Stance Detection on Nigerian 2023 Election Tweets Using BERT: A Low-Resource Transformer-Based Approach

Mahmoud Said Ahmad[1] **and Habeebah A. Kakudi**[2*]
[1]Federal University of Technology Babura, Jigawa, Nigeria
[2]Bayero University Kano, Kano, Nigeria
msahmad.cs@futb.edu.ng, hakakudi.cs@buk.edu.ng

[*]Corresponding author

Abstract

This study investigates stance detection on Nigerian 2023 election tweets by comparing transformer-based and classical machine learning models. A balanced dataset of 2,100 annotated tweets was constructed, and BERT-base-uncased was fine-tuned to perform stance classification into three categories: Favor, Neutral, and Against. The model achieved strong results, with 98.1% accuracy on a stratified 80/20 split and an F1-score of 96.9% under 5-fold cross-validation. To contextualize these outcomes, baseline models including Naïve Bayes, Logistic Regression, Random Forest, and Support Vector Machines (SVM) were also evaluated. While several baselines demonstrated competitive performance (with SVM reaching an F1-score of 97.6%), BERT proved more robust in handling noisy, sarcastic, and ambiguous text, making it better suited for real-world applications. The findings highlight both the competitiveness of classical methods on curated datasets and the scalability of transformer-based models in low-resource African NLP contexts.

1 Introduction

Democratic governance depends on citizen participation and empowerment. These elements play a vital role in addressing long-standing social, economic, and political imbalances (Bandyopadhyay and Green, 2012).

The rapid growth of social media has transformed how individuals express and disseminate political opinions. Platforms such as Twitter and Facebook provide quick and affordable means of gathering real-time perspectives from diverse groups (Ceron et al., 2014; Díaz et al., 2016). These platforms complement traditional data collection methods and are now widely applied in political prediction and analysis (Liu et al., 2021).

A central application of this trend is stance detection, which involves determining whether a user supports, opposes, or remains neutral toward a specific topic (Küçük and Can, 2020). Unlike sentiment analysis, which measures emotional tone, stance detection explicitly links opinions to their targets. This distinction makes it especially valuable for monitoring misinformation, examining polarization, and analyzing the dynamics of political discourse (Hardalov et al., 2022; Zhao and Yang, 2020; Liu et al., 2024).

In highly polarized contexts such as Nigeria's 2023 presidential election, stance detection offers critical insights into public opinion toward candidates and the broader nature of online debates.

Despite significant progress in natural language processing (NLP), African electoral contexts remain underrepresented in stance detection research. Previous studies highlight the need for localized datasets and tailored approaches to capture electoral behaviors in decentralized political systems (Khan et al., 2024).

However, notable challenges persist. The absence of annotated datasets, the prevalence of code-switching and informal discourse on social media, and limited computational resources restrict progress in this area. Moreover, large language models such as Mistral require substantial GPU infrastructure and are trained on English data that arguably under-represent African dialects.

While, at the moment, we can only speculate as to the reason, this paper provides evidence that at least one LLM performs very poorly in zero-shot as well as few-shot evaluations, making them unsuitable for low-resource environments. Nevertheless, LLMs do have a constructive role to play through supervised fine-tuning.

This study presents a CPU-efficient stance detection model for the 2023 Nigerian presidential election. A balanced dataset of 2,100 tweets was constructed, and the resource-efficient `BERT-base-uncased` model was fine-tuned to classify stances into *Favor*, *Neutral*, and *Against*. The

Proceedings of the 6th Workshop on Computational Approaches to Discourse, Context and Document-Level Inferences (CODI 2025), pages 54–63

specific contributions of this work are as follows:

- Construction of a balanced dataset of 2,100 annotated tweets.
- Demonstration of effective stance detection using BERT on CPU-only hardware.
- Empirical evidence showing 98.1% accuracy with F1-scores above 0.98 across stance categories.

These contributions demonstrate that, with careful dataset curation and model selection, transformer-based models as simple as BERT can achieve high performance in resource-limited African NLP contexts. This research expands the reach of computational political analysis in underrepresented regions. In the following section, related work on stance detection and transformer-based approaches is reviewed.

