IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 October 2022, accepted 8 November 2022, date of publication 21 November 2022,
date of current version 29 November 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3223703*

SURVEY

# A Systematic Review on Language Identification of Code-Mixed Text: Techniques, Data Availability, Challenges, and Framework Development

**AHMAD FATHAN HIDAYATULLAH[1,2], ATIKA QAZI[3], DAPHNE TECK CHING LAI[1], (Member, IEEE), AND ROSYZIE ANNA APONG[1]**

[1]School of Digital Science, Universiti Brunei Darussalam, Gadong BE1410, Brunei Darussalam
[2]Department of Informatics, Universitas Islam Indonesia, Yogyakarta 55584, Indonesia
[3]Centre for Lifelong Learning, Universiti Brunei Darussalam, Gadong BE1410, Brunei Darussalam

Corresponding authors: Atika Qazi (atikaqazium@gmail.com) and Ahmad Fathan Hidayatullah (21h2501@ubd.edu.bn)

This work was supported by the Universiti Brunei Darussalam and Ministry of Education (MoE) Brunei Darussalam.

**ABSTRACT** The mix of native language with other languages (code-mixing) in social media has posed a severe challenge for language identification (LID) systems. It has encouraged research on code-mixed LID solutions. Four things have been identified in this study, such as techniques, challenges, and dataset availability with corresponding quality criteria and developed a comprehensive framework for code-mixed LID. Also, we identified gaps and future work opportunities in tackling code-mixed LID challenges. Based on our analysis of reviewed studies, we outlined key points for future research in code-mixed LID. We demonstrated a taxonomy of applied techniques for code-mixed LID and highlighted the different technique variants. In code-mixed LID tasks, we discovered four significant challenges: ambiguity, lexical borrowing, non-standard words, and intra-word code-mixing. This systematic literature review recognised 32 code-mixed datasets available for LID. We proposed five features to describe the quality criteria datasets, such as the number of instances or sentences, percentage of code-mixed types in the data, number of tokens, number of unique tokens, and average sentence length. Finally, we synthesised the methodologies and proposed a conceptual framework for subsequent studies through our literature analysis.

**INDEX TERMS** Code-mixed text, code-mixing, language identification, social media, machine learning, deep learning.

## I. INTRODUCTION

With the advent of social media, human interaction has become limitless. Social media platforms have become an integral and inseparable part of human life. We can connect with people from all over the world through social media to exchange and spread information. For instance, we can leverage social media to increase customer engagement and thus generate brand exposure, leads, sales, and revenue in the business domain.

In social media, individuals write posts without adhering to the standard language of communication [1]. For example, multiple languages in a single sentence or utterances within social media texts. It is common for people who live in a multilingual culture and know many languages to switch from one language to another [2]. People often express their thoughts on social media in mixed languages using their native language and English [3]. In linguistics, this is known as code-mixing, which refers to the embedding of linguistic units from one language into the usage of another language by using phrases, words, and morphemes [4].

Code-mixing is commonly encountered during spoken and written communication in multilingual communities [5], [6],

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

for example, Indonesian-English [7], [8], Malay-English [9], Persian-English [10], Hindi-English [11], and English-Bengali [12]. Code-mixing can be divided into intra-sentential, intra-word, and inter-sentential. Intra-sentential code-mixing is a term that refers to occurrences of mixing languages within a sentence. Intra-word code-mixing refers to the mixing of languages in a word. Inter-sentential code-mixing happens when languages are mixed across sentences. Because of variances in spelling and grammar, code-mixing in social media material is a daunting task in natural language processing [7]. Consequently, code-mixed text requires more pre-processing tasks than monolingual text data [13].

One of the pre-processing tasks that is frequently applied in analysing code-mixed text is language identification (LID). LID refers to the automatic identification of languages used in a document [14], [15]. LID is crucial for downstream natural language processing (NLP) applications, such as sentiment analysis and machine translation [15], [16], [17], [18]. Applying LID for such NLP applications may significantly impact the system's performance [11].

Most LID studies, however, focus on identifying a single language at the document or sentence level. Determining languages in a code-mixed text, therefore, remains an unresolved problem. Performing LID tasks at the document or sentence level are frequently inadequate for extracting critical information from the text [18]. Also, relying on language tags at the document or sentence level makes language detector systems fail to detect language correctly due to the mixed language in the sentences [19], [20]. Thus, researchers were motivated to shift their focus from document or sentence level to token-level language identification.

One notable gap in current research is the need of code-mixed datasets for low-resource languages. Low-resource languages are less common and studied as a result of scarce resources [21]. Regarding code-mixed data, language pairs involving languages from South Asia (Hindi and Bengali) and English are prevalent [2]. Exploring additional language pairs for low-resource languages is highly encouraged, accordingly. The new language pair datasets are necessary to help solve code-mixed LID problems in languages commonly used but lacking resources. Apart from that, the lexical look-up or dictionary-based approach cannot cope with the presence of borrowed words or code-mixing [22]. Another problem is the failure to get context information due to ambiguity and irregular phonetic typing in the code-mixed text [16], [23], [24].

We found a few literature reviews related to code-switching and code-mixing text with different research focuses, such as the application of code-mixing [25], a survey on the code-switched dataset [2], and sentiment analysis of a code-mixed text [24], [26], [27]. To the best of our knowledge, there has not been a comprehensive literature review that explicitly highlights the latest techniques of LID and reviews its challenges for code-mixed texts. Such a survey would benefit relevant researchers in NLP and text processing.

This systematic review aims to examine the current state of research in the LID field for code-mixed texts. The objectives of this study are: (1) to investigate the most recent techniques developed for solving LID tasks for code-mixed content; (2) to explore the resolved and unresolved challenges associated with LID tasks for code-mixed text; (3) to investigate the availability and quality of code-mixed datasets for LID; and (4) to develop a general framework for code-mixed LID.

The remainder of this paper is organised as follows. Section 2 describes the research methodology. The result and discussion of this study are explained in Section 3. Section 4 presents the implications of this literature study. Finally, the conclusion is described in Section 5. A list of abbreviations used in this literature review paper is presented in Table 1.

**TABLE 1. List of abbreviations used in this paper.**

| List of abbreviations | |
|---|---|
| ANN | Artificial Neural Network |
| BERT | Bidirectional Encoder Representations from Transformers |
| BLSTM | Bidirectional Long Short-Term Memory |
| BLSTM-CRF | Bidirectional Long Short-Term Memory-Conditional Random Fields |
| CNN | Convolutional Neural Network |
| CNN-BLSTM | Convolutional Neural Network- Bidirectional Long Short-Term Memory |
| CNN-BLSTM-CRF | Convolutional Neural Network- Bidirectional Long Short-Term Memory- Conditional Random Fields |
| CRF | Conditional Random Fields |
| ELECTRA | Efficiently Learning an Encoder that Classifies Token Replacements Accurately |
| GloVe | Global Vector |
| GRU | Gated Recurrent Unit |
| HMM | Hidden Markov Model |
| KNN | K-Nearest Neighbors |
| LID | Language identification |
| LSTM | Long Short-Term Memory |
| MNN | Multichannel Neural Network |
| NLP | Natural language processing |
| PRISMA | Preferred Reporting Items for Systematic Reviews and Meta-Analysis |
| RNN | Recurrent Neural Network |
| SegRNN | Segmental Recurrent Neural Network |
| SLR | Systematic Literature Review |
| SVC | Support Vector Classifier |
| SVM | Support Vector Machine |
| TF-IDF | Term Frequency-Inverse Document Frequency |
| Word2Vec | Word to vector |
| XGB | Extreme Gradient Boosting |
| XML | Extensible Markup Language |

## II. RESEARCH METHODOLOGY

This section consists of a system of guidelines for designing and analysing the studies of language identification in code-mixed data. This literature study follows the systematic literature review methodology by [28] and [29]. The review's

strategy includes establishing the study population (studies of language identification in code-text data), identifying resources from where the population is sourced, listing search string keywords, and determining the inclusion and exclusion criteria to generate the population relevant to this study. The research methodology is conducted by applying the following review strategies: (1) designing research questions; (2) searching related studies from databases using defined search strings; (3) applying predetermined inclusion and exclusion criteria; and (4) applying quality assessment criteria.

### A. RESEARCH QUESTION

This study aims to answer the following research questions to highlight critical practical aspects of language identification of code-mixed text. The four research questions addressed in this literature review are as follows:

*RQ1:* Which techniques and features have been used for code-mixed LID text in bilingual and multilingual?

The response to RQ1 allows us to learn about the techniques used to solve code-mixed language identification task. Examining previously used methods will provide insight into the state-of-the-art, advantages and limitations of LID in code-mixed data. The findings will demonstrate the most recommended technique and features for dealing with code-mixed text LID. Also, we expect to obtain some additional features applied in LID for code-mixed text.

*RQ2:* What are the challenges in LID of code-mixed text?

The RQ2 aims to identify the open challenges LID for code-mixed text. Understanding the challenges and the current state of the art is necessary for determining the research gaps in the previous studies that are currently not addressed or answered adequately. The findings would also assist in directing future work, considering resolved and unresolved issues in code-mixed LID.

*RQ3:* What datasets are available for LID of code-mixed text? What are the quality criteria for the dataset?

This RQ3 aims to determine the availability of code-mixed datasets and quality criteria for language identification using code-mixed text from various languages. The investigation of the availability of code-mixed datasets allows us to determine how many datasets are bilingual and multilingual code-mixed datasets. We will also learn about the popular mixed languages studied and those less studied. Answering RQ3 allows us to know the source benchmark datasets and prepare the scope of our experiments for evaluating our LID methodology in code-mixed text. The dataset quality criteria provide a set of properties and policies to determine the dataset's quality and completeness. We can evaluate the dataset's quality by identifying the relevant properties to measure as our proposed quality criteria.

*RQ4:* What is the standard workflow for language identification of code-mixed text for future research?

The RQ4 allows us to know the future directions of code-mixed LID research. To answer RQ4, we propose a framework developed for the code-mixed LID task. The framework can be leveraged as a standard guideline for those researching code-mixed LID.

### B. DATA SOURCES AND SEARCH STRATEGY

In this work, we referred to [28] and [30] to find related articles through electronic databases. We selected five electronic sources to gather our references. Through the electronic sources, we investigated all available materials pertaining to the objectives of this systematic literature review [31]. Search strings (keywords) were developed to collect related research papers responding to the research questions. The search strings were developed using critical terms within the topic field and the purpose of the review [32]. The selected electronic sources and search strings for this literature study are provided in Table 2.

**TABLE 2. Electronic sources and search string keywords.**

| | |
|---|---|
| Electronic Source | • ACM Digital Library<br>• Google Scholar<br>• IEEE Xplore<br>• ScienceDirect<br>• Springer |
| Search String | • language identification AND code mix<br>• language identification AND code switch<br>• language identification code mix AND social media<br>• language identification code switch AND social media<br>• challenges AND language identification AND code mix<br>• challenges AND language identification AND code switch<br>• code mix OR code switch |

### C. INCLUSION AND EXCLUSION CRITERIA

This section discusses the inclusion and exclusion criteria applied in our literature study. Meta-data and abstracts of papers were reviewed to determine which studies should be included in the review and removed irrelevant articles [33]. The following criteria were applied for inclusion: (I1) Studies published between 2016 and 2021; (I2) full-text papers; (I3) papers written in English; (I4) papers related to language identification for code-mixing or code-switching text. We excluded those articles that did not satisfy the inclusion criteria from the study. Also, any publications that did not match any of the excluded criteria were excluded.

The inclusion and exclusion criteria for this study are presented in Table 3. The following are the exclusion criteria to eliminate irrelevant papers: (E1) papers not written in the English language; (E2) papers that do not focus on natural language processing fields; (E3) papers that do not discuss language identification for code-mixing or code-switching text; (E4) grey literature, such as working papers, dissertation/theses, and research reports.

**TABLE 3.** Inclusion and exclusion criteria.

| Inclusion Criteria | Exclusion Criteria |
|---|---|
| • Papers published within the period 2016-2021<br>• Full-text papers<br>• Papers written in English<br>• Papers related to language identification for code-mixing or code-switching text | • Papers not written in English<br>• Papers not related to natural language processing<br>• Papers not related to language identification for code-mixing or code-switching text<br>• Grey literature |

### D. QUALITY ASSESSMENT (QA)

The QA [28] is used in this systematic literature review to determine the strength of the selected studies [34]. The QA was developed using tools such as a checklist of all aspects or queries required to be applied to each study. The following questions were developed as the QA criteria for each study:

1. (C1) Does the paper describe the code-mixed dataset clearly?
2. (C2) Are the techniques clearly explained in the paper?
3. (C3) Does the research paper explain the challenges in LID of code-mixed text?
4. (C4) Are the findings clearly stated in the paper?

