International Journal of Information Management Data Insights 5 (2025) 100330

Contents lists available at ScienceDirect

International Journal of Information Management Data Insights

journal homepage: www.elsevier.com/locate/jjimei

Deep learning techniques for sentiment analysis in code-switched Hausa-English tweets

Yusuf Aliyu [a,*], Aliza Sarlan [b], Kamaluddeen Usman Danyaro [a,c,*], Abdullahi Sani abd Rahman [a], Aminu Aminu Muazu [a,d], Mustapha Yusuf Abubakar [e]

[a] Department of Computer and Information Science, Universiti Teknologi PETRONAS, Seri Iskandar 32610 Perak, Malaysia
[b] Center for Foundation Studies, Universiti Teknologi PETRONAS, Seri Iskandar, Perak, 32610, Malaysia
[c] Centre for Cyber-Physical Systems (C2PS), Universiti Teknologi PETRONAS, Seri Iskandar, Perak 32610, Malaysia
[d] Computer Science Department, Faculty of Natural and Applied Science, Umaru Musa Yar'adua University, Katsina, Nigeria
[e] Computer Science Department, School of Technology, Kano State Polytechnic, Kano 700231, Nigeria

ARTICLE INFO

Keywords:
Sentiment analysis
Low-resource
Code-switched
Hausa language
Word-embedding
Transformer

ABSTRACT

Social media serve as a crucial platform for expressing opinions and perspectives. Its texts often characterised by code-switching or mixed languages in multilingual setting. This results in a diverse and complex linguistic context, which can negatively affect the accuracy of sentiment analysis for low-resource languages such as Hausa. Prior research has predominantly concentrated on sentiment analysis within single-language data rather than code-switched data. This paper proposes an efficient hyperparameter tuning framework and a novel stemming algorithm for the Hausa language. The framework leverages word embeddings to determine the polarity scores of code-mixed tweets and enhances the accuracy of sentiment analysis models in low-resource language. The extensive experiments demonstrate the framework's efficiency and reveal a superior performance of transformer models over conventional deep learning models. The framework achieves a balance between accuracy and computational efficiency, making it suitable for deployment in practical applications. Compared to state-of-the-art transformer models, our framework significantly reduces computational costs while maintaining competitive performance. Notably, the AfriBERTa model achieves outstanding results, with an F1-score of 0.92 and an accuracy of 0.919, surpassing current baseline standards. These findings have broad implications for social media monitoring, customer feedback analysis, and public sentiment tracking, enabling more inclusive and accessible NLP tools for underrepresented linguistic communities.

1. Introduction

The rise of social media led to a constant influx of data in various linguistic formats, reflecting its users' diverse languages and cultures (Amjad et al., 2021). This multilingual context data presents opportunities for different organisations to gain insight from it. With the exponential growth of user-generated content on social media platforms such as X, formerly referred to as Twitter. Researchers and industries increasingly leverage this wealth of information for applications such as sentiment analysis, trend prediction, emotion detection, hate speech detection and social media monitoring. Sentiment analysis (SA) is referred to as opinion mining (Ganganwar & Rajalakshmi, 2019; Yue et al., 2019). It is an area of research that examines individuals' attitudes, emotions, appraisals, and opinions towards entities and their various aspects as expressed through written text (Al Shamsi & Abdallah, 2023). This field of study aims to analyse and understand people's sentiment toward a particular topic or entity by extracting relevant information from textual data (Liu, 2015). Moreover, the growing importance of SA extended beyond high-resource languages such as English, Chinese, Spanish, French and Arabic. It found its increasing importance in multilingual settings. These languages remain the dominant language studied in SA, and little effort is made to study low-resource languages. Social media has led to a notable rise in the population of individuals conversing in these languages and who prefer to use them for written communication (Londhe et al., 2021). Their expressions can be found in a variety of forms, including text, image, info-graphic, audio, video, and emoticons (Meena et al., 2023). However, extracting and interpreting the sentiments in their expression

* Corresponding authors.
E-mail addresses: yusuf_22005103@utp.edu.my (Y. Aliyu), kamaluddeen.usman@utp.edu.my (K.U. Danyaro).

https://doi.org/10.1016/j.jjimei.2025.100330

Available online 5 April 2025

brings a unique challenge known as code-mixed or code-switched.

Code-switching is the ability to use two or more languages within a single conversation of any communication (Konate & Du, 2018; Mahadzir, 2021; Song et al., 2019; Srinivasan & Subalalitha, 2023). The text shows the tendencies for intra-sentential code-mixing (within the sentence) and inter-sentential code-mixing (across the sentence), inventive spelling, lexical borrowings and phonetic typing (Konate & Du, 2018). Similarly, traditional low-resource SA models were mostly designed for monolingual datasets (Roy, 2024). The models usually struggle to handle the linguistic diversity and semantic complexity of the code-switched text (Kuwanto et al., 2024). This resulted in reduced performance. Moreover, the challenges become more evident when focusing on multilingual data (Meena & Mohbey, 2023) and low-resource languages like Hausa. Despite being widely spoken (Magueresse et al., 2020; Suleiman et al., 2019), the limited availability of linguistic resources poses a significant obstacle to effective SA applications (Abdulmumin & Galadanci, 2019; Abubakar et al., 2021; Ibrahim et al., 2022; Muhammad et al., 2022). Addressing these limitations requires innovative approaches and tailored models that capture the nuances of the Hausa language, enabling more accurate and insightful SA. With the increasing use of social media and the reliance on digital platforms for communication, analysing low-resource language text sentiment is more important than ever. In addition, deep learning (DL) algorithms offer a promising solution (Meena et al., 2023). It has the ability to leverage large datasets to classify sentiment with high accuracy.

Most prior studies overlook the code-switching factor in SA models for low-resource languages. Furthermore, existing methods and models for SA in low-resource languages are constrained by computational inefficiency and reliance on large, labelled datasets. Unlike existing methods, this study introduces a framework that combines pre-trained embeddings with contextual feature extraction to enhance sentiment classification in code-switched data. The method adopts a quantitative approach that focuses on a computational method. However, integrating word similarity and contextual features has the potential to significantly enhance model performance for code-switched text. These linguistic features, coupled with the fact that much of the content circulating online is produced in low-resource languages, underpin the compelling need for more rigorous research in this area. Code-switching practice can have a detrimental impact on the precision of SA results. As a result, it becomes quite challenging to produce suitable approaches for the SA process. This research focuses on addressing these challenges by performing SA in code-switching text within low-resource like Hausa language. It aims to investigate the impact of code-switching tweets and enhance the accuracy performance of SA in low-resource language. While balancing computational efficiency, a critical need for underrepresented linguistic communities. The present study offers the following contributions to the research field:

- The study proposes a hyperparameter framework that effectively handles the linguistic complexities of code-switched text by integrating pre-trained embeddings (e.g., FastText, GloVe) with contextual features extracted from sequence models.
- The study introduces novel dictionary-word pairs combined with a rule-based stemming algorithm tailored for code-switched text to improve the pre-processing pipeline. Which handles linguistic variations effectively.
- The study evaluates the framework's performance and demonstrates its effectiveness against state-of-the-art benchmark techniques on standard datasets.

The subsequent sections of this paper are organized as follows: Literature Review discusses other research studies regarding sentiment classification techniques; the Methodology describes the intended process flow, data, preprocessing, experimentation, proposed hyperparameters optimization, deep learning models, feature extraction, statistical significant test and evaluation metric. Results and Discussion present the results obtained and the evaluation and the conclusion section provide the conclusion of the study and the key findings.

2. Related works

This section explores recent advancements in SA, particularly in the context of low-resource languages. Scholars have applied several methods leveraging machine learning (ML) and DL, to analyse sentiments in different languages.

Some scholars have tried to use low-resource language-related tweet data to identify and categorise objectionable text polarity. Abubakar et al. (2021) presents a strategy for carrying out multilingual SA between English and Hausa tweets using the skip-gram (SG) Algorithm of ML and evaluates the effectiveness of Support Vector Machine (SVM), Naive Bayes (NB) and Maximum Entropy (MaxEnt) classifiers on the obtained datasets. The results show that the classifiers achieve an accuracy of 56% using the SVM classifier pure Hausa dataset. Similarly, the study of Rakhmanov & Schlippe (2022) performed the comparative analysis of the monolingual and cross-lingual approach of sentiment analysis on the Hausa English language dataset. The authors utilised Google Translate to translate a large portion of the data before running the machine model. Furthermore, they achieved a reasonable level of accuracy with Hausa-English machine translation, but it should be noted that the dataset is domain-specific and may not be generalisable to other areas.

Additionally, one of the significant challenges in low-resource SA is the insufficient availability of word embedding corpora, which is essential for developing effective NLP solutions. The study by Abdulmumin & Galadanci (2019), undertaken to overcome this drawback, they propose word embedding models using Word2Vec's CBoW and SG models. These models were employed to forecast the 10 most analogous words for 30 randomly chosen Hausa words. The outcomes revealed an accuracy of 88.7% for CBoW and 79.3% for SG. A study by Ogueji et al. (2021) introduced AfriBERTa, an African version of Bidirectional Encoder Representation from Transformer (BERT) trained from scratch to support 11 African languages, to mitigate the challenge of presenting low-resource languages for text classification tasks. Another research conducted by Muhammad et al. (2022), presents NaijaSenti, which marks as the first large-scale human-annotated dataset for SA. It is intended to cope with the difficulties with of availability of resources poses for African languages. The study compared various pre-trained models including multilingual BERT (mBERT), Cross-lingual language model robust (XLM-R), multilingual DeBERTa V3, and AfriBERTa and fine-tuned their dataset in language adaptive manners.

Shehu et al. (2024) introduced a novel approach combining Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Hierarchical Attention Networks (HAN) with a lexicon-based dictionary and a customized stemming technique to enhance the performance of the bag-of-words method for Hausa SA. Despite the innovative integration of these components, the method achieved a modest accuracy of 68.48%, highlighting significant potential for further optimization. Moreover, the bag-of-words representation exhibits inherent limitations, particularly in capturing the syntactic and semantic complexities of textual expressions, which may lead to suboptimal sentiment classification outcomes. Moreover, the study performed by the authors Yusuf et al. (2023), fine-tuned different transformers for sentiment classification of the Hausa language. The multilingual pre-trained language models are, RoBERTa, XLM- R and mBERT. Their findings show that mBERT base-cased achieves the peak F1 of 0.73. In a similar research by Konate & Du (2018) explores the area of SA in the low-resource language known as the Bambara-French language. To this end, the researchers perform an experiment on Bambara-French social network code-mixed data using CNNs and LSTM-based DL architectures on which appropriate frameworks are built specifically for code-mixed social network data. Their approach entails the creation of sentence

and comment representation using both characters and word embeddings using fixed word embeddings. Most significantly, their cross-validation indicates that using a one-layer CNN model in particular results in better performance, compared to other models, in terms of accuracy at 83.3%.

Similarly, Singh et al. (2020) presents a datasets benchmark to identify polarity on social media datasets and find offensive sentiment phrases in the Nepali language. Moreover, the experiment was done using genism (Řehůřek & Sojka, 2010) and trained 300-dimension SG fast-text word embeddings on mono and multilingual datasets. However, the classification task achieved a good accuracy of 0.805 using Bidirectional (BiLSTM) models. Kanclerz et al. (2020) performed an experiment using the transfer learning approach for polarity classification techniques; they used two alternative neural network architectures of BiLSTM and CNN. The authors used Language agnostic embeddings, they conducted evaluated on their cross-lingual model using the translated PolEmo 1.0 (Kocoń et al., 2019) sentiment corpus test datasets from different languages. The concludes that when it comes to comparing different models with single-domain datasets, the use of BiLSTM network model together with language-neutral phrase embeddings displayed a better performance on the task. Similar work by Jamatia et al. (2020) addressed sentiments extraction from code-mixed text as well as the relatively low predictive capacity of traditional ML learning models to their DL counterparts – specifically, models LSTM, CNN, and BiLSTM. Applying code-mixed data from Hindi-English and Bengali-English, the experiment showed that attention-based models outperformed traditional models by the contribution of 20–60%. Also, when compute on monolingual English data with the monolingual only dataset, the proposed method had an impressive accuracy of 72.6%.

Many language models have been trained for the Vietnamese language (VL) focused on monolingual and multilingual model variants. However, these models differ in terms of structure, and resulted differently in their performance during the fine-tuning as state by Thin et al. (2023). They also noted in their study that more research has been done in comparing these models at one time, but not many of them have been thoroughly evaluated using the same SA datasets. To fill this gap, they propose an approach to fine-tuning that differs in existing VL models for SA. In their experiments, they found that the single-Vietnamese model PhoBERT and Vietnamese Text-to-Text model ViT5 out-performed other models and set new records on five Vietnamese SA datasets. Additionally, a related study by Vo et al. (2023) analysed sentiments in the same language but was more focused on the educational domain. They focused on evaluating social attitudes towards the University of Phan Thiet, with a special building a sentiment corpus on the university. The overall investigation was carried out using DL models which include LSTM, BERT, DistilBERT, and PhoBERT. Experimental outcomes show PhoBERT model yielded the best results, and it have an F1-score of 89.68%. This has so illustrated how PhoBERT is well suited to handle SA issues within a low resource language learning environment

A number of research have been done using different methods and approaches. However, The limitation observed in several studies is the use of traditional ML techniques as utilised by Abubakar et al. (2021) in their models. These methods often yield accuracy below benchmark standards. In the same way, the use of word embedding in a domain-specific data with the use of translators as seen in these studies (Konate & Du, 2018; Rakhmanov & Schlippe, 2022; Sabri et al., 2021) and (Abubakar et al., 2021). It is constrained of data for specific domain and is only efficient on the available translation tools or Machine Neural Translation MNT models (Isbister et al., 2021). Moreover, word embedding is known to pay little attention to word dependency as well as other contextual characteristics of the text. It also sometimes fails in the polysemy disambiguation as commented by Bensoltane & Zaki (2021), during the process of correcting associate meaning for a word with a different context (Bensoltane & Zaki, 2021). Transfer learning techniques have advantages over starting SA from scratch or initiating training a model. The technique consumes a lot of time during training (Tao & Fang, 2020). However, recent research indicates that there is still much potential for improvement of transfer learning despite its effectiveness in a multilingual corpus (Pires et al., 2019).

