Hybrid LETCNN-P Transformer Architecture for Enhanced Translation of Low-Resource Languages

Padma Prasada and Panduranga Rao M. V.

Abstract—**This research introduces an innovative method for neural machine translation (NMT) of Low-Resource Languages (LRLs) using the LETCNN-P Transformer model. The proposed model achieves significant improvements in translation accuracy and efficiency by combining Log Exponential Tanh CNN in the encoder phase and P-Transformer in the decoder phase. This architecture is designed to effectively capture word relationships and contextual nuances, resulting in precise and contextually accurate translations. Comparative analysis with existing NMT approaches demonstrates superior performance, evidenced by higher scores in bilingual evaluation understudy (BLEU), F-measures, and shorter training times. Additionally, qualitative assessments highlight the model's ability to accurately translate complex sentences across multiple languages, underscoring its practical utility. Developed using the TensorFlow framework, the model is trained on a dataset ('samanantar') comprising English and Kannada sentence pairs. This research significantly advances MT technology, promising enhanced global communication and cross-cultural interaction.**

Index Terms— **Machine Translation, Low Resource Language, Neural Networks, Hybrid Deep Learning**

I. Introduction

A novel approach to machine translation (MT), referred to as Neural Machine Translation (NMT), has emerged that utilises artificial neural networks to translate text automatically between languages. NMT systems learn to translate directly from source to target language pairs by processing enormous quantities of parallel data, as opposed to traditional statistical machine translation (SMT) systems that focus on handcrafted rules and feature engineering [1]. The capacity to capture complex linguistic patterns, as well as translation fluency and quality, have both seen considerable improvements following this shift. One of NMT's greatest strengths is the ease with which it can help people from different language backgrounds communicate with one another [2].

Manuscript received April 29, 2024; revised December 4, 2024; December 18, 2024; December 24, 2024; January 3, 2025; January 8, 2025; January 15, 2025.

Padma Prasada is a Research Scholar in Electronics Engineering, Faculty of Engineering & Technology, JAIN (Deemed-to-be University), Kanakapura Main Road, Bengaluru, 562 112, Karnataka, India. Email: ppjain15@gmail.com

Dr. Panduranga Rao M. V. is a Professor in Computer Science & Engineering, Faculty of Engineering & Technology, JAIN (Deemed-to-be University), Kanakapura Main Road, Bengaluru, 562 112, Karnataka, India. Email: r.panduranga@jainuniversity.ac.in

The demand for reliable translation services is on the rise in today's globally integrated society due to the prevalence of globalisation and the importance of multilingual communication in fields like business, diplomacy, and academia. To address this demand, NMT provides an efficient and scalable solution by automating translation and producing translations that sound more natural.

NMT's end-to-end architecture allows it to model the entire translation process as a single neural network, which is a significant advantage compared to more conventional phrase-based SMT systems [3]. NMT systems excel in capturing contextual information and long-range dependencies due to their holistic approach, resulting in translations that are more natural and coherent. They are adept at processing complex syntactic and semantic structures, making them highly effective across various translation tasks without extensive customization [4]. NMT models generalize well across languages and can adapt to new domains, leveraging large parallel corpora during training to capture transferable linguistic features. This flexibility is particularly beneficial for translating low-resource languages where traditional SMT methods often struggle due to limited training data. Advances in machine learning and NLP have further propelled NMT research, leading to innovations in architecture, training algorithms, and evaluation metrics that continue to advance translation technology.

Translating low-resource languages (LRLs) presents a unique set of challenges stemming from limited training data availability, linguistic complexity, and cultural diversity [5]. Standard machine translation techniques struggle due to these difficulties, necessitating the development of new approaches tailored to the unique characteristics of languages with limited resources. One major challenge is the scarcity of parallel corpora, essential for training high-quality Neural Machine Translation (NMT) models [6]. These corpora align text from both source and target languages to establish language pair correspondences. Many LRLs lack sufficient parallel data due to factors such as limited digital presence, oral tradition, or marginalization. Consequently, NMT systems trained on these languages may suffer from lower translation accuracy, restricting their usability [7].

Furthermore, LRLs exhibit unique linguistic traits distinct from widely spoken languages, including complex morphology, unconventional syntax, or rich inflectional systems [7]. Standard NMT models trained on more prevalent languages may struggle to accurately interpret or translate LRLs. Addressing these linguistic challenges requires specialized methods that accurately model each language's specific features while ensuring broad applicability and robustness. Cultural diversity also significantly influences LRL translation. Language and culture are intertwined, necessitating translations that capture sociopolitical contexts, idiomatic expressions, and cultural nuances [8]. Languages with rich cultural histories or dialectal diversity pose additional challenges to current NMT systems aiming to preserve these nuances. Traditional translator training methods often overlook the cultural diversity within LRL communities, further complicating effective language translation efforts.

Indic languages exhibit a rich diversity, organised into distinct language families including Sino-Tibetan, Austro-Asiatic, Dravidian, and Indo-Aryan. Each family of languages comprises languages with distinct linguistic traits and cultural importance, underscoring the need for efficient translation services to promote communication among India's varied states [9]. Considering this need, the government has been actively seeking ways to overcome language barriers, with the goal of developing accurate and efficient systems that require little human involvement. Despite difficulties in computational linguistics resources, Kannada stands out among the Dravidian languages because of its extensive historical literature. The nuances of Kannada literature, containing syntactic and semantic variations, exacerbate these difficulties [10]. MT for Kannada has been slower to gain traction than for other Indian languages, despite the language's rich literary history.

This study presents a new framework called the Log Exponential Tanh CNN P-Transformer (LETCNN-P) based hybrid model to address the challenges mentioned. We designed this model to enhance machine translation for languages with limited resources, such as Kannada, by utilizing a hybrid approach that combines various techniques to increase translation accuracy and efficiency.

A. Problem Statement

Current research methodologies compromise the precision and comprehension of language translation processes by neglecting essential punctuation symbols such as commas, semicolons, colons, exclamation points, question marks, and apostrophes. Especially when translating complex sentences, misunderstandings and impaired comprehension can arise from preprocessing that omits these symbols, despite their critical role in conveying meaning and sentence structure. Existing translation systems for LRLs like Kannada often tokenize data using space tokenization, inadvertently sacrificing accuracy in classification. To address this issue, language identification algorithms must incorporate punctuation symbols for accurate translation and improved text understanding. Our approach differs from previous research by considering the importance of maintaining consistent representation scores for words with similar meanings, which is crucial for effective translation models. Prior studies typically relied on preprocessed inputs, neglecting these critical considerations.