### E. STUDY SELECTION PROCESS

This section explains the selection process to determine relevant studies that fulfil all the research questions. The study selection task is done in four phases: identification, screening, eligibility, and included studies. In this study selection stage, we utilised PRISMA flow diagram as a reporting guideline [35].

Figure 1 illustrates the PRISMA flow diagram of the systematic review protocol in this work. In the identification stage, we searched the literature from five electronic databases using predefined keywords and obtained 233 research papers. Firstly, we screened the retrieved research by removing duplicate papers. A total of 70 papers were removed in the screening stage. We applied inclusion and exclusion criteria to 163 articles, and a total of 86 studies were eliminated. After that, the rest of the 77 articles were assessed using the five quality criteria, and we excluded 37 papers in this stage. Finally, a list of 40 research papers (referred to as selected studies) was included in this literature review, 8 papers (20%) from journals and 32 papers (80%) from conferences. Table 4 shows the number of selected papers in each stage.

## III. RESULT AND DISCUSSION

This section presents the findings of the primary studies on LID for code-mixed text. We divided our discussion into three subsections in response to the respective research questions explained earlier. The first subsection addresses RQ1 regarding existing LID techniques in bilingual and multilingual code-mixed text. In the second subsection, we present our findings concerning RQ2, which investigates the challenges of code-mixed LID. Subsequently, we provide our findings regarding dataset availability and quality criteria for evaluating code-mixed LID tasks. Finally, we provide a framework in response to RQ4, which is explained in the last subsection.

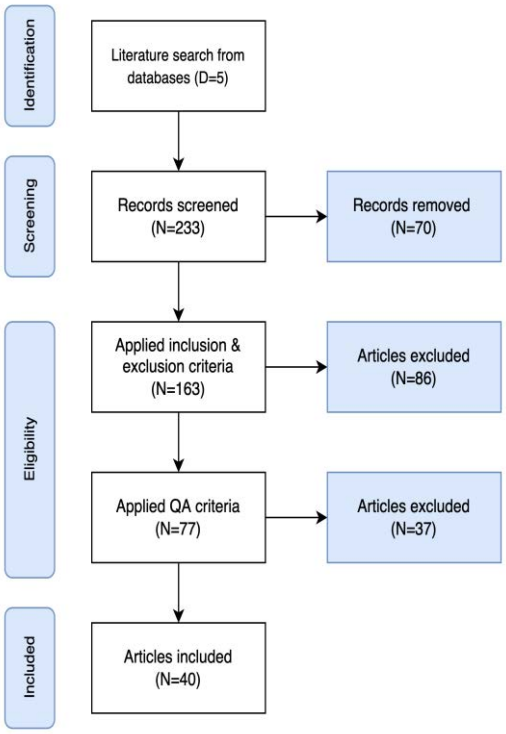


**FIGURE 1.** PRISMA flow diagram of the study selection process.

**TABLE 4.** Number of reviewed papers.

| Database | Papers | Round 1 (Inclusion & Exclusion Criteria) | | Round 2 (QA Criteria) | |
|---|---|---|---|---|---|
| | | Included | Excluded | Included | Excluded |
| ACM Digital Library | 19 | 5 | 14 | 2 | 3 |
| Google Scholar | 68 | 35 | 33 | 20 | 15 |
| IEEE Xplore | 42 | 19 | 23 | 12 | 7 |
| ScienceDirect | 19 | 9 | 10 | 1 | 8 |
| Springer | 15 | 9 | 6 | 5 | 4 |
| Total | 163 | 77 | 86 | 40 | 37 |

### A. (RQ1) WHICH TECHNIQUES AND FEATURES HAVE BEEN USED FOR LID OF CODE-MIXED TEXT IN BILINGUAL AND MULTILINGUAL?

In the following part, we examined the characteristics of existing techniques used in code-mixed LID from the 40 selected studies. We intend to highlight and discuss the existing techniques and their properties to update researchers for future work.

### 1) APPROACHES AND APPLIED TECHNIQUES

Figure 2 depicts the taxonomy of approaches and applied techniques implemented for code-mixed LID from the selected studies. Based on our investigation, two primary approaches were identified, machine learning and non-machine learning. Machine learning can be divided into two main categories, supervised and unsupervised. We identified three groups for the supervised one: non-neural network-based (12 unique techniques), neural network-based (9 unique techniques), and hybrid technique (2 unique techniques).

Support Vector Machine (SVM) and Conditional Random Fields (CRF) were the most utilised supervised technique. SVM and CRF have been implemented in 14 studies, followed by Naïve Bayes in 12 studies. Logistic Regression and Random Forest were used in 8 research, respectively. Decision Tree and K-Nearest Neighbour (KNN) were applied in 6 and 3 studies. We found 2 studies that applied AdaBoost and HMM. The remaining methods (XGBoost, Linear Discriminant Analysis, and Quadratic Discriminant Analysis) were utilised in one study severally.

We found several neural network-based techniques, such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and RNN variants, such as Long Short-Term Memories (LSTM), Gated Recurrent Unit (GRU), Bidirectional LSTM (BLSTM), Segmental Recurrent Neural Network (SegRNN), and Transformer-based. There are five methods in the Transformer-based, namely XLM-RoBERTa, ELECTRA, BERT, DistilBERT, and Camem-BERT. We encountered three unique techniques for the unsupervised approach: Hidden Markov Model (HMM), Morfessor, and the unsupervised dictionary-based approach. As for non-machine learning, two techniques were recognised, rule-based and lexicon-based. We came across three previous studies that implemented rule-based, and two studies used lexicon-based.

Table 5 summarises relevant literature for code-mixed language identification. We summarised the following information: year of publication, languages identified, applied techniques, language identification level, and reported the best performance. The technique with the best performance in the applied techniques column is highlighted in bold and italics. The best performance is presented to demonstrate the effectiveness of the technique used for that specific dataset or language identification problem. We identified the best performance based on the best achievement of the applied techniques in the reported literature. Other methods are also compared but not presented in the table. The best performance from each study is given based on the highest accuracy, F1 score, precision, or recall reported in the investigated papers. The discussion of each approach will be described in detail in the following subsections.

#### *a: MACHINE LEARNING APPROACH*

This section describes the utilisation of both supervised and unsupervised approaches. Among the 40 papers, we found that supervised approaches were used more often than unsupervised approaches. The supervised learning approach requires annotated training data as a model to predict output for the new data [67], [68].

SVM was the most widely used technique by researchers in language identification tasks. SVM is often implemented due to its capability to build an efficient classifier model and produce good performance [46]. From the selected studies, SVM has shown impressive performance. Veena et al. [46] utilised a linear kernel SVM classifier and could achieve an accuracy of 93% for word-level Malayalam-English and 95% for Tamil-English code-mixed LID. Chaitanya et al. [47] incorporated several machine learning methods with Word2Vec embedding for Hindi-English. Based on their experiments, the SVM using Skip-gram reached the highest accuracy of 67.34%. SVM and word embedding were also implemented by Sarma et al. [18] in their study. Their work demonstrated that the SVM using word embedding obtained better results than Naïve Bayes and Convolutional Neural Network (CNN) with an F1 score of 90.61%.

Kalita and Saharia [20] applied linear kernel SVM with N-gram and dictionary features to identify Assamese-English code-mixed language. They obtained 89.51% accuracy in word-level identification. Shanmugalingam et al. [50] presented that SVM with linear kernel performed the best with an accuracy of 89.46% for Tamil-English code-mixed LID. In Kazi et al. [58], they implemented the Support Vector Classifier (SVC), one of the SVM variants. The result showed that the SVM with RBF kernel and N-gram features obtained the best accuracy of 92%.

Code-mixed LID task can be categorised as a sequence tagging problem, and Conditional Random Fields (CRF) can be adopted to solve it. CRF is a statistical modelling method commonly utilised in sequence tagging problems, such as named entity tagging, POS tagging, and language identification. While an ordinary classifier may predict a label for a single sample without considering its neighbours, CRF can take context into account to make more accurate predictions [38]. In this literature study, we found that 8 of 14 papers utilised CRF with satisfying results.

Lamabam and Chakma [38] developed a code-mixed LID system for Manipuri-English using CRF with characters as features and achieved an F1 score of 90%. CRF method has been implemented by [41] to build a tweet-level and token-level LID of code-mixed text for Spanish-English and Arabic-Modern Standard Arabic (MSA). The result showed that CRF gave good results, with an F1 score of 83% for the tweet level and 94.9% in overall token-level accuracy.

In Phadte and Wagh [44], CRF outperformed SVM and Random Forest techniques with an accuracy of 94%. Gundapu and Mamidi [49] experimented with four machine learning approaches, namely Naïve Bayes, Random Forest, Hidden Markov Model (HMM), and CRF. Among these classifiers, the CRF presented the best accuracy of 91.28%. Yirmibeşoğlu and Eryiğit [52] obtained 95.6% micro-F1 using CRF with character-level N-grams. Mave et al. [15]

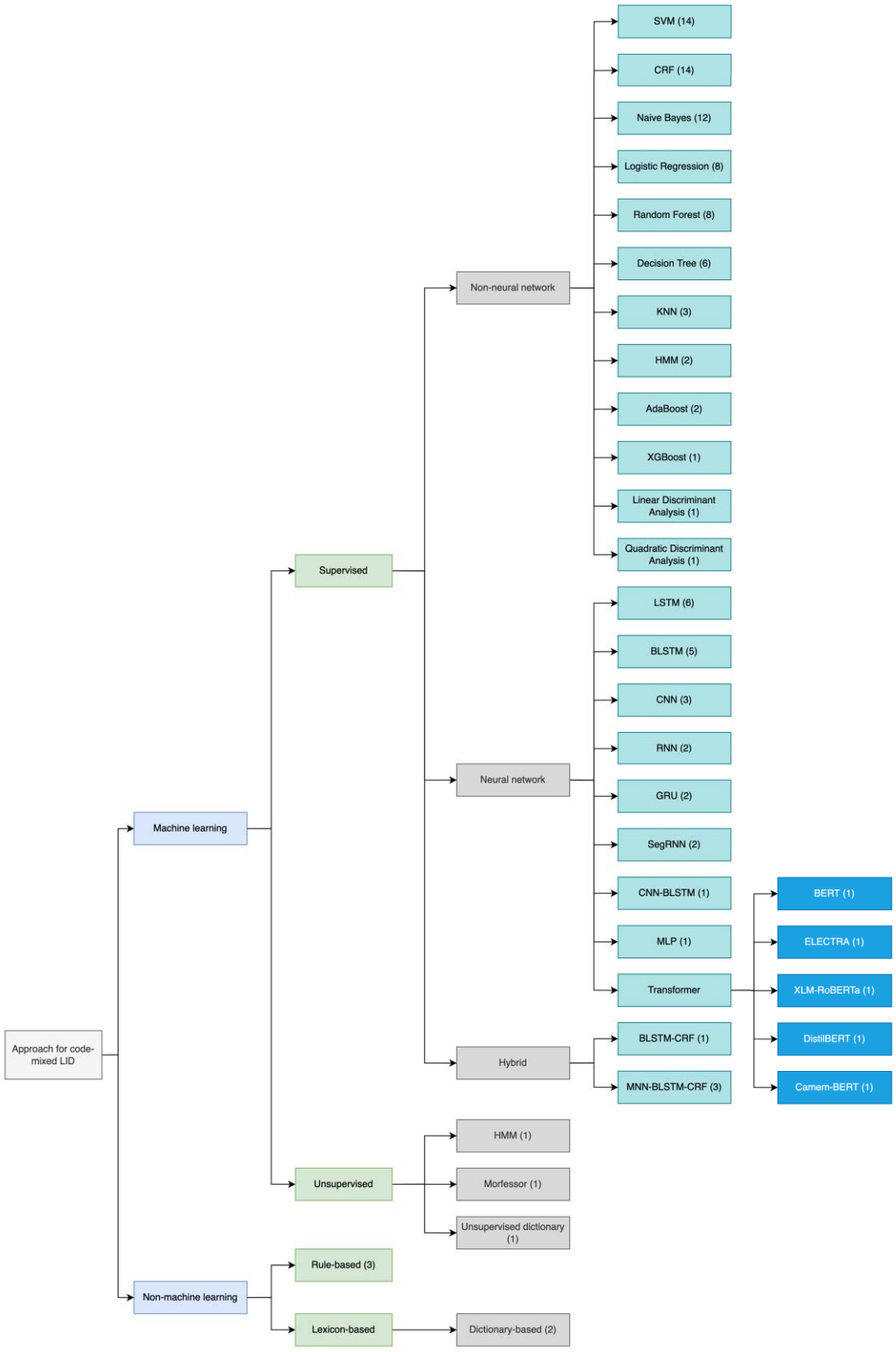


**FIGURE 2.** Taxonomy of applied techniques for code-mixed LID with their number of studies in bracket.

**TABLE 5.** Languages, applied techniques, LID level, and reported performance. The table is sorted by publication year, from 2016 to 2021.