Another limitation observed across these studies in the Hausa language is the phenomenon of code-switching. Although, studies of Muhammad et al. (2022) and Wang et al. (2023) acknowledge the practice of code-switching in their work. they have not explicitly addressed this challenge, and this can complicate SA especially, resulting to possible wrong sentiment scores. To overcome this affirmation of code-switching in Hausa, the following strategies have been addressed in this study. However, combining pre-trained embeddings with the DL model was proven to be more reliable (Hasib et al., 2021). To enhance sentiment classification in code-switched data, a better fine-tuning of the model's parameters and regularisation can achieve higher accuracy of sentiment expressions. Table 1 shows the summaries of the related studies together with their respective contribution and limitations to low-resource sentiment analysis. It is shown that word embeddings from different languages can be integrated as an effective approach in the code-switched task (James et al., 2022; Winata et al., 2021). This method provides a specialised approach that offers an effective framework for language modelling due to their ability to comprehend word semantic and sequential data (Zhang et al., 2018). Each word is semantically linked to its neighbouring words in the vector space. Additionally, DL models excel in capturing long-range dependencies (Demotte et al., 2020), thereby enhancing the model's adaptability. Leveraging word embeddings is applied to generate additional samples that exhibit lexical similarity (Tan et al., 2022).

3. Methods

This section outlines our methodology for low-resource SA. The method consists of several phases. It includes data exploratory analysis, preprocessing, our novel stemming and the proposed efficient hyperparameter tuning framework. Integration of an advanced word embedding approach and contextual feature methodologies for feature mapping in deep learning are presented. These joinings are proposed to address the need for the essential multilingual capabilities of the model to build on existing semantic embeddings.

3.1. Dataset

This study utilises Twitter datasets obtained from Github,[1] which is publicly available. These datasets have been made available as open-source resources specifically designed for research purposes and come pre-labelled. The dataset's rigorous annotation process involves three native speakers per language, aged 20–45, with expertise in computer science and linguistics. The training was conducted using the LightTag (Perry, 2021) tool through three iterations of 100 tweets. Data annotation was executed in batches of 1000 tweets, with adjudication of disagreements and the introduction of a unique majority vote approach in cases of subjective disagreement. The study disclosed varying inter-annotator agreements across languages. Continuous monitoring and adjustments were implemented in over 30 batches, with human evaluations of 200 tweets confirming corpus reliability. Table 2 illustrates the sample of the dataset with the code-mixed practices.

Furthermore, The data set consists of 16,849 instances of tweet accounts, where 5574 instances are categorised as positive, 5467 as negative, and 5808 as neutral. This distribution indicates a reasonable representation of the three categories, which helps minimise the potential biased predictions toward any particular categories or label. Fig. 1 displays a distribution of the dataset's class categories. Moreover, making it comparable to prior work by Abubakar et al. (2021) and (Rakhmanov & Schlippe, 2022), which used a dataset of similar

[1] https://github.com/hausanlp/NaijaSenti/tree/main/data

Table 1
Comparative analysis summary of SA approaches in low-resource languages.

Ref	Contribution	Technique/Model	Language	Metric (f1-score)	Limitation
(Abubakar et al., 2021)	Sentiment analysis in monolingual and multilingual data text.	SG, SVM, MxtEnt and NB	Hausa and English	68%	Word dependencies may be overlooked, leading to significant information gaps.
(Rakhmanov & Schlippe, 2022)	Sentiment analysis in monolingual and cross-lingual approaches to classify student comments in course evaluations.	LSTM, MLP, BERT, RoBERTa	Hausa-English	91.3%	The dataset is domain-specific, limited solely to the educational domain, thus restricting its applicability to other areas of study.
(Abdulmumin & Galadanci, 2019)	Proposed of word embedding corpus in Hausa language to use for sentiment analysis.	CBoW and SG	Hausa	88.7% and 79.3%	Corpus creation for research progress.
(Ogueji et al., 2021)	Proposed multilingual language models for text in low resource.	mBERT XLM-R	11 different languages	90.86%	The text classification models proposed is tailored for specific tasks other than sentiment analysis.
(Muhammad et al., 2022)	Introduced large-scale human-annotated Twitter sentiment dataset.	mBERT, XML-R, and AfriBERT	Hausa, Igbo, Yoruba and Pidgin	81.5%	The phenomenon of code-switching is acknowledged, no explicit measures have been undertaken to address it within the context discussed.
(Wang et al., 2023)	SemEval-2023 Task 12: Sentiment Analysis for African Languages competition	AfroXLMR	Multilingual sentiment classification	75.06%	No proactive measures have been implemented to address the code-switching-related problem.
(Jamatia et al., 2020)	Proposed social media code-mixed challenges in Hindi-English and Bengali-English.	BiLSTM, CNN, BERT	Hindi-English and Bengali-English	63.3%	The observed low accuracy underscores the potential values of constructing data augmentation systems.
(Thin et al., 2023)	Presented comprehensive assessment of transformer models on identical sentiment analysis datasets in Vietnamese	mBERT,XLM-r,mT5,ViT5, PhoBERT, viBERT4news, viBERT, and viELECTRA	Vietnamese	64.65%	The study inadequately addresses the guidance on their practical application in other languages.
(Konate & Du, 2018)	Social media code-mixed or code -switching in Bambara low-resource language.	LSTM, BiLSTM, CNN	Bambara-French	83.3%	The study overlooks word dependency as a factor in its analysis.
(Sabri et al., 2021)	Polarity ratings of tweets in a Persian-English code-mixed data set.	BiLSTM	Persian-English	66.17%	Depend on the accuracy of the translation
(Shehu et al., 2024)	Present a combine models with Hierarchical Attention Networks and rule-based stemming algorithms for Hausa language sentiment analysis.	CNN,RNN and HAN	Hausa	68.48%	The study overlooks word dependency and contextual semantic of an expression. They used a domain specific data which upon which the method cannot be generalized to other domain

linguistic characteristics but smaller size and in educational domain. However, unlike prior work, this dataset includes additional noise (e.g., misspellings, code-mixed) to reflect real-world scenarios.

The code-mixed tweets were identified through the utilisation of the open-source Detect Language API, accessible at https://detectlanguage.com. This API was employed to assess the extent of code-mixing within the datasets, distinguishing between Hausa and English tokens or words. The findings were instrumental in understanding the distribution of code-mixed content. Fig. 2 depicts in a visual representation the linguistic composition of the tweets, emphasising the prevalence of code-mixed content in the tweet.

3.2. Preprocessing

The preprocessing stage involved cleaning the data to ensure the correct format before training. It is an essential component of the process (Hasib et al., 2021). It includes data cleaning, data stemming and tokenization processes as follows:

3.2.1. Noise removal

The preprocessing begins with removing the Username, links, special characters, and digits. This process resulted in a clean and well-structured dataset token. It served as a foundation for accurate sentiment analysis to capture a fine cultural nuance in code-switched-language expressions. Fig. 3 Present the noise cleaning in the preprocessing stage.

3.2.2. Stemming

Stemming is also a common preprocessing step employed to enhance the effectiveness of the sentiment classification system (Al-Saqqa et al., 2019). It standardises code-switched text by converting words to their root forms. This process aligns tokens across different languages, as it enhances their consistency and improves representation in the embedding layer. Stemming also plays a crucial role in NLP tasks by reducing the dimensionality of the vector space ((Al Shamsi & Abdallah, 2022; Rajalakshmi et al., 2023)). Developing a stemmer for resource-poor languages presents significant challenges due to their limited linguistic resources (Jabbar et al., 2023). Therefore, selecting an appropriate stemming method depends on several factors, including the language's features, dataset size, and the specific problem. For low-resource languages, linguistic knowledge-based approaches or hybrid methods often yield better results (Jabbar et al., 2023).

However, the Hausa language has no standard tools in NLTK for performing word stemming. Bashir et al. (2015) and Bimba et al. (2016) performed stemming algorithm tools in Hausa and reported significant stemming errors of under and over-stemming. They employed a rule-based approach that is highly specific and may not generalise well to handle data in other language contexts. An improved method was proposed by Musa et al. (2022) using the same rule-based. Their method lacks the fine-grained control of linguistic rules, potentially missing some nuanced stemming patterns.

A modified method was introduced as a comprehensive Hausa stemming algorithm based on the dictionary word pairs and a rule-based. We created words-pairs of dictionaries, and more than 500 words were paired with their respective root words. The proposed stemming algorithm was designed to address the limitations of existing methods. As it incorporated a dictionary-based approach. The algorithm ensures accurate root word extraction while minimizing errors. The rule-based mechanism for prefix and suffix removal complements the dictionary by handling edge cases where words are not present in the dictionary. Furthermore, the algorithm's development involved close collaboration with Hausa linguistic professionals to ensure its validity

Table 2
Data sample with related code-switched practice.

No	*Tweet*	*Label*	*Remark(tweet)*
00110	@user Wato Sai ina Ga kamar hukumomin kasar nan Basu gano rikicin Katsina Zamfara da Sokoto ba rikicin **bandits** bane, wallahi **faction** ne na Boko Haram. Idan zasu yi da gaske suyi, **they are deciving us**. Zancen banza sulhu da Yan iska😣😣	Negative	Hausa
00166	@user BBC iyayen **propaganda**... Da Boko Haram din dakuma **IS** din duka Mutane dayane, Kuma iyayen gidansu dayane su **France, England** da **America**. @user @user @user @user @user ba abunda Alumma sukeso kuke yadawaba illah kuna yada abunda iyayen gidanku sukeso 😏	Negative	Hausa
03686	@user Kowa anan zai nuna shi **he is pure while he isa devil himself** wasunku masu zagin @user anan shahararrun yen iskane wasuma yen luwadi ne wasu kawalai ne kannanku wasu karuwai ne irin shigar dasukeyi ma ya baci sune zuwa **party,club** amma kunaku ba iskanci sukeba **idiots**😠😠😠	Negative	Hausa
05077	@user Adaiye a hankali dan gudun **Demolition** na **Venue**, dan wancen inn bashida dade koh kadan 🤣🤣🤣	Neutral	Hausa
05132	@user @user @user @user @user @user @user @user @user guy ***yau da Sa'a ka fito, na kai shekara uku ina*** posting pictures dinta on many events and occasions of her life (birthdays, awards, and some hit movies she produce) ***amma wlh bata taba koda yi min*** like **ba**, though a Instagram ***ne*** 😠😠😠	Neutral	English
05138	@user @user @user Pls someone should tell us d name of the skul ***sabida zamuna zu bi biya muga ya dan*** Governor ***yake ana damun shi*** 😂	Neutral	English
09598	@user Don't tell me this is happening in Algeria 😱 Hunger strike ***bayan*** Allah yace """""""""" and don't kill yourselves""""""""""? ***Allah ya shirye mu da kokoyo da kafirai***	Positive	English
09506	@user Looking beautiful Rahma, for your information guy's ***Wallahi*** I guarantee you ***Wallahi*** who ever say something wrong about Rahma again we will just say ***Allah ya isa, Da Buro'uba ku wayasan me kukeyi a gefe? Allah Shi Toni asirinku*** FOOLS. 😠 back to school. 👆	Positive	English
09574	@user Wow u luck beautiful ***barakallah Masha Allah, Allah ya qara daukaka da nisan kwana***. my best favorite actor... ♥	Positive	English

*red indicate Hausa tweet with English mixed. While ***blue*** shows English tweets with Hausa mixed.

*red indicate Hausa tweet with English mixed. While ***blue*** shows English tweets with Hausa mixed.

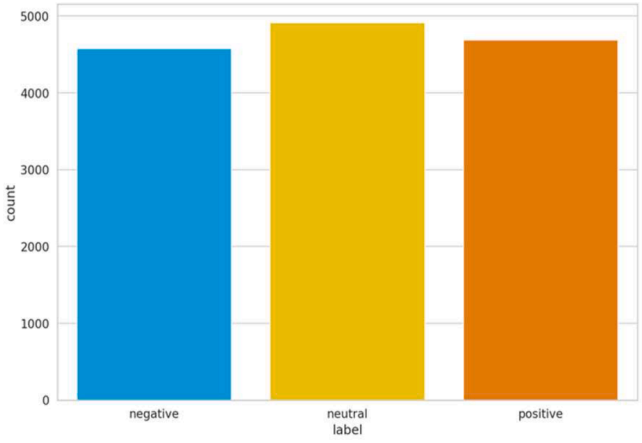

Fig. 1. Tweets Class distributions.

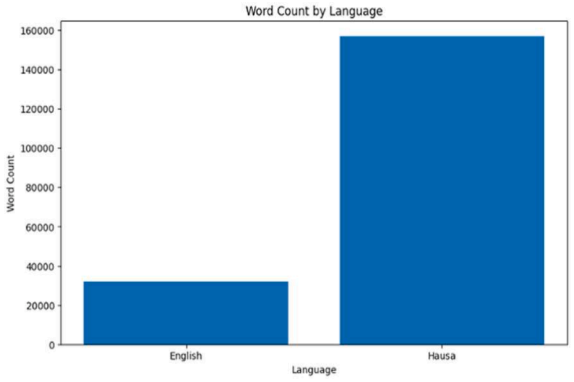

Fig. 2. Linguistic Composition of Code-Mixed Tweets.

and adaptability to the nuances of the Hausa language. These enhancements make the algorithm robust and suitable for sentiment analysis tasks in low-resource language. The performance of the proposed stemming algorithm was rigorously evaluated through a manual assessment conducted by a Hausa linguistic professional. It offered significant qualitative information about the way the algorithm works. Word-roots pair inspections were conducted to maintain the inter-rater reliability. Table 3 presents the sample of all the stem dictionary-word pairs and related stemming features.

Furthermore, the HausaStemmer algorithm was evaluated qualitatively using 88 simple words and 39 complex words, each matched with their expected stems collected from the Hausa linguistic experts. The algorithm achieved 86.36% accuracy for simple words, showing strong performance in straightforward cases. However, its accuracy for complex words was at 51.28%, this is due to the unique structure of certain words. For instance, compound words like *'yar-malam* (teacher's daughter) were overstemmed to *ya alam* ("without meaning"). Similarly, *iri-iri* (varieties) was reduced to *iri* (seed), leading to a loss of meaning. Another issue arose with proper nouns, such as Sabitu (a person's name), which was incorrectly stemmed to Sabit. Table 4 summarizes the proportions of correct and incorrect stems for both word categories.

Table 5 provides examples of original words, their expected stems, and the algorithm's output. These challenges can be mitigated by expanding the dictionary and incorporating logic for handling special cases, such as compound words and proper nouns. Despite its limitations, the HausaStemmer algorithm shows significant potential. With these enhancements, its performance and applicability in NLP tasks are expected to improve further.

