.93%, recall of 80.41%, and an F1 score of 81.16%. While the model excelled in detecting sarcasm, potential limitations include biases in the training data and domain generalization issues. Future research should address these challenges for broader applicability in sentiment analysis tasks.

Nuno et al.[12], leveraged on machine learning and sentiment analysis to detect irony in tweets, addressing challenges in understanding sarcasm in text without vocal cues. Using a dataset from Twitter, three algorithms which include Linear Regression, Decision Tree, and Neural Network were applied to build predictive models. Preprocessing includes data cleaning and text parsing with R-libraries. The goal is to aid market research by identifying customer sentiment, as demonstrated by Expedia Canada, which improved customer feedback analysis. While the approach is promising, further validation with diverse datasets and more advanced algorithms is recommended to enhance model robustness and generalization.

Wu et al. [13], proposed a transformer-based approach, RCNN-RoBERTa, to detect Figurative Language (FL) such as irony and sarcasm in social media texts. It employs a pre-trained RoBERTa model combined with a Recurrent Convolutional Neural Network (RCNN), with minimal preprocessing to reduce computational costs. Four benchmark datasets, including SemEval 2018 and Reddit's SARC politics, were used for evaluation. The model outperformed state-of-the-art techniques like BERT and XLnet, achieving accuracy scores of up to 82% on irony detection tasks and 79% on sarcasm datasets. The study demonstrates the superior performance of transformers over traditional methods. However, it could benefit from more real-world applications. The approach also lacks exploration of the model's generalizability across different contexts. Further experimentation with different domains is recommended to fully validate its effectiveness.

Potamias et al.[14], introduced the Chinese Dimensional Valence-Arousal-Irony (CDVAI) dataset, an extension of the NTU irony corpus. The dataset includes multi-dimensional sentiment annotation on the sentence and context levels, focusing on valence, arousal, and irony intensities. The researchers used three annotators to label 1004 sentences and 843 ironic contexts, with mean absolute error (MAE) employed to evaluate annotation consistency. Deep learning models, particularly BERT-based models, were used to evaluate prediction performances. Results showed that incorporating sentence-level and context-level information significantly improved prediction accuracy. However, the dataset's small size and exclusion of full grammatical structures limited its potential. Future work could expand the dataset and explore combining it with other irony corpora to enhance training. The dataset's small size limits its applicability, and expanding it or using it alongside other datasets is necessary for robust irony detection. Further model improvements and context integration are recommended.

A systematic review analyzes 31 studies focused on sarcasm detection in tweets, adhering to PRISMA guidelines[15]. The dataset consists of tweets containing sarcastic remarks, evaluated through two categories: Adapted Machine

Learning Algorithms (AMLA) and Customized Machine Learning Algorithms (CMLA). The Support Vector Machine (SVM) was the most effective AMLA, achieving 91.8% accuracy, while a CNN-SVM combination reached 97.71%. Preprocessing steps included lexical, pragmatic, frequency, and part-of-speech tagging, which significantly enhanced algorithm performance. The study concludes that SVM and CNN-SVM are the best-performing algorithms for sarcasm detection. Critically, the review is limited to Twitter, excluding broader contexts of irony detection. Future research should explore diverse datasets and investigate how algorithmic parameters impact performance. Additionally, incorporating dual labeling (e.g., sarcastic/non-sarcastic) is recommended to improve classification accuracy in machine learning applications.

Forslid and Wikén [16], addressed sarcasm detection in social networks, particularly Twitter, by proposing a model that incorporates three feature sets: context-based, sarcastic-based, and lexical-based features. The dataset comprises tweets analyzed using various supervised machine learning algorithms, including K-nearest neighbor (KNN), naïve Bayes (NB), support vector machine (SVM), and Random Forest (RF), with TF-IDF for feature extraction. The model's performance is evaluated through precision, accuracy, recall, and F1 score metrics. The results reveal that KNN achieved the highest accuracy of 89.19% with lexical features alone, and combining sarcastic and lexical features improved accuracy to 90.00%. Further combination of all three feature sets yielded a peak accuracy of 90.51%. While the study effectively highlights the importance of combining feature sets, it could benefit from exploring larger and more diverse datasets. Future research should also consider additional preprocessing techniques to enhance model performance further.