| Reference | Year | Language | Applied Techniques | LID Level | Reported Best Performance (Accuracy/F1/Precision /Recall) |
|---|---|---|---|---|---|
| [36] | 2016 | Hindi-English, Bengali-Hindi-English, Gujarati-Hindi-English | Naïve Bayes | Word | Accuracy: 97.6% |
| [37] | 2016 | Spanish-English, Arabic-Modern Standard Arabic | CNN-BLSTM | Sentence, Word | F1: 95.1% (English) |
| [38] | 2016 | Manipuri-English | J48, ***CRF*** | Word | F1: 90% |
| [39] | 2016 | Swahili-English | SVM | Word | Accuracy: 99.3% |
| [40] | 2016 | Spanish-English, Arabic-Modern Standard Arabic | ***LSTM-CRF***, LSTM, CRF | Tweet, Word | Accuracy: 96.7%, F1: 90% (Spanish-English) |
| [41] | 2016 | Spanish-English | CRF | Tweet, Word | Accuracy: 94.9% |
| [42] | 2016 | Dutch-Limburgish | Morfessor (Morphological Segmentation) | Word, sub-word | Precision: 98.5%, recall: 75.7%. (Dutch); Precision: 95.6%, recall 78.8% (Limburgish) |
| [43] | 2017 | Kannada-English | Multinomial Naïve Bayes, ***Bernoulli Naïve Bayes***, SVM, Random Forest, Logistic Regression, CRF | Word | Accuracy: 94.8%, Precision: 96.3%, Recall: 95.2% |
| [44] | 2017 | Konkani-English | ***CRF***, SVM, Random Forest, Unsupervised-dictionary-based | Word | Accuracy: 94% |
| [45] | 2017 | Dutch-English, French-English, Portuguese-English, German-English, Turkish-English, Spanish-English, Dutch-Turkish | ***HMM***, Dictionary-based | Word | Accuracy: 98.5% (Turkish-English) |
| [46] | 2017 | Tamil-English, Malayalam-English | SVM | Word | Accuracy: 95.45%, F1: 94.77 % (Tamil-English) |
| [47] | 2018 | Hindi-English | ***SVM***, Random Forest, Logistic Regression, Gaussian Naïve Bayes, KNN, AdaBoost | Word | Accuracy: 67.34% |
| [48] | 2018 | Spanish-English, Dutch-English, Turkish-German | Dictionary-based | Word | F1: 96.3% and 98.3% (Spanish-English) |
| [49] | 2018 | Telugu-English | Naïve Bayes, Random Forest, ***CRF***, HMM | Word | Accuracy: 91.28% |
| [20] | 2018 | Assamese-English | SVM | Word | F1: 88.9% |
| [11] | 2018 | Bengali-English, Hindi-English | MNN-BLSTM-CRF | Word | Accuracy: 93.32% (Hindi-English) |
| [18] | 2018 | Assamese-Hindi-Bengali-English | SVM, Naïve Bayes, CNN | Sentence, Word | F1: 90.61% |
| [50] | 2018 | Tamil-English | Naïve Bayes, ***SVM***, Logistic Regression, Random Forest, Decision Tree | Word-level | Accuracy: 89.46% |
| [51] | 2018 | Hindi-English | RNN, GRU, ***LSTM***, CRF | Word | F1: 93.4%, Accuracy: 96.1% |
| [52] | 2018 | Turkish-English | CRF | Word | F1: 95.6% |
| [15] | 2018 | Hindi-English, Spanish-English | ***CRF***, BLSTM, LSTM | Word | F1: 98% |
| [53] | 2018 | Bengali-English, Hindi-English, Bengali-Hindi-English | ***CRF***, LSTM, BLSTM | Word | Accuracy: 91.54%, F1: 91.02% (Hindi-English) |
| [23] | 2019 | Bengali-English | SVM, ***LSTM*** | Word | Accuracy: 92.35% |
| [22] | 2019 | Dutch-English | Rule-based, ***Decision Tree***, SVM, AdaBoost, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Logistic regression, neural network | Word | F1: 95.37% |
| [7] | 2019 | Indonesian-English | CRF | Word | Accuracy: 90.11%, F1: 89.58% |
| [54] | 2019 | Spanish-Wixarika, Turkish-German | CRF, ***SegRNN***, BLSTM, BLSTM-CRF | Word, sub-word | F1: 98.7% (Turkish-German) |
| [55] | 2019 | French-Italian-Spanish-English | CRF | Word, sentence | Accuracy 97.77%, precision: 95%, recall: 95.5%, F1: 95%. |
| [3] | 2019 | Sinhala-English | SVM, Naïve Bayes, Logistic Regression, ***Random Forest***, Decision Tree | Word | Accuracy: 90.5% |

**TABLE 5.** *(Continued.)* **Languages, applied techniques, LID level, and reported performance. The table is sorted by publication year, from 2016 to 2021.**

| | | | | | |
|---|---|---|---|---|---|
| [56] | 2019 | Sinhala-English | XGB Classifier, ***CRF***, SVM, KNN, Random Forest, LSTM, GRU, RNN, CNN | Word, sentence | Accuracy: 92.1%, F1: 94% |
| [16] | 2020 | Punjabi-English | ***Logistic Regression***, Decision Tree, Gaussian Naïve Bayes | Word | Accuracy: 86.63%, F1: 88% |
| [57] | 2020 | Malay-English | Rule-based | Sentence | Accuracy: 88.11% |
| [58] | 2020 | Gujarati-Hindi-English | Multinomial Naïve Bayes, ***SVC***, Decision Tree, KNN, Logistic Regression, Random Forest | Sentence | Accuracy: 92% |
| [59] | 2020 | Gujarati-English | HMM, Naïve Bayes | Word | N/A |
| [60] | 2020 | Hindi-English | MNN-BLSTM-CRF | Word | F1: 93.97% |
| [61] | 2021 | Hindi-English | MNN-BLSTM-CRF | Word | F1: 93.97% |
| [62] | 2021 | Bodo-English | Decision Tree, ***Naïve Bayes***, Multilayer Perceptron | Word | F1: 65.9%, precision: 76.2%, recall: 69.3% |
| [63] | 2021 | English-Assamese-Hindi-Bengali | ***CNN***, BLSTM, SVM, Logistic Regression | Word | F1:91.03% |
| [64] | 2021 | Malayalam-English | XLM-Roberta, ***ELECTRA***, BERT, DistilBERT, Camem-BERT | Word | F1: 99.33%, Accuracy: 99.41%, Precision: 99.37%, Recall: 99.31% |
| [65] | 2021 | Arabic-English | Naïve Bayes, BLSTM, ***SegRNN*** | Word | F1: 94.84% |
| [66] | 2021 | Hindi-English | Rule-based | Word | F1: 87.99% |

found that the CRF model's performance presented better results than the deep learning models (LSTM and BLSTM). The result showed that they could provide an F1 score of 98% and 96% for the Hindi-English language pair. Barik et al. [7] demonstrated code-mixed LID for Indonesian-English using a small dataset from Twitter. In their work, the CRF obtained an 89.58% F1 score and an accuracy of 90.11%. Finally, in Mishra and Sharma [55], CRF accurately identified multilingual code-mixed with 97.77% accuracy and 95% F1 score.

Naïve Bayes was the third most common technique for code-mixed LID with 12 studies. It is often used as a baseline model due to its simplicity. It uses the Bayes probability theorem to forecast the class of unknown datasets and the model assumes no relationship exists between the input features [65]. The naïve Bayes algorithm has proven to perform well in several studies. Gupta et al. [36] utilised the supervised learning and edit distance method. The result showed that combining edit distance and Naïve Bayes on the N-gram Markov model could perform well, particularly when detecting language from misspelt words. Lakshmi and Shambhavi [43] employed two different Naïve Bayes algorithms, Multinomial and Bernoulli Naïve Bayes. The Bernoulli Naïve Bayes combined with TF-IDF, and dictionary module outperformed the other methods (SVM, Random Forest, and Logistic Regression) with accuracy, precision, and recall of 94.8%, 96.3%, and 95.2%, respectively. A study by Kalita et al. [62] showed that Naïve Bayes outperformed Decision Tree and Multilayer Perceptron, achieving F1 of 65.9%, precision of 76.2%, and recall of 69.3%.

Logistic Regression was applied in eight of the selected studies. Bansal et al. [16] used Logistic Regression for English-Punjabi code-mixed LID. Based on the experiments, Logistic Regression outperformed Decision Tree and Gaussian Naïve Bayes in word-level code-mixed LID with an accuracy of 86.63% and an F1 score of 88%.

Random Forest was also one of the popular machine learning techniques implemented in eight studies. Among these studies, Shanmugalingam and Sumathipala [3] revealed that Random Forest performed the best among the other machine learning techniques with an accuracy of 90.5% for word-level Sinhala-English code-mixed LID. Based on their experiments, the Random Forest model could identify Sinhala and English languages quite well, with an F-measure of 94.9% for Sinhala and 75.8% for English. However, the Random Forest model yielded unsatisfactory results with an F-measure of 51.3% for tokens other than Sinhala and English, such as named entities, acronyms, universal, mixed, and other language tags.

For the neural network-based, we found the following various neural-network techniques, such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and some variants of RNN, such as Long Short-Term Memories (LSTM), Gated Recurrent Unit (GRU), Bidirectional LSTM (BLSTM), and Segmental Recurrent Neural Network (SegRNN).

Mager et al. [54] and Sabty et al. [65] proposed SegRNN in their studies to address sub-word LID for intra-word code-switching. The SegRNN models appear to perform better on language pairs with more intra-word CS, whereas pipeline approaches may perform equally well on language pairs with fewer mixed words [54]. In [65], SegRNN provided high results compared to Naïve Bayes and BLSTM. The SegRNN obtained a 94.84% F1 score for intra-word labelling and 99.17% for segmentation.

Another RNN variation technique, LSTM, has shown satisfactory performance in identifying Hindi-English and Bengali-English code-mixed text [23], [51], [53]. In [51], the LSTM architecture could give a high average F1 score of 93.4% and an average accuracy of 96.1% across the three classes. LSTM with pre-trained word embeddings outperformed CRF and BLSTM in Bengali-English code-mixed LID [53]. Samih et al. [40] experimented with LSTM and CRF to improve the LID performance. The results showed that integrating character and word representation with a char-word LSTM and adding CRF produced the highest overall accuracy of 96.3%.

In text processing problems, CNN is often implemented to extract text features before applying a machine learning algorithm. The convolution layer in CNN building blocks extracts text features by applying a convolutional filter or kernel to each window in the sequence of text. Sarma et al. [63] experimented with CNN and BLSTM, and CNN showed the best performance among the other techniques with an F1 score of 91.03%. Some studies combined two ANN modules, CNN and LSTM or CNN and BLSTM, in their neural network architecture.

Jaech et al. [37] incorporated the CNN and BLSTM for word-level LID in Spanish-English code-mixed text. In their architecture, the convolutional layer provides word vectors from the input characters by transforming them into vectors. Next, the BLSTM maps the word vector sequence to a language tag. BLSTM was selected due to its capability to capture long sequence dependencies. In BLSTM, the context of the observed sequence will be considered during the word-level identification process. The result achieved an F1 score of 95.1% for English and 94.1% for Spanish.

We encountered three research by [11], [60], and [61] for the hybrid technique that combined the non-neural network and neural network techniques. They proposed similar architecture consisting of two modules: a multichannel neural network (MNN) and BLSTM-CRF. The MNN comprises three one-dimension convolution layers and one LSTM layer. One-dimension convolution layer cells were used to capture the N-gram representation of the input text. Additionally, the BLSTM-CRF module aims to capture the context of the input text.

Mandal and Singh [11] experimented on two code-mixed data (Bengali-English and Hindi-English). They implemented 2, 3, and 4 kernels in their multichannel architecture to seize the N-gram representations. Their study revealed that the combination of multichannel and BLSTM-CRF achieved an accuracy of 93.28% and 93.32% for Bengali-English and Hindi-English severally. Another work by [60] identified Hindi-English code-mixed text from social media platforms. In their work, they gained the best F1 score of 93.97%. Gupta et al. [61] implemented 3-gram word embedding in their MNN-BLSTM-CRF architecture. In their work, they acquired the best result with an F1 score of 93.97%.