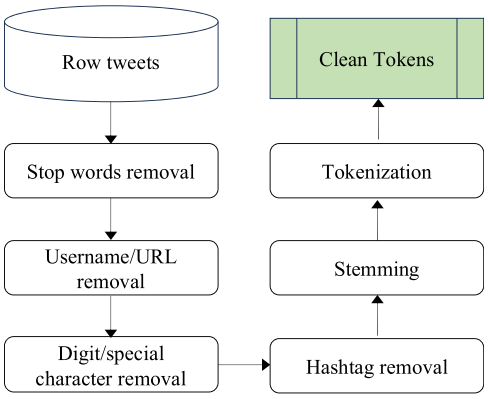

Fig. 3. Noise Preprocessing Procedure for the Tweets.

Table 3
stem sample of dictionary word-pairs.

Stemming Algorithm Features	Sample
Dictionary word-pairs	Murhu-murh,karya-kary, karnuka-karn,karnukan-karn, sark-sark,sarakai-sarak,manazarci-manazar,masunci, sunc, masunciya-sunc,bakano-kan,bakaniya-kan, baduddire-gudd,gidaje-gid, bishiya-bish, bishiyoyi-bishiy,falake-falk,rumfuna-runf, benenaye-benen,rafuna-raf,dakuna-dak, Dalibai-dalib,dalibi-dalib, daliba-dalib, yara-yar,yarinya-yar, mabuɗi-buɗ, mayanka-yank, mawaƙi-waƙ,banufiya-nufiy,bakatsine-katsin,Gurgu-gurg, Amincin-aminc,kawaici-kawaic,samartaka-samartak, …
Prefixes	ma, tsatt,yan, mai, ba, ta,abin, wuri-n
Suffixes	ai,in,r,ana,una, ao, ar, awa, aye, cce, che, chi, ci, fa, fen, u, ilu, ina, ir, ka, ke, ki, ku, mchi, mci, mtaka, n, nchi, nci, nda, ni, ake,nka,nku, nna, nni, nnu, nsa, nsu, nta, ntka, nu, obi, ochi, odi, ofi, ogi, oli, ori, oshi, oti, owa, oyi, rko, ru, sa, shi, suwa,je
Stopwords	a, amma, ba, ban, ce, cikin, da,don, ga, in, ina, ita, ji, ka, ko, kuma, lokacin, ma, mai, na, ne, ni, sai, shi, su, suka, sun, ta, tafi, take, tana, wani, wannan, wata, ya, yake, yana, yi, za

Table 4
proportion of correct and incorrect stem for each category.

Category	Total Words	Correct Stems	Incorrect Stems	Accuracy (%)
Simple Cases	88	76	12	86.36%
Complex Cases	39	20	19	51.28%

Table 5
HausaStemmer Evaluation Based on Simple and Complex Case.

Categories	Input word	Expected Stem	HausaStemmer Output	Remark
Simple cases	atamfa	Atamf	atam	overstemmed
	agogo	Agog	agogo	understemmed
	takadda	Takadd	takad	overstemmed
	dori	Dor	NaN	overstemmed
	gwanda	Gwand	gwan	overstemmed
Complex cases	mafauchi	Fauch	fauc	overstemmed
	yar-malam	yar-malam	ya lam	overstemmed
	iri-iri	Iri-iri	iri	overstemmed
	dumu-dumu	dumu-dumu	dum	overstemmed
	bara-gurbi	bara-gurb	gurb	overstemmed

Additionally, it was compared with an existing work of Musa et al. (2022) to assess its competitiveness. The proposed approach showed good word stemming as it minimizes under stemming as all words have to go through a dictionary test before going for the rule-based. The findings show the algorithm's resilience in changing language regimes coupled with the exploitation of word semantics in SA. The algorithm is presented in Table 6.

3.2.3. Tokenization

Following the application of the stemming algorithm, the text undergoes tokenization using the NLTK and WordPiece tokenizer. The resulting stemmed and tokenized text is subsequently fed into the embedding layer for feature extraction.

3.3. Embeddings

Embeddings are numerical representations of words or texts where words with similar meanings are closer together in the embedding space (Shanmugavadivel et al., 2022). They are heavily employed in NLP tasks as usual (Mutinda et al., 2023). They can still capture semantic relationships between words (Thara & Poornachandran, 2022). These embeddings are learned from large amounts of text data by which algorithms can better interpret and analyze language understanding. In our SA task, the choice of pre-trained embeddings is strategically aligned with the specific requirements of our goal. Embeddings excel in capturing significant linear substructures by effectively utilising global statistics from the word-word co-occurrence matrix (Toshevska et al., 2020). This characteristic is particularly advantageous for SA. To illustrate this, consider the example where "*King - Man + Woman*" results in a vector close to "Queen" (Toshevska et al., 2020). This goes to illustrate how embeddings can capture semantics relationships with deep learning models. The current models of word embeddings are applied to improve the results and representation of the low-resource language, especially developed for code-mixed text. This model enables the learning of contextual connections between words and facilitates comprehension of the training data. This work utilises the most popular embedding techniques including Word2vec (Mikolov et al., 2013), fastText (Bojanowski et al., 2017) and Glove (Pennington et al., 2014).

3.4. Deep learning models

Deep learning refers to the use of artificial neural networks comprises of more than one layer to perform feature learning tasks (Zhang et al., 2018). These tasks involve extraction and search for patterns from large collection data across various fields. Deep learning-based methods have emerged as a promising approach for SA (Araújo et al., 2020). This study utilises some of the popular types of deep neural networks (DNN) that have shown great potential in achieving high accuracy in SA. They are divided into two based on their deep Neutral Network DNN architectures (CNN, LSTM and GRU) and Attention-based mechanism architectures (Transformers).

3.4.1. Convolutional neural network (CNN)

(CNN), utilises the DNN architecture, which consists of convolutional and max pooling layers that filter and combine features. These layers are followed by fully connected layers for the class label classification (Pathak et al., 2020). CNN extracts and reduces the complexity of features during the training (Dang et al., 2020; Kim, 2014). The CNN model utilized in this study was 1D-CNN architecture. The model structure employs a dropout layer applied to the embedding vectors and the input tweets as in Eq. (1) before entering the convolutional layer.

$$E_t = W \cdot e_t \tag{1}$$

Where: E_t denotes the vector for the input token t and $W \in \mathbb{R}^{d \times d'}$ represents the trainable weight matrix obtained during the training

Table 6
Stemming algorithm.

Table 6 Stemming Algorithm

Algorithm 1: Comprehensive Hausa Stemming

Input:
- InputText (the raw text to be stemmed)
- Dictionary (a dictionary of word-root pairs)
- StopwordsList (a list of stopwords)
- PrefixesList (a list of prefixes)
- SuffixesList (a list of suffixes)

Output:
- SuffixesList (a list of suffixes)

Begin

Remove punctuation.
Tokenize InputText into words and store them in a list called WordsList.
Initialize an empty list called StemmedWordsList to store the stemmed words.
    For each Word in WordsList ***do*:**
        If *Word is in StopwordsList* **then:**
            Skip it and continue to the next Word.
        Else:
            Check if the Word is in the Dictionary.
            If *Word is in Dictionary,* **then**:
                Append the corresponding root to StemmedWordsList.
            Else:
                -Proceed to the next rule.
                -Check for prefixes in PrefixesList and remove them if found.
                -Check for suffixes in SuffixesList and remove them if found.
                -Append the modified Word to StemmedWordsList.
Join the words in StemmedWordsList to form StemmedText.
Return StemmedText;

process. It transforms embedding vectors to higher dimensional space. The dropout layer is intended to prevent overfitting, all relevant embedding vectors will be randomly turned off at any given time to mitigate overfitting. Subsequently, a convolutional layer works on the output of the dropout layer employing convolutional operations with kernel matrices and bias vectors. The convolutional layer extracts various sentiment-related features from the input as in Eq (2):

$$c_{ij}^{l} = f\left(\sum_{k\in K} c_{i+k,j+k}^{l-1} \cdot W_k^l + b_j^l\right) \tag{2}$$

Where: c_{ij}^{l} Stands for the value of the output of the feature map at location (i,j) and layer depth level of l. f Refers to the activation function applied to introduce non-linearity that enables the network to learn complex pattern. The results of the operation is the summation over all the element k in the convolutional windows K. $c_{i+k,j+k}^{l-1}$ it denotes a convolution operation between the input tensor and the kernel tensor. The b_j^l Denotes biases of the terms. Subsequently, in the convolution layer there is a max pooling layer with pooling windows to perform, aggregate information and reduce dimensionality. This layer computes its output based on the maximum value within each pooling window. The max-pooling layer further refines the extracted features from the convolutional layer as in Eq. (3):

$$c_{ij}^{l} = f\left(\underset{m\in M}{max} c_m^{l-1} + b_j^l\right) \tag{3}$$

where: c_{ij}^{l}: stands for the output of the max-pooling layer and f represents and is applied to the pooled values to introduce non-linearity, refining the extracted features. $max_{m\in M}$ shows the maximum operation on all elements. c_m^{l-1}: Represents the input to the max-pooling layer usually, the output of the previous layer. The last layer connected with the max-pooling layer has been included to make the extracted features more refined. This layer adjacent layer interfaces between convolutional and output. It forms the max-pooling layer and converts low dimensions feature vectors to a form that is ideal for sentiment classification. Lastly, SoftMax function is used to the output layer to determine the sentiment of the input sentence of each class label. The architecture of the CNN model structure is depicted in Fig. 4(a).

3.4.2. Long short-term memory (LSTM)

LSTM is an RNN which helps to learn and analyze sequential data by modelling long term dependencies (Demotte et al., 2020). It consists of a chain of repeating modules, where the repeating module in a standard RNN usually has a simple structure. However, in the case of LSTM, the repeating module has more complex as the layers work in a different fashion (Zhang et al., 2018). Similar to CNN the structure of LSTM model follows the same pattern. has been introduced in our model allowing for a dropout for refining the vectors derived from the embedding layers, introducing a dropout to refine the vectors generated by the embedding layers. Subsequently, these vectors are passed into the LSTM layer, which produces the hidden states during the training computations. Eq. (4) gives the computation of the input gate activation vector in the LSTM model. This gate controls the flow of information from the input vector and previous hidden states to the current cell state at a time step t.

$$i_t = \sigma(\mathrm{x}_t \cdot \mathrm{W}_{ix} + \mathrm{h}_{t-1} \cdot \mathrm{W}_{ih} + \mathrm{c}_{t-1} \cdot \mathrm{W}_{ic} + \mathrm{b}_i) \tag{4}$$

In the LSTM model, the forget gate determines which parts of the previous cell state c_{t-1} should be retained or discarded at the current time step t. The forget gate activation vector f_t is computed as in Eq. (5).

$$f_t = \sigma(\mathrm{x}_t \cdot \mathrm{W}_{fx} + \mathrm{h}_{t-1} \cdot \mathrm{W}_{fh} + \mathrm{c}_{t-1} \cdot \mathrm{W}_{fc} + \mathrm{b}_f) \tag{5}$$

Additionally, The cell update involves integrating new information from the input gate, forget gate and the cell state candidate. It retains relevant information over time. The updated cell state c_t at time step t is computed as in Eq. (6).

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{6}$$

The output gate determines how much of the updated cell state c_t should be passed as hidden h_t state at the current time t step. It

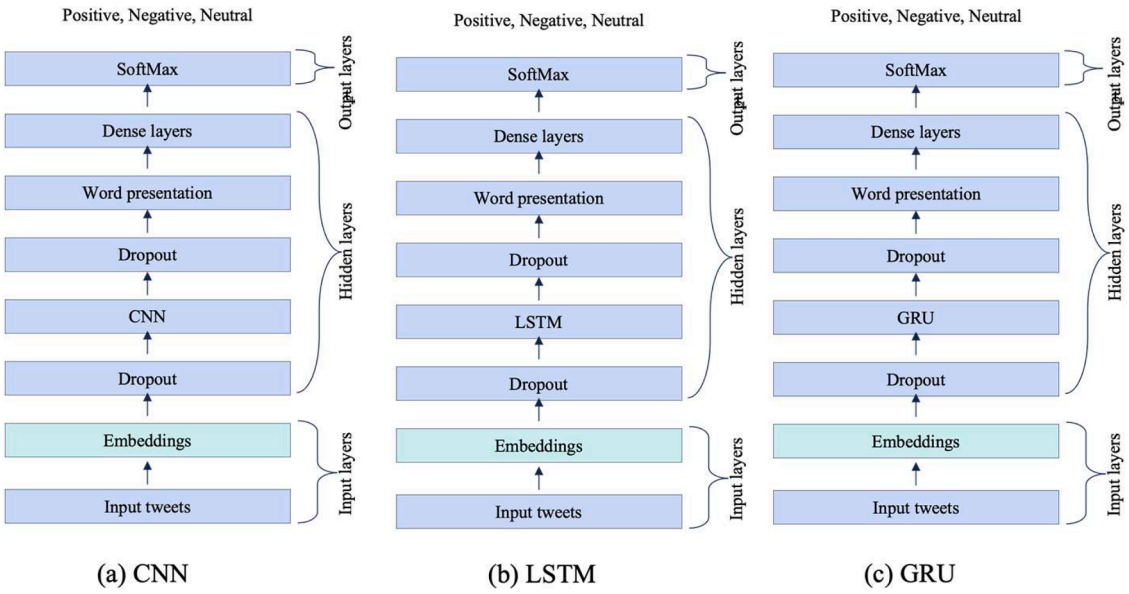

Fig. 4. The model structure architecture utilised.

explained the detail computation for output gate activation vector o_t in the LSTM model, presented in Eq. (7)

$$o_t = \sigma\left(x_t \cdot W_{ox} + h_{t-1} \cdot W_{oh} + c_t \cdot W_{oc} + b_o\right) \tag{7}$$

The hidden state h_t at the time t step is computed by applying the hyperbolic tangent (tanh) to the updated cell state c_t, scaled by the output gate activation as shown in Eq. (8). Fig. 4(b) shows the graphical structure of the LSTM model

$$h_t = o_t \cdot \tanh(c_t) \tag{8}$$

3.4.3. Gated recurrent unit (GRU)

GRU is also a RNN that solely takes current time information, ignoring any data carried in the spatial and temporal sequence data structure (Zouzou & Azami, 2021). The GRU model also utilises gating mechanisms like LSTM but with a simpler architecture. (Eq. (9)) compute the update gate z_t in a GRU model.

$$z_t = \sigma(W_z \cdot [X_t; h_{t-1}] + b_z) \tag{9}$$

Where: z_t stands for the update gate activation vector at time t. It computes the update gate activation via applying a sigmoid function to the dot product of $[X_t; h_{t-1}]$ and the weight matrix W_z as well as applying a sigmoid function on σ with bias. The graphical structure of the GRU model is illustrated in Fig. 4(c). We introduce another dropout layer to the output of the CNN, LSTM and GRU models layer, followed by the additional dense layer on top of the dropout layer to generate the sentence representation as in Eq. (10).

$$\boldsymbol{s} = \text{relu}(\boldsymbol{h}_t) \tag{10}$$

The sentence representation is then passed to a softMax classification layer for classifying the sentence as positive, or neutral, or negative. Eq. (11) fosters the computation of the probabilities of the different classes employable from the model. Thus, the SoftMax function guarantees that all chance estimations are summed up to one which makes it useful in the classification problem when a model has to predict probability of a class.