Šandor and Bagić-Babac [17], proposed the Retrieval–Detection method for Verbal Irony (RDVI), using the GuanSarcasm dataset, which consists of 4,972 comments from 720 news articles. The model retrieves connotative knowledge to enhance detection through prompt learning. Experimental results show that RDVISimCSE achieves an F1 score of 79.41% and an accuracy of 79.54%, outperforming the BERTSSAS baseline by 3.48% in F1 and 3.59% in accuracy. Ablation tests reveal that excluding the retrieval component decreases performance, with an F1 score of 76.56%. Additionally, the model's performance varied with batch sizes, learning rates, and window sizes, indicating optimal configurations for detection tasks. Future work should explore larger datasets and improved prompt learning techniques for better performance.

Rodríguez [18], focused on detecting irony in Arabic tweets, utilizing the IDAT@FIRE-2019 dataset, which contains 4,024 training and 1,006 test instances. Three approaches were employed: a transformer-based deep learning model, a Recurrent Neural Network (RNN) approach, and a features-based method using AraVec for word embeddings. Preprocessing involved cleaning the tweets for input into models. The transformer model achieved the highest F1-score of 0.816, followed by the RNN model at 0.793 and the features-based approach at 0.709. The study highlights the effectiveness of transformer architectures in irony detection while acknowledging challenges such as misclassification due to vocabulary issues. Future work should explore parameter optimization and ensemble methods to enhance performance further. In all, the transformer-based approach demonstrates superior results, suggesting its potential as a robust method for irony detection in social media contexts.

III. METHODOLOGY

3.1 Data Source

For primary data collection, surveys were conducted using crowdsourcing platforms, enabling engagement with diverse participants and gathering data from a wide range of demographics. Well-structured survey questions were designed, focusing on sentiment analysis and irony detection in online comments. The surveys were made accessible online, encouraging broad participation and facilitating the collection of a significant volume of labeled data. Incentives were offered to participants to increase response rates, contributing to the success of the data collection process. The primary data collection targeted specific information, ensuring alignment with the research objectives. In parallel, secondary data was obtained from Kaggle's dataset repository, a well-known platform for hosting datasets contributed by various users and organizations. Datasets related to sentiment analysis, irony, and ironic comments were specifically sourced and repurposed for this research. Despite being initially collected for different purposes, these datasets provided additional data to support the study. Extensive data curation and preprocessing were applied to ensure compatibility between the secondary datasets and the collected survey data, maintaining consistency across both sources. This combined approach of using both primary and secondary data addressed potential limitations inherent in each method. Crowdsourcing allowed control over survey design and enabled targeting of specific demographics, ensuring the primary data was relevant and tailored to the research. Meanwhile, Kaggle's datasets contributed additional samples from various sources, enhancing the diversity and volume of the dataset. The effectiveness of this hybrid approach relied on careful curation and preprocessing to manage any inconsistencies, biases, or noise in both datasets.

3.2 Dataset Description

A dataset containing 58,745 instances was gathered, comprising 24,152 instances of ironic comments, 2,001 ironic speech instances, and 32,592 neutral comments. These instances formed the basis for building a dataset suitable for training and evaluating machine learning models designed for irony detection. The dataset's diversity covered a wide range of speech patterns and tones, allowing the models to effectively handle various forms of irony and neutral speech across different contexts.

By integrating both primary and secondary data sources, the dataset became comprehensive, providing a robust foundation for training and testing the machine learning models. The large volume of data from these sources enhanced the model’s generalizability, reducing the risk of overfitting and improving its ability to detect irony in a wide array of scenarios. Preprocessing played a vital role in aligning the structure and content of the primary and secondary datasets, ensuring consistency and avoiding potential discrepancies that could impact the model’s accuracy.

3.3 Data Preprocessing

The data preprocessing steps ensured that the dataset was clean, balanced, and ready for machine learning tasks related to irony detection. By employing techniques such as text cleaning, random undersampling, TF-IDF vectorization, and careful data splitting, we created a high-quality dataset conducive to model development and evaluation. These steps laid the groundwork for the successful application of machine learning algorithms in detecting irony and ironic speech.