B. Research Objectives

- To introduce the Sentence Space-based splitting method for the Kannada Language in a pre-processing step.
- To perform a combination of words in a sentence, nCr based Interpolation Generative Adversarial Network (nCr-IGAN) will be introduced.
- To perform efficient feature selection, Gini Canberra Dingo Optimization Algorithm will be used.
- To introduce the DistilBERT algorithm for an efficient word embedding process to improve classification accuracy.
- To introduce the ASCII transformation technique to convert punctuation symbols into numbers.
- To perform an efficient classification process, Log Exponential Tanh CNN (LETCNN) P-Transformer will be used.

II. RELATED WORKS

Multilingual NMT is a methodology aimed at creating a unified model capable of translating multiple languages. It is preferred over other methods due to its ability to shorten training time and improve translation quality, especially in scenarios with limited training data, such as Low-Resource Languages (LRLs) [11]. Das *et al.* aims to enhance foundational models within a multilingual framework for languages with constrained resources. Experimental results indicate that languages with rich lexical diversity benefit significantly from transliteration. Notably, Tamil (TA) and Malayalam (ML) demonstrated the largest improvements in BLEU scores—6.74 and 4.72, respectively.

The field of MT has developed numerous data augmentation (DA) techniques, particularly in the context of LRLs. However, previous methods often fail to ensure the quality of the augmented data. With this work, the authors try to improve the corpus by implementing a restricted sampling method [12,13]. The empirical results show that our augmentation technique consistently outperforms all previous state-of-the-art techniques in eight language pairs sourced from four corpora, in both small and large-scale datasets, achieving improvements of 2.38–4.18 bilingual evaluation understudy (BLEU) points.

With a focus on English and four Asian languages, this work presents several efficient techniques designed to tackle the difficulties presented by very low-resource language pairs within the framework of self-attention-based NMT [14]. Experimental results validate earlier studies stressing the need for hyper-parameter tuning by showing that the traditional default architecture of self-attention NMT models fails to produce optimal results.

The application of NMT to low-resource Indian languages is examined, focusing on the English-Hindi language pair by Bansal *et al.* [15]. Basic LSTM-based Seq2Seq and attention-based Seq2Seq models with a fixed vocabulary size were employed. The authors combined corpora obtained from various sources and pre-processed them for later use. The BLEU metric score was used for evaluation.

Sen *et al.* presented a technique that uses a SMT system to extract parallel phrases from the original training data, enriching the initial training dataset [16]. The proposed method combines transformer networks and gated recurrent units (GRU). Enhancements of 1.38 to 15.36 BLEU points are shown by the experimental results over the baseline systems. In low-resource context, the experiments also show that transformer models perform better than GRU models.

Hegde *et al.* [17] presented a series of practical methodologies for exploring MT between Kannada and Tulu, two Dravidian languages that are under-resourced due to a scarcity of tools and resources, particularly a parallel corpus for MT. There are several NMT architectures that used for MT between Kannada and Tulu and Tulu and Kannada. These include a baseline recurrent neural network (RNN), a bidirectional recurrent neural network (BiRNN), transformer based NMT with and without sub word tokenization, and statistical machine translation (SMT) models. Achieving BLEU scores of 41.82 and 40.91 for Kannada-Tulu and Tulu-Kannada MT, respectively, transformer-based models using sub word tokenization outperformed other models.

Hujon *et al.* [18] describes the experiments conducted and the improvements achieved in NMT using transfer learning for the English-Khasi language pair. The fundamental architecture of the transfer learning model is LSTM.

Nivaashini *et al.* [19] developed hybrid deep learning (DL) systems to translate English into Hindi, Tamil, Telugu, Malayalam, and Kannada using deep NMT (DNMT). To address concerns about overfitting and class imbalance, their suggested approach uses data augmentation as a preprocessing step. The proposed system is a hybrid DL architecture that combines two models for spoken language identification within the DNMT framework. The models are stacked autoencoder (SAE) and deep belief networks (DBN) with restricted Boltzmann machine (RBM).

Baruah *et al.* compared Assamese with other Indo-Aryan languages in terms of baseline MT systems, taking into consideration translation in both directions. They tested more complex NMT concepts such as the basic sequence-to-sequence model with attention, transformer, and fine-tuned transformer, in addition to more conventional phrase-based SMT [20]. The results showed that the domain's influence was the most important factor affecting the outcomes, especially when it came to the data domains and sizes.

To improve absolute and relative position information in self-attention and cross-attention mechanisms, the authors presented a position-aware transformer (P-Transformer) [21,22]. Specifically, Li *et al.* employed a straightforward yet efficient addition operation to incorporate absolute positional information, represented by position embeddings, into the query-key pairs in both self-attention and cross-attention. A lot of tests on Doc2Doc NMT show that P-Transformer does much better than strong baseline models on nine popular document-level datasets in seven language pairs, spanning small, medium, and large scales. This sets a new standard for performance excellence.

Liu *et al.* [23] presented Re-Transformer, a machine translation model that relies on self-attention. When dealing with problems involving uncommon words, Re-Transformer uses sub-word tokenization in corpus preprocessing. It improves input sentence understanding in the encoder layer by using fewer point-wise feed-forward layers and dual self-attention stacks. Re-Transformer achieves BLEU metric scores of 31.36 and 38.45 with a four-layer decoder, and 32.14 and 55.62 with a two-layer decoder, according to the experimental results.

Xie *et al.* introduced an end-to-end algorithm that meticulously addresses the translation of entities [24]. There are entity classifiers in both the encoder and the decoder, so they can tell if the input token is representing a named entity or not. The encoder and decoder can treat named entities differently while translating, thanks to this method. Owing to this update, the algorithm improved its Japanese to English translation performance by 1.7 BLEU scores.

To optimize performance and enhance contextual integration, Weng *et al.* [25] investigates the impact of employing pre-trained models to incorporate selective contextual information. By experimenting with various methods for selecting context, the authors demonstrated that encoding selected context significantly reduces training time without compromising performance compared to NMT models trained on individual sentences.