With the advent of the transformers-based technique proposed by Vaswani et al. [69], the transformers have quickly become a popular and reliable technique for natural language processing, outperforming the prior neural networks such as CNN and RNN [70]. Recent work by Thara and Poornachandran [64] presented a transformer-based technique called Bidirectional Encoder Representations from Transformers (BERT). The authors built a word-level code-mixed LID system in Malayalam-English. They experimented with five BERT model architectures; BERT, DistilBERT, ELECTRA, XLM-RoBERTa, and CamemBERT. Overall, ELECTRA performed the best, with an F1 score of 99.33% and an accuracy of 99.41%.

Moreover, three out of 40 selected studies were found to be using the unsupervised approach. Nguyen and Cornips [42] carried out a study on Dutch-Limburgish using an unsupervised morphological segmentation approach. They utilised Morfessor tools to analyse text by slicing the words into smaller units.

Phadte and Wagh [44] built a word-level LID for Konkani-English text by applying an unsupervised approach with dictionaries. They compared the unsupervised dictionary-based technique with other supervised LID, such as CRF, SVM, and Random Forest. The supervised approach performed better than the unsupervised dictionary-based technique based on the experiments. Rijhwani et al. [45] presented an unsupervised word-level code-mixed LID for seven different language pairs without manual data annotation for the training process. The automatic annotation process was carried out using the Hidden Markov Model (HMM) and dictionary as baselines. In their study, the HMM was implemented in an unsupervised manner.

#### *b: NON-MACHINE LEARNING*

Kent and Claeser [22] incorporated a dictionary-based approach in a rule-based code-mixed LID system. They mentioned that there would be many word misclassification if the system relied on a basic dictionary without adding rules. Kasmuri and Basiron [57] proposed a rule-based approach, and their research focused on distinguishing between code-switching and monolingual sentences in an English-Malay dataset. The rule-based approach was used with five dictionaries as look-up tools to identify the language in their work. The rule-based solution employed a ratio of word presence in a phrase with a 90% threshold for monolingual communication and various codes-switching ratios. The study has shown that the rule-based technique produced a good performance with more than 87% accuracy. Nguyen et al. [66] employed a rule-based approach for Hindi-English code-switched LID. Their study utilised a word list for each language to help identify the language for each token.

Although machine learning techniques are widely utilised, we found some studies that applied lexicon-based using dictionaries to solve code-mixed LID. From the examined papers, we encountered one study that applied a dictionary-based technique. Claeser et al. [48] proposed a lexicon-based classification using Wikipedia as a dictionary to identify code-switching from Twitter data. Their dictionary-based

technique has correctly identified the language of word sequences and abbreviated words, such as *'jajaja'* and *'omg'*. However, the system could not determine some irregular tokens.

In our investigation, we found an issue of ambiguity in the dictionary look-up technique. For instance, the language tag is sometimes incorrect when a particular word exists in more than one dictionary [36]. Some of the selected studies utilised dictionaries together with machine learning techniques. In the previous studies, the dictionaries were employed as features. The discussion regarding dictionaries as features will be presented in the features section.

#### 2) FEATURES

Selecting appropriate features is crucial for enhancing the performance of code-mixed LID systems [3]. Feature extraction enables us to generate more accurate data, which the model will produce good results. We listed some essential features used by researchers for code-mixed LID, such as N-gram, word embeddings, and dictionary features. Based on the reviewed literature, most studies implemented more than one type of feature to solve code-mixed LID tasks.

##### *a: N-GRAM*

N-gram was used as a feature in 16 out of 40 studies. We found two different N-gram techniques being applied from the selected studies: word or token N-gram and character N-gram. Word or token N-gram has been used by [20], [45], and [62]. In the word-level code-mixed LID task, the character N-gram is more popular than the word N-gram, especially for identifying the language in code-mixed script [15], [16], [58]. In Piergallini et al. [39], it was observed that the use of the Swahili regular expression with the character N-gram was redundant. They suggested utilising N-gram features and capitalisation features to improve the LID performance. The capitalisation feature identifies capitalised initial letters and whether they occurred at the beginning of a sentence. Veena et al. [46] stated that an adequate number of embedding data could be sufficiently applied to develop word-level features for code-mixed LID. Additionally, a few studies observed that the character N-gram successfully solved LID for code-mixed text [52], [56].

##### *b: WORD EMBEDDING*

The word embedding or word vector representation technique could help represent word similarity and context from texts. Xia [41] trained sub-word information of the input text using enhanced skip-gram word vector models. An experiment by [47] has proven that the skip-gram could improve the performance of their LID system. Sarma et al. [18] employed word embedding to help detect the language of a new word. In Jamatia et al. [53], the pre-trained word embeddings could improve the performance of the proposed code-mixed LID system. Another study by Shekhar et al. [60] and Gupta et al. [61] demonstrated that word embedding performed better than the character embedding technique. The word embeddings can identify language separation by detecting the word origin and mapping it to the correct language label.

##### *c: CHARACTER EMBEDDING*

Character embedding is an embedding feature vector generated by splitting words into characters [46]. Applying character embedding for code-mixing LID can capture the morphological features of the words and make them more sensitive toward out-of-vocabulary problems [15], [71]. Mandal and Singh [11] employed character embeddings of length 15 fed into the multichannel neural network layer. In [46], the vector size of character embedding was set to 100. Mave et al. [15] combined character and word embedding representations by applying two LSTM layers. The LSTM layers were used to train fixed-dimensional representations from the embedding layers. In [60] and [61], character embedding was also employed with word embedding.

##### *d: TF-IDF (TERM FREQUENCY-INVERSED DOCUMENT FREQUENCY)*

TF-IDF is the most advanced count vectorizer technique to convert text data into a form of vector as an input to the classifier [58]. In natural language processing, TF-IDF is frequently applied with N-gram features. Smith and Thayasivam [56] trained their model by employing TF-IDF into several types, word-level TF-IDF, character N-gram TF-IDF, and N-gram TF-IDF. Mishra and Sharma [55] adopted TF-IDF to model the context of the sentence in a particular discussion. However, the TF-IDF feature has less impact on performance than N-gram features [58].

##### *e: DICTIONARY FEATURES*

Treating the dictionary as a feature has been an effective method applied in code-mixed LID. Piergallini et al. [39] utilised English and Swahili dictionary features combined with two other features: capitalisation and regular expression features. Kalita and Saharia [20] incorporated three different dictionaries with N-gram features. Bansal et al. [16] used dictionaries as features to express the presence of words.

To summarise the RQ1, we have demonstrated a taxonomy of applied techniques for code-mixed LID, highlighting the different technique variants. We identified that machine learning, mainly supervised approaches, is more widely used than the non-machine learning approaches to solving code-mixed LID problems. SVM and CRF are the most popular and recommended non-neural network techniques. For the neural network-based technique, the Multichannel CNN-BLSTM-CRF has proven excellent performance. Due to its impressive performance, we came across the transformer-based technique as a robust technique for code-mixed LID. In terms of applied features, we obtained four crucial features from the reviewed primary studies: N-gram, word embedding, dictionary features, and TF-IDF.

### B. (RQ2) WHAT ARE THE CHALLENGES IN LID OF CODE-MIXED TEXT?

Detecting mixed-language text is a hot topic in natural language processing research. Most existing language detectors do not identify mixed-language texts. Identifying multiple languages in code-mixed text requires different techniques from multiple languages applied to a document [20]. Moreover, code-mixed text that uses various levels of combinations (sentence, clause, word, and sub-word) makes LID more complicated [15] than text expressed in one language. Using dictionary look-up, identifying language from code-mixed text has shown poor performance due to spelling variations, losing the context of the words, and failure to differentiate some borrowed words [22], [72]. Accordingly, conducting LID in the code-mixed text is more challenging than in the non-code-mixed text.

#### 1) AMBIGUITY

The ambiguity existed in several LID of code-mixed text studies, for example, Punjabi-English [16], Hindi-English [15], [36], [61], Malayalam-English [64], Bengali-English [36], Gujarati-English [36], [59], Spanish-English [15], [37], [41], [73], Dutch-English [22], Turkish-German & Spanish-Wixarika [54], Modern Standard Arabic-Arabic [41], [73], Konkani-English [44], Swahili-English [39], English-Assamese-Hindi-Bengali [63], Sinhala-English [56]. The annotation of mixed languages becomes increasingly complicated when the languages are closely related [16].

For bilingual code-mixed, addressing ambiguity is challenging in language annotation when a particular word exists in two or more languages. This phenomenon has two conditions: a single word with the same meaning and a single word with different meanings in two or more languages. A problem exists if the word has a different meaning for two or more languages because the meaning would be different based on the language identified by the system. Moreover, language ambiguity makes trilingual code-mixed text more challenging to identify than bilingual code-mixed text [53]. Table 6 shows examples of word ambiguity for several language pairs.

**TABLE 6. Example of ambiguity between languages.**

| Language Pair | Word | Meaning |
|---|---|---|
| Punjabi-English [16] | Sat | Punjabi: greeting<br>English: V2 of sit, or Saturday in short |
| Gujarati-English [59] | Mate | Gujarati: for<br>English: partner |
| Bengali-English [23] | Choke | Bengali: eyes<br>English: have severe difficulty in breathing because of a constricted or obstructed throat or a lack of air |
| Hindi-English [11] | Age | Hindi: ahead<br>English: the length of time that a person has lived |
| Swahili-English [39] | Wake | Swahili: hers, Its (possessive), his/her<br>English: emerge or cause to emerge from a state of sleep |

#### 2) LEXICAL BORROWING

Another challenge related to ambiguity is lexical borrowing [63]. Lexical borrowing is defined as transferring or copying a particular lexical item from one language to the lexicon of another language [22], [74]. We identified two examples of lexical borrowing words in Dutch-English and Hindi-English language pairs. For example, the word *sociaal* (Dutch) and *social* (English) [22]; *pajama* (Hindi) and *pyjama* (English) [11]. It can be seen from the examples that the words have almost similar spelling. In this case, the LID system may identify similar words to the correct language tag for words with phonetic similarities but different spellings.

However, an issue arises when the words have the exact lexicon similarity. Due to such lexicon similarity, it is difficult for a language detection system to distinguish between code-switching and borrowing words. The exact similarity in the spelling of a particular word means that the word is valid in multiple languages. Therefore, the correct tag of the word will depend on the context, which is the other surrounding words. For instance, the word *'school'* is valid in Dutch and English [22]. In Twitter, the system will detect the word *'school'* as Dutch instead of English if it is surrounded by Dutch words and vice versa.

#### 3) NON-STANDARD WORDS

Non-standard words are quite common in social media texts due to the informal use of the language. In the following, we identified some non-standard words from the investigated papers. We categorised the non-standard words into four types, such as non-standard spelling [7], [15], [56], abbreviated words [3], [36], [38], [44], [48], [56], [64], exaggerated words [3], [7], [23], [38], [44], [46], [48], [49], [50], [64], and mixing characters with numbers or special characters [3], [23], [38], [49]. Table 7 describes some examples of non-standard words found in code-mixed text LID.

**TABLE 7. Non-standard word examples in code-mixed text LID.**

| Type of Non-standard Word | Example |
|---|---|
| Non-standard spelling [7, 56] | *Prends* or *prenzz* (friends), *plis* (please), *kalo* for *'kalau'* (Indonesian language, meaning 'if' in English) |
| Mixing word and numeric or special characters [3, 7, 23, 38, 49] | *ri8* (right), *2morrow* (tomorrow), *ni8t* (night), *orang2* (Indonesian language, meaning people in English) |
| Word exaggeration [3, 7, 23, 38, 44, 46, 48-50, 64] | *goood* (good), *Pleasssseee* (please), *cooool* (cool), *helloooo* (hello) |
| Abbreviated words [3, 38, 44, 48, 56, 64] | *bght* (brought or bought), *tkt* (ticket), *flm* (film), TC (take care) |

#### 4) INTRA-WORD CODE-MIXING

Since word-level code-mixed LID becomes a common task, determining code-mixed at the sub-word level is a

more demanding task. Only a few studies have addressed intra-word code-mixing issues [65]. Intra-word code-mixing occurs when speakers incorporate languages within a token or word [38], [54]. This happens when a prefix or suffix from one language is added to another language. Table 8 provides examples of intra-word code-mixing.

**TABLE 8. Intra-word code-mixing examples.**

| Language | Example |
|---|---|
| Indonesian (ID)-English (EN) [7] | *ngevote → nge (ID) + vote (EN)*<br>*jobnya → job (EN) + nya (ID)* |
| Assamese (AS)-English (EN) [20] | *servicetu → service (EN) + tu (AS)* |
| Bodo (BD)-English (EN) [62] | *publickho → public (EN) + kho (BD)* |
| Turkish (TR)-German (GE) [48] | *schatzim → schatz (GE) + im (TR)* |
| Spanish (SP)-Wixarika (WX) [54] | *pecansadoxi → pe (WX) + cansado (SP) + xi (WX)* |
| Kannada (KN)-English (EN) [43] | *plangalannu → plan (EN) + galannu (KN)* |
| Telugu (TL)-English (EN) [49] | *classlo → class (EN) + lo (TL)* |

In the Indonesian language, people sometimes add to an English word an informal prefix *(nge-)* in a verb or suffix *(-nya)* in a noun [7]. Similar examples can also be found in other language pairs, such as Assamese-English [20], Bodo-English [62], Spanish-Wixarika [54], Dutch-Limburgish-English [42], Turkish-English [52], Turkish-German [48], Telugu-English [49], Kannada-English [43], and Manipuri-English [38].