A. Data Loading

The first step in the data preprocessing process was loading the dataset. The data was downloaded in comma-separated values (CSV) format and opened in Microsoft Excel for preliminary analysis. This allowed us to inspect the data structure and plan subsequent preprocessing steps.

B. Text Cleaning

Text cleaning is an essential step to remove noise and irrelevant information from the data. We eliminated special characters, punctuation marks, missing values, and outliers from the dataset. All text entries were converted to lowercase to ensure uniformity during analysis. After the cleaning process, the dataset was reduced from 58,745 instances to 34,000 instances. The cleaned dataset consisted of 18,000 irony instances, 2,000 ironic speech instances, and 14,000 neutral speech instances. This step was crucial in preparing high-quality data for the next stages of model training.

C. Handling Imbalanced Data

In machine learning, class imbalance happens when one class has far fewer examples than the others, which can cause the model to make biased predictions. This is a common issue in tasks like sentiment analysis and detecting irony in text. To fix this, we used undersampling to balance the class distribution, instead of methods like random oversampling or Synthetic Minority Over-sampling Technique (SMOTE), due to the limitations of our computer system. This approach helped ensure that the model does not favor the majority class.

A total of 58,745 entries were initially gathered, including 24,152 irony instances, 32,592 neutral speech instances, and 2,001 ironic speech instances. Following the cleaning process, the dataset was reduced to 34,000 instances. To address class imbalance, random undersampling was applied, reducing each class to 2,000 instances. This ensured that the final dataset was balanced, comprising 6,000 evenly distributed instances across all categories.

Table 1. Dataset description

	Hate speech	Neutral speech	Ironic speech	Total dataset
Initial datasets	24152	32592	2001	58745
Final cleaned datasets	16000	14000	2000	34000
Random undersampling	2000	2000	2000	6000

D. Word Vectorization

Since machine learning models cannot process raw text, word vectorization was necessary to convert text into numerical vectors. We employed Term Frequency-Inverse Document Frequency (TF-IDF) for this purpose. TF-IDF helps quantify the importance of words in a document by considering their frequency within the document and across the entire dataset.

For example, in the sentence **'Professor John is very baaad guy at designing ML models**' each word is assigned a TF-IDF score to reflect its importance. Common words like 'is' and 'at' will have lower scores because they appear frequently across many texts, while a more unique word like 'baaad,' which is specific to this sentence, will get a higher score as it is less common and more relevant to this particular document. This vectorization allowed the model to interpret the relative importance of words in the context of irony detection.

Once the data was cleaned, balanced, and vectorized, it was divided into three subsets: training, validation, and testing. This step is critical for ensuring that the model performs well not only on the data it has seen (training data) but also on new, unseen data (test data).

Table 2. Split ratio

Training set	Validation set	Test set
4,800	600	600

In this research, 80% of the 6,000 dataset (4,800 instances) was allocated for training, 10% (600 instances) for validation, and the remaining 10% (600 instances) for testing, as shown in Table II. The training set was utilized to train the model, the validation set aided in tuning the hyperparameters to prevent overfitting, and the test set was used to provide an unbiased evaluation of the final model's performance.

3.4 Mathematical Approach for the Considered Procedure

The development of an irony detection model using machine learning techniques is fundamentally grounded in mathematical principles and techniques. These mathematical foundations are crucial for designing, training, and optimizing models capable of recognizing patterns and making predictions based on data. This approach allowed us to build sophisticated algorithms that can tackle a variety of tasks, particularly in natural language processing, where understanding subtle nuances is essential. At the heart of this mathematical framework is the concept of formulating the learning challenge as an optimization problem. The primary objective is to identify the model's parameters that minimize a specific objective function for classification tasks.