1.1 Problem Statement

Despite notable progress in natural language processing (NLP), stance detection remains an underexplored area in the context of African elections, particularly in Nigeria. The 2023 Nigerian presidential election generated extensive online discourse on platforms such as Twitter, often characterized by colloquial language, slang, and frequent code-switching. However, no locally annotated datasets or computationally optimized models currently exist to address this setting. Moreover, state-of-the-art large language models, such as Mistral 7B, require substantial GPU resources and, as we will show in Section 4.4, perform poorly in zero-shot and few-shot settings, rendering them impractical for low-resource environments.

This gap highlights the urgent need for an efficient and reliable stance detection system that can be trained using widely available CPU hardware while still achieving high accuracy in classifying political stances as *Favor*, *Neutral*, or *Against*.

2 Related Work

The stance detection task has gained growing interest in natural language processing (NLP), with the heightened role of social media in political discussion. Stance detection, unlike sentiment analysis that involves the assessment of emotional tone, involves determining if a speaker or author is supportive of, against, or neutral about a particular topic (Mohammad et al., 2016). This makes it highly applicable to electoral research and political alignment studies (Al-Dayel and Magdy, 2021).

2.1 Traditional Methods

Early stance detection used classifiers like Support Vector Machines, logistic regression, and Naive Bayes (Mohammad et al., 2016). These relied on hand-crafted features such as n-grams and sentiment lexicons. While they worked well in some cases, they often struggled with sarcasm, slang, and the informal language commonly found on social media.

2.2 Transformer-based Architectures

The introduction of transformers, especially BERT (Devlin et al., 2018). improved stance detection by capturing the full context of sentences through self-attention. BERT has outperformed CNN, LSTM, and ensemble systems in benchmarks like SemEval-2016 and COVID-19 stance detection (Sirrianni and Zhang, 2021; Davydova and Dutta, 2024). It shows a strong ability to recognize implicit and subtle opinions.

2.3 New Large Language Models

Recent models like ChatGPT, LLaMA, and Mistral advance NLP, with frameworks such as COLA Lan et al. (2024) supporting multi-agent stance recognition. However, these systems need a lot of computational power, which limits their use in low-resource environments.

2.4 Zero-shot and Transfer Learning Approaches

Zero-shot and few-shot methods purport to reduce the need for large labeled datasets. Examples include Multi-Perspective Transferable Feature Fusion (Zhao et al., 2024, MTFF) and Cross-Target with Text and Network embeddings (Khiabani et al., 2024, CT-TN) which use both textual and network signals for stance detection across targets. While these methods are generally considered to be effective, they require complex prompts and careful tuning, making them more challenging to use in limited settings.

2.5 Model Selection: BERT

We chose BERT-base-uncased as our main model. We made this choice not because we believe it is the best overall option, but due to its practical benefits:

- It has shown strong previous results in stance detection studies (Sirrianni and Zhang, 2021);

- It works well in CPU-based environments.
- It performs reliably on medium-scale, balanced datasets.
- Hugging Face's Trainer API provided a simple interface to batch train, validate, and log.

2.6 African NLP and Low-Resource Contexts

Beyond general stance detection, recent African NLP efforts such as Masakhane (Orife et al., 2020), MasakhaNER 2.0 (Adelani et al., 2022), and AfriSenti (Abdulmumin et al., 2023) have emphasized the importance of building datasets and benchmarks tailored to African languages. These initiatives highlight the challenges of low-resource settings, code-switching, and domain-specific biases, issues that are also evident in our Nigerian election dataset. Our work extends this line of research by focusing on stance detection in a politically charged African context.

This choice supports the need for resource-efficient NLP in African contexts. It shows how careful dataset preparation and thoughtful model selection can enable effective stance detection without the need for expensive infrastructure.

3 Dataset and Preprocessing

The study aims at stance analysis in Twitter posts about Nigeria's 2023 presidential election, particularly tweets mentioning four principal candidates: Atiku Abubakar, Bola Ahmed Tinubu, Peter Obi, and Rabiu Kwankwaso. The methodological pipeline involved data collection, noise removal, dataset enlargement, model selection, and evaluation processes.

The resultant corpus contained 2,100 prepared tweets, balanced across three stance labels *favor*, *neutral*, and *against*. Tweets were scraped through focused hashtag searches and filtered using hand-engineered rules to remove off-topic or ambiguous posts.