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{11}$$

Where $x_i$represents the raw score or logit for class i, and the denominator is the sum of the exponentiated scores over all classes. Fig. 4 illustrates the models' architectures.

3.4.4. Multilingual BERT (mBERT)

mBERT is a multilingual adaptation of the BERT transformer (Muhammad et al., 2022; Pires, Schlinger & Garrette, 2019), which undergoes pre-training on a diverse dataset comprising 104 languages (Xu, Durme & Murray, 2021). The model was implemented following the BERT base version with 12 transformer blocks, 768 hidden dimensions, 12 attention heads, and a maximum sequence length of 512 tokens (Wu & Dredze, 2020).

3.4.5. Robustly optimized BERT (RoBERTa)

RoBERTa is also a pre-trained model with the same architecture as BERT but is specifically crafted to improve upon BERT's performance (Shanmugavadivel et al., 2022) (Liu et al., 2019). Roberta used byte-level Byte-Pair Encoding, in addition to using a byte-level BPE vocabulary but with a larger subword set consisting of 50,000 (Tan et al., 2022). However, the primary objective of the base layers in RoBERTa is to produce a meaningful word embedding serving as the feature representation. This facilitates the subsequent advanced layers in effortlessly extracting the necessary information (Sirisha & Chandana, 2022).

3.4.5.1. Cross-Lingual language model with RoBERTa (XLM-R). XLM-R is a masked language model based on transformers (Conneau et al., 2020). It enhances the initial XLM-100 model, designed to support multiple languages, including Hausa (Kumar & Albuquerque, 2021; Muhammad et al., 2022). It attains cutting-edge performance on various cross-lingual benchmarks and is a viable alternative for low-resource natural language processing (NLP) (Bansal et al., 2022; Kumar & Albuquerque, 2021).

3.4.6. African language version of BERT (AfriBERTa)

AfriBERTa is a transformer-based multilingual language model that underwent training on 11 low-resource African languages including Hausa (Ogueji et al., 2021). Even with its more modest parameter size of 110 million, it achieves performance comparable to XLM-R on datasets for African languages (Alabi et al., 2022). Transformer models have an exceptional ability to capture contextual relationships and long-range dependencies, which are integral in understanding sentiment nuances in natural language.

The choice of the above models was based on the course of the model's multilingualism (Conneau et al., 2020). The selected models were chosen because they have been developed for the NLP challenge

and have been shown to perform well in multilingual settings (Alabi et al., 2022). They are also trained in different low-resource languages. The architectures of BERT models are similar and constructed based on the transformer architecture (Vaswani et al., 2017). Thus, Fig. 5 illustrates the BERT transformer architecture employed in this study for text classification.

However, the Transformer architecture is a DL model that leverages a self-attention mechanism to capture the contextual relationship between words in a sequence. It forms the foundation of the models such as mBERT, AfriBERT, RoBERTa, XLM-R and others. The transformer architecture comprises distinct components, such as an encoder for text input comprehension. BERT transformer model adopts a singular encoder design with 12 or 24 layers. Each variant introduces a large feedforward network and multiple attention heads. The input processing consists of the [CLS] token and a word sequence that is followed by [SEP] as a separator of the following sequences (Bilal & Almazroi, 2023). Both layers implement self-attention and feedforward networks to all tokens present in low dimensional space. This is because the output is a 768-dimensional hidden vector after passing through the layers. Last, the SoftMax layer supports detection for the predicted probability for classification activities (Devlin et al., 2019).

The process of transforming input data for use in encoder models of BERT tokenisation involves using a transformer tokenizer. Similarly, the tokenised inputs were then segmented. During the passage through encoder blocks, each input sequence is represented by a matrix of dimensions, as shown in Eq. (12) with positional encoding providing positional information. The encoder comprises multiple blocks (N blocks) that collaboratively encode the input representations into the output.

$$\left(\text{Input}_{\text{Length}} \times \text{Embedding}_{\text{Dim}}\right) \tag{12}$$

The encoder's architecture is fundamentally based on multi-head attention, which performs multiple attention calculations using different weight matrices and combines the results obtained. Each attention calculation, or head, is denoted with a subscript i corresponding to its weight matrices. The outcomes of these heads are concatenated, forming a matrix with dimensions as in Eq. (13)

$$\text{Input}_{\text{Length}} \times (h \times d_v) \tag{13}$$

This is followed by a linear transformation using a weight matrix W_0 of dim $(h \times d_v) \times$ Embedding Dimension, yielding a final output matrix of dimensions of (Eq. (12)). This is expressed as in Eq. (14):

$$\text{MultipleAtensionHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W_0 \tag{14}$$

Where $\text{head}_i = \text{Attention}\left(QW_{Q_i}, KW_{K_i}, VW_{V_i}\right)$ and Q, K, and V represent different input matrices, and each head is defined by unique projection matrices W_{K_i}, W_{Q_i}, and W_{V_i} in Eq. (15a,b,c). These matrices have dimensions (Embedding Dimension $\times d_k$) for W_{K_i} and W_{Q_i}, then (Embedding Dimension $\times d_v$) for W_{V_i}. The input matrix X is projected through these weight matrices to estimate the heads, resulting in (Eq. (15)).

$$XW_{K_i} = K_i \quad \text{with a dimension} \quad \left(\text{Input}_{Length} \times d_k\right) \tag{15a}$$

$$XW_{Q_i} = Q_i \quad \text{with a dimension} \quad \left(\text{Input}_{Length} \times d_k\right) \tag{15b}$$

$$XW_{V_i} = V_i \quad \text{with a dimension} \quad \left(\text{Input}_{Length} \times d_v\right) \tag{15c}$$

The scaled dot-product attention is computed as in (Eq. (16))

$$\text{Attention}\ (Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{16}$$

This formula uses the dot product of K_i and Q_i to assess token similarities. For token projections m_i and n_j via K_i and Q_i, the dot product is given in (Eq(17)):

$$m_{ij} = \cos\left(m_i, n_j\right).\frac{\| m_i\|_2 .\ \| n_j\|_2}{\sqrt{d_k}} \tag{17}$$

Scaling by $\sqrt{d_k}$ and applying the softmax function row-wise ensures that the row values sum to 1. The final attention value is obtained by multiplying this result with V_i, forming the head (Prottasha et al., 2022).

3.5. Feature extraction

In the feature extraction stage, we convert textual data into numerical form and integrate word embedding vectors. The framework begins with the representation of input tweets and word embeddings. Specifically, a tweet composed of N words, denoted as $T = \{w_1, w_2, w_3 \ldots w_n,\}$ is processed, and each word w_i is transformed into a numerical vector, represented as in (Eq. (18)). Simultaneously, the designated class y is converted into its vector, denoted in (Eq. (19)).

$$e_i = E(w_i) \tag{18}$$

$$v_a = E(y) \tag{19}$$

The context embedding vectors (Eq. (18)) are combined with pretrained embeddings from Glove; FastText and Word2Vec to compute word similarity. Here, $E \in \mathbb{R}^{v \times d}$ refers to the embeddings, with d representing the dimension of the word embedding vectors. The vocabulary size is denoted as v, and N represents the number of words in

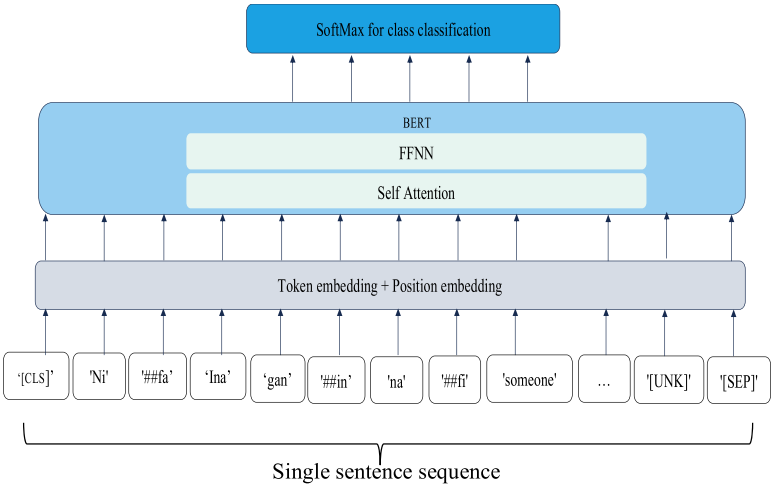

Fig. 5. BERT architecture for text classification (Devlin et al., 2019; Rajeshwari & Kallimani, 2022).

the tweet. Additionally, each word is represented as a dense vector capturing semantic relationships.

In the attention mechanism architecture, we utilise a transformer Tokenizer, leveraging the word piece tokenizer to handle out-of-vocabulary (OOV) terms by decomposing them into sub-words, thereby enhancing our text-processing capabilities in the features extraction step. We begin with fine-tuning the mBERT model for sentiment classification. The tokenised input text sequences were augmented with a special classification token [CLS] at the beginning and a separation token [SEP] at the end of each token sequence. We obtained token embeddings for each sub-word using the contextual embedding matrix, which was combined with segment embeddings to distinguish between tokens from the first and second sentences. Position embeddings were utilised to represent the position of each token in the input sequence by the transformer attention-head encoder layer. The mBERT input representation, consisting of the token embeddings, segment embeddings, and position embeddings, was fed into a transformer encoder and feedforward neural network (FFNN) for feature representation. SoftMax layer saved as an aggregation for class label classification.

Similarly, for Roberta, XLM-R and AfriBERTa models, we employed the same consistent fine-tuning approach as mBERT. All the models require input sequences of fixed lengths for efficient processing; a normal practice in handling such cases is known as padding. The [PAD] token is typically used to fill the empty spaces in shorter sequences, making them equivalent in length to the longest sequence in the input tokens, until the desired fixed length is reached. Additionally, any unidentified token is denoted by [UNK].

3.6. Hyperparameter tuning

To ascertain optimal, efficient parameter tuning, a series of training was conducted involving different parameter combinations using mBERT and was validated with f1-score metric.

Table 7 presents a detailed result obtained from hyperparameter tuning considered during the model optimisation process. In each row of the table shows the model performance with different learning rates and batch size combinations. Each of the selected hyperparameters was intentionally set to some value in order to investigate the effects of the changes on the results of the selected sentiment analysis task. The goal was to find out which parameters allow achieving reasonable levels of both model convergence and computational efficiency at the same time. Analysing the learning rate and f1-score gives more elaborate patterns, especially in the F1 scores obtained from the different learning rates and the different batch sizes. To determine the impact of these hyperparameter choices, we cautiously performed the experiments. Nevertheless, in the present investigation, we found out that, 1e-5 of the learning rate paired with a batch size of 32 gave the best F1 scores. These values were then used in the other experiments with the remaining models as well.

We determined a maximum sequence length parameter of 120 by analysing the distribution of the tweets in our datasets. To reduce the need for unnecessary padding and the need for computational resources. The number was carefully selected to capture the main ideas of the tweets. For an illustration of the maximum sequence length of tweets. Fig. 6 provides a visual representation of the maximum token length observed in tweets. Additionally, most prior work optimized models used a fixed learning rate. However, we employed a different learning rate to adaptively adjust the rate, which yielded improved convergence in our experiments until choosing the learning rate of 1e-5 based on the result which was trained on various parameter combinations as shown in Table 7 during the training phase.

Table 7
Hyperparameter combination and performance.

Sn	Learning rate	Batch size		
		16	32	64
1	5e-5	0.17	0.17	0.17
2	3e-5	0.51	0.17	0.41
3	2e-5	0.60	0.73	0.66
4	1e-5	0.69	0.73	0.72

To select the dropout value we performed a systematic dropout tuning from 0.1, to 0.5 to see the effectiveness of each of these. This cross-fold training was intended to determine the dropout rate of given network, which serve to minimize overfitting to improve model generalisation. In this process, we find that the dropout value of 0.3 was found to give better performance accuracy than the higher dropout rates. To achieve such balance, we make a conscious effort to select the best practices as our chosen value with evidence of its effectiveness on model performance. For optimisation, the commonly used Adam optimiser was used since it is one of the most popular used, appraised for its capabilities, fast and its effective training of deep learning models. The same parameter setting was used across other models as well, as shown in Table 8 where the selected hyperparameters are displayed.

4. Experiments

To evaluate the effectiveness of our proposed framework, we performed a series of experiments. This section describes the framework, experimental setup, ablation study and evaluation methods used to assess the model's performance. The structured presentation ensures clarity in methodology and criteria, highlighting the robustness of our approach. Fig. 7 illustrates the proposed method and efficient hyperparameter tuning framework of deep learning for SA in code-switch text. However, the conventional hyperparameter tuning methods rely on exhaustive grid or random search. This framework employs a systematic combination of dropout regularisation, learning rate tuning, and batch size selection tailored specifically for the complexities of code-switched data. Through leveraging insights from transformer-based architectures and integrating contextual embeddings. The framework minimizes computational overhead while optimizing model accuracy performance. These design considerations uniquely address the challenges posed by low-resource language settings, as demonstrated in our comparative analysis.

The dataset initially undergoes a pre-processing step in our proposed framework through data cleaning and language identification systems (LIDs). The feature extraction step utilises word embedding from FastText, Glove, and Word2vec in combination with the LSTM, CNN, and GRU networks to capture word similarity and relationship. Subsequently, the transformer models are used to capture contextual relationships from the training data. During the model training process, the dataset is split into training, validation and test data sets to select the right model parameters, especially for identifying code-mixed cross-language tweets and language structure from context. Before proceeding to the ultimate output, a statistical experiment is undertaken by the SoftMax layer to evaluate notable variances among various models' class probability for efficient parameter tuning. The framework decision-making process is based on the average of predicted probabilities. The aim of introducing the framework is to enhance overall accuracy performance. Further elaboration on the experimental evaluation is provided in the experiment section.