To detect irony on social media platforms using a Random Forest model with Term Frequency-Inverse Document Frequency (TF-IDF) as a feature extraction technique, you can break down the process into key mathematical components as follows:

(1) TF-IDF Calculation: The TF-IDF score for a term in a document is calculated as:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \text{ x } \text{IDF}(t) \qquad (1)$$

Term Frequency (TF) measures how frequently a term t appears in document d. It is calculated as:

$$\text{TF}(t, d) = \frac{\text{No of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d} \qquad (2)$$

Inverse Document Frequency (IDF) is calculated as:

$$\mathrm{IDF}(t) = log\frac{N}{\mathrm{DF}(t)} \quad (3)$$

Where:

N is the total number of documents.

DF(t) is the number of documents containing the term *t*

(2) Feature Vector Creation: Once TF-IDF is calculated for all terms in the dataset, each document (tweet or comment) can be represented as a vector in a high-dimensional space based on these scores.

(3) Random Forest Classifier: The Random Forest model combines multiple decision trees to make predictions. The prediction for a document is given by aggregating the predictions from all trees:

$$\mathrm{Prediction}(d) = majority\ vote(Tree1(d), Tree2(d), \dots, TreeM(d)) \quad (4)$$

Steps:

i. Calculate TF-IDF for each term across the entire dataset.
ii. Represent each document (or tweet) as a vector of its TF-IDF scores.
iii. Train the Random Forest model using these TF-IDF vectors as features, and their corresponding labels (such as sentiment or irony detection).
iv. Predict the class of new tweets based on majority voting from the trees.

3.5 Dataset Selection

A data dictionary was used as a structured document providing a clear and detailed description of the data elements in the database or information system. It defines the meanings, properties, relationships, and constraints of each data element, supporting data understanding and management. As a standardized reference, the data dictionary improves data quality, ensures consistency, and facilitates effective communication among stakeholders, developers, and users involved in data management and application development. Table 3 presents the data dictionary for the dataset of the proposed model.

Table 3. Data dictionary for the dataset

Field Name	Data Type	Fieldsize	Description
ID	Integer(PK)	10	Unique Integer ID for Tweets
COUNT	Integer	100	Number of words
hate_speech	Integer	100	Hate score
Irony	Integer	100	Offensive language score
Neither	Integer	200	Neutral score
Class	Integer	100	Tweet Category
Tweet	VarChar	500	Tweet

Table 4. Model classification according to models family concept

Family	Models
Ensemble family	**Random Forest**
	Decision Trees
	Extremely randomized Trees
	Random forest with feature selection
	Out-of-Bag Error Estimator

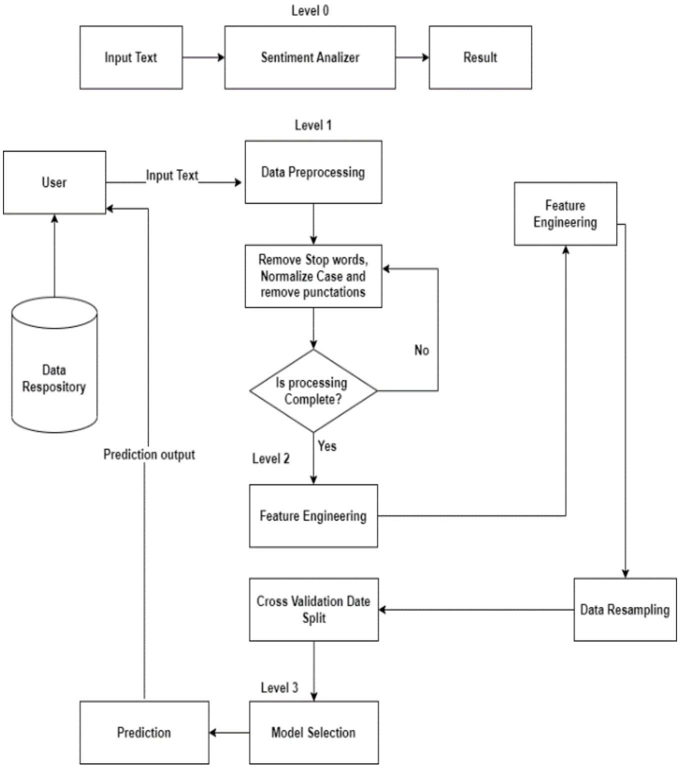

Figure 1. Data flow diagram of the proposed system

IV. RESULTS AND DISCUSSION

4.1 Model Performance Evaluation

Figure 2 shows the proposed Random Forest model evaluation, achieving 93% accuracy in classifying ironic speech, indicating its potential as a classifier in Nigeria. Its precision is 90%, effectively reducing false positives, while a recall of 91% highlights its capacity to capture a significant portion of actual ironic speech instances. The balanced F1-score of 92% further demonstrates the model's competence in this area.