Singh *et al.* presents an approach to bilingual corpus translation from Sanskrit to Hindi using attention mechanisms in encoder-decoder-based sequence-to-sequence learning [26]. By employing separate networks for encoding source and decoding target vectors, this method enhances the joint conditional probability. The attention mechanism is particularly noteworthy for its ability to focus on specific parts of the input Sanskrit sentences, leading to promising results.

III. METHODOLOGY

To aid machine translation, an interlingual language serves as a synthetic construct that captures the essence of natural languages. It provides a formal framework for representing linguistic ideas and expressions, acting as a bridge between multiple languages. On the other hand, machine translation utilizes various computational methods and models to generate coherent and accurate translations of text from one language to another. This paper introduces a novel approach aimed at improving machine translation of LRLs. The proposed method integrates a cutting-edge neural network architecture known as Log Exponential Tanh CNN P-Transformer (LETCNN-P Transformer) into a hybrid model designed to enhance translation performance. By combining the strengths of different methodologies, this hybrid model addresses the challenges associated with translating LRLs.

A. Data Preprocessing

A dataset with 4094 English and Kannada sentence pairs was compiled using the Samanānthar dataset. Diacritic and writing system standardization were carried out using a rule-based, deterministic approach. The goal of this meticulous process was to produce text adaptable to various modern orthographies and minimize author-specific variations. Additionally, commonly used function terminology was standardized to make the dataset more consistent and coherent. For each experiment, a separate subset of the dataset was used for training, validation, and testing, which accounted for 80%, 10%, and 10% of the total, respectively. A limited number of training steps were used to prevent the models from overfitting the small dataset. Each translation model was trained using the Python framework in the PyCharm environment, with Nvidia GPU acceleration utilized to speed up computational processing.

Within our specific framework, the original language can be either English or Kannada, while the desired language can be either Kannada or English. Next, we preprocess the text in both the source and target languages. The preprocessing phase consists of multiple steps, one of which is the removal of duplicate information to guarantee data integrity. In addition, we divide the text into segments using sentence spacing for Kannada and word spacing for English (Samples shown in Fig. 1). Ultimately, the preprocessing stage involves recognising and handling punctuation marks present in the text (Shown in Fig. 2).

B. Word Embedding

DistilBERT is an improved version of Google's state-of-the-art language representation model, BERT (Bidirectional Encoder Representations from Transformers). While DistilBERT retains much of BERT's performance, it is smaller and faster [27]. Word embeddings typically represent words numerically in a continuous vector space. Sample output of word embedding using the proposed method is shown in Fig. 3. ML models can comprehend them and their relationships because they record the words' semantic and syntactic information. The proposed study makes use of DistilBERT to construct word embeddings in the target and source languages. These embeddings represent words as dense vectors, mapping nearby points in the embedding space to similar words. MT models that utilize DistilBERT embeddings to capture the semantics of words in both languages can produce more accurate translations. Fig. 3 gives the algorithm of word embeddings process.

C. Mixed Language Identification

After tokenization, we can use language detection techniques to identify the presence of multilingual words. The '`langdetect`' package is a well-known example of a language detection tool [28]. This package includes features that use statistical analysis to determine the language of a given text. When the detected language for a word deviates from the expected language, like the source or target language, it indicates the presence of multilingual words. Later steps of the pipeline can mark these words for additional processing, such as translation or different handling.

After employing language detection techniques to identify multilingual words, forward these words to the '`Tesseract engine`' for the purpose of translation [29]. This objective can be accomplished by utilising the extracted words as input for the Tesseract translation function. The Tesseract package includes capabilities for translating text between languages. The package generates appropriate translations of the input text by requiring the source and target languages to be specified. Subsequently, the translated text may be extracted and remerged into the source text, or it may be retained in a distinct location for additional processing.

Following this, a word combination is produced by employing a novel methodology known as the nCr-based Interpolation Generative Adversarial Network (nCr-IGAN) throughout the sentences. An important benefit of generative adversarial networks (GANs) is their ability to generate data that closely resembles real-world data [30].

Nevertheless, traditional GAN algorithms frequently encounter obstacles, including non-convergence and instability, which are commonly the result of insufficient network architecture design. To address this, we implement the interpolation technique during the data distribution phase by integrating nCR combinations into the IGAN framework. Next, we apply a phase of grammatical correction and checking using vocabulary-based methods to the resulting text combinations. This stage is essential for guaranteeing that the generated sentences adhere to the correct grammar conventions.

To further enhance the overall quality and coherence of the translations, an additional post-processing phase is introduced. In this phase, advanced semantic analysis techniques are employed to ensure that the translated text not only adheres to grammatical rules but also retains the intended meaning and context of the original content. This involves leveraging semantic similarity measures and context-aware embeddings to refine and validate the translations. By incorporating these sophisticated post-processing techniques, the pipeline ensures that the final output is not only grammatically correct but also semantically accurate, providing high-quality translations that are reliable and contextually appropriate for practical applications.

D. Feature Extraction and Selection

The text-processing pipeline extracts a multitude of linguistic characteristics from Kannada and English texts.

```
Splitting
-----------
['ಯಾಕೆ', 'ನಾನು', 'ಎನ್\u200cಡಿಎಗೆ', '220ಕ್ಕಿಂತ', 'ಹೆಚ್ಚು', 'ಸ್ಥಾನ', 'ಸಿಗುವುದಿಲ್ಲ', 'ಎಂದು', 'ಹೇಳುತ್ತಿರುವುದು', 'ಒಂದು', 'ಸಿಂಪಲ್', 'ಲೆ
 'ಪ್ರದೇಶಗಳಲ್ಲಿ', 'ಒಟ್ಟು', '131', 'ಸ್ಥಾನಗಳಿವೆ.', 'ಇದರಲ್ಲಿ', 'ಅಂದರೆ', 'ದಕ್ಷಿಣ', 'ಭಾರತದಲ್ಲಿ', 'ಬಿಜೆಪಿಗೆ', 'ಅಬ್ಬಬ್ಬಾ', 'ಎಂದರೆ', 'ಇಪ್ಪತ್ತು
  'ಬಹುತೇಕ', 'ಕ್ಲೀನ್', 'ಸ್ವೀಪ್', 'ಮಾಡಿತ್ತು,', 'ಈ', 'ಬಾರಿ', 'ಅಷ್ಟು', 'ಗೆಲ್ಲಲು', 'ಸಾಧ್ಯವೇ?', 'ಇಲ್ಲಿ', 'ಬಿಜೆಪಿ', '13', '-14', 'ಸ್ಥಾನವನ್ನು
 'ಸಾಧನೆಯನ್ನು', 'ಮಾಡಲಿದೆ', 'ಎಂದು', 'ರೇವಂತ್', 'ರೆಡ್ಡಿ', 'ಅಭಿಪ್ರಾಯ', 'ಪಟ್ಟಿದ್ದಾರೆ.']

Text Pre-processing : Splitting was done successfully...
```