4.1. Experiment setup

In the experiment, we employed Python language and the TensorFlow 1.13.1 library. The model used torch of the second version 2.0.1+cu118 library in multilabel sentiment classification. The implementation was done on Google Collaboratory, and the GPU hardware accelerator was used. The usage of tweets was analyzed with the help of NumPy, Pandas, sci-kit learn, transformers, language detect API, and seaborn libraries. The results obtained for the proposed efficient hypermeter tuning framework also support the effectiveness of the

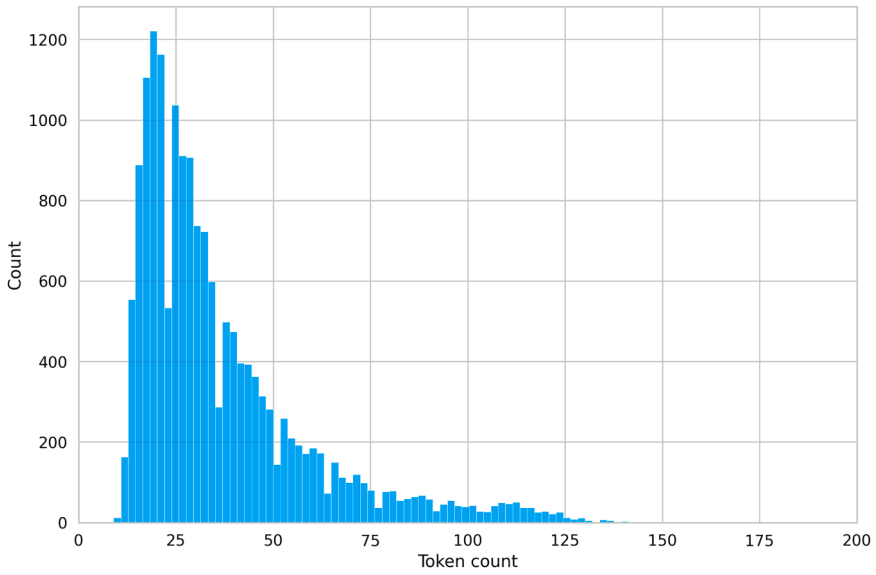

Fig. 6. Tweets token maximum sequence length.

Table 8
Hyperparameters used in this study.

Sn	Parameter	Value
1	Batch size	32
2	Max Sequence length	120
3	Learning rate	1e-5
4	Dropout	0.3
5	Optimizer	Adam

particular emphasis is placed on metrics associated with NLP. These metrics include Precision, Recall, F1-Score and Accuracy. Each experiment involves the use of the training, validating, and testing datasets of the proposed sentiment analysis techniques performance being determined. Metrics in general derive from the confusion matrix. It is a construct that comprises critical parameters like true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) values.

4.2.1. Accuracy

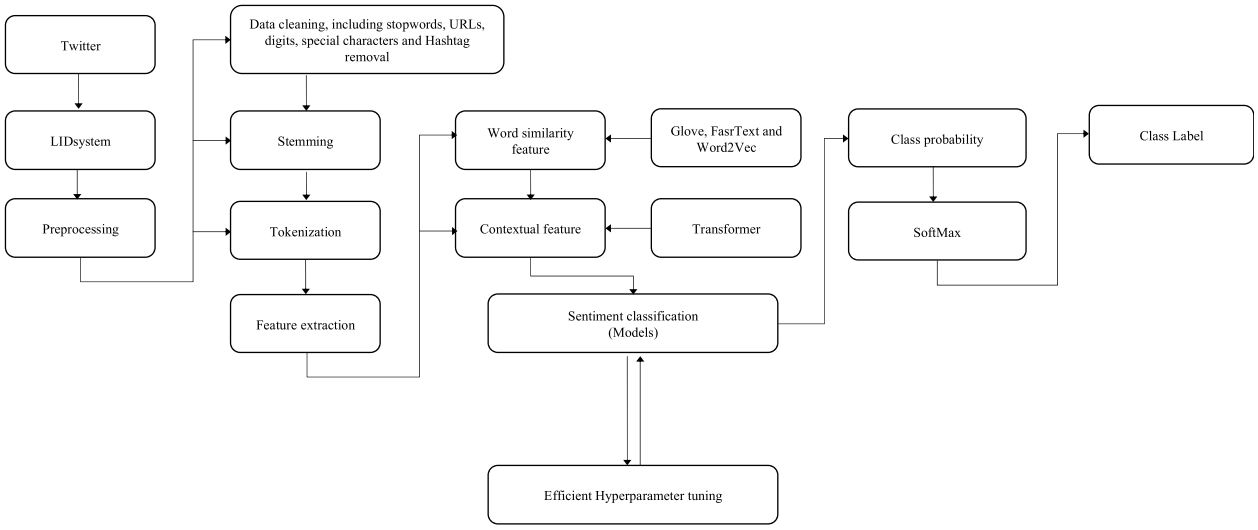

Fig. 7. Proposed efficient hyperparameter tuning framework.

employed approach. The code can be accessible at a Github repository.[2]

4.2. Evaluation metric

Evaluation metrics constitute a subset of performance assessment tools utilised to measure the efficiency of machine learning or statistical models' performance. However, within the scope of this study,

Accuracy measures the effectiveness of pre-classified with the actual classification of the tweets. The idea of accuracy testing is simply to show the viability of the models in estimating data with higher accuracy than other models. Accuracy can sometimes be misleading in cases of class imbalance, as it may overemphasise the performance of majority classes. The accuracy formula is given as in (Eq. (20)):

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{20}$$

[2] https://github.com/yusuf-003/HausaCodemixed

4.2.2. Recall

Recall counts the accurately predicted positive labels out of the total positive labels. Recall testing aims to appraise the models to appropriately recall correctly classified data. The recall formula is given as in (Eq. (21)):

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

4.2.3. Precision

Precision evaluates the proportion of predicted data or samples assigned to a specific class, ensuring that the predicted samples genuinely pertain to that class. A high precision indicates fewer false positives. This makes it particularly important for tasks where misclassifications as positive have significant consequences. The formula is given as in (Eq. (22)):

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

4.2.4. F1-score

F1-score shows the relationship between precision and recall and for this reason, it varies inversely with aims to achieve the right balance between the two. Furthermore, the F-score serves as a harmonic mean (Ghafoor et al., 2021). It efficiently reduces the problem of the trade-off between precision and recall in the assessment of the performance of a model. The formula is given in (Eq. (23)):

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{23}$$

4.3. Statistical significant test of the model performance

To assess the statistical significance of the differences in model performance, The Friedman Test was deployed after the experiment. It is a non-parametric statistical test suitable for comparing the performance of multiple models across paired datasets. The test evaluates the null hypothesis that all models perform equally well. The significant differences was further applied by the Nemenyi Post-Hoc Test to identify which specific model pairs exhibit statistically significant performance differences. These statistical methods provide robust comparisons of models, particularly in scenarios where assumptions of normality cannot be guaranteed.

4.4. Design of ablation study

To evaluate the specific impact of each component of the proposed framework, an ablation study was conducted systematically by removing individual components. We examined the effects of the preprocessing steps and the embedding methods. The ablation experiments were conducted using the using CNN model, which was selected as the first baseline performance.

5. Results and discussion

Multiple experiments were conducted to assess the performance and effectiveness of our proposed efficient hypermeter framework across different deep-learning models. Specifically focusing on the code-switch dataset. The training epoch is set to a maximum of 20. Throughout the experiment, the dataset was randomly divided into three for training. Thus, train, validation, and test. 90% of the data was allocated for training, amounting to a finalised set of 15,164 tweets, while the validation set for 5% accounted for 842. Lastly, 5% was separated for testing, resulting in 843 tweets data. The importance of splitting the data into three parts is that it allows for a more robust evaluation of the model. The model was trained on the training data, and its performance was evaluated on the validation set. The hyperparameters of the model are then tuned based on the performance of the validation set. For all experiments, accuracy was measured together with the weighted averages of precision, recall and F1 score.

Table 9 presents the performance of the deep learning models on Hausa-English code-switch tweet data obtained from the experiment. It presents the evaluation metric on unseen data. The framework utilises Glove, Word2Vec and FastText embeddings. An experiment of the first three models, CNN, LSTM, and GRU, without code-mixed consideration was conducted. We employed evaluated metrics on the unseen dataset split for validation. We observed the models yield slightly lower results with an overall best of 0.42 accuracy by the CNN model. These performance results underscore the baseline performance in SA. While putting the code-switch consideration, the validation offers a greater precision of 0.64 by the GRU_Glove embeddings. However, the models with embedding come with higher computational costs. On the other hand, pre-trained transformers with attention-based mechanism models notably outperformed non-pretrained or classical deep-learning models. It is also important to note that the models, in general, show slightly better performance and give more meaningful information to be used while ensuring a higher rate of metric evaluation. In the f1-score, AfriBERTa emerged as the top performance model with an f1-score of 0.92. This showcases the viability of the attention-based mechanism of the transformer framework in language understanding for sentiment classification in codemixed text.

Similarly, Fig. 8 illustrates the training and validation loss curves among the twelve deep learning models evaluated for sentiment classification. It shows a unique pattern of performance. Notably, the CNN_fasttext model has the least validation loss, slightly behind LSTM_fasttext and even less than GRU_fasttext. From this observation, it can be inferred that the models may perhaps perform better on the intended task. From the graphs of validation loss, we can observe the decline across epochs suggesting that all the models effectively learnt from the training data. However, discrepancies in loss reduction rates and the magnitude of final loss values highlight variations in model efficiency. Models exhibiting smaller gaps between training and validation loss curves are less prone to overfitting, indicating better generalisation capabilities. Therefore, the code-switch factor affects the accuracy of sentiment analysis in low-resource sentiment analysis as indicated in the experiment with CNN, LSTM and GRU, when the code-switched factor was not considered. However, it is crucial to consider additional factors, such as the performance on unseen data of individual model's training and validation curve during the learning process.

Fig. 9 illustrates the graphical representations of the best

Table 9
The evaluation performance of the deep learning model on Hausa-English code-switched tweets data.

Model	Embeddings	Evaluation Metric			
		Precision	Recall	Accuracy	F1-score
CNN	–	0.43	0.42	**0.42**	0.34
LSTM	–	0.36	0.36	0.36	0.27
GRU	–	0.32	0.34	0.34	0.23
CNN	Glove	0.61	0.58	0.58	0.57
LSTM	Glove	0.62	0.60	0.60	0.60
GRU	Glove	**0.63**	0.62	0.62	0.62
CNN	FastText	0.60	0.57	0.57	0.56
LSTM	FastText	**0.62**	0.60	0.60	0.59
GRU	FastText	0.61	0.58	0.58	0.52
CNN	Word2Vec	0.61	0.58	0.58	0.58
LSTM	Word2Vec	0.63	0.60	0.61	0.61
GRU	Word2Vec	**0.63**	0.62	0.62	0.62
mBERT	Contextual	0.74	0.73	0.73	0.73
Robert	Contextual	0.76	0.75	0.75	0.75
XLM-R	Contextual	0.75	0.74	0.74	0.75
Afriberta*	Contextual	0.92	0.92	0.92	**0.92**

The **bold** indicates the best performance metric in a category and * indicates overall best metric.

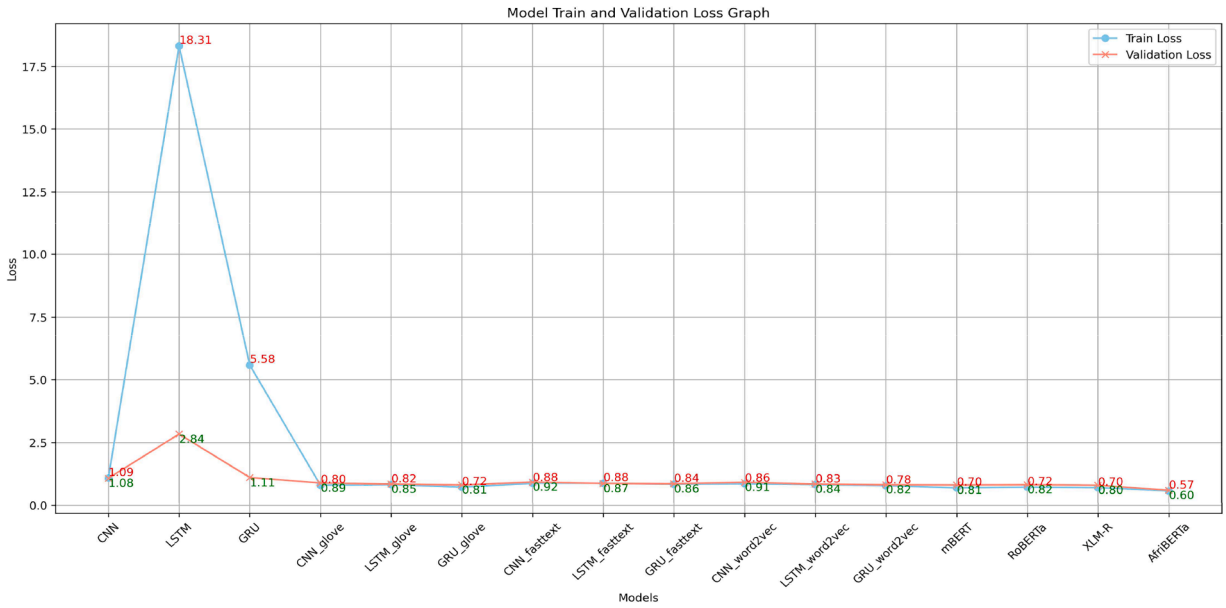

Fig. 8. Models train and validation loss curve.

performance train and validation accuracy curve of the models experiment, thus Fig. 9(a)CNN, (b) GRU_Glove, (c) LSTM_FastText, (d) GRU_Word2Vec and (e) AfriBERTa. However, both models' training and validation accuracy curves indicate a steady improvement in accuracy as training progresses. However, the model without embedding or code-mixed consideration Fig. 9(a) (CNN) reveals a wider train and accuracy curve. Suggesting overfitting during the feature learning led to a poor generalisation of the unseen data. Likewise, the models with embeddings gain an enhancement in learning Fig. 9(b)(c) and (d). This implies that both models were able to extract the features from the data set and were able to generalize on new samples. Nonetheless, there is a slight variation of the accuracy scores between the training and validation data sets especially in Fig. 9(b)(c)(d) at the 5th to the 10th epoch, and a consistent decrease of validation over time as in Fig. 9(b) and (d). This suggests that overfitting could be a problem with the case of the model being too intricate to fit the training data; that may affect its ability to generalize on more instances in the embedding than what is visible with the training instances except with the FastText embedding which shows better learning. As for the AfriBERTa model, the train and accuracy curve in Fig. 9(e) reflects much lesser overfitting, and this might have been made possibly by the huge amount of data used in its pretraining.