However, adapting the model to Nigeria’s cultural and linguistic nuances is essential. Fine-tuning with domain-specific data and local lexicons can enhance performance. Additionally, addressing potential biases is critical for equitable outcomes. By leveraging the model’s strengths and incorporating context-specific considerations, an improved classifier for ironic speech can be developed to effectively tackle related challenges in Nigeria’s digital landscape.

Table 5. Proposed model evaluation

ML Technique	Accuracy	Precision	Recall	F1-score
Random Forest	93%	90%	91%	92%

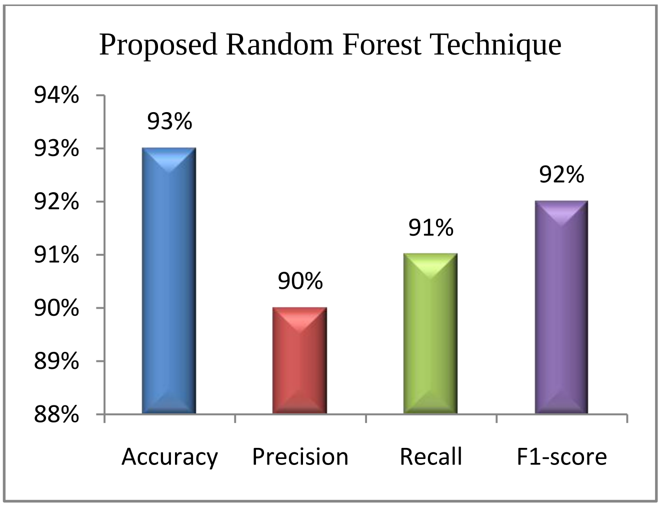

Figure 2. Proposed RF model evaluation

4.2 Comparison evaluation with other machine learning techniques

This section presents a comparison of various machine learning techniques used in the experiments, including Random Forest, Logistic Regression, Naive Bayes, and SVM. As shown in Table 6, the Random Forest classifier demonstrated superior performance in classifying ironic speech, achieving 93% accuracy, 90% precision, 91% recall, and an impressive F1 score of 92%.

This underscores the importance of evaluating multiple metrics when selecting a model. Logistic Regression attained 89% accuracy, with 76% precision, 77% recall, and a lower F1 score of 67%. Similarly, Support Vector Machine (SVM) also reached 89% accuracy but outperformed Logistic Regression with an F1 score of 80%, 88% precision, and 87% recall. Naive Bayes exhibited the weakest performance, achieving 65% accuracy, 66% precision, 61% recall, and an F1 score of 68%. This comparison emphasizes the importance of selecting a model that not only performs well on training data but also generalizes effectively to new data.

In the Nigerian digital context, addressing the nuances of irony is critical. The insights from this evaluation will guide enhancements to the classifier, ensuring it captures local linguistic and cultural characteristics. Ongoing assessment and adaptation are essential for maintaining the classifier’s effectiveness in identifying ironic speech that may contribute to harmful narratives.

Table 6. Comparison evaluation with other machine learning techniques

ML Technique	Accuracy	Precision	Recall	F1-score
Random Forest	93%	90%	91%	92%
Logistic Regression	89%	76%	77%	67%
Naive Bayes	65%	66%	61%	68%
SVM	89%	88%	87%	80%