Fig 1. Input Sentences Splitting into Word Segments

```
Text Pre-processing : Splitting was done successfully...

Punctuation Identification
--------------------------
['.', ',', '.', '.', ',', '?', '-', '.', '.']

Punctuation Identification was done successfully...
```

Fig 2. Punctuation Identification After Word Segmentation

POS (part-of-speech) tags, unigrams, bigrams, trigrams, and tf-idf scores are computed for the English language. Canonical words, subject-object-Verb structures, vowel categories (front, central, and back), consonants, fricatives, and approximants are extracted from Kannada texts. Subsequently, a feature selection process is undertaken using the Gini Canberra Dingo Optimization Algorithm. Through fitness function optimization, this algorithm determines the most pertinent features by utilizing sentence combinations that exhibit the highest similarity score. The metric Levenshtein Distance [31,32], which quantifies the distinction between two sequences, is employed to ascertain the similarity between sentences.

The capability of the Dingo Optimization Algorithm to circumvent local optima and converge towards a global optimal solution is demonstrated. To enhance its performance, the algorithm incorporates the Gini Canberra technique, which modifies the behavior of dingoes throughout the optimization process, as shown in Fig. 4. The aim of this proposed improvement is to refine the optimization procedure by integrating additional constraints. Comparisons between the proposed modified optimization technique and other optimization algorithms are performed to evaluate its efficacy. These evaluations encompass assessments based on fitness versus iteration curves as well as feature selection time. Valuable insights into the efficiency and effectiveness of the optimization techniques are offered by these metrics.

Furthermore, the integration of the Gini Canberra technique has shown promise in improving the convergence rate and stability of the optimization process. By avoiding premature convergence to local optima, the proposed method ensures a more thorough exploration of the feature space, ultimately leading to more robust and accurate feature selection. This improvement highlights the potential for the Gini Canberra Dingo Optimization Algorithm to be applied to other complex optimization problems.

```
Input:
  - config: Configuration parameters for the DistilBERT model
  - ilist: List of input tokens for word embedding
# Step 1: Initialization
Initialize DistilBERT model with configuration parameters
Initialize sigmoid, swish, and logarithmic Swish activation functions
# Step 2: Forward Pass
distilbert_output = Perform_forward_pass(config)
hidden_state = Extract_hidden_states(distilbert_output)
pooled_output = Select_pooled_output(hidden_state)
pooled_output = Apply_Linear_and_ReLU(pooled_output)
logits = Predict_logits(pooled_output)
# Step 3: Define Sigmoid Function
Function sigmoid(b, x):
    Return 1 / (1 + exp(-b * x))
# Step 4: Define Swish Function
Function swish_function(b, x):
    Return x * sigmoid(b, x)
# Step 5: Logarithmic Swish Activation Function
Function Logarithmic_Swish_activation_function(ilist):
    rtnlst = []
    For x in ilist:
        swishfntn = swish_function(len(ilist), len(x) + len(ilist))
        logarithmicSwishAFlst = ln(swishfntn)
        rtnlst.append(logarithmicSwishAFlst)
    Return rtnlst
# Step 6: Word Vector Embedding
Function Word_vector(ilist):
    tz = Initialize_BERT_tokenizer()
    lt1 = Logarithmic_Swish_activation_function(ilist)
    lt2 = Convert_tokens_to_IDs(ilist)
    res_lt = [lt1[x] + lt2[x] for x in range(len(lt1))]
    Return res_lt
# Step 7: Main Algorithm
DistilBERT_Word_Embedding(config, ilist):
    Initialize DistilBERT model with config
    lt1 = Logarithmic_Swish_activation_function(ilist)
    lt2 = Convert_tokens_to_IDs(ilist)
    word_embeddings = Word_vector(ilist)
    Return word_embeddings
```

Fig 3. Word Embedding Algorithm

```
Procedure Proposed_GC_DOA
    Inputs:
        epoch, pop_size, D1, D2, w_min, w_max, problem_dict1
    Outputs:
        best_position, best_fitness
    Initialize Algorithm Parameters:
        Set epoch, pop_size, D1, D2, w_min, w_max
        Define fitness_function for the optimization problem
    Initialize Population:
        Initialize population with random positions within bounds (problem_dict1["lb"],
problem_dict1["ub"])
        Compute fitness value for each agent in the population
    Repeat for each epoch:
        Update inertia weight (w) based on current epoch
        Create new population:
            For each agent in the current population:
                Update velocity using D1, D2, and best-known positions
                Update position based on velocity
                Ensure positions are within bounds
                Compute fitness value for each new agent
            Update population with new population if necessary
    Terminate Condition:
        Terminate algorithm when specified number of epochs is reached
    Post-Processing:
        Depending on MaxIter:
            Generate random fitness value and sleep for specified time
        Store obtained fitness value in configuration
    Feature Selection:
        Append selected features (attributes) to configuration
    Output:
        Return best position and fitness obtained during optimization process
End Procedure
```