Similarly, a confusion matrix is utilised to evaluate the effectiveness of the model's misclassification, as depicted in Fig. 10. The class labels are assigned as 0 for negative, 1 for neutral and 2 for positive. Fig. 10(a) CNN without embedding or code-mixed consideration. It achieved 547 accurately predicted negative instances, 824 instances accurately predicted neutral class instances and only 31 correct instances of positive labels with a high rate of misclassification of each class label. However, CNN finds it very challenging to predict the actual value of the label class due to the language complexity and code-switched nature of the data. Furthermore, When the embeddings are incorporated in models such as Fig. 10(b) GRU_Glove. There is an observable increase in the correct label classification. The GRU_Glove correctly predicted 407 negative tweet instances, 704 neutral instances, and 660 positive instances in their correct label class. This indicates the effectiveness of embeddings in capturing the word similarities of the dataset.

However, Fig. 10(c) AfriBERTa model confusion matrix demonstrated an impressive correct class label classification. The model almost predicts all the label classes correctly, with only a few misclassification errors. It incorrectly classified only 4 instances of the correct negative class label as positive and 23 instances of tweets of the correctly negative class as neutral. It incorrectly classifies 17 actual neutral classes as negative and 11 actual neutral class labels as positive. The model also misclassified only 6 actual positive class labels as negative and only 7 actual positive class labels as neutral. Therefore, the AfriBERTa model, as a pre-trained transformer-based model, outperforms all the others. This is attributed to its contextual mechanism and to being pre-trained on large data in different languages and domains.

6. Error analysis

Error analysis was conducted to evaluate the model's strengths and weaknesses. We manually reviewed predictions from the test set categorising errors based on ambiguous linguistic challenges (code-switching) and dataset noise. Representative examples of success and failure cases are presented in Table 10 to provide insights into the model's behaviour. However, Using the gold set data consists of 2677 instances. The model incorrectly classified only 244 instances in a total of 2677. However, the error analysis revealed that the model performs well on tweets with short-word sentiment markers but struggles with code-mixed words with long phrases. Future work will explore integrating external resources, such as sentiment lexicons, to address these limitations.

7. Evaluation of the proposed framework

In the evaluation of the proposed efficient hyperparameter tuning framework in code-switching tasks. We Investigated its applicability to other low-resource language code-switch datasets. To ensure fair comparisons, all baseline models were re-trained and evaluated under identical experimental settings. We analysed the performance across various language pairs datasets, drawing comparisons with prior studies (Chakravarthi et al., 2020), (Raihan et al., 2023) and (Hassan Muhammad et al., 2022). The experiment involved applying the proposed efficient hyperparameter tuning, specifically, the transformer models, which exhibited the best performance f1-score in our previous experiments in different language datasets. Additionally, we focused on evaluating our framework based on three pre-trained models: mBERT, XMR-R, and AfriBERTta which have been trained on more low-resource languages.

Our framework aimed to optimise model performance by

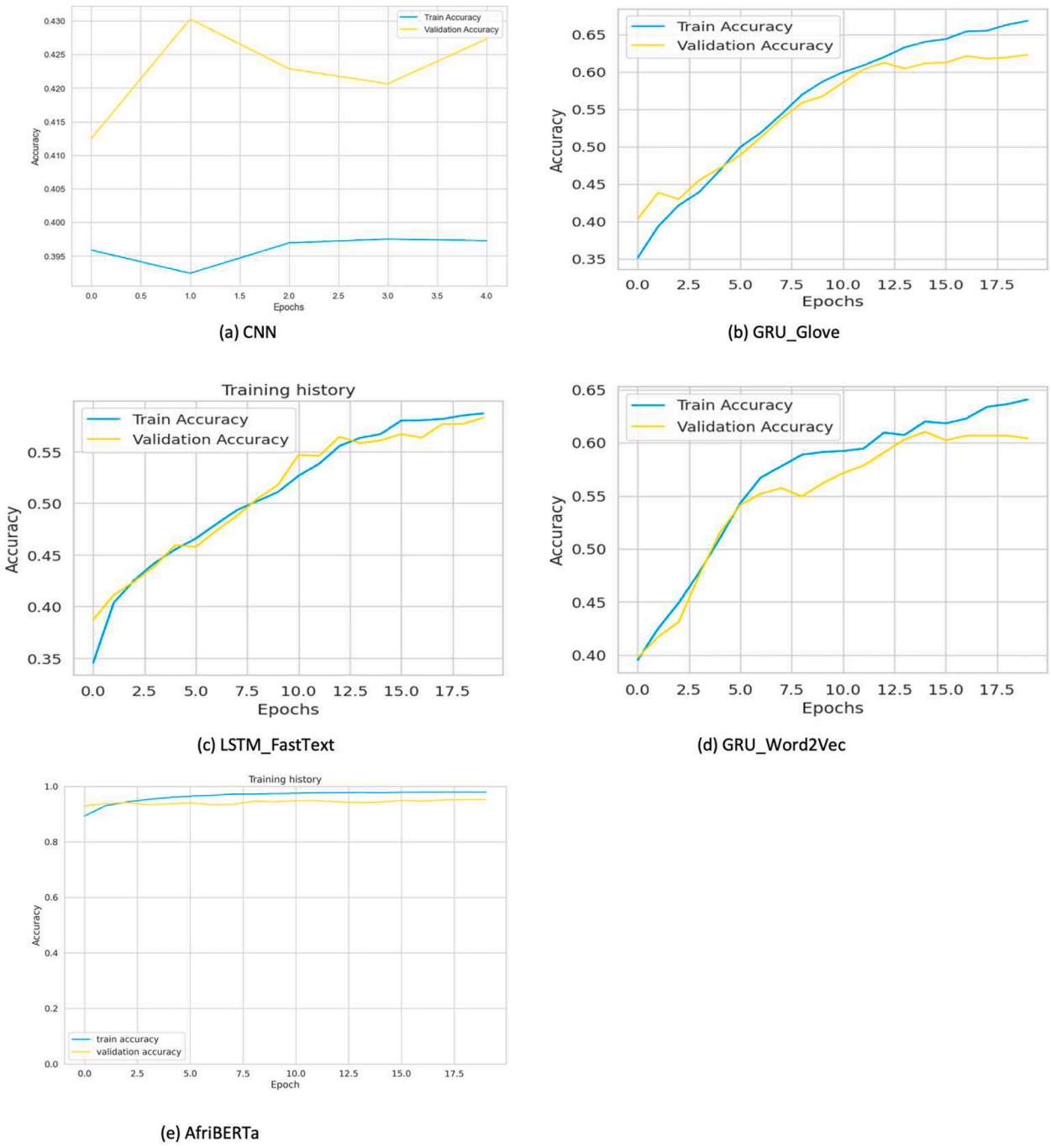

Fig. 9. Models train and validation accuracy curve.

systematically adjusting hyperparameters. The evaluation, therefore, used performance metrics that gave a comparison of our framework with previous methodologies to include precision, recall, accuracy and the F1-score. For instance, in the Malayalam-English dataset (Chakravarthi et al., 2020), the proposed framework using BERT reported an F1-score of 0.82 in a similar study, the F1-score observed was 0.75. Similarly, in the Bangla-English-Hindi dataset (Raihan et al., 2023), The designed framework using mBERT and XLM-R achieved an F1-score of 0.83 and 0.96, respectively which were higher than this related study. In addition, in the Igbo-English dataset (Hassan Muhammad et al., 2022), we used here, with the help of AfriBERTa our framework showed a significantly higher F1-score of 0.94 compared to the f1-score of 0.81 in a related study. Therefore, these results highlight the importance of performing systematic hyperparameter tuning in deep learning algorithms that handle different grammar language sets. Table 11 presents the evaluation of the related study data using a weighted average.

Additionally, statistical significance testing was conducted to validate the observed performance improvements of the proposed framework. The Friedman test was carried out across the F1-scores values obtained for each model. It indicated statistical significant differences among the models (χ^2=11.862, $p = 0.018$). Pairwise comparisons using the Nemenyi post-hoc test revealed that AfriBERTa significantly outperformed models such as CNN (No Embeddings), LSTM (No Embeddings), and GRU (No Embeddings), as shown in Table 12. However, differences between certain embedding-based models were not statistically significant, as indicated by p-values above 0.05.

Moreover, the statistical significance analysis validates the superiority of AfriBERTa in low-resource sentiment analysis tasks. Its

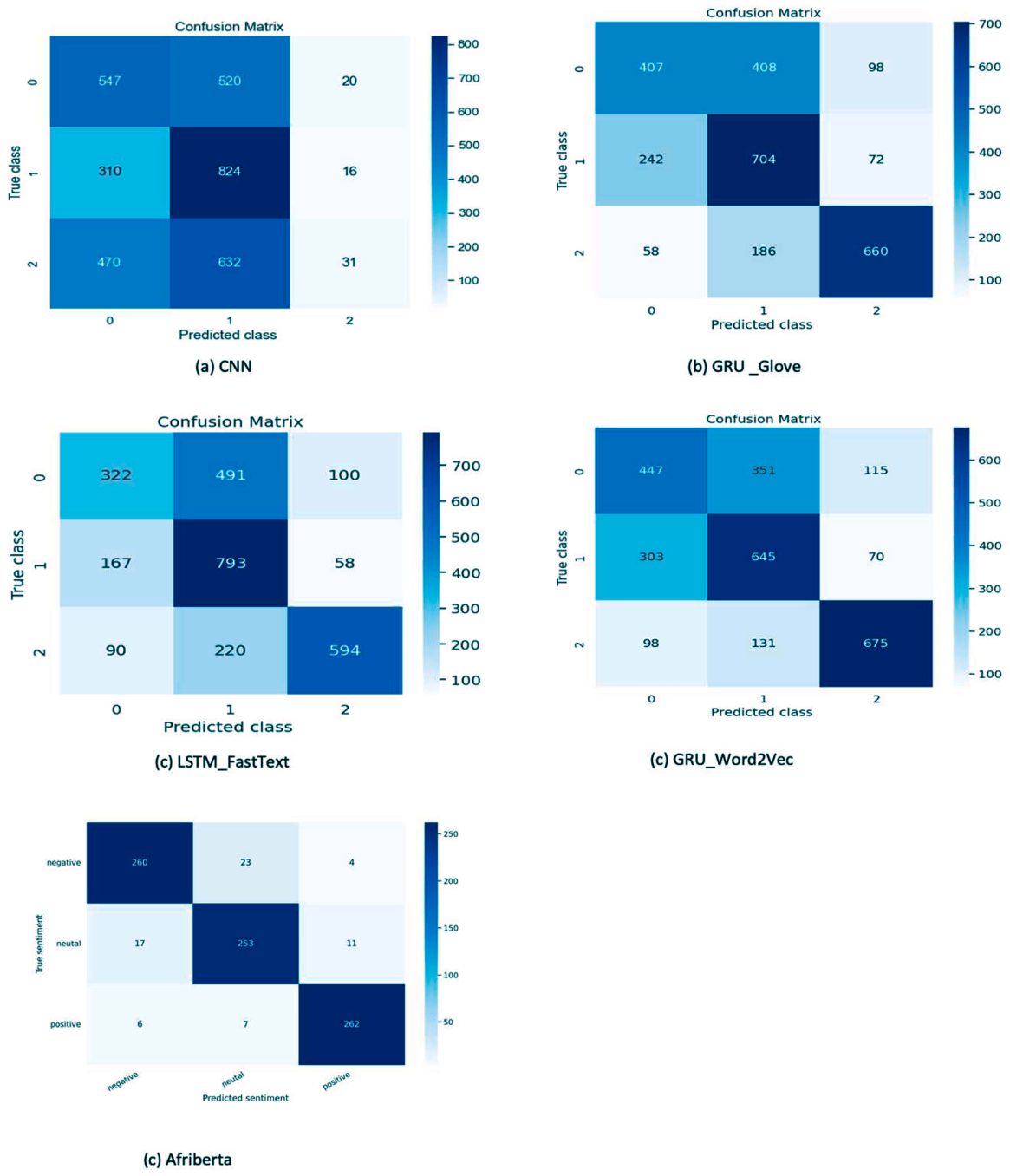

Fig. 10. Models' confusion matrix for misclassification.

performance, driven by contextual embeddings, was consistently higher than that of non-embedding models and significantly better than most embedding-based models. This aligns with the hypothesis that transformer-based architectures excel in handling code-switched datasets.

The ablation study results in Table 13 indicate the importance of each component in the proposed framework. The embeddings, particularly Glove had the most significant impact on CNN model performance, with a decrease of −0.23 in the F1-score when removed. In the preprocessing steps such as stemming also contributed to performance improvements, though to a lesser extent of −0.19 in the F1-score. However, stemming may lead to semantic loss, building a model with a transformer model overcomes this limitation due to it contextual embedding and long dependencies handling.

The combination of word similarity and contextual features enhances the model's ability to capture nuanced relationships between words. For example, similarity measures help identify semantically related terms, while contextual embeddings ensure that these relationships are considered in the broader context of the sentence as shown in Table 10 ablation study. It confirms the significance of combining these features, with a performance drop. Overall, our proposed framework demonstrated superior F1-score results than existing methodologies, which proves the effectiveness of using the proposed approach to improve the performance of models for code-switching scenarios. Similarly, the proposed framework opens new possibilities for real-world application in underrepresented linguistic communities as it will foster inclusivity and technological accessibility to users.

Table 10
Error analysis of the model predicted success and failure cases.

Example of tweet	True Label	Predicted Label	Remark on the model Behavior
@user Ranka shidade **nd then MR**. G.M.K **we are waiting** yar gata **song** 🙏🙏🙏🙏🙏🙏	Positive	Positive	Correctly identified sentiment with short words code-mixed.
@user @user Wai **wait**. Maganar **case** din @user da @user bai tabbata ba kuwa **imagine how she hug** @user **at the first pic** 🤔	Neutral	Neutral	Correctly identified sentiment with short words code-mixed.
@user Wannan faa Africa ba makaryaci kamarsa 😂 😂 😂 Koh kun san labarin **Club** din kwallon da in aka sa mutum se ya tsufa baAyi mai **passing** ba???	Negative	Negative	Correctly identified sentiment with short words code-mixed.
@user Aunty sadau Just tank god say na Lagos if na arewa u they drive and mistakenly enter go-slow in minutes ***Zaki ji bakia nan yaran malam sun sace ki***. The next thing u go hear is buhari ***yayi alawadai da abin da ya faru finish***.....god save our country 👏	Positive	Negative	Struggles with code-switched with long phrases.
@user Yakamata yasan cewasu makiyayan yan'kasane kuma daga cikin dokar kasar akwai (**right to freedom of movement**) saide indan sunyi masa abunda baidace ba ajamusu kunne 👂 👂	Neutral	Negative	Struggles with code-switched with long phrases.
@user Gaskiya Buhari ya yaki talauci dongashi kayi kyau sosai kuma duk kayi kiba dakuma kuzari gwamnati tayi dadi. Zanzo ace Muga **pictures** dinka na 2012,13,14,15 a **comparing** dana yanxu. Buhari yana aiki wadansu suna jin dadi. **Long live with the president, long live Nig** 👍 👍 👍 💚 💚 💚	Negative	Positive	Struggles with code-switched with long phrases.