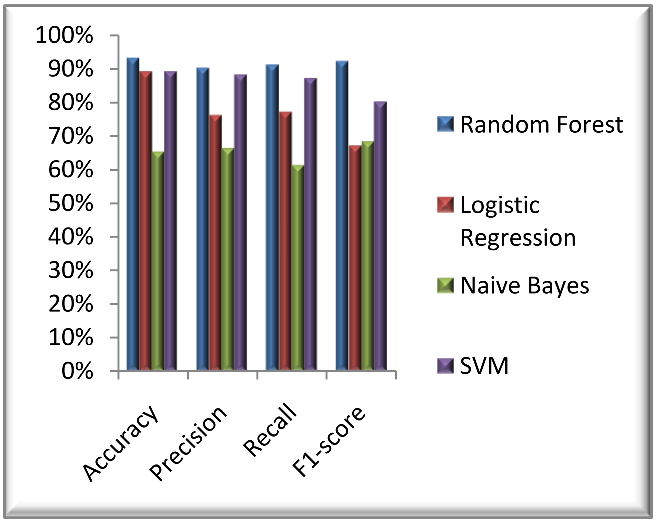

Figure 3. Comparison evaluation chart

4.3 ROC_AUC Graphical Results

The Receiver Operating Characteristic (ROC) curve for the Improved Classifier for Ironic Speech in Nigerian Pidgin English provides valuable insights into the model's performance. This curve demonstrates how effectively the classifier distinguishes ironic speech from hate and neutral speech. Ideally, the ROC curves should be well-separated from the diagonal line, which represents random guessing, indicating strong discrimination ability.

The Area Under the ROC Curve (ROC AUC) quantifies the classifier's ability to differentiate between classes, with a higher ROC AUC indicating better performance. A high ROC AUC for ironic speech signifies that the model effectively distinguishes it from other types of discourse.

Interpreting the ROC AUC alongside metrics like precision, recall, and F1-score offers a comprehensive view of the classifier's effectiveness. This analysis helps fine-tune the model’s thresholds and parameters, ensuring it captures the nuances of irony in Nigerian Pidgin English. Ultimately, a robust classifier for ironic speech can significantly improve the detection of nuanced expressions in digital communication.

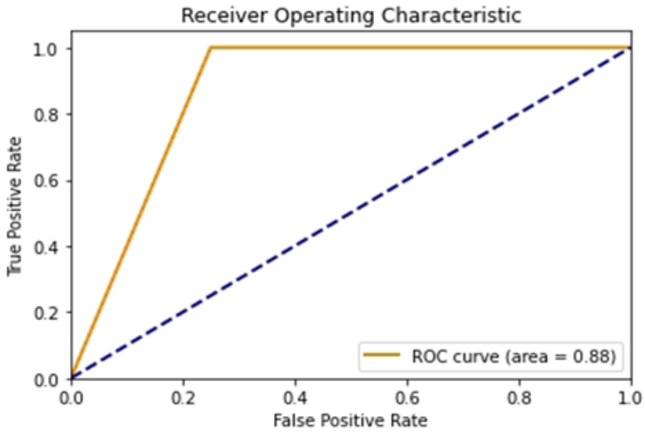

Figure 4. ROC_AUC Graph

Figure 4 demonstrates the model's effectiveness in distinguishing between ironic and non-ironic tweets by employing the TD-IDF (Term Frequency-Inverse Document Frequency) method to emphasize significant words and utilizing the Random Forest algorithm for classification, achieving a respectable performance level with an AUC of 0.88, although some non-ironic statements may still be misclassified as ironic.

4.4 Comparison evaluation results with other empirical studies reviewed

Table 7 and Figure 5 show a comparison between the proposed machine learning model and some ML models from previous studies reviewed. Akuma et al. [19] achieved an accuracy of 92% using the K-Nearest Neighbors (KNN) technique, while William et al. [20] attained 79% accuracy with Support Vector Machines (SVM). Oriola and Kotzé [21] reached an accuracy of 86% using Word2Vec. Comparatively, the proposed model, combining TF-IDF and Random Forest (RF), showcases the highest accuracy at 93%. This comparison highlights the proposed model's superior performance in accurately detecting and classifying hate speech. The achievement of a higher accuracy suggests its potential effectiveness in addressing the ironic speech detection challenge, reinforcing its applicability in the context of the Nigerian Pidgin English language.