To sum up the RQ2, we have identified four main challenges often found in code-mixed LID tasks: ambiguity, lexical borrowing, non-standard words, and intra-word code-mixing. These challenges are prevalent in social media text and becoming a problem in code-mixed LID.

Ambiguity happens when a particular word or token is recognised in two or more languages. There are two issues relating to ambiguity, a word with a similar meaning and a word with a different meaning for two or more languages. Another challenge identified is lexical borrowing. In this study, the problem of lexical borrowing arises when a word has the exact spelling. For non-standard words, four challenges have been identified: non-standard spelling, mixing between word and numeric or special characters, word exaggeration, and abbreviated words. The last challenge is intra-word code-mixing which occurs when two or more languages are mixed in a word or token.

### C. (RQ3) WHAT DATASETS ARE AVAILABLE FOR CODE-MIXED TEXT? WHAT ARE THE QUALITY CRITERIA FOR THE DATASET?

#### 1) DATASET AVAILABILITY

In this section, we analysed the code-mixed dataset based on four perspectives: (1) the number of mixed languages in the datasets, considering only datasets that are bilingual or multilingual, (2) the datasets that are English code-mixed or non-English code-mixed, (3) the language family combination, and (4) source of datasets.

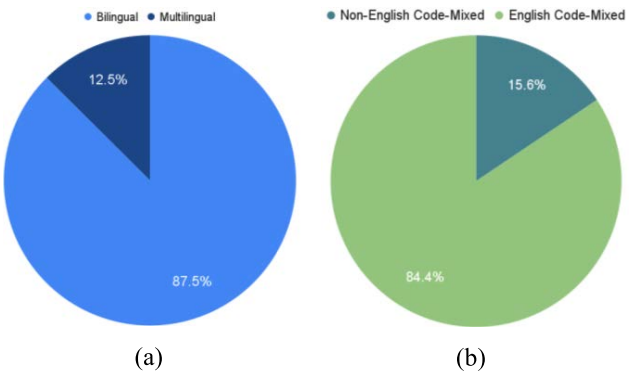


**FIGURE 3. Data distribution: (a) bilingual and multilingual; (b) English and Non-English.**

Figure 3 illustrates the data distribution from the selected studies. From the left side (a), bilingual code-mixed data dominates with 87.5% (28 datasets), while the percentage of multilingual data is only 12.5% (4 datasets). The multilingual code-mixed data include Bengali-Hindi-English, English-Assamese-Hindi-Bengali, English-French-Italian-Spanish, and Gujarati-Hindi-English. As shown in Figure 3 (b), the ratio between English code-mixed and non-English code-mixed is 84.3% (27 datasets) and 15.6% (5 datasets), respectively. All non-English code-mixed data are bilingual, and these data are of the language pairs; Arabic-Modern Standard Arabic, Dutch-Limburgish, Dutch-Turkish, Spanish-Wixarika, and Turkish-German.

Among the 32 datasets, the Hindi-English was the most frequent language pair with 9 studies. Spanish-English is the second most studied language pair with 5 studies, followed by Bengali-English and Dutch-English with three studies. Two studies each focused on the following mixed languages: Turkish-English, Turkish-German, Malayalam-English, Tamil-English, Assamese-Hindi Bengali-English, Sinhala-English, and Arabic-Modern Standard Arabic.

We also grouped the available code-mixed dataset based on the language family combination. To identify the language family, we referred to a study conducted by [75]. Overall, we found 12 language family combinations as follows: Austronesian & Germanic, Dravidian & Germanic, Germanic & Trans Eurasian, Germanic & Germanic, Indo-Aryan & Germanic, Italic & Germanic, Italic & American, Niger-Congo & Germanic, Semitic & Germanic, Semitic & Semitic, Sino-Tibetan & Germanic, and Trans Eurasian & Germanic. Most of the code-mixed data were combined with English, which belonged to the Germanic language family. Germanic is a part of Indo-European languages and is mainly spoken in the north of Europe, such as in England, Germany, and the Netherlands [76].

The most studied language family was the combination between Indo-Aryan and Germanic with ten language mix

combinations. Indo-Aryan is a branch of the Indo-European language spoken mainly by people in South Asia [77]. The Indo-Aryan language family consists of Assamese, Bengali, Gujarati, Hindi, Konkani, Punjabi, and Sinhala. The Dravidian and Italic mixed with Germanic language families were the second most studied in the dataset with four language mix combinations. From the investigation, we found some datasets categorised as part of the Dravidian language family, such as Kannada, Malayalam, Tamil, and Telugu language [77].

As for the Italic language family, we acquired French, Italian, Spanish, and Portuguese. We identified three mixed language combinations belonging to the combination of the Germanic language family: Dutch-English, Dutch-Limburgish, and German-English. In the Austronesian language family, we discovered Indonesian and Malay intermingled with English. We found two languages, Bodo and Manipuri, which are classified as the Sino-Tibetan language family. In terms of the Trans Eurasian family, we found Turkish, which was mixed with English and German. We also identified one language family combination for Germanic & Trans Eurasian, Italic & American, Niger-Congo & Germanic, Semitic & Germanic, and Semitic & Semitic. Table 9 presents the code-mixed dataset grouped by the language family combination.

We encountered eight unique data sources from the inspected 32 datasets, such as Twitter, Facebook, WhatsApp, YouTube comments, chat messages, blogs, frequently asked questions data, and interviews and internet forums. Twitter is the most used platform with 24 studies, followed by Facebook (21 studies), WhatsApp (8 studies), and YouTube comments (2 studies). Chat messages, blogs, FAQ data, interviews and internet forums were utilised in one study, respectively. Table 10 shows the source of code-mixed LID datasets from the investigated papers.

### 2) CODE-MIXED DATASET QUALITY CRITERIA

Good quality data is necessary for conducting research. We attempted to determine the properties representing the quality of code-mixed data. To identify the quality criteria of the dataset, we applied the study by Jose et al. [2]. A set of items was defined, including the number of instances, percentage of code-mixed data, number of tokens, number of unique tokens, and average sentence length.

The number of instances and the number of tokens indicate the size of the corpus. The number of words provides further insight into the corpus's structure, especially for language tagging tasks, such as identification, named entity recognition, and POS tagging. The percentage of code-mixed data shows the diversity of code-mixed, code-mixed types, and the ratio of the types in the entire dataset. The quantity of unique tokens represents the vocabulary size of the dataset. This allows us to discern the richness of text in the data. Finally, the average sentence length indicates completeness and grammatical complexity since the longer the sentence, the more complicated the syntactic and semantic structure [2].

**TABLE 9. Code-mixed dataset availability grouped by the language family combination.**

| Language Family Combination | Language Data |
|---|---|
| Austronesian & Germanic | Indonesian-English [7] |
| | Malay-English [57] |
| Dravidian & Germanic | Kannada-English [43] |
| | Malayalam-English [46, 64] |
| | Tamil-English [46, 50] |
| | Telugu-English [49] |
| Germanic & Trans Eurasian | Dutch-Turkish [45] |
| Germanic & Germanic | Dutch-English [22, 45, 48] |
| | Dutch-Limburgish [42] |
| | German-English [45] |
| Indo-Aryan & Germanic | Assamese-English [20] |
| | Assamese-Hindi-Bengali-English [18, 63] |
| | Bengali-English [11, 23, 53] |
| | Bengali-Hindi-English [53] |
| | Gujarati-English [59] |
| | Gujarati-Hindi-English [58] |
| | Hindi-English [11, 15, 36, 47, 51, 53, 60, 61, 66] |
| | Konkani-English [44] |
| | Punjabi-English [16] |
| | Sinhala-English [3, 56] |
| Italic & Germanic | French-English [45] |
| | French-Italian-Spanish-English [55] |
| | Portuguese-English [45] |
| | Spanish-English [15, 37, 40, 45, 48] |
| Italic & American | Spanish-Wixarika [54] |
| Niger-Congo & Germanic | Swahili-English [39] |
| Semitic & Germanic | Arabic-English [65] |
| Semitic & Semitic | Arabic-Modern Standard Arabic [37, 40] |
| Sino-Tibetan & Germanic | Bodo-English [62] |
| | Manipuri-English [38] |
| Trans Eurasian & Germanic | Turkish-English [45, 52] |
| | Turkish-German [48, 54] |

From the examined studies, we found that all the studies described the number of instances except three studies: Bodo-English [62], Kannada-English [43], and Punjabi-English [16]. The number of tokens is presented by 23 out of 32 datasets. This high ratio indicated the importance of information regarding the number of instances and tokens from a particular dataset. Eleven studies presented a percentage of code-mixed from the dataset. We observed

**TABLE 10. Source of code-mixed LID datasets.**

| Data Source | Reference |
|---|---|
| Facebook | [3, 15, 16, 18, 20, 23, 43, 44, 46, 47, 49, 50, 53, 54, 56, 60-63, 65, 66] |
| Twitter | [7, 11, 15, 16, 22, 23, 37, 38, 40-43, 45, 47-49, 51-55, 60, 61, 65, 66] |
| YouTube comments | [58, 64] |
| Chat messages | [59] |
| Blogs | [57] |
| FAQ (Frequently Asked Questions) dataset | [36] |
| Interviews and internet forums | [39] |
| WhatsApp | [23, 47, 49, 53, 60, 61, 65, 66] |

that the number of unique tokens was reported in 6 code-mixed datasets from 5 papers, such as Assamese-English [20], Bodo-English [62], Hindi-English [15], Sinhala-English ()[3], Spanish-English [15], and Tamil-English [50]. Moreover, papers by [42] and [59] reported the average sentence length in their study. Table 11 provides the quality criteria of the datasets.

Further, 2 out of 32 datasets fulfil 4 out of 5 criteria. Kalita and Saharia [20] described their Assamese-English with 1,012 instances, 227,329 tokens, and 5,977 unique tokens. They presented the percentage of the code-mixed based on three groups: intra-sentential (26.69%), inter-sentential (69.26%), and intra-word (4.05%). Mave et al. [15] provided four criteria for Spanish-English and Hindi-English language pairs. Unfortunately, both studies by [20] and [15] did not report the information regarding the average sentence length of their dataset.

Two main aspects have been discussed in the RQ3, dataset availability and dataset quality criteria. This literature review study has recognised 32 code-mixed datasets available for LID. The bilingual code-mixing datasets dominate 87.5%, and only a few multilingual datasets are available for code-mixed text. In addition, we discovered that 84.38% of datasets are mixed with English. This finding is acceptable since English is an international language and has become an integral part of the education system in many countries. Moreover, 10 of 32 datasets (31.25%) were the Indo-Aryan language families combined with English.

We also proposed five features to describe the quality criteria dataset. The features are the number of instances or sentences, percentage of code-mixed types in the data, number of tokens, number of unique tokens, and average sentence length. Those five items can be used as a standard criterion for researchers to build an excellent quality of the created dataset for future research.

### *D. (RQ4) WHAT IS THE STANDARD WORKFLOW FOR LANGUAGE IDENTIFICATION OF CODE-MIXED TEXT FOR FUTURE RESEARCH?*

We unified the methodologies studied through our literature analysis and proposed a framework for researchers to use. The framework consists of two parts, model development and a code-mixed LID system. The model development generates a classification model. The code-mixed LID system predicts language labels from the input. The model development is divided into seven stages: data collection, pre-processing, data annotation, quality criteria assessment, feature extraction, classification modelling, and evaluation. In the data collection stage, the language pair of interest is chosen. The code-mixed data is gathered by defining keywords or topics to search data from various sources, such as reviews, chats, social media, and speech transcription. The collected data is then stored in a storage.

Subsequently, data pre-processing tasks are carried out by removing duplicates or irrelevant data. The tokenisation task is then conducted by splitting text data (sentences, tweets, comments, or documents) into words. Moreover, case-folding can be applied to convert the words into the same case form, like lowercase.

The next stage is data annotation. Data annotation is one of the essential processing tasks in a language identification system [16]. Before annotating the data, we must first define the labels for the dataset. The annotation process can be done manually [65], [78] or semi-automatically. A shared task and crowdsourcing are the most common methods for manual data annotation. The semi-automated method combines manual annotation with a dictionary or machine learning techniques. Before moving on to the next stage, we must evaluate the data to ensure that our labelled data is valid. In addition, a quality criteria assessment is conducted in this stage to provide the excellent quality of the dataset.