*__red__ indicate Hausa tweet with English mixed. While ***blue*** shows English tweet with Hausa mixed

*__red__ indicate Hausa tweet with English mixed. While ***blue*** shows English tweet with Hausa mixed.

Table 11
evaluation comparison of the efficient hyperparameter tuning framework for low-resource language code-switching datasets comparison.

Related work	Language (Dataset)	Model (Used)	Evaluation Metric(weighted avg)			
			Precision	Recall	Accuracy	F1-score
(Chakravarthi et al., 2020)	Malayalam-English	BERT	0.73	0.73	–	0.75
Ours		BERT	0.83	0.84	0.84	**0.82**
(Raihan et al., 2023)	Bangla-English-Hindi	mBERT	–	–	–	0.74
		XLM-R	–	–	–	0.77
Ours		mBERT	0.84	0.82	0.82	0.83
		XLM-R	0.95	0.96	0.96	**0.96**
(Hassan Muhammad et al., 2022)	Igbo-English	AfriBerta	–	–	–	0.81
Ours		AfriBerta	0.94	0.94	0.94	**0.94**

(-) Indicates that the metric was not reported in the respective studies.

Table 12
Nemenyi results.

Model Pair	F1-Score Diffrence	p-value	Significant?
AfriBERTa Vs CNN	0.58	<0.0001	Yes
AfriBERTa Vs LSTM	0.66	<0.0001	Yes
AfriBERTa vs. GRU	0.69	<0.0001	Yes
AfriBERTa vs. RoBERTa	0.17	0.015	Yes

Table 13
The framework ablation study.

Framework Component	Precision	Recall	Accuracy	F1-Score	Difference (F1)
Full Framework	0.61	0.57	0.58	0.57	–
Without Stemming	0.39	0.40	0.41	0.38	−0.19
Without Embedding	0.42	0.42	0.42	0.34	−0.23

7.1. Comparison of the state-of-the-performance

To compare the efficiency of our proposed hyper-parameter tuning framework, we compare our results with the state of the art in sentiment analysis for low-resource languages. Specifically, through our experiments, we show that our framework surpasses state-of-art models of the same domain on several benchmarks.

In the SemEval-2023 Task 12: For example, in Sentiment Analysis for African Language (Hassan et al., 2023), a well-known competition on this subject, the highest achieved F1-Score on the Hausa dataset was 82.62%. Interestingly, our proposed framework surpasses this benchmark, achieving an impressive F1 score of 92.0%, this confirms the effectiveness of the approach in capturing nuanced sentiment in code-mixed Hausa tweets.

In addition, we compare our proposed framework to the AfriSenti benchmark (Muhammad et al., 2023), a Twitter Sentiment Analysis benchmark for African Languages that revealed the best F1 score of 67.20 %. Still, our framework outperforms this benchmark, thereby underlining our research's capacity to address the intricacies related to sentiment analysis in a multilingual environment. The details of the performance metrics are compared with the state-of-the-art benchmark in Table 14. It also further demonstrates the effectiveness of the present efficient hyperparameter tuning framework we proposed above. It is clear from these results that the current approach is adequate and efficient. Combining word embedding techniques with deep learning approaches to handle the problems of sentiment analysis in the Hausa-Informal Twi code-mixed tweets. The enhancements proved

Table 14
Performance comparison of sentiment analysis for Hausa Language.

Reference	(Methods related work)	F1-score
(Muhammad et al., 2023)	SemEval-2023 Task 12	82.62
(Muhammad et al., 2023)	AfriSenti (Benchmark)	67.20
Ours	**Proposed (framework)**	**92.00**

against existing solutions support our framework as novel contributions and usefulness for the low-resource languages sentiment analysis.

7.2. Computational efficiency analysis

Computational efficiency was evaluated based on the following metrics in Table 15. Training and inference times were measured using NVIDIA Tesla T4 GPU with a batch size of 32 on google corlab. Memory usage was monitored using PyTorch's built-in profiling tools. Listing all the best performance model.

The proposed framework demonstrates a balance between computational efficiency and performance. Among the best performing models tested. the LSTM_FastText model exhibited the shortest training time of approximately (4.5 s per epoch) and the fastest inference time of approximately (0.35 ms per input). This makes it highly efficient for real-time non-memory requirement applications. However, the CNN model has the smallest parameter count (50,905). This makes it more lightweight but with reduced contextual understanding and loss of semantics. Additionally, transformer-based models AfriBERTa require significantly more computational resources during the training. It was noted it has a training time of 256 s per epoch and 112 M parameters. Despite its computational cost AfriBERTa's performance justifies its usage for tasks requiring high contextual understanding. However, the runtime comparison in Table 8 highlights the efficiency of the proposed framework in handling low-resource SA tasks. While models like AfriBERTa provide superior accuracy, their high computational cost can make them less practical for real-time applications. In contrast, LSTM (FastText) and CNN models have lower runtime demands, making them better suited for resource-limited environments.

8. Study's implication

This research has a significant real-world application in different areas. The framework's models can be used in social media monitoring, customer feedback analysis and public sentiment tracking. For instance, businesses operating in multilingual regions can use it to analyse customer reviews and improve their services. Similarly, governments and NGOs can use it to measure public opinion during elections or health campaigns in code-switched low-resource settings. The study carries substantial significant promises for future researchers exploring the area of low-resource sentiment analysis. Despite prior work on the selection of parameters for code-switch dataset tasks, there has been no comprehensive study of the best parameter setting combinations tailored specifically for low-resource code-switch data. this work offers significant value as it presents a framework for fine-tuning pre-trained deep-learning algorithms. The framework outlines general preprocessing steps to illustrate practical instruction for emerging researchers, the basic need to know is highlighted for each step while its relevance in classification-modelling for sentiment analysis to obtain optimal solutions in low-resource areas is emphasized. The findings in the performance comparisons of the wide range of deep learning models offer important insights for researchers and application developers interested in selecting the most appropriate methods in various problems. Moreover, our results also emphasize the importance of employing word embedding method to fix the issue of the overfitting model and boosting the achievement of accuracy. Notably our approach introduces a stemming algorithm at the pre-processing step that seems particularly suited for low resource languages. The ability to apply the provided algorithm to these low-resource languages with similar features, means that a good starting point for developing sentiment analysis tools in such languages is provided.

Table 15
Computational efficiency.

Model	Training Time (per epoch)	Inference Time (per input)	Parameters
CNN	~20s	~0.59ms	50,905
LSTM_FastText	~4.5s	~0.35ms	1,079,619
GRU_glove	~50s	~1.4ms	1,061,187
GRU_Word2Vec	~53s	~1.76ms	1,061,187
AfriBERT	~256s	–	112M

9. Limitations and direction for future research

Besides strengthening confidence in the efficacy of the proposed framework, our work also underscores the value of the preprocessing approach, such as stemming, in the improvement of general DL models. This research decidedly contributes to the SA research in the low resource linguistic context. The study's efficient hyper-parameter tuning framework exhibits remarkable generalisation to other languages facing similar challenges. Similarly, the proposed framework highlights it suitability for real-world deployment in resource-constrained settings. Models such as LSTM_FastText and CNN offer competitive efficiency in both training and inference while maintaining a lower parameter count. These characteristics make them well-suited for applications requiring fast response time.

However, dedicated resources, such as dictionaries-word pairs may be necessary for stemming in each language. The framework also relies on pre-trained embedding, which requires computational resources during training. Additionally, the trade-off between efficiency and performance must be considered. Although transformer-based models thus, AfriBERTa are computationally expensive. They outperform simpler architectures in tasks requiring deeper contextual understanding and huge dataset analysis. Future work will further focus on optimising transformer-based models to balance efficiency and performance. The framework could be extended with data augmentation in transformer models to encompass a wider array of languages and domains. We seek to amplify the applicability and impact of sentiment analysis tools in real-world scenarios, which will foster the development of effective sentiment analysis models in linguistically diverse settings.

10. Conclusions

There has been an upsurge in the last couple of years in SA and text classification. This study addresses a critical gap in SA, by focusing on code-switched low-resource language text. The research not only advanced academic understanding but also provides a practical tool for real-world application. Traditional approaches have predominantly relied on basic machine learning classifiers and deep learning models. The study introduces an efficient hyperparameter tuning framework design to improve the accuracy of the existing DL models. The proposed framework offers a structured approach for tailoring tools to accommodate the distinct linguistic features for specific languages. It combines multiple models utilizing pre-trained embedding vectors which, include Glove, FastText, and Word2vec. Moreover, the framework was evaluated using Hausa-English code-switched tweets. Furthermore, its applicability was validated across other low-resource language settings. The results demonstrate that transformer-based models outperform conventional DL techniques. Notably, AfriBERTa emerges as a particularly effective model for Hausa-English code-switched data, demonstrating superior generalization capabilities and achieving higher accuracy in sentiment classification. Although this study focuses on a Hausa-English dataset, the framework's adaptability to other code-switched languages requires further validation. Future work will

involve testing the model on diverse datasets and larger corpora, to evaluate its robustness across various NLP applications. The study contributions underscore the importance of inclusive AI solution that serve underrepresented linguistic communities.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used Grammarly and quillbot in order to check misspelling and grammatical error. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

CRediT authorship contribution statement

Yusuf Aliyu: Writing – original draft, Methodology, Formal analysis. **Aliza Sarlan:** Writing – review & editing, Supervision. **Kamaluddeen Usman Danyaro:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Abdullahi Sani abd Rahman:** Writing – review & editing. **Aminu Aminu Muazu:** Writing – review & editing, Formal analysis, Data curation. **Mustapha Yusuf Abubakar:** Data curation.

Declaration of competing interest

The authors therefore wish to confirm that they have no financial or personal interests in the work performed in this paper.

Acknowledgements

We express our gratitude to PTDF (Petroleum Technology Development Fund) and Universiti Teknologi PETRONAS (UTP) for their assistance. Their help was instrumental in helping us share the findings of our work, identify the new trends, and most importantly, to build the essential professional network within our field. As we value acknowledgment, we appreciate the investment that is being made in our scholarly and real growth. The work which underpins this paper was funded by the Integration of Metocean Data for Intelligent Operations and Automation Using Large Language Models (LLMs). Cost Center: YUTP-PRG 015PBC-035.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.jjimei.2025.100330.

References

Abdulmumin, I., & Galadanci, B. S. (2019). hauWE: Hausa words embedding for natural language processing. In *2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf)* (pp. 1–6). https://doi.org/10.1109/NigeriaComputConf45974.2019.8949674, 14-17 Oct. 201.

Abubakar, A. I., Roko, A., Bui, A. M., & Saidu, I. (2021). An enhanced feature acquisition for sentiment analysis of English and Hausa tweets. *International Journal of Advanced Computer Science and Applications(IJACSA), 12*(9). https://doi.org/10.14569/IJACSA.2021.0120913

Alabi, J.O., Adelani, D.I., Mosbach,.M., & Klakow,.D. (2022)."Multilingual language model Adaptive Fine-Tuning: A study on African languages," *arXiv preprint* arXiv:2204.06487.

Al-Saqqa, S., Awajan, A., & Ghoul, S. (2019). Stemming effects on sentiment analysis using large Arabic multi-domain resources. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (pp. 211–216). https://doi.org/10.1109/SNAMS.2019.8931812, 22-25 Oct. 201.

Al Shamsi, A. A., & Abdallah, S. (2022). Sentiment analysis of Emirati dialect. *Big Data and Cognitive Computing, 6*(2), 57 [Online]Available https://www.mdpi.com/2504-2289/6/2/57.

Al Shamsi, A. A., & Abdallah, S. (2023). Ensemble stacking model for sentiment analysis of Emirati and Arabic dialects. *Journal of King Saud University - Computer and Information Sciences, 35*(8), Article 101691. https://doi.org/10.1016/j.jksuci.2023.101691, 2023/09/01.

Amjad, M., Ashraf, N., Zhila, A., Sidorov, G., Zubiaga, A., & Gelbukh, A. (2021). Threatening language detection and target identification in Urdu tweets. *IEEE access : practical innovations, open solutions, 9*, 128302–128313. https://doi.org/10.1109/ACCESS.2021.3112500

Araújo, M., Pereira, A., & Benevenuto, F. (2020). A comparative study of machine translation for multilingual sentence-level sentiment analysis. *Information Sciences, 512*, 1078–1102. https://doi.org/10.1016/j.ins.2019.10.031, 2020/02/01.

Bansal, V., Tyagi, M., Sharma, R., Gupta, V., & Xin, Q. (2022). A transformer based approach for abuse detection in code mixed indic languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*. https://doi.org/10.1145/3571818

Bashir, M., Rozaimee, A., Binti, W., & Wan Isa, W. M. (2015). A word stemming algorithm for hausa language. *IOSR Journal of Computer Engineering, 17*, 2278–2661. https://doi.org/10.9790/0661-17362531, 06/2.

Bensoltane, R., & Zaki, T. (2021). Comparing word embedding models for Arabic aspect category detection using a deep learning-based approach. In , *297*. *E3S web of conferences* (p. 01072). EDP Sciences.

Bilal, M., & Almazroi, A. A. (2023). Effectiveness of fine-tuned BERT model in classification of helpful and unhelpful online customer reviews. *Electronic Commerce Research, 23*(4), 2737–2757. https://doi.org/10.1007/s10660-022-09560-w, 2023/12/0.

Bimba, A., Idris, N., Khamis, N., & Noor, N. F. M. (2016). Stemming Hausa text: using affix-stripping rules and reference look-up. *Language Resources and Evaluation, 50*(3), 687–703. https://doi.org/10.1007/s10579-015-9311-x, 2016/09/0.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics, 5*, 135–146.

Chakravarthi, B.R., Jose,.N., Suryawanshi,.S., Sherly,.E., & McCrae, J.P. (2020)."A sentiment analysis dataset for code-mixed Malayalam-English," *arXiv preprint* arXiv:2006.00210.

Conneau, A., et al. (2020). *Unsupervised cross-lingual representation learning at scale*. ACL.

Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. *Electronics, 9*(3), 483 [Online]Available https://www.mdpi.com/2079-9292/9/3/483.

Demotte, P., Senevirathne, L., Karunanayake, B., Munasinghe, U., & Ranathunga, S. (2020). Sentiment analysis of Sinhala News comments using sentence-State LSTM networks. In *2020 Moratuwa Engineering Research Conference (MERCon)* (pp. 283–288). https://doi.org/10.1109/MERCon50084.2020.9185327, 28-30 July 202.