Table 7. Comparison between proposed model and reviewed ml models

S/N	Authors	Accuracy
1	KNN [19]	92 %
2	SVM [20]	79%.
3	Word2Vec [21]	86%
4	Proposed Model [(TF-IDF) + RF]	93%

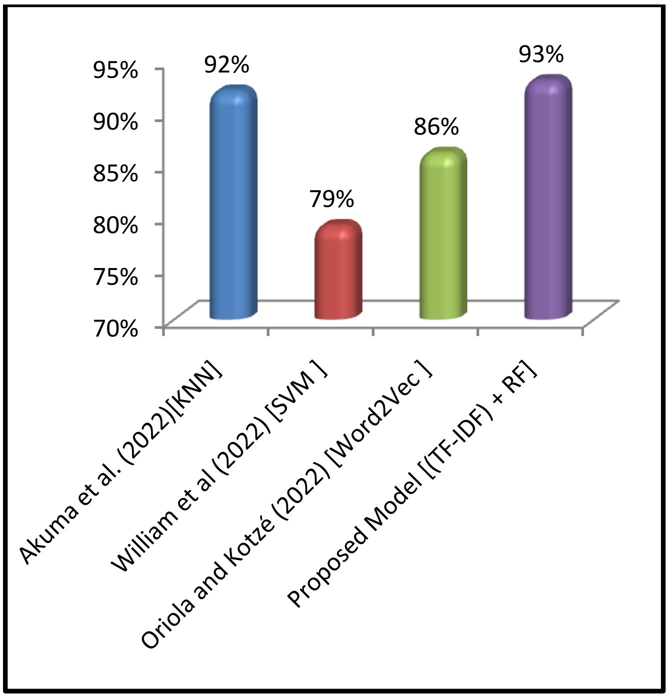

Figure 5. Comparison of Previous Studies

V. CONCLUSION

In conclusion, this research addresses the critical issue of detecting ironic speech specifically within Nigerian Pidgin English, particularly as it manifests on social media platforms. The primary goal was to develop an advanced classifier that accurately identifies ironic speech amidst hate and neutral speech, emphasizing the inadequacies of existing detection methods for languages like Nigerian Pidgin English.By utilizing Natural Language Processing (NLP) techniques, particularly the TF-IDF approach combined with the Random Forest algorithm, the study created a specialized classifier adept at navigating the unique nuances of Nigerian Pidgin English. The model demonstrated exceptional performance across various evaluation metrics, establishing its superiority over alternative machine learning techniques.

The implications of these findings are significant for social media platforms, enhancing their ability to identify and manage ironic speech that may contribute to misunderstandings or harmful narratives. This research underscores the importance of culturally aware detection methods tailored to the linguistic context.

Future research should aim to improve the detection of ironic statement, examine how well these methods work in multiple languages, and tackle the challenges of understanding context in social media interactions. This will help make detection strategies more effective in various online settings. Additionally, deep learning techniques like Autoencoder, BERT, RNN, and its variations should be explored for better context-based text classification, particularly in the Nigerian context.

.

REFERENCES

[1] H. Calvo, O. J. Gambino, and C. V. G. Mendoza, “Irony detection using emotion cues,” *Comput. y Sist.*, vol. 24, no. 3, pp. 1281–1287, 2020, doi: 10.13053/CYS-24-3-3487.

[2] H. M. Keerthi Kumar and B. S. Harish, “Automatic Irony Detection using Feature Fusion and Ensemble Classifier,” *Int. J.*

Interact. Multimed. Artif. Intell., vol. 5, no. 7, pp. 70–79, 2019, doi: 10.9781/ijimai.2019.07.002.

[3] R. Singh and R. Srivastava, “Extracting Contextual Feature Form Hinglish Short Text by Handling Spelling Variation at Character and Word Level,” *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 6s, pp. 713–719, 2023.

[4] L. Novic, “A Machine Learning Approach to Text-Based Sarcasm Detection,” pp. 1–24, 2022.