Feature extraction is performed to convert the text into numerical representations. The converted numerical data is used as input for the data modelling task. Several techniques for feature extraction can be used at this stage, such as N-gram [20], [45], [62], term frequency-inverse document frequency (TF-IDF) [49], [58], word embedding [47], character embedding [11], [15], [46], [60], [61], word to vectors (Word2Vec) [47], and global vector (GloVe) [47], [55].

The transformed texts are then processed in the subsequent stage, classification modelling. In our framework, the code-mixed LID is a classification problem where every word is labelled to its corresponding language tag [78]. The classification modelling process aims to derive conclusions from the training data and predict the class label. Training, validation, and testing sets are sampled from the dataset in this stage. The training set is a subset of data fed into any machine learning or deep learning algorithm to uncover the dataset's hidden patterns. The validation set assesses the trained model, and the results from validation are used for fine-tuning until the best result is achieved. The model's performance is then determined by evaluating the best result on the testing set. We can use evaluation metrics, such as accuracy, precision, recall, and F-score, to assess the model's performance.

Finally, the best model generated from the classification modelling stage is used as a classification model for the code-mixed LID system. The system receives user input in a word,

**TABLE 11.** Code-mixed datasets with their quality criteria.

| Languages | Reference | Instances | Percentage of Code-Mixed | Tokens | Unique Tokens | Average Sentence Length |
|---|---|---|---|---|---|---|
| Arabic-Modern Standard Arabic | [37, 40] | 11,241 | N/A | 227,329 | N/A | N/A |
| Arabic-English | [65] | 2,507 | N/A | 30,321 | N/A | N/A |
| Assamese-English | [20] | 1,012 | Intra-sentential: 26.69% Inter-sentential: 69.26% Intra-word: 4.05% | 26,444 | 5,977 | N/A |
| Assamese-Hindi-Bengali-English | [18, 63] | 28,968 | N/A | N/A | N/A | N/A |
| Bengali-English | [11, 23, 53] | 15,264 | N/A | N/A | N/A | N/A |
| | [53] | 6,685 | N/A | 86,739 | N/A | N/A |
| | [11] | 3,000 | N/A | N/A | N/A | N/A |
| Bengali-Hindi-English | [53] | 13,421 | N/A | N/A | N/A | N/A |
| Bodo-English | [62] | N/A | Intra-sentential: 31.59% Inter-sentential: 65.15% Intra-word: 3.36% | 15,022 | 6,602 | N/A |
| Dutch-English | [48] | 1,284 | N/A | 16,050 | N/A | N/A |
| | [22] | 1,250 | N/A | 18,250 | N/A | N/A |
| | [45] | 115 | N/A | 1,351 | N/A | N/A |
| Dutch-Limburgish | [42] | 10,434 | N/A | N/A | N/A | 2.664 |
| Dutch-Turkish | [45] | 1,463 | N/A | 27,918 | N/A | N/A |
| French-English | [45] | 98 | N/A | 1,249 | N/A | N/A |
| French-Italian-Spanish-English | [55] | 387 | N/A | N/A | N/A | N/A |
| German-English | [45] | 99 | N/A | 1,461 | N/A | N/A |
| Gujarati-English | [59] | 1,200 | N/A | 8,210 | N/A | 8-10 |
| Gujarati-Hindi-English | [58] | 6,324 | N/A | 41,075 | N/A | N/A |
| Hindi-English | [36] | 1,000 | N/A | N/A | N/A | N/A |
| | [47] | 3,000 | N/A | N/A | N/A | N/A |
| | [53] | 6,736 | 0.11% | 256,519 | N/A | N/A |
| | [11] | 3,000 | N/A | N/A | N/A | N/A |
| | [51] | 2,079 | N/A | 35,374 | N/A | N/A |
| | [15] | 7,421 | 43.62% | 146,722 | 23,998 | N/A |
| | [60] | 11,740 | Mixed words: 0.08%, Ambiguous: 0.27% | N/A | N/A | N/A |
| | [61] | 3,071 | N/A | 46,276 | N/A | N/A |
| | [66] | 41,144 | N/A | N/A | N/A | N/A |
| Indonesian-English | [7] | 825 | N/A | N/A | N/A | N/A |
| Kannada-English | [43] | N/A | N/A | 12,750 | N/A | N/A |
| Konkani-English | [44] | 34,036 | 14.94% | 60,118 | N/A | N/A |
| Malay-English | [57] | 6,543 | 46% | N/A | N/A | N/A |
| Malayalam-English | [46] | 3,000 | N/A | 31,803 | N/A | N/A |
| | [64] | 50,000 | 3.2% | 775,430 | N/A | N/A |
| Manipuri-English | [38] | 1,000 | N/A | N/A | N/A | N/A |
| Portuguese-English | [45] | 106 | N/A | 1,496 | N/A | N/A |
| Punjabi-English | [16] | N/A | N/A | 40,000 | N/A | N/A |
| Sinhala-English | [3] | 1,500 | N/A | 9,797 | 4,475 | N/A |
| | [56] | 1,900 | N/A | 11,795 | N/A | N/A |
| Spanish-English | [37] | 144,000 | N/A | N/A | N/A | N/A |
| | [40] | 28,827 | N/A | N/A | N/A | N/A |

**TABLE 11.** *(Continued.)* **Code-mixed datasets with their quality criteria.**

| | | | | | | |
|---|---|---|---|---|---|---|
| | [45] | 3,065 | N/A | 25,616 | N/A | N/A |
| | [48] | 1,028 | N/A | 7,133 | N/A | N/A |
| | [15] | 25,130 | 34.75 | 294,261 | 35,153 | N/A |
| Spanish-Wixarika | [54] | 985 | 16.55% | 8,000 | N/A | N/A |
| Swahili-English | [39] | 230,539 | N/A | N/A | N/A | N/A |
| Tamil-English | [46] | 3,000 | N/A | 35,693 | N/A | N/A |
| | [50] | 3,000 | N/A | 25,516 | 13,058 | N/A |
| Telugu-English | [49] | 1,987 | N/A | 29,503 | N/A | N/A |
| Turkish-English | [45] | 100 | N/A | 1,130 | N/A | N/A |
| | [52] | 391 | N/A | 5,430 | N/A | N/A |
| Turkish-German | [48] | 145 | N/A | 1,720 | N/A | N/A |
| | [54] | 1,029 | 15.66% | 17,000 | N/A | N/A |

sentence, paragraph, or document. The input is tokenised and transformed before it is fed into the classifier model. Tokens with the corresponding predicted labels are the system's output. Figure 4 depicts a comprehensive picture of the framework for code-mixed LID.

## IV. IMPLICATIONS

### A. THEORETICAL IMPLICATIONS

This systematic review contributes to the theoretical advancement of the code-mixed LID. First, this study identifies the techniques utilised to solve code-mixed LID problems. For the non-neural network techniques, SVM and CRF algorithms are the most recommended techniques. MNN-BLSTM-CRF can be considered an alternative technique for future studies due to its excellent performance. Moreover, transformer-based techniques have demonstrated more impressive performance than neural network-based and machine learning models [79], [80]. Transformers-based is a context-sensitive embedding method that can perceive the word from its context. A language identification system using such a technique proposed by Thara and Poornachandran [64] demonstrated impressive performance for Malayalam-English code-mixed data. Other bilingual and multilingual code-mixed data can be evaluated using such a technique.

Second, previous studies typically built the code-mixed LID model using a supervised approach. However, we need sufficient annotated data to build the dataset for a supervised approach. Since humans carry out the data annotation process, the human annotation process is time-consuming and exhausting, especially for large datasets. Developing a large dataset can be conducted using the pseudo-labelling technique for future research. Pseudo-label is a part of semi-supervised learning for labelling more unlabelled datasets using a small number of labelled data [81], [82]. Additionally, the pseudo-label technique can improve the model performance [83].

Third, we have investigated the four main challenges in code-mixed LID: ambiguity, lexical borrowing, out-of-vocabulary, and intra-word code-mixing. These challenges in the code-mixed text may lead to incorrect language predictions in the LID system. Generally, inaccurate word tag predictions may be caused by the following factors; rare and noisy word forms and noisy context [63]. Neighbouring words that express the context of a text's body is critical to identify the words' language correctly. Therefore, future studies are encouraged to develop a code-mixed LID system capable of dealing with the challenges. For example, combining information from external resources such as dictionaries and knowledge bases can solve these challenges and enhance the LID system's performance [63].

### B. PRACTICAL IMPLICATIONS

This study offers notable practical implications for subsequent researchers in their future studies. First, most previous researchers claimed to have obtained a good LID performance in terms of accuracy, precision, recall, or F1 score while using non-standardised datasets to train their classifiers. Therefore, the results from such studies cannot be directly compared and may be unreliable. Furthermore, current LID systems evaluated on one dataset may be tailored for this dataset only and cannot be generalised to other datasets. Therefore, standardisation of code-mixed datasets as a benchmark for LID is needed.

Second, the opportunity to develop a new code-mixed dataset is still widely open. Our findings showed high percentages of bilingual code-mixed data, especially English code-mixed text. Also, most of the available datasets are dominated by Indo-Aryan language families. We observed research opportunities in the following code-mixed data: multilingual data, building non-English code-mixed data and building code-mixed data for low-resource languages. These can help make toward a standard for evaluating code-mixed LID systems.

Third, we incorporated our findings into a conceptual framework for developing a code-mixed LID model. In the future, the framework can be beneficial for developing theories, conducting empirical research, and practical application in code-mixed LID-related studies. The framework provides general steps that can be used as a standard practical guideline for budding researchers in code-mixed LID studies. To build

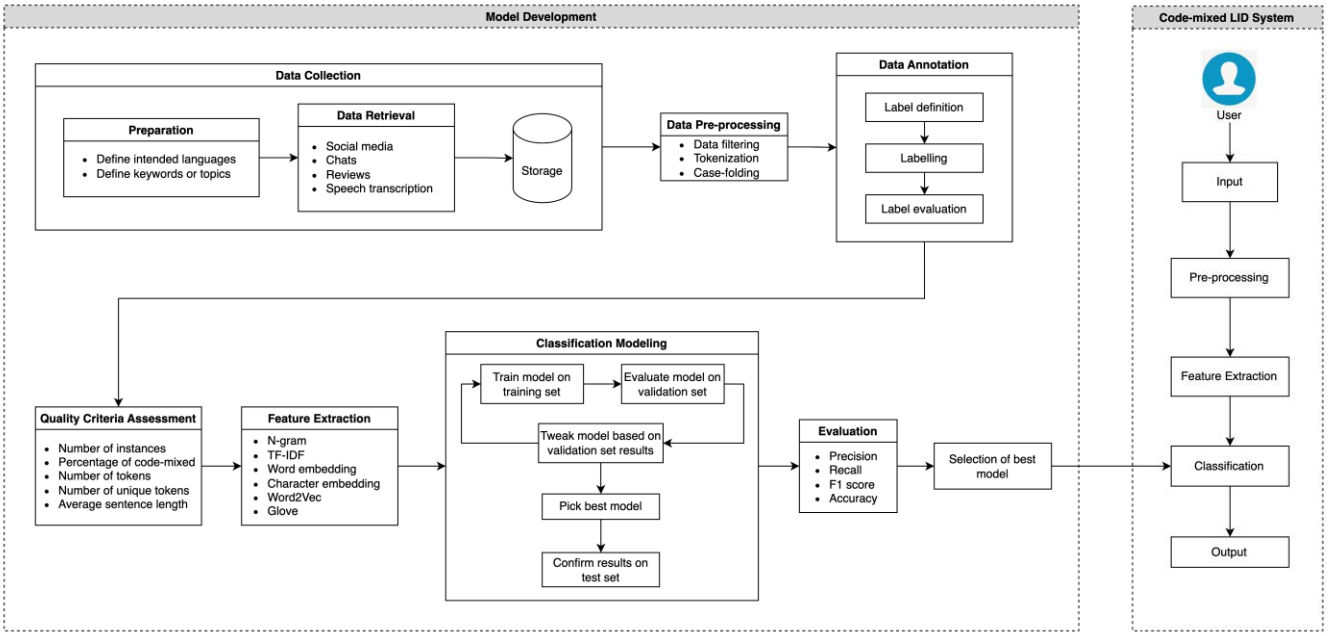


**FIGURE 4.** Framework for code-mixed LID.

a code-mixed LID for new languages, researchers should pay more attention to the following things in the framework: pre-processing, data annotation, feature extraction, and classification modelling. Pre-processing helps to select the relevant things from the raw dataset. Data annotation determines the identified labels. The feature extraction process assists in extracting the necessary part of the texts. Finally, researchers apply the designed training scenario in classification modelling to gain the best model.