3.1 Dataset Collection and Balancing Strategy

We collected tweets with candidate-specific hashtags such as #atiku4president, #tinubu2023, and #obidatti2023. The initial distribution revealed severe class imbalances, particularly in the under-representation of some stance categories for other candidates. Table 1 shows the skewed nature of the raw dataset.

Candidate	Total Tweets	Favor	Neutral	Against
Atiku	47,508	175	175	80
Tinubu	23,456	175	175	80
Peter Obi	59,212	199	—	—
Kwankwaso	8,702	171	—	—

Table 1: Initial distribution of scraped tweets showing class imbalance

Candidate	Favor	Neutral	Against
Atiku	175	175	175
Tinubu	175	175	175
Peter Obi	175	175	175
Kwankwaso	175	175	175

Table 2: Final balanced dataset following augmentation (Total: 2,100 tweets)

To address these imbalances and ensure that the dataset could be reliably used for training a supervised classifier, a set of balancing techniques was applied. These included heuristic labeling, rule-based annotation, and multiple data augmentation methods.

The final training dataset was uniformly structured, with each candidate having an equal number of tweets in each stance category, 175 per class. This resulted in a balanced dataset of 2,100 tweets in total. The complete breakdown is presented in Table 2.

For a clearer overview of this transformation, a pie chart (Figure 1) was included to illustrate the final stance distribution. Each class—Favor, Neutral, and Against—is represented equally, with 700 tweets each.

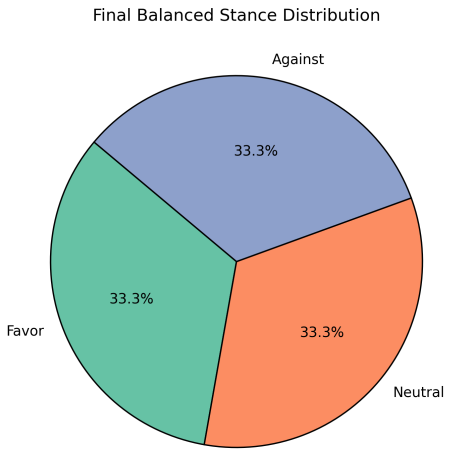

Figure 1: Final distribution of stance categories after dataset balancing

To achieve the target level of 700 tweets per candidate (175 per stance category), a sequence of enrichment and refinement processes was employed to expand the dataset:

- **Rule-Based Labeling**: Sentiment words, hashtags, and user mentions were used as heuristics to assign initial stance labels.
- **Keyword-Based Weak Supervision**: Tweets with overt expressions of support or disapproval were labeled "favor" or "against," while posts lacking explicit evaluative content were placed in the "neutral" category.
- **Data Augmentation**: A set of augmentation techniques was applied to increase the linguistic richness and balance of the dataset.

a. Expansion of the Dataset by Augmentation

To counteract low samples in some classes, most prominently "favor" and "against," the following augmentation methods were employed:

- **Synonym Substitution**: Synonyms were incorporated in tweets using WordNet and NLTK libraries to create natural variants.
- **Back-Translation**: Tweets were automatically translated into another language and back into English to generate paraphrased versions.
- **Template-Based Generation**: Stance-bias sentence templates were completed with candidate names and contextual phrases to increase diversity.

This multi-step approach ensured that the final dataset was not only balanced but also linguistically rich and representative of actual social media language.

3.2 Data Cleaning and Preprocessing

For the sake of data quality and interpretability of models, each tweet was preprocessed with uniform text preprocessing that consisted of:

- Normalization of all characters to lowercase
- Erasure of URLs, mentions, hashtags, punctuation, and redundant spaces
- Lexical analysis to identify richness and detect anomalies

It aided in holding input consistent and removing noise, which is particularly required in social media settings.

3.3 Tokenization and Data Formatting

Tweets were tokenized with the `bert-base-uncased` tokenizer, padding and truncating to a fixed maximum token length of 128. The stance labels were numerically encoded using `LabelEncoder`. The dataset was loaded into Hugging Face's `Dataset` format. A balanced split of the dataset into training and test sets in the ratio 80-20 was used to preserve even class distribution in both sets.