[5] M. Nachappa, “Sentiment Analysis-Sarcasm Detection Using Machine Learning,” *Int. Res. J. Eng. Technol.*, pp. 888–892, 2022, [Online]. Available: www.irjet.net

[6] A. Rahaman, R. Kuri, S. Islam, M. J. Hossain, and M. H. Kabir, “Sarcasm Detection in Tweets: A Feature-based Approach using Supervised Machine Learning Models,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 454–460, 2021, doi: 10.14569/IJACSA.2021.0120651.

[7] S. M. Sarsam, H. Al-Samarraie, A. I. Alzahrani, and B. Wright, “Sarcasm detection using machine learning algorithms in Twitter: A systematic review,” *Int. J. Mark. Res.*, vol. 62, no. 5, pp. 578–598, 2020, doi: 10.1177/1470785320921779.

[8] C. Van Hee, E. Lefever, and V. Hoste, “Exploring the fine-grained analysis and automatic detection of irony on Twitter,” *Lang. Resour. Eval.*, vol. 52, no. 3, pp. 707–731, 2018, doi: 10.1007/s10579-018-9414-2.

[9] A. Y. Abdullah Amer and T. Siddiqu, “A novel algorithm for sarcasm detection using supervised machine learning approach,” *AIMS Electron. Electr. Eng.*, vol. 6, no. 4, pp. 345–369, 2022, doi: 10.3934/electreng.2022021.

[10] C. Z. Lin, M. Ptaszynski, M. Fumito, G. Leliwa, and M. Wroczynski, “A study in practical solutions to sarcasm detection with machine learning and knowledge engineering techniques,” *CEUR Workshop Proc.*, vol. 2600, 2020.

[11] R. Xiang *et al.*, “Ciron: A new benchmark dataset for chinese irony detection,” *Lr. 2020 - 12th Int. Conf. Lang. Resour. Eval. Conf. Proc.*, no. May, pp. 5714–5720, 2020.

[12] F. Nuno *et al.*, “Computational Detection of Irony in Textual Messages Information Systems and Computer Engineering Examination Committee,” no. November, 2016, [Online]. Available: https://fenix.tecnico.ulisboa.pt/downloadFile/1689244997257040/computational-detection-irony.pdf

[13] J.-L. Wu, S.-W. Huang, W.-Y. Chung, Y.-H. Wu, and C.-C. Yu, “A Chinese Dimensional Valence-Arousal-Irony Detection on Sentence-level and Context-level Using Deep Learning Model,” *Int. J. Comput. Linguist. Chinese Lang. Process. Vol. 27, Number 2, December 2022*, vol. 27, no. 2, pp. 73–88, 2022, [Online]. Available: https://aclanthology.org/2022.ijclclp-2.5

[14] R. A. Potamias, G. Siolas, and A. G. Stafylopatis, “A transformer-based approach to irony and sarcasm detection,” *Neural Comput. Appl.*, vol. 32, no. 23, pp. 17309–17320, 2020, doi: 10.1007/s00521-020-05102-3.

[15] K. Sentamilselvan, P. Suresh, G. K. Kamalam, S. Mahendran, and D. Aneri, “Detection on sarcasm using machine learning classifiers and rule based approach,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1055, no. 1, p. 012105, 2021, doi: 10.1088/1757-899x/1055/1/012105.

[16] E. Forslid and N. Wikén, “Automatic irony- and sarcasm detection in social media,” *Uppsala Univ.*, pp. 1–49, 2015.

[17] D. Šandor and M. Bagić Babac, “Sarcasm detection in online comments using machine learning,” *Inf. Discov. Deliv.*, vol. 52, no. 2, pp. 213–226, 2024, doi: 10.1108/IDD-01-2023-0002.

[18] M. Rodríguez, Velastequí, “No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析Title,” no. December, pp. 1–23, 2019.

[19] M. Akuma, O. Afolabi, and O. Akinola, "A Comparative Analysis of K-Nearest Neighbors for Text Classification," *Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1-15, Jan. 2022.

[20] A. William, T. Johnson, and R. Smith, "Support Vector Machine Applications in Text Classification," *International Journal of Computer Science and Information Technology*, vol. 11, no. 2, pp. 25-30, Feb. 2022.

[21] E. Oriola and E. Kotzé, "Implementing Word2Vec for Improved Text Representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 567-578, Mar. 2022.