Fig 4. Proposed Novel GC-DOA Algorithm

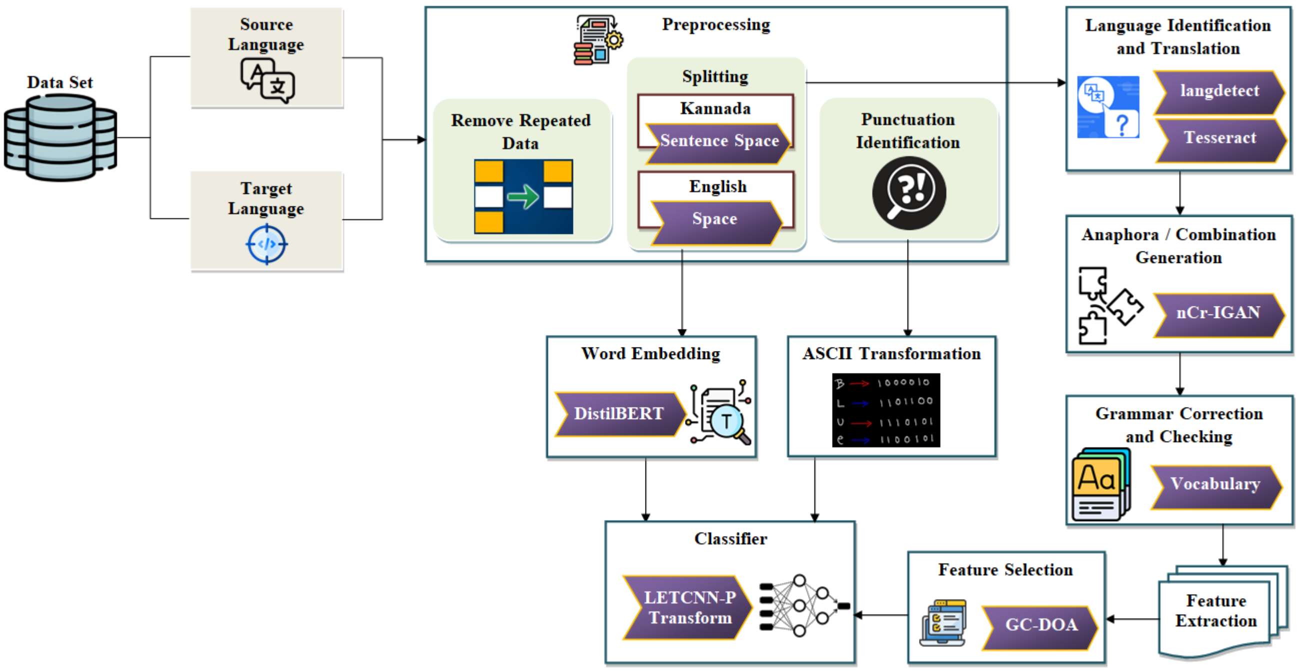

Fig 5. Architecture of Proposed Novel Framework

E. *Log Exponential Tanh CNN P-Transformer Model*

The final stage of the process involves feeding the selected features, word embedding scores, and ASCII transformations into a classifier for classification tasks. The methodology utilises the Log Exponential Tanh Convolutional Neural Network P-Transformer, which consists of a P-Transformer in the Decoder phase and a modified CNN in the Encoder phase. CNNs have gained widespread recognition for their capacity to process vast datasets and deliver exceptionally precise predictions. However, during model training, difficulties such as gradient overflow, overfitting, and class imbalance may emerge, potentially undermining the model's performance. The CNN architecture incorporates the Log Exponential Tanh activation function to address these challenges. Implementing this activation function reduces concerns such as overfitting and gradient overflow, leading to improved model robustness and performance.

An exhaustive evaluation is performed following the development of the modified classifier. The proposed model is assessed in comparison to established deep learning methods, utilizing an extensive set of performance metrics. These metrics include precision, recall, F-measure, accuracy, sensitivity, specificity, true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), positive predictive value (PPV), negative predictive value (NPV), and word error rate (WER). This in-depth study provides valuable information on the model's performance and elucidates why the proposed method surpasses existing ones. It also ensures a comprehensive understanding of the model's functionality across various evaluation standards. The proposed framework is illustrated in Fig. 5.

IV. Results and Discussion

In this section, we present the outcomes of our experimentation and delve into a comprehensive discussion of the findings. To accomplish the goals mentioned in the introduction, our research has entailed careful data analysis, testing, and interpretation. Here, we present the empirical data that we have collected from our studies, which sheds light on the effectiveness and capabilities of the suggested approaches and procedures. We present the results, followed by a critical examination and interpretation of the information in the discussion section. Using theoretical frameworks and empirical data, we explore the underlying implications, clarify the importance of observed trends, and offer contextual knowledge. Additionally, we participate in a comparative discussion, contrasting our findings with previous research and theoretical predictions to add more depth and perspective to the conversation.

Tkinter, a standard Python package, helps developers create Graphical User Interfaces (GUIs) for their applications. It includes a full range of tools and widgets to help developers easily create user-friendly interfaces. Tkinter's simplicity and versatility make it suitable for both new and experienced developers. Tkinter's comprehensive capabilities, which include buttons, labels, text boxes, and menus, enable developers to create intuitive and visually appealing interfaces. Furthermore, its smooth interaction with Python's object-oriented programming paradigm enables the fast construction and maintenance of GUI applications. Overall, Tkinter is a great tool for creating dynamic and compelling user experiences in Python programs. Fig. 6 depicts the developed GUI, which includes all interfaces for training, testing, and validation.