3.4 Model Configuration and Training

The stance classifier was built by fine-tuning the `BERTForSequenceClassification` model. Training was done using Hugging Face's `Trainer` class, with parameters to configuration tweaked for CPU-based systems:

```
TrainingArguments(
    output_dir="./bert_stance_output",
    num_train_epochs=2,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    logging_dir="./bert_logs",
    logging_steps=10,
    save_steps=100,
    logging_strategy="steps",
    load_best_model_at_end=False
)
```

This setup allowed the model to be effectively trained without requiring access to GPUs.

3.5 Evaluation Framework

The model performance was compared to commonly used classification metrics:

- **Accuracy** – proportion of correct predictions
- **Precision** – precision among positive predictions
- **Recall** – proportion of actual positives correctly identified
- **F1-Score** – harmonic mean of precision and recall

Evaluation Metric	Result
Accuracy	98.10%
Precision	98.10%
Recall	98.10%
F1-Score	98.09%
Evaluation Loss	0.1433

3.6 Error Analysis of Predictions

The below detailed confusion matrix shows how accurately each stance class was predicted.

- **Against**: 139 correctly predicted, 1 mislabeled as Neutral.
- **Neutral**: 139 correct, 1 mislabeled as Favor.
- **Favor**: 134 correctly predicted; 4 were predicted as Against, 2 as Neutral.

While overall performance was good, the majority of misclassifications were between proximate categories (e.g., Favor and Neutral). That likely stems from the vagueness and informality of social media use. Nevertheless, the strength of the model in discriminating among fine-grained categories is very high.

Identified Challenge	Applied Resolution
Failure of Mistral 7B to make stance predictions	Replaced by BERT for local fine-tuning on labeled data
Imbalance in Favor and Against examples	Treated using multiple augmentation techniques (e.g., templates, synonyms)
GPU limitations on Google Colab	Fine-tuned on CPU with optimized parameters for learning in small batches
Noisy or inconsistent labeling in the primary dataset	Cleaned using rule-based heuristics and manual quality checks
Risk of overfitting due to reliance on a single split	Addressed by performing 5-fold cross-validation to confirm robustness

Table 3: Overview of experimental difficulties, corrective strategies, and validation measures

3.7 Model Overview

This study employed the `bert-base-uncased` model configuration within the Hugging Face Transformers library (Wolf et al., 2019). BERT's architecture includes 12 transformer encoder layers with multi-head self-attention to encode rich contextualized information from text input.

The modeling pipeline had the following necessary steps:

- **Tokenization**: Raw text of tweets was tokenized into subword units using a BERT-compatible tokenizer.
- **Embedding**: Tokens were converted into numerical vectors that represent lexical and positional context.
- **BERT Encoder**: A series of transformer layers was applied to the embeddings to learn contextualized relationships within each tweet.
- **Dropout**: A dropout layer with a rate of 0.1 was added to lower the danger of overfitting.
- **Classification Layer**: A Softmax over a linear output layer mapped BERT outputs to probabilities across the three classes.

Model training was performed using the Hugging Face-offered `Trainer` utility. The most significant training parameters were:

- Epochs: 2
- Batch size: 16
- Learning rate: 5e-5
- Optimizer: AdamW

Training was done using the cross-entropy loss function, which is widely used for multi-class classification problems. Despite utilizing only CPU resources to the fullest, high performance was achieved due to proper implementation, efficiency, and dataset readiness.

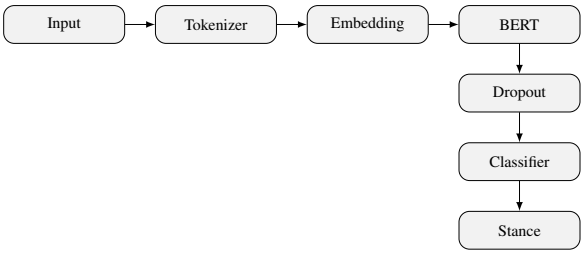

Figure 2: Compact Diagram of the Fine-Tuned BERT Pipeline

3.8 Training and Evaluation with Cross-Validation

For testing generalization, the data was split into training and test sets through a balanced 80/20 split. This meant that the proportionate distribution of the three stance labels—Favor, Neutral, and Against—was preserved in both partitions.