Devlin, J., Chang, M.-W., Lee,.K., & Toutanova,.K. (2019)."BERT: pre-training of deep bidirectional transformers for language understanding," *ArXiv,* vol. abs/1810.04805.

Ganganwar, V., & Rajalakshmi, R. (2019). Implicit aspect extraction for sentiment analysis: A survey of recent approaches. *Procedia Computer Science, 165*, 485–491. https://doi.org/10.1016/j.procs.2020.01.010, 2019/01/01.

Ghafoor, A., et al. (2021). The impact of translating resource-rich datasets to low-resource languages through multi-lingual text processing. *IEEE access : practical innovations, open solutions, 9*, 124478–124490. https://doi.org/10.1109/ACCESS.2021.3110285

Hasib, K. M., Habib, M. A., Towhid, N. A., & Showrov, M. I. H. (2021). A novel deep learning based sentiment analysis of Twitter data for US airline service. In *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)* (pp. 450–455). https://doi.org/10.1109/ICICT4SD50815.2021.9396879, 27-28 Feb. 202.

Hasib, K. M., Towhid, N. A., & Alam, M. G. R. (2021). Online review based sentiment classification on Bangladesh airline Service using supervised learning. In *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)* (pp. 1–6). https://doi.org/10.1109/ICEEICT53905.2021.9667818, 18-20 Nov. 202.

Hassan, M.S. et al., (2023). "SemEval-2023 task 12: sentiment analysis for African languages (AfriSenti-SemEval)," *arXiv preprint* arXiv:2304.06845.

Hassan Muhammad, S., et al. (2022). *NaijaSenti: A nigerian twitter sentiment corpus for multilingual sentiment analysis*.

Ibrahim, U. A., Boukar, M. M., & Suleiman, M. A. (2022). Development of Hausa dataset a baseline for speech recognition. *Data in Brief, 40*, Article 107820. https://doi.org/10.1016/j.dib.2022.107820, 2022/02/01.

Isbister, T., Carlsson,.F., & Sahlgren,.M. (2021)."Should we stop training more monolingual models, and simply use machine translation instead?," *arXiv preprint* arXiv:2104.10441.

Jabbar, A., Iqbal, S., Tamimy, M. I., Rehman, A., Bahaj, S. A., & Saba, T. (2023). An analytical analysis of text stemming methodologies in information retrieval and natural language processing systems. *IEEE Access : Practical Innovations, Open Solutions, 11*, 133681–133702. https://doi.org/10.1109/ACCESS.2023.3332710

Jamatia, A., Swamy, S. D., Gambäck, B., Das, A., & Debbarma, S. (2020). Deep learning based sentiment analysis in a code-mixed English-Hindi and English-bengali social Media corpus. *International Journal on Artificial Intelligence Tools, 29*(05), Article 2050014. https://doi.org/10.1142/s0218213020500141

James, J., et al. (2022). Language models for code-switch detection of te reo Māori and English in a low-resource setting. *Findings of the association for computational linguistics: Naacl 2022* (pp. 650–660).

Kanclerz, K., Miłkowski, P., & Kocoń, J. (2020). Cross-lingual deep neural transfer learning in sentiment analysis. *Procedia Computer Science, 176*, 128–137. https://doi.org/10.1016/j.procs.2020.08.014, 2020/01/01.

Kim, Y. (2014)."Convolutional neural networks for sentence classification," *arXiv preprint* arXiv:1408.5882.

Kocoń, J., Zaśko-Zielińska, M., & Miłkowski, P. (2019). *Multi-Level analysis and recognition of the text sentiment on the example of consumer opinions*.

Konate, A., & Du, R. (2018). Sentiment analysis of code-mixed Bambara-French social Media text using deep learning techniques. *Wuhan University Journal of Natural Sciences, 23*(3), 237–243. https://doi.org/10.1007/s11859-018-1316-z, 2018/06/0.

Kumar, A., & Albuquerque, V. H. C. (2021). Sentiment analysis using XLM-R transformer and zero-shot transfer learning on resource-poor indian language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process., 20*(5), Article 90. https://doi.org/10.1145/3461764

Kuwanto, G., Agarwal,.C., Winata, G.I., & Wijaya, D.T. (2024)."Linguistics theory meets LLM: code-switched text generation via equivalence constrained large language models," *arXiv preprint* arXiv:2410.22660.

Liu, B. (2015). Introduction. *Sentiment analysis: Mining opinions, sentiments, and emotions* (pp. 1–15). Cambridge: Cambridge University Press.

Liu, Y. et al., (2019). "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint* arXiv:1907.11692.

Londhe, D. D., Kumari, A., & Emmanuel, M. (2021). Challenges in multilingual and mixed script sentiment analysis. In *2021 6th International Conference for Convergence in Technology (I2CT)* (pp. 1–6). https://doi.org/10.1109/I2CT51068.2021.9418087, 2-4 April 202.

Magueresse, A., Carles, V., & Heetderks, E. (2020). *Low-resource languages: A review of past work and future challenges*.

Mahadzir, N. (2021). Sentiment analysis of code-mixed text: A review. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12*, 2469–2478. https://doi.org/10.17762/turcomat.v12i3.1239, 04/1.

Meena, G., & Mohbey, K. K. (2023). Sentiment analysis on images using different transfer learning models. *Procedia Computer Science, 218*, 1640–1649. https://doi.org/10.1016/j.procs.2023.01.142, 2023/01/01.

Meena, G., Mohbey, K. K., & Kumar, S. (2023). Sentiment analysis on images using convolutional neural networks based Inception-V3 transfer learning approach. *International Journal of Information Management Data Insights, 3*(1), Article 100174. https://doi.org/10.1016/j.jjimei.2023.100174, 2023/04/01.

Mikolov, T., Chen,.K., Corrado,.G., & Dean,.J. (2013)."Efficient estimation of word representations in vector space," *arXiv preprint* arXiv:1301.3781.

Muhammad, S. H., et al. (2022). *NaijaSenti: A nigerian twitter sentiment corpus for multilingual sentiment analysis*.

Muhammad, S.H. et al., (2023). "SemEval-2023 task 12: sentiment analysis for African languages (AfriSenti-SemEval)," *arXiv preprint* arXiv:2304.06845.

Muhammad, S.H. et al., (2023). "Afrisenti: A twitter sentiment analysis benchmark for african languages," *arXiv preprint* arXiv:2302.08956.

Musa, S., Obunadike, G. N., & Yakubu, M. M. (2022). An improved hausa word stemming algorithm. *Fudma Journal of Sciences, 6*(1), 291–295. https://doi.org/10.33003/fjs-2022-0601-899, 04/0.

Mutinda, J., Mwangi, W., & Okeyo, G. (2023). Sentiment analysis of text reviews using Lexicon-enhanced Bert embedding (LeBERT) model with convolutional neural network. *Applied Sciences, 13*(3), 1445 [Online]Available https://www.mdpi.com/2076-3417/13/3/1445.

Ogueji, K., Zhu, Y., & Lin, J. (2021). Small data? No problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning* (pp. 116–126).

Pathak, A. R., Agarwal, B., Pandey, M., & Rautaray, S. (2020). Application of deep learning approaches for sentiment analysis. In B. Agarwal, R. Nayak, N. Mittal, & S. Patnaik (Eds.), *Deep learning-based approaches for sentiment analysis* (pp. 1–31). Singapore: Springer Singapore.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Perry, T. (2021)."Lighttag: text annotation platform," *arXiv preprint*arXiv:2109.02320.

Pires, T. J. P., Schlinger, E., & Garrette, D. (2019). How multilingual is Multilingual BERT?. In *Annual Meeting of the Association for Computational Linguistics*.

Prottasha, N. J., et al. (2022). Transfer learning for sentiment analysis using BERT based supervised fine-tuning. *Sensors, 22*(11), 4157 [Online]Available https://www.mdpi.com/1424-8220/22/11/4157.

Raihan, M.N., Goswami,.D., Mahmud,.A., Anstasopoulos,.A., & Zampieri,.M. (2023)."SentMix-3L: A bangla-english-Hindi code-mixed dataset for sentiment analysis," *arXiv preprint* arXiv:2310.18023.

Rajalakshmi, R., Selvaraj, S., R, F. M., Vasudevan, P., & M, A. K. (2023). HOTTEST: hate and offensive content identification in Tamil using transformers and enhanced STemming. *Computer Speech & Language, 78*, Article 101464. https://doi.org/10.1016/j.csl.2022.101464, 2023/03/01.

Rajeshwari, S. B., & Kallimani, J. S. (2022). Development of optimized linguistic technique using similarity score on BERT model in summarizing Hindi text documents. *Innovative data communication technologies and application* (pp. 767–781). Singapore: Springer Nature Singapore.

Rakhmanov, O., & Schlippe, T. (2022). *Sentiment analysis for hausa: Classifying students' comments*.

Řehůřek, R., & Sojka, P. (2010). *Software framework for topic modelling with large corpora* (pp. 45–50).

Roy, P. K. (2024). Deep Ensemble Network for sentiment analysis in Bi-lingual low-resource languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process., 23*(1), Article 8. https://doi.org/10.1145/3600229

Sabri, N., Edalat, A., & Bahrak, B. (2021). Sentiment analysis of persian-english code-mixed texts. In *2021 26th International Computer Conference, Computer Society of Iran (CSICC)* (pp. 1–4). https://doi.org/10.1109/CSICC52343.2021.9420605, 3-4 March 202.

Shanmugavadivel, K., Sathishkumar, V. E., Raja, S., Lingaiah, T. B., Neelakandan, S., & Subramanian, M. (2022). Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data. *Scientific reports, 12*(1), 21557. https://doi.org/10.1038/s41598-022-26092-3, 2022/12/1.

Shehu, H. A., et al. (2024). Unveiling sentiments: A deep dive into sentiment analysis for low-resource languages—A case study on Hausa texts. *IEEE access : practical innovations, open solutions, 12*, 98900–98916. https://doi.org/10.1109/ACCESS.2024.3427416

Singh, O. M., Timilsina, S., Bal, B. K., & Joshi, A. (2020). Aspect based abusive sentiment detection in Nepali social Media texts. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (pp. 301–308). https://doi.org/10.1109/ASONAM49781.2020.9381292, 7-10 Dec. 202.

Sirisha, U., & Chandana, B. (2022). Aspect based sentiment and emotion Analysis with ROBERTa, LSTM. *International Journal of Advanced Computer Science and Applications, 13*. https://doi.org/10.14569/IJACSA.2022.0131189, 01/0.

Song, M., Park, H., & Shin, K.-S. (2019). Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in Korean. *Information Processing & Management, 56*(3), 637–653. https://doi.org/10.1016/j.ipm.2018.12.005, 2019/05/01.

Srinivasan, R., & Subalalitha, C. N. (2023). Sentimental analysis from imbalanced code-mixed data using machine learning approaches. *Distributed and Parallel Databases, 41*(1), 37–52. https://doi.org/10.1007/s10619-021-07331-4, 2023/06/0.

Suleiman, M., Aliyu, M. M., & Zimit, S. (2019). Towards the development of Hausa language corpus. *Int. J. Sci. Eng. Res, 10*, 1598–1604.

Tan, K. L., Lee, C. P., Anbananthen, K. S. M., & Lim, K. M. (2022). RoBERTa-LSTM: A hybrid model for sentiment analysis with transformer and recurrent neural network. *IEEE access : practical innovations, open solutions, 10*, 21517–21525. https://doi.org/10.1109/ACCESS.2022.3152828

Tao, J., & Fang, X. (2020). Toward multi-label sentiment analysis: A transfer learning based approach. *Journal of Big Data, 7*(1), 1. https://doi.org/10.1186/s40537-019-0278-0, 2020/01/0.

Thara, S., & Poornachandran, P. (2022). Social media text analytics of Malayalam–English code-mixed using deep learning. *Journal of Big Data, 9*(1), 45. https://doi.org/10.1186/s40537-022-00594-3, 2022/04/2.

Thin, D. V., Hao, D. N., & Nguyen, N. L.-T. (2023). Vietnamese sentiment analysis: an overview and comparative study of fine-tuning pretrained language models. *ACM Trans. Asian Low-Resour. Lang. Inf. Process., 22*(6), Article 166. https://doi.org/10.1145/3589131

Toshevska, M., Stojanovska, F., & Kalajdjieski, J. (2020). *Comparative analysis of word embeddings for capturing word similarities*.

Vaswani, A., et al. (2017). Attention is all you need. *Advances in neural information processing systems, 30*.

Vo, Q.-T., Tran,.P., & Tran,.T. (2023). Sentiment analysis for a low-resource language: A study on a Vietnamese university, The 4th International Conference of Computer Science and Renewable Energies (ICCSRE'2021). 14, pp. 1115–1124, https://www.e3s-conferences.org/articles/e3sconf/abs/2021/73/e3sconf_iccsre21_01072/e3sconf_iccsre21_01072.html.

Wang, M., Adel,.H., Lange,.L., Strötgen,.J., & Schütze,.H. (2023)."NLNDE at SemEval-2023 task 12: adaptive pretraining and source language selection for low-resource multilingual sentiment analysis," *arXiv preprint* arXiv:2305.00090.

Winata, G.I., Cahyawijaya,.S., Liu,.Z., Lin,.Z., Madotto,.A., & Fung,.P. (2021)."Are multilingual models effective in code-switching?," *arXiv preprint*arXiv:2103.13309.

Wu, S., & Dredze, M. (2020). *Are all languages created equal in multilingual bert* (pp. 120–130).

Xu, H., Durme, B. V., & Murray, K. (2021). BERT, mBERT, or BiBERT? A study on contextualized embeddings for neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Yue, L., Chen, W., Li, X., Zuo, W., & Yin, M. (2019). A survey of sentiment analysis in social media. *Knowledge and Information Systems, 60*(2), 617–663. https://doi.org/10.1007/s10115-018-1236-4

Yusuf, A., Sarlan, A., Danyaro, K. U., & Rahman, A. S. B. A. (2023). Fine-tuning multilingual transformers for Hausa-English sentiment analysis. In *2023 13th International Conference on Information Technology in Asia (CITA)* (pp. 13–18). https://doi.org/10.1109/CITA58204.2023.10262742

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *WIREs Data Mining and Knowledge Discovery, 8*(4), e1253. https://doi.org/10.1002/widm.1253

Zouzou, A., & Azami, I. E. (2021). Text sentiment analysis with CNN & GRU model using GloVe. In *2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS)* (pp. 1–5). https://doi.org/10.1109/ICDS53782.2021.9626715