## V. CONCLUSION

This systematic literature review has presented the current state of studies in code-mixed LID and proposed a framework for future research. This review included 40 primary studies published from 2016 until 2021. Three main aspects of LID for code-mixed text were investigated, e.g., 1) techniques, 2) challenges, and 3) data availability with corresponding quality criteria.

Findings revealed that in some neural network-based studies, the multichannel CNN incorporated with BLSTM and CRF had shown excellent performance in solving code-mixed LID problems. As for the non-neural network techniques, SVM and CRF are recommended to be applied. Due to its remarkable performance, the transformed-based technique can be considered one of the most robust techniques for code-mixed LID. Subsequently, we encountered four significant challenges in code-mixed LID tasks: ambiguity, lexical borrowing, non-standard words, and intra-word code-mixing. From the examined papers, this study identified 32 code-mixed datasets for LID containing 87.5% (28 studies) bilingual and 12.5% (4 studies) multilingual. Among the 32 datasets, the ratio between English code-mixed and non-English code-mixed is 84.3% (27 datasets) and 15.6% (5 datasets). Furthermore, this research setting defined five quality criteria to determine dataset quality evaluation as a benchmark to generate quality datasets for future studies. Finally, based on a detailed analysis of the recent literature and following the systematic approach, a framework for code-mixed LID is developed as a standard guideline for researchers.

## ACKNOWLEDGMENT

The authors would like to thank the Universiti Brunei Darussalam (UBD) and the Ministry of Education (MoE) Brunei Darussalam.

## REFERENCES

[1] A. F. Hidayatullah, "Language tweet characteristics of Indonesian citizens," in *Proc. Int. Conf. Sci. Technol. (TICST)*, Pathum Thani, Thailand, Nov. 2015, pp. 397–401, doi: 10.1109/TICST.2015.7369393.

[2] N. Jose, B. R. Chakravarthi, S. Suryawanshi, E. Sherly, and J. P. McCrae, "A survey of current datasets for code-switching research," in *Proc. 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Coimbatore, India, Mar. 2020, pp. 136–141, doi: 10.1109/ICACCS48705.2020.9074205.

[3] K. Shanmugalingam and S. Sumathipala, "Language identification at word level in Sinhala-English code-mixed social media text," in *Proc. Int. Res. Conf. Smart Comput. Syst. Eng. (SCSE)*, Colombo, Sri Lanka, Mar. 2019, pp. 113–118, doi: 10.23919/SCSE.2019.8842795.

[4] C. Myers-Scotton, "Common and uncommon ground: Social and structural factors in codeswitching," *Lang. Soc.*, vol. 22, no. 4, pp. 475–503, Dec. 1993.

[5] D. Gautam, P. Kodali, K. Gupta, A. Goel, M. Shrivastava, and P. Kumaraguru, "CoMeT: Towards code-mixed translation using parallel monolingual sentences," in *Proc. 5th Workshop Comput. Approaches Linguistic Code-Switching*, 2021, pp. 47–55.

[6] J. Xu and F. Yvon, "Can you traducir this? Machine translation for code-switched input," in *Proc. 5th Workshop Comput. Approaches Linguistic Code-Switching*, 2021, pp. 84–94.

[7] A. M. Barik, R. Mahendra, and M. Adriani, "Normalization of Indonesian-English code-mixed Twitter data," in *Proc. 5th Workshop Noisy User-Generated Text (W-NUT)*, Hong Kong, 2019, pp. 417–424, doi: 10.18653/v1/d19-5554.

[8] A. N. Rizal and S. Stymne, "Evaluating word embeddings for Indonesian–English code-mixed text based on synthetic data," in *Proc. 4th Workshop Comput. Approaches Code Switching (LREC)*, Marseille, France, 2020, pp. 26–35.

[9] M. J. Fuadvy and R. Ibrahim, "Multilingual sentiment analysis on social media disaster data," in *Proc. Int. Conf. Electr., Electron. Inf. Eng. (ICEEIE)*, Bali, Indonesia, Oct. 2019, pp. 269–272, doi: 10.1109/ICEEIE47180.2019.8981479.

[10] N. Sabri, A. Edalat, and B. Bahrak, "Sentiment analysis of Persian-English code-mixed texts," in *Proc. 26th Int. Comput. Conf., Comput. Soc. Iran (CSICC)*, Tehran, Iran, Mar. 2021, pp. 1–4, doi: 10.1109/CSICC52343.2021.9420605.

[11] S. Mandal and A. K. Singh, "Language identification in code-mixed data using multichannel neural networks and context capture," in *Proc. EMNLP Workshop W-NUT: 4th Workshop Noisy User-Generated Text*, Brussels, Belgium, 2018, pp. 116–120, doi: 10.18653/v1/W18-6116.

[12] A. Jamatia, S. D. Swamy, B. Gambäck, A. Das, and S. Debbarma, "Deep learning based sentiment analysis in a code-mixed English-Hindi and English-Bengali social media corpus," *Int. J. Artif. Intell. Tools*, vol. 29, no. 5, Aug. 2020, Art. no. 2050014, doi: 10.1142/s0218213020500141.

[13] R. Srinivasan and C. N. Subalalitha, "Sentimental analysis from imbalanced code-mixed data using machine learning approaches," *Distrib. Parallel Databases*, pp. 1–16, Mar. 2021, doi: 10.1007/s10619-021-07331-4.

[14] T. Jauhiainen, M. Lui, M. Zampieri, T. Baldwin, and K. Lindén, "Automatic language identification in texts: A survey," *J. Artif. Intell. Res.*, vol. 65, pp. 675–782, Aug. 2019, doi: 10.1613/jair.1.11675.

[15] D. Mave, S. Maharjan, and T. Solorio, "Language identification and analysis of code-switched social media text," in *Proc. 3rd Workshop Comput. Approaches Linguistic Code-Switching*, Melbourne, VIC, Australia, 2018, pp. 51–61.

[16] N. Bansal, V. Goyal, and S. Rani, "Experimenting language identification for sentiment analysis of English Punjabi code mixed social media text," *Int. J. E-Adoption*, vol. 12, no. 1, pp. 52–62, Jan. 2020, doi: 10.4018/ijea.2020010105.

[17] M. Dhar, V. Kumar, and M. Shrivastava, "Enabling code-mixed translation: Parallel corpus creation and MT augmentation approach," in *Proc. 1st Workshop Linguistic Resour. Natural Lang. Process.*, 2018, pp. 131–140.

[18] N. Sarma, S. R. Singh, and D. Goswami, "Word level language identification in Assamese-Bengali-Hindi-English code-mixed social media text," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Bandung, Indonesia, Nov. 2018, pp. 261–266.

[19] A. Das and B. Gambäck, "Code-mixing in social media text: The last language identification frontier?" *Traitement Automatique des Langues*, vol. 54, no. 3, pp. 41–64, 2013.

[20] N. J. Kalita and N. Saharia, "Language identification on code-mix social text," in *Proc. Int. Conf. Comput. Commun. Syst.* (Lecture Notes in Networks and Systems). Singapore: Springer, 2018, pp. 433–440, doi: 10.1007/978-981-10-6890-4_42.

[21] B. P. King, "Practical natural language processing for low-resource languages," Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. Michigan, Ann Arbor, MI, USA, 2015.

[22] S. Kent and D. Claeser, "Incorporating code-switching and borrowing in Dutch-English automatic language detection on Twitter," in *Proc. Future Technol. Conf. (FTC)* (Advances in Intelligent Systems and Computing), vol. 880. Cham: Springer, 2018, pp. 418–434, doi: 10.1007/978-3-030-02686-8_32.

[23] S. D. Das, S. Mandal, and D. Das, "Language identification of Bengali-English code-mixed data using character & phonetic based LSTM models," in *Proc. 11th Forum Inf. Retr. Eval.*, Kolkata, India, Dec. 2019, pp. 60–64, doi: 10.1145/3368567.3368578.

[24] N. H. M. Et. al, "Sentiment analysis of code-mixed text: A review," *Turkish J. Comput. Math. Educ. (TURCOMAT)*, vol. 12, no. 3, pp. 2469–2478, Apr. 2021.

[25] S. Thara and P. Poornachandran, "Code-mixing: A brief survey," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Bengaluru, India, Sep. 2018, pp. 2382–2388.

[26] K. S. N. Tan and T. M. Lim, "A review on sentiment analysis of resource-scarce languages and code-mixed social media text," in *Proc. Conf. Int. Conf. Digit. Transformation Appl. (ICDXA)*, 2020.

[27] C. Tho, Y. Heryadi, L. Lukas, and A. Wibowo, "Code-mixed sentiment analysis of Indonesian language and Javanese language using Lexicon based approach," in *Proc. 2nd Annu. Conf. Sci. Technol. (ANCOSET)*, Malang, Indonesia, vol. 1869, 2021, pp. 1–7, doi: 10.1088/1742-6596/1869/1/012084.

[28] S. Keele, "Guidelines for performing systematic literature reviews in software engineering," Dept. Comput. Sci., Univ. Durham, Durham, U.K., Tech. Rep. Ver.2.3, 2007.

[29] A. Qazi, R. G. Raj, G. Hardaker, and C. Standing, "A systematic literature review on opinion types and sentiment analysis techniques: Tasks and challenges," *Internet Res.*, vol. 27, no. 3, pp. 608–630, 2017, doi: 10.1108/IntR-04-2016-0086.

[30] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, 2007, doi: 10.1016/j.jss.2006.07.009.

[31] A. Qazi, H. Fayaz, A. Wadi, R. G. Raj, N. A. Rahim, and W. A. Khan, "The artificial neural network for solar radiation prediction and designing solar systems: A systematic literature review," *J. Cleaner Prod.*, vol. 104, pp. 1–12, Oct. 2015.

[32] A. Qazi, F. Hussain, N. A. Rahim, G. Hardaker, D. Alghazzawi, K. Shaban, and K. Haruna, "Towards sustainable energy: A systematic review of renewable energy sources, technologies, and public opinions," *IEEE Access*, vol. 7, pp. 63837–63851, 2019, doi: 10.1109/ACCESS.2019.2906402.

[33] A. Qazi, N. Hasan, O. Abayomi-Alli, G. Hardaker, R. Scherer, Y. Sarker, S. Kumar Paul, and J. Z. Maitama, "Gender differences in information and communication technology use & skills: A systematic review and meta-analysis," *Educ. Inf. Technol.*, vol. 27, pp. 4225–4258, Oct. 2021.

[34] A. Qazi, G. Hardaker, I. S. Ahmad, M. Darwich, J. Z. Maitama, and A. Dayani, "The role of information & communication technology in elearning environments: A systematic review," *IEEE Access*, vol. 9, pp. 45539–45551, 2021, doi: 10.1109/ACCESS.2021.3067042.

[35] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, and S. E. Brennan, "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *Systematic Rev.*, vol. 10, no. 1, pp. 1–11, 2021.

[36] B. Gupta, G. Bhatt, and A. Mittal, "Language Identification and disambiguation in Indian mixed-script," in *Proc. Int. Conf. Distrib. Comput. Internet Technol. (ICDCIT)*, vol. 9581. Cham: Springer, 2016, pp. 113–121, doi: 10.1007/978-3-319-28034-9_14.

[37] A. Jaech, G. Mulcaire, M. Ostendorf, and N. A. Smith, "A neural model for language identification in code-switched tweets," in *Proc. 2nd Workshop Comput. Approaches Code Switching*, Austin, TX, USA, 2016, pp. 60–64.

[38] P. Lamabam and K. Chakma, "A language identification system for code-mixed English-Manipuri social media text," in *Proc. IEEE Int. Conf. Eng. Technol. (ICETECH)*, Coimbatore, India, Mar. 2016, pp. 79–83.

[39] M. Piergallini, R. Shirvani, G. S. Gautam, and M. Chouikha, "Word-level language identification and predicting codeswitching points in Swahili-English language data," in *Proc. 2nd Workshop Comput. Approaches Code Switching*, Austin, TX, USA, 2016, pp. 21–29.

[40] Y. Samih, S. Maharjan, M. Attia, L. Kallmeyer, and T. Solorio, "Multilingual code-switching identification via LSTM recurrent neural networks," in *Proc. 2nd Workshop Comput. Approaches Code Switching*, Austin, TX, USA, 2016, pp. 50–59.

[41] M. X. Xia, "Codeswitching language identification using subword information enriched word vectors," in *Proc. 2nd Workshop Comput. Approaches Code Switching*, Austin, TX, USA, 2016, pp. 132–136.

[42] D. Nguyen and L. Cornips, "Automatic detection of intra-word code-switching," in *Proc. 14th SIGMORPHON Workshop Comput. Res. Phonetics, Phonol., Morphol.*, Berlin, Germany, 2016, pp. 82–86.