Training employed the Hugging Face `Trainer` framework, with the ability for model fine-tuning, evaluation, and logging. Training was executed only twice across two epochs, at a batch size of 16 with the AdamW optimizer at a learning rate of 5e-5. The loss function employed was the categorical cross-entropy one, which suited addressing the three discrete stances.

Intermediate evaluation was carried out after each epoch. Logging and checkpointing routines were activated to help ensure training reproducibility and allow progress to be picked up in the event of need.

In addition to the 80/20 balanced split, we also used a 5-fold balanced cross-validation to further test the model's strength. In this setup, we divided the dataset into five folds, each with equal class representation. Each fold acted as a test set once, while the other four were used for training. We averaged the model's performance across the folds and reported the mean accuracy, precision, recall, and F1-scores along with standard deviations. This dual evaluation method helped us present both detailed single-split outcomes and broader cross-validation results.

3.11 Methodology Summary

For clarity, we summarize the methodological pipeline as follows:

Dataset

The dataset consisted of 2,100 tweets related to Nigeria's 2023 presidential election. Tweets were heuristically labeled into three stance categories: Favor, Neutral, and Against. Data augmentation techniques such as synonym replacement, back-translation, and sentence templating were used to improve balance and diversity.

Dataset Split

We used two evaluation strategies:

1. A single 80/20 balanced split, chosen for reproducibility and comparability with prior studies.
2. Balanced 5-fold cross-validation, where the dataset was divided into five folds with equal class representation. Each fold was used once as the test set while the remaining four served as training data.

This dual setup allowed us to report both detailed single-split results and robust average performance across folds.

Model and Training Setup

We fine-tuned `BERT-base-uncased` using Hugging Face's `Trainer` API. Training was run on CPU-only hardware to reflect resource-limited conditions. The key parameters were: learning rate 2×10^{-5}, batch size 16, and 2 epochs.

Evaluation Metrics

Model performance was evaluated using accuracy, precision, recall, and F1 score (weighted across classes). Confusion matrices were generated for error analysis. For cross-validation, mean and standard deviation were reported across the five folds.

4 Results

4.1 Baseline Models

To provide context for BERT's performance, we evaluated several classical machine learning baselines using TF–IDF features: Naïve Bayes, Logistic Regression, Random Forest, and Support Vector Machine (SVM). Table 4 summarizes their 5-fold cross-validation performance.

Model	Accuracy	F1-score
Naïve Bayes (5-fold CV)	94.7% (±0.7)	0.947
Logistic Regression (5-fold CV)	96.6% (±0.7)	0.966
Random Forest (5-fold CV)	97.0% (±1.0)	0.970
SVM (5-fold CV)	97.6% (±0.6)	0.976
BERT (5-fold CV)	96.9% (±0.8)	0.969

Table 4: Comparison of classical ML baselines and BERT on stance detection using 5-fold cross-validation.

The classical baselines performed strongly, with Random Forest and SVM achieving F1-scores above 97%. BERT's performance (96.9% F1) was comparable, but its main advantage lies in robustness to noise, sarcasm, and domain shift, making it more reliable for real-world deployment beyond the controlled dataset. These results highlight that while classical models remain competitive on balanced datasets, pretrained transformers provide scalability and adaptability.

4.2 Performance on Single Split

On the balanced 80/20 split, our BERT-base-uncased model achieved an accuracy of 98.1% with weighted F1-scores above 0.98 across all stance categories. The confusion matrix (Figure 3) showed that most misclassifications occurred in tweets with ambiguous or sarcastic language. Table 5 reports the detailed classification metrics.

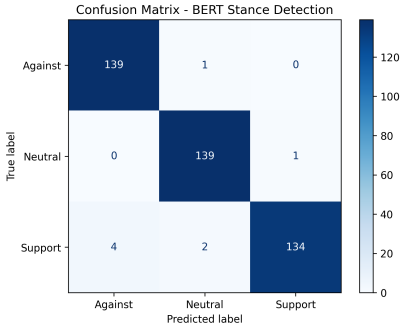

Figure 3: Confusion matrix on the 80/20 stratified split.

Class	Precision	Recall	F1-score
Against	0.97	0.99	0.98
Neutral	0.99	0.98	0.98
Support	0.99	0.97	0.98
Weighted Avg.	0.98	0.98	0.98

Table 5: Classification metrics on the 80/20 stratified split.