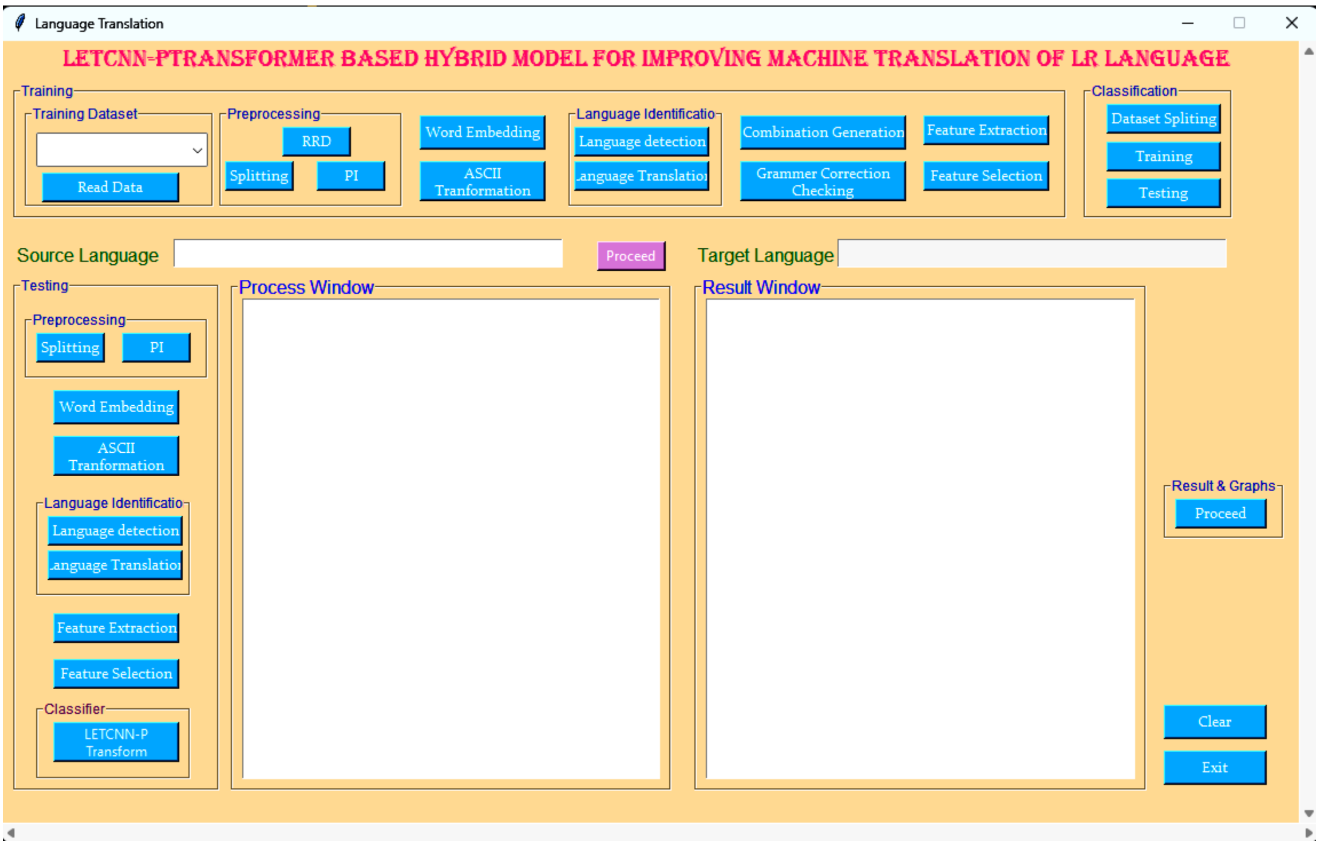

Fig 6. GUI Screen Snapshot

A. Quantitative Result Analysis

In the context of MT evaluation, the sensitivity score is important in MT evaluation because it indicates the model's capacity to correctly identify meaningful translations, or the proportion of true positives detected by the model out of all actual positive cases in the data. A higher sensitivity score shows that the model is good at capturing and incorporating relevant translations into its output, reducing the risk of missing important information during the translation process. It refers to the model's ability to interpret and accurately translate the meaning of the source text into the target language, thereby improving the overall quality and efficacy of the translation output. A high sensitivity score indicates the model's ability to capture the substance and nuances of the input text, leading to more trustworthy and complete translations. The proposed model’s sensitivity score comparison is illustrated in Fig. 7.

The F-measure comparison in Fig. 8 provides compelling insights into the performance of several translation models, with the proposed LETCNN-PT technique standing out as the best among them. With an F-measure of 98.88, LETCNN-PT beats existing models like CNN, RNN, DBN, and DNN, which have F-measures ranging from 91.54 to 96.77. This substantial superiority demonstrates the LETCNN-PT approach's efficiency in striking a harmonious balance between precision and recall, leading to more