[43] B. S. Sowmya Lakshmi and B. R. Shambhavi, "An automatic language identification system for code-mixed English-Kannada social media text," in *Proc. 2nd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solution (CSITSS)*, Bengaluru, India, Dec. 2017, pp. 1–5, doi: 10.1109/CSITSS.2017.8447784.

[44] A. Phadte and R. Wagh, "Word level language identification system for Konkani-English code-mixed social media text (CMST)," in *Proc. 10th Annu. ACM India Compute Conf. (ZZZ-Compute)*, Bhopal, India, 2017, pp. 103–107, doi: 10.1145/3140107.3140132.

[45] S. Rijhwani, R. Sequiera, M. Choudhury, K. Bali, and C. S. Maddila, "Estimating code-switching on Twitter with a novel generalized word-level language detection technique," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, Vancouver, BC, Canada, 2017, pp. 1971–1982, doi: 10.18653/v1/P17-1180.

[46] P. V. Veena, M. A. Kumar, and K. P. Soman, "An effective way of word-level language identification for code-mixed Facebook comments using word-embedding via character-embedding," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Udupi, India, Sep. 2017, pp. 1552–1556.

[47] I. Chaitanya, I. Madapakula, S. K. Gupta, and S. Thara, "Word level language identification in code-mixed data using word embedding methods for Indian languages," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Bangalore, India, Sep. 2018, pp. 1137–1141, doi: 10.1109/ICACCI.2018.8554501.

[48] D. Claeser, D. Felske, and S. Kent, "Token level code-switching detection using Wikipedia as a lexical resource," in *Proc. Int. Conf. German Soc. Comput. Linguistics Lang. Technol.* Cham, Switzerland: Springer, 2018, pp. 192–198.

[49] S. Gundapu and R. Mamidi, "Word level language identification in English Telugu code mixed data," in *Proc. 32nd Pacific Asia Conf. Lang., Inf. Comput.*, Hong Kong, 2018.

[50] K. Shanmugalingam, S. Sumathipala, and C. Premachandra, "Word level language identification of code mixing text in social media using NLP," in *Proc. 3rd Int. Conf. Inf. Technol. Res. (ICITR)*, Moratuwa, Sri Lanka, Dec. 2018, pp. 1–5, doi: 10.1109/ICITR.2018.8736127.

[51] K. Singh, I. Sen, and P. Kumaraguru, "Language identification and named entity recognition in Hinglish code mixed tweets," in *Proc. ACL, Student Res. Workshop*, Melbourne, VIC, Australia, 2018, pp. 52–58.

[52] Z. Yirmibeşoğlu and G. Eryiğit, "Detecting code-switching between Turkish-english language pair," in *Proc. EMNLP Workshop W-NUT: 4th Workshop Noisy User-Generated Text*, Brussels, Belgium, 2018, pp. 110–115, doi: 10.18653/v1/W18-6115.

[53] A. Jamatia, A. Das, and B. Gambäck, "Deep learning-based language identification in English-Hindi-Bengali code-mixed social media corpora," *J. Intell. Syst.*, vol. 28, no. 3, pp. 399–408, Jul. 2019, doi: 10.1515/jisys-2017-0440.

[54] M. Mager, Ö. Çetinoglu, and K. Kann, "Subword-level language identification for intra-word code-switching," in *Proc. Conf. North*, Minneapolis, MN, USA, vol. 1, 2019, pp. 2005–2011.

[55] A. Mishra and Y. Sharma, "Language identification and context-based analysis of code-switching behaviors in social media discussions," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Los Angeles, CA, USA, Dec. 2019, pp. 5951–5956, doi: 10.1109/BigData47090.2019.9006032.

[56] I. Smith and U. Thayasivam, "Language detection in Sinhala-English code-mixed data," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Shanghai, China, Nov. 2019, pp. 228–233, doi: 10.1109/IALP48816.2019.9037680.

[57] E. Kasmuri and H. Basiron, "Segregation of code-switching sentences using rule-based technique," *Int. J. Advance Soft Comput. Appl.*, vol. 12, no. 1, pp. 1–16, 2020.

[58] M. Kazi, H. Mehta, and S. Bharti, "Sentence level language identification in Gujarati-Hindi code-mixed scripts," in *Proc. IEEE Int. Symp. Sustain. Energy, Signal Process. Cyber Secur. (iSSSC)*, Gunupur Odisha, India, Dec. 2020, pp. 1–6, doi: 10.1109/iSSSC50941.2020.9358837.

[59] D. Patel and R. Parikh, "Language identification and translation of English and Gujarati code-mixed data," in *Proc. Int. Conf. Emerg. Trends Inf. Technol. Eng. (ic-ETITE)*, Vellore, India, Feb. 2020, doi: 10.1109/ic-ETITE47903.2020.410.

[60] S. Shekhar, D. K. Sharma, and M. M. Sufyan Beg, "An effective bi-LSTM word embedding system for analysis and identification of language in code-mixed social media text in English and Roman Hindi," *Computación y Sistemas*, vol. 24, no. 4, Dec. 2020, doi: 10.13053/cys-24-4-3151.

[61] Y. Gupta, G. Raghuwanshi, and A. Tripathi, "A new methodology for language identification in social media code-mixed text," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl.* (Advances in Intelligent Systems and Computing), vol. 1141. Singapore: Springer, 2021, pp. 243–254, doi: 10.1007/978-981-15-3383-9_22.

[62] N. J. Kalita, A. G. Agarwala, and J. Das, "Word level language identification on code-mixed English-bodo text," in *Proc. 6th Int. Conf. Comput. Manage. Math. Sci. (ICCM)*, Nirjuli, India, vol. 1020, 2021, pp. 1–6, doi: 10.1088/1757-899X/1020/1/012027.

[63] N. Sarma, R. S. Singh, and D. Goswami, "SwitchNet: Learning to switch for word-level language identification in code-mixed social media text," *Natural Lang. Eng.*, vol. 28, no. 3, pp. 337–359, 2022, doi: 10.1017/s1351324921000115.

[64] S. Thara and P. Poornachandran, "Transformer based language identification for Malayalam-English code-mixed text," *IEEE Access*, vol. 9, pp. 118837–118850, 2021, doi: 10.1109/access.2021.3104106.

[65] C. Sabty, I. Mohamed, Ö. Çetinoglu, and S. Abdennadher, "Language identification of intra-word code-switching for Arabic–English," *Array*, vol. 12, Dec. 2021, Art. no. 100104, doi: 10.1016/j.array.2021.100104.

[66] L. Nguyen, C. Bryant, S. Kidwai, and T. Biberauer, "Automatic language identification in code-switched Hindi-English social media text," *J. Open Humanities Data*, vol. 7, pp. 1–13, Jun. 2021.

[67] P. K. Jain, R. Pamula, and G. Srivastava, "A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews," *Comput. Sci. Rev.*, vol. 41, Aug. 2021, Art. no. 100413, doi: 10.1016/j.cosrev.2021.100413.

[68] A. Goyal, V. Gupta, and M. Kumar, "Recent named entity recognition and classification techniques: A systematic review," *Comput. Sci. Rev.*, vol. 29, pp. 21–43, Aug. 2018, doi: 10.1016/j.cosrev.2018.06.001.

[69] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[70] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, and S. Shleifer, "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2020, pp. 38–45.

[71] R. Joshi and R. Joshi, "Evaluating input representation for language identification in Hindi-English code mixed text," 2020, *arXiv:2011.11263*.

[72] M. A. Ansari and S. Govilkar, "Sentiment analysis of mixed code for the transliterated Hindi and Marathi texts," *Int. J. Natural Lang. Comput.*, vol. 7, no. 2, pp. 15–28, Apr. 2018, doi: 10.5121/ijnlc.2018.7202.

[73] G. Molina, F. AlGhamdi, M. Ghoneim, A. Hawwari, N. Rey-Villamizar, M. Diab, and T. Solorio, "Overview for the second shared task on language identification in code-switched data," in *Proc. 2nd Workshop Comput. Approaches Code Switching*, Austin, TX, USA, 2016, p. 2016.

[74] M. Haspelmath, "Lexical borrowing: Concepts and issues," in *Loanwords in the World's Languages: A Comparative Handbook*. Berlin, Germany: De Gruyter Mouton, 2009, pp. 35–54.

[75] T. C. Cole, E. Siebert-Cole, D. Medan, A. V. Podgurenko, M. Zhang, E. Selvi, A. Sokolova, V. N. Godin, J. Satoła-Staśkowiak, and S. Alghazo, "The international 'language phylogeny poster' (LangPP) Project: Teaching tools for a first overview on the evolution of languages," Tech. Rep., 2022. [Online]. Available: https://www.researchgate.net/publication/360343127_The_International_Language_Phylogeny_Poster_LangPP_Project_Teaching_Tools_for_a_First_Overview_on_the_Evolution_of_Languages

[76] M. L. Louden, "Minority Germanic languages," in *The Cambridge Handbook of Germanic Linguistics*. Cambridge, U.K.: Cambridge Univ. Press, 2020, pp. 807–832.

[77] J. Kaur and J. R. Saini, "A study and analysis of opinion mining research in indo-aryan, dravidian and tibeto-burman language families," *Int. J. Data Mining Emerg. Technol.*, vol. 4, no. 2, pp. 53–60, 2014.

[78] S. S. V. Kusampudi, A. Chaluvadi, and R. Mamidi, "Corpus creation and language identification in low-resource code-mixed Telugu-English text," in *Proc. Conf. Recent Adv. Natural Lang. Process.-Deep Learn. Natural Lang. Process. Methods Appl.*, 2021, pp. 744–752.

[79] R. Hanslo, "Deep learning transformer architecture for named entity recognition on low resourced languages: State of the art results," 2021, *arXiv:2111.00830*.

[80] M. Al-Yahya, H. Al-Khalifa, H. Al-Baity, D. AlSaeed, and A. Essam, "Arabic fake news detection: Comparative study of neural networks and transformer-based approaches," *Complexity*, vol. 2021, pp. 1–10, Apr. 2021, doi: 10.1155/2021/5516945.

[81] A. F. Hidayatullah, A. M. Hakim, and A. A. Sembada, "Adult content classification on Indonesian tweets using LSTM neural network," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*, Bali, Indonesia, Oct. 2019, pp. 235–240, doi: 10.1109/ICACSIS47736.2019.8979982.

[82] A. Hande, K. Puranik, K. Yasaswini, R. Priyadharshini, S. Thavareesan, A. Sampath, K. Shanmugavadivel, D. Thenmozhi, and B. Raja Chakravarthi, "Offensive language identification in low-resourced code-mixed dravidian languages using pseudo-labeling," 2021, *arXiv:2108.12177*.

[83] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Represent. Learn. (ICML)*, Atlanta, GA, USA, 2013, vol. 3, no. 2, p. 896.

**AHMAD FATHAN HIDAYATULLAH** received the bachelor's degree (B.Eng.) in informatics from the Universitas Islam Negeri Sunan Kalijaga, Yogyakarta, Indonesia, in 2010, and the master's degree (M.Cs.) in computer science from the Universitas Gadjah Mada, Yogyakarta, in 2014. He is currently pursuing the Ph.D. degree with the Computer Science Program, School of Digital Science, Universiti Brunei Darussalam, with The UBD Graduate Scholarship (UGS). He is also an Assistant Professor at the Department of Informatics, Faculty of Industrial Engineering, Universitas Islam Indonesia. His research interests include text mining, natural language processing, and data science. He is also actively involved in research at the Center of Data Science, Universitas Islam Indonesia.

**ATIKA QAZI** received the Ph.D. degree from the Universiti Malaya, Kuala Lumpur, Malaysia. Her research interests include opinion mining, sentiment analysis, big data analytics, information systems, lifelong learning, ICT, and renewable energy. She is actively involved as a reviewer of ISI journals.

**DAPHNE TECK CHING LAI** (Member, IEEE) received the B.Sc. degree in computer science from the University of Strathclyde, in 2004, the M.Sc. degree from the University of Kent, Canterbury, in 2006, and the Ph.D. degree from the University of Nottingham, U.K., in 2014. She worked at Hosei University, Japan, in 2018, under the Hosei International Fund Foreign Scholars Fellowship. She is currently a Senior Assistant Professor at the School of Digital Science, Universiti Brunei Darussalam. She has published and reviewed about 40 articles. Her research interests include data mining, computational intelligence, and evolutionary computation.

**ROSYZIE ANNA APONG** received the B.Sc. degree in computer science from the University of Strathclyde, in 2004, the M.Sc. degree in multimedia and internet computing from Loughborough University, in 2016, and the Ph.D. degree in computer science from The University of Manchester, in 2018. She is currently a Lecturer at the School of Digital Science, Universiti Brunei Darussalam. Her research interests include text mining, the Internet of Things, and information retrieval.