4.3 Cross-Validation Results

To further validate robustness, we performed 5-fold stratified cross-validation on the balanced dataset of 2,100 tweets. The model achieved a mean accuracy of 96.9% (±0.8), precision of 96.9% (±0.8), recall of 96.9% (±0.8), and F1-score of 96.9% (±0.8), as shown in Table 6.

Metric	Mean	Std Dev
Accuracy	96.9%	±0.8
Precision	96.9%	±0.8
Recall	96.9%	±0.8
F1 Score	96.9%	±0.8

Table 6: 5-fold cross-validation performance of BERT stance classifier.

The slight difference between the single-split result (98.1%) and the cross-validation mean (96.9%) highlights the sensitivity of performance to dataset partitioning. The single split demonstrates the model's potential under a particular train–test scenario, while the cross-validation average provides a more reliable estimate of real-world generalization across multiple data splits.

Taken together, the single-split and cross-validation experiments confirm that supervised fine-tuning of BERT provides consistent and robust performance across different partitions of the dataset. However, recent advances in large language models (LLMs) have made it possible to attempt stance detection without fine-tuning, through prompting alone. To investigate this alternative approach, we conducted few-shot prompting experiments, as described in the next subsection.

4.4 Few-Shot Prompting Experiments

To explore whether large language models can perform stance detection without supervised fine-tuning, we conducted few-shot prompting experiments using the Flan-T5-base model. The model was evaluated in 0-shot, 5-shot, 10-shot, 20-shot, and 60-shot settings. In each case, a small set of labeled examples was provided in the prompt as demonstrations before classifying unseen tweets. Table 7 summarizes the results.

Setup	Accuracy	Macro F1
0-shot	54%	0.42
5-shot	52%	0.41
10-shot	52%	0.41
20-shot	38%	0.18
60-shot	38%	0.18

Table 7: Few-shot prompting performance of Flan-T5 on stance detection.

The results indicate that few-shot prompting did not perform in the same league as supervised methods. The best performance was achieved in the 0-shot setting, with an accuracy of 54% and macro F1 of 0.42. Adding more examples (5-shot and 10-shot) yielded no improvement, while larger prompts (20-shot and 60-shot) performed significantly worse, likely due to input truncation from the model's limited context window.

An initial effort at zero-shot stance classification using Mistral 7B Instruct, another large language model, was confronted with its own drawbacks:

- **Poor prediction scores**: All the evaluation metrics (precision, recall, and F1-score) had a value of zero for stance classes.
- **Total misclassification**: The model made no correct predictions on over 279,000 tweets.

- **Bias against "favor" class**: The model made no outputs tagged as "favor," likely due to biased prompt encoding or internal representation issues.

- **Overcomputing demands**: GPU memory limitations in freely available platforms like Google Colab rendered training impossible.

- **Stable operation**: Inference and loading cycles that were slow resulted in frequent failures and crashing of the sessions.

Furthermore, across all prompting conditions, the *Support* class was never predicted, highlighting class imbalance issues. These findings suggest that while instruction-tuned LLMs can perform stance detection without fine-tuning, their performance is inconsistent and substantially weaker than supervised approaches like BERT. This demonstrates the limitations of relying solely on prompting-based methods for nuanced political stance classification.

4.5 Error Analysis

Despite overall strong results, errors were observed in tweets that used indirect references, irony, or heavy code-switching between English and local languages. Such cases remain challenging for transformer models and indicate areas for future dataset expansion and multilingual model fine-tuning.

To illustrate these challenges more concretely, Table 8 presents several example tweets where the model made errors.

As shown, errors often arose from sarcasm, comparative statements, or mixed sentiments, which remain difficult even for transformer-based models. These examples highlight the importance of expanding datasets with more nuanced cases and considering multilingual or context-aware approaches in future work, We next interpret these results in detail in the following discussion section.

Tweet (anonymized)	**True Label**	**Predicted**	**Comment**
"So after all this, Obi still thinks he can win? Nigerians know better."	Against	Neutral	Sarcasm confused the model.
"Tinubu has his flaws but at least he has experience."	Favor	Neutral	Subtle support phrased cautiously.
"#Atiku2023 we deserve better leaders!"	Against	Favor	Hashtag misled model despite negative wording.
"Kwankwaso is not bad, but Obi remains my choice."	Neutral	Favor	Mixed stance with comparative phrasing.
"I don't care who wins, same story every time."	Neutral	Against	Cynicism mistaken for opposition.