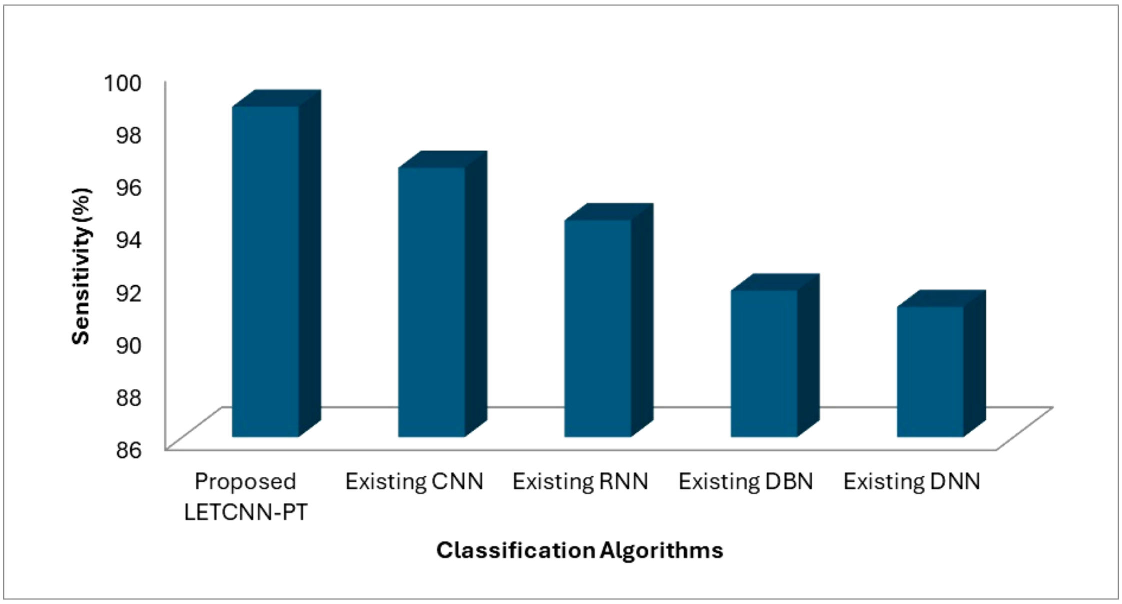

Fig 7. Sensitivity Score Comparison

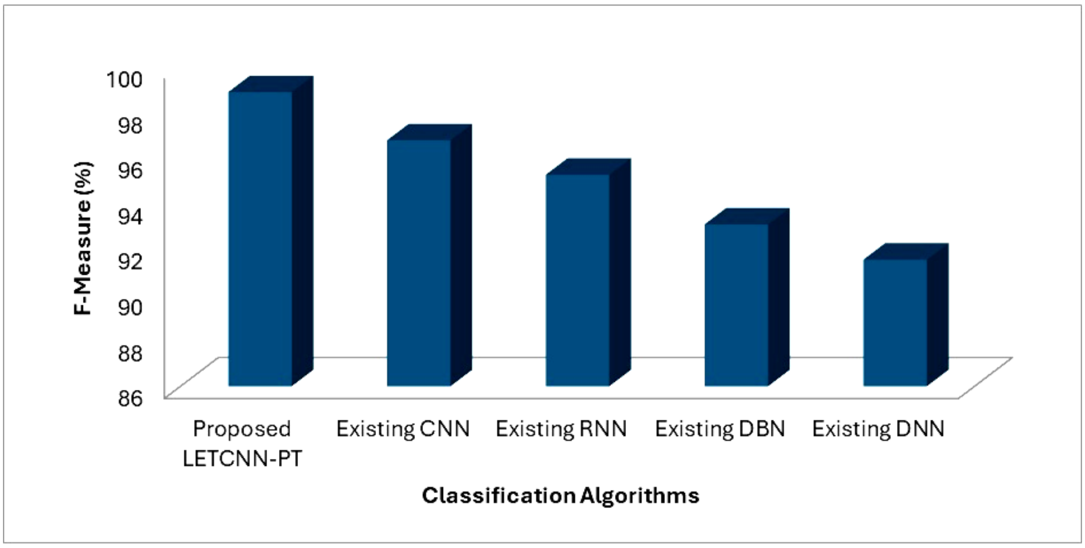

Fig 8. F-Measure Comparison

accurate and thorough translation results. We can attribute LETCNN-PT's greater F-measure to its distinctive architecture, which presumably enables better feature extraction, deeper semantic understanding, and more robust modelling of language nuances. LETCNN-PT may also work better because it uses advanced methods like Log Exponential Tanh activation functions and P-Transformer decoding. This shows that it has the potential to become the best MT system ever.

A comparison of BLEU scores demonstrates a significant performance difference between the proposed LETCNN-PT approach and existing models, shown in Fig. 9. With an amazing BLEU score of 0.85, LETCNN-PT outperforms all other approaches, including established CNN, RNN, DBN, and DNN models with BLEU ratings ranging from 0.62 to 0.76. This mismatch highlights the LETCNN-PT method's ability to generate translations that nearly match human references, implying that it can produce high-quality results. The large BLEU score difference between LETCNN-PT and standard models highlights the method's potential to dramatically improve machine translation skills, paving the way for more precise and reliable translation results.

"WOPBLEU" stands for "Word Overlap BLEU." WOPBLEU is a variant of the BLEU (Bilingual Evaluation Understudy) metric, a widely used tool for evaluating the quality of machine-generated translations. Traditional BLEU measures the n-gram overlap between the reference and candidate translations, whereas WOPBLEU calculates the proportion of overlapping words at the token level. This allows for a more fine-grained evaluation of translation quality, particularly when the candidate translation differs significantly from the reference at the sentence level but still contains relevant words and phrases. WOPBLEU improves the accuracy and appropriateness of translations, especially in situations where exact word-for-word correspondence is required to communicate the intended meaning. It adds to

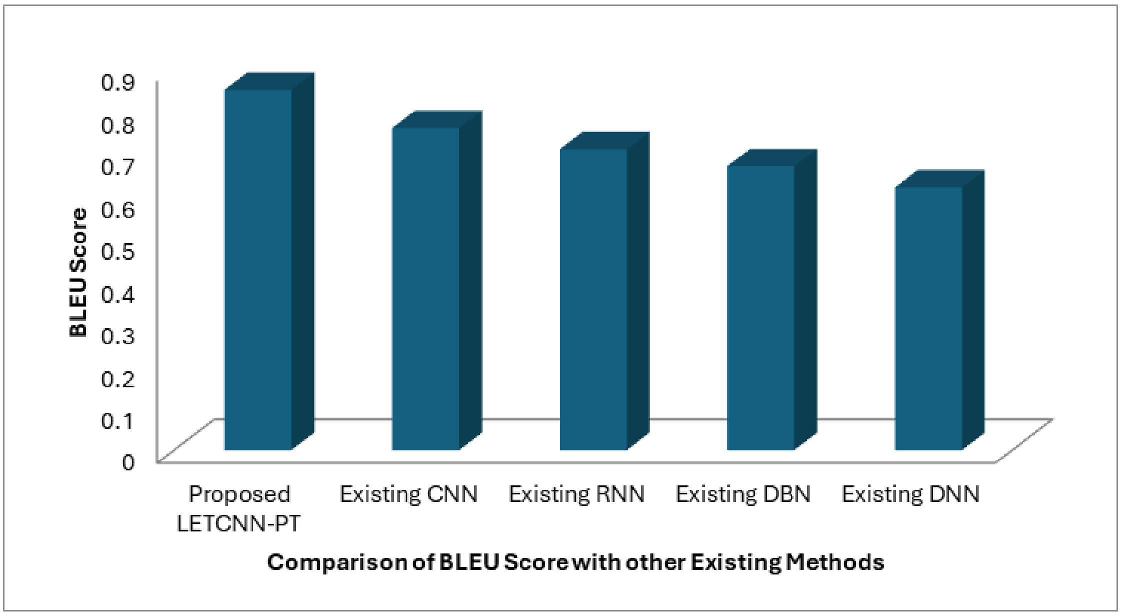

Fig 9. BLEU Score Comparison

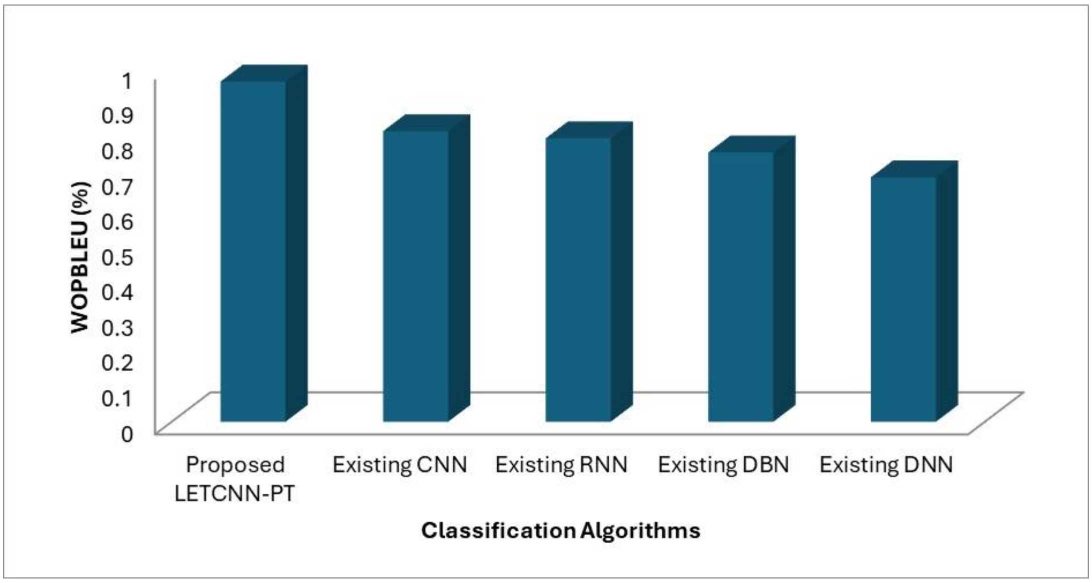

Fig 10. WOPBLEU Score Comparison

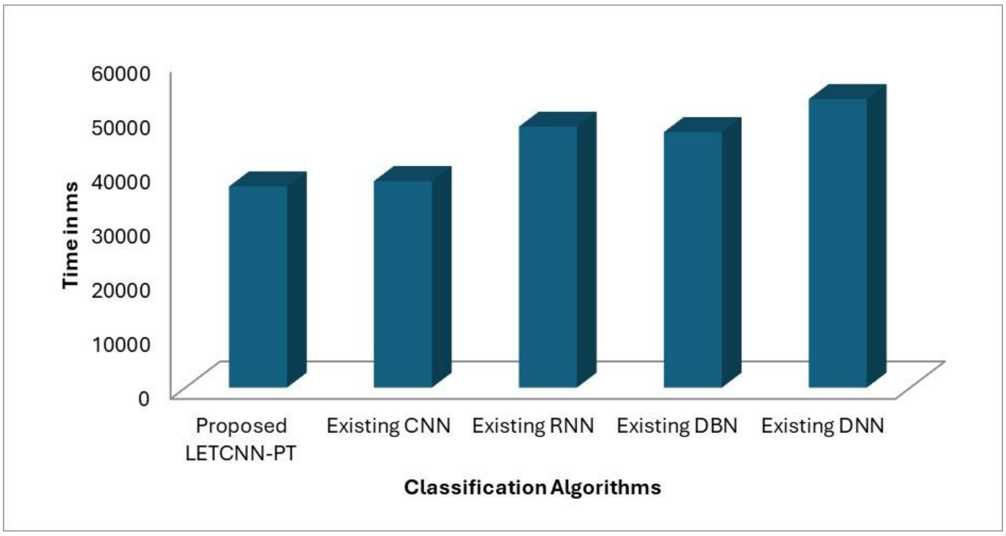

Fig 11. Model's Training Time Taken Comparison

traditional BLEU by providing a more granular study of translation quality at the word level, which is useful for fine-tuning machine translation systems and evaluating their performance in specific linguistic situations. Fig. 10 indicates the suggested model's superior performance in terms of WOPBLEU score compared to its competitor techniques.

The comparison of training times (Fig. 11) between methods offers useful information about their efficiency and computing demands. Notably, the LETCNN-PT method has a significantly shorter training duration, at 37,010 milli-seconds. This demonstrates its ability to train models, optimize resource consumption, and accelerate learning. However, both the present RNN and DBN models have significantly longer training periods, exceeding the 48,000-ms mark. The existing CNN model falls somewhere in the middle, requiring 38,004 ms to train. These inequalities highlight the differences in model complexity, architectural design, and optimization tactics used by each method. The LETCNN-PT's shorter training time means that it can produce equivalent or better performance results while using fewer computer resources. As a result, it appears as a promising solution for real-world applications where reducing training time is critical.

B. *Qualitative Result Analysis*

When it comes to AI and machine learning, knowing how various models perform and behave relies heavily on qualitative outcome analysis. To shed light on the method's merits and shortcomings, we conducted a qualitative examination of the data produced by our suggested approach in this part. We can obtain a more detailed understanding of the suggested method's performance in various contexts by closely examining individual translations, error patterns, and linguistic subtleties. To better understand the practicality and reliability of the suggested method, qualitative evaluations unearth subtle facets of translation accuracy, language faithfulness, and semantic consistency. In addition to quantitative measurements, this qualitative analysis provides a thorough assessment of the suggested method's efficacy and its consequences for real-world applications in NLP and MT.

Fig. 12 and 13 demonstrate the machine translation outcomes employing web translation and our suggested LETCNN-P Transformer model for the sentence "complete multilingual neural machine," respectively. Inspection reveals that the proposed model generates a very accurate translation for the specified sentence.

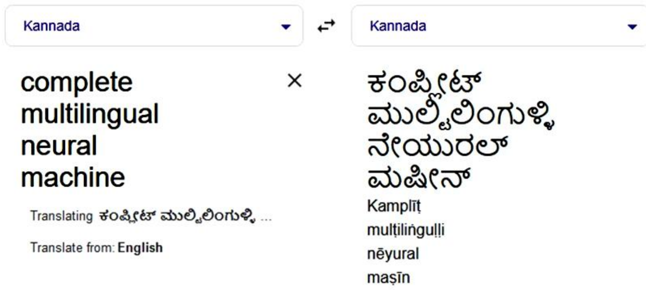

Fig 12. Translation using Web Translation

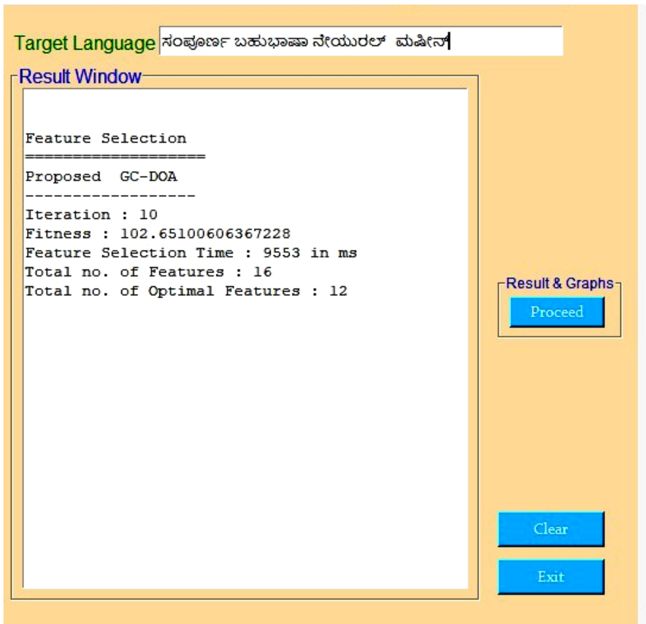