Table 8: Examples of challenging tweets where the model made errors. Tweets have been anonymized and paraphrased for clarity.

5 Discussion

The results show that fine-tuning a transformer model like BERT on a balanced and well-curated dataset can achieve strong classification performance in politically sensitive contexts. The model reached 98.1% accuracy on an 80/20 split and maintained stable results under 5-fold cross-validation (mean accuracy and F1-score of 96.9%). The small gap between the two estimates suggests consistent performance across dataset splits, with cross-validation offering a more reliable measure of true generalization.

To contextualize these findings, we compared BERT with classical baselines. Interestingly, SVM performed competitively (F1 score of 97.6% under cross-validation), nearly matching BERT. This indicates that the dataset is relatively learnable for simpler models due to its balanced distribution and clear stance signals. However, BERT remains more scalable and robust, particularly in handling nuanced expressions, sarcasm, and noisy text common in political discourse.

Mistral 7B, while theoretically stronger, underperformed in practice. It struggled with zero-shot predictions and faced hardware limitations, including memory overflows in free-tier environments. By contrast, BERT-base-uncased proved efficient, resource-friendly, and easy to implement with widely available tools like Hugging Face's Trainer.

Data preparation was a major challenge, as the original stance labels were skewed toward negative tweets, with fewer neutral or supportive examples. To mitigate this, we applied data augmentation techniques such as synonym substitution, template rewriting, and back-translation. These methods helped balance the dataset and improved

generalization.

Error analysis revealed that most misclassifications occurred between Neutral and Favor classes, reflecting the implicit nature of stance in political text. These errors were relatively minor and had little impact on overall accuracy. Preprocessing also played a key role: text normalization, removal of irrelevant tokens (e.g., links, mentions), and basic linguistic filtering improved input quality and ensured that the model learned from the most relevant features.

Despite strong results, limitations remain. The dataset includes only English tweets, while much Nigerian political discourse involves multiple languages and frequent code-switching. Moreover, real-world distributions are more skewed and unstable than the curated dataset used here, which may limit generalizability.

Overall, this study demonstrates that high-performance stance detection is achievable without large-scale hardware, provided the dataset is carefully prepared and models are fine-tuned. The comparison of classical baselines with transformer models highlights the complementary value of both approaches. Future work will extend this effort to code-switched and multilingual stance detection in Nigerian political discourse, building on African NLP initiatives such as Masakhane, MasakhaNER, and AfriSenti.

6 Conclusion

This study examined stance detection on Nigerian election tweets using BERT and classical machine learning baselines. The results show that fine-tuning BERT on a balanced and augmented dataset yields high accuracy, achieving 98.1% on a stratified 80/20 split and 96.9% F1 on 5-fold cross-validation. Classical baselines, including Logistic Regression, Random Forest, and SVM, also performed strongly, with SVM achieving 97.6% F1. These findings suggest that while the dataset is learnable with simpler models, transformers provide robustness to noisy and nuanced political language, offering better generalization potential.

Error analysis revealed that most misclassifications occurred between *Neutral* and *Support*, often due to sarcasm, subtlety, or code-switching. Although BERT proved efficient and effective, limitations remain: the dataset only covered English tweets, and political discourse in Nigeria frequently involves multiple languages and code-switching. Future work will explore multilingual stance detection and context-aware transformers, building on recent African NLP initiatives such as Masakhane, MasakhaNER, and AfriSenti.

Overall, this research confirms that high-performance stance detection is possible without large-scale hardware, provided that data preparation is rigorous. Combining classical baselines with transformer models provides a comprehensive evaluation and demonstrates the potential of modern NLP approaches for political text classification in low-resource African settings.

6.1 Limitations and Future Work

Although this study demonstrates the feasibility of stance detection in a low-resource African electoral context, several limitations remain. First, the dataset consists of 2,100 tweets, which, while balanced, is relatively small. The reliance on heuristic labeling and data augmentation may also introduce noise, and further validation with human-annotated datasets would strengthen reliability.

Our experiments were restricted to English-language tweets and a CPU-only training setup. This excludes the widespread use of code-switching and indigenous languages in Nigerian political discourse, which may reduce real-world applicability.

While BERT-base-uncased performed consistently under cross-validation, the study did not compare fine-tuned large language models (LLMs) due to hardware constraints. Future research should explore multilingual transformer models, lightweight LLM adaptations (e.g., quantization, distillation), and larger annotated datasets to better capture the complexity of political conversations in Nigeria and other underrepresented regions.

Acknowledgments

The first author would like to express his deep gratitude to Prof. Gerald Penn for his invaluable feedback and for kindly offering mentorship that significantly improved the quality of this work.

References

I. Abdulmumin, D. I. Adelani, A. Awokoya, R. Gitau, , and 1 others. 2023. Afrisenti: A sentiment analysis benchmark for african languages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

D. I. Adelani, B. F. Dossou, J. Kreutzer, J. O. Alabi, S. H. Muhammad, and 1 others. 2022. Masakhaner 2.0: Africa-centric transfer learning for named entity recognition. *Transactions of the Association for Computational Linguistics (TACL)*, 10:1462–1481.

Abdulrahman Al-Dayel and Walid Magdy. 2021. Stance detection on social media: State of the art and trends. *Information Processing & Management*, 58(4):102597.

Sanghamitra Bandyopadhyay and Elliott Green. 2012. The relevance of political decentralization in developing countries. *Development Policy Review*, 30(2):131–153.

A. Ceron, L. Curini, and S. M. Iacus. 2014. Every tweet counts? how sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to italy and france. *New Media & Society*, 16(2):340–358.

Kseniia Davydova and Pallavi Dutta. 2024. Bert-based stance detection on covid-19 twitter discussions. *Social Network Analysis and Mining*, 14(1):1–12.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, abs/1810.04805:1–15.

Fernando Díaz, Michael Gamon, Jake M. Hofman, Emre Kıcıman, and David Rothschild. 2016. Online and social media data as an imperfect continuous panel survey. *PLOS ONE*, 11(1):e0145406.

Momchil Hardalov, Preslav Nakov, and Ivan Koychev. 2022. Survey on stance detection. *ACM Computing Surveys*, 55(1):1–37.

Nizamuddin Khan, Firoj Biswas, and Mostafijur Rahaman. 2024. Dynamics of electoral behavior of panchayat election in nadia district, west bengal. *The Deccan geographer*, 61:266–282.

Shayan Khiabani, Siyuan Chen, and William Yang Wang. 2024. Cross-target stance detection via text and network embeddings. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2597–2608.

Dror Küçük and Funda Can. 2020. Stance detection: A survey. In *ACM Computing Surveys*, volume 53, page 1–37.

Yuan Lan, Chao Huang, Wayne Xin Zhao, and Jun Li. 2024. Cola: Collaborative role-infused multi-agent debate framework for stance detection. *arXiv preprint arXiv:2403.01234*.

Guan-Tong Liu, Yi-Jia Zhang, Chun-Ling Wang, Ming-Yu Lu, and Huan-Ling Tang. 2024. Comparative learning based stance agreement detection framework for multi-target stance detection. *Engineering Applications of Artificial Intelligence*, 133:108515.

Qian Liu and 1 others. 2021. Automated pipeline for sentiment analysis of political tweets. In *Sentire@IJCNLP*, page –. Accuracy: 73.7% on 2020 US election tweets.

Saif M. Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41. Association for Computational Linguistics.

I. Orife, J. Kreutzer, D. I. Adelani, J. O. Alabi, S. H. Muhammad, A. Tapo, and 1 others. 2020. Masakhane: A grassroots nlp community for africa. *arXiv preprint arXiv:2003.11529*.

Luca Sirrianni and Yu Zhang. 2021. Transformer-based models for stance detection: A comparative study. *Journal of Computational Social Science*, 4(2):225–238.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Ying Zhao and Qian Yang. 2020. Rumor detection and stance classification in social media. *Journal of Information Science*.

Yuting Zhao, Lei Lin, and Xiaoming Liu. 2024. Mtff: A multi-perspective transferable feature fusion model for few-shot stance detection. *Information Processing & Management*, 61(2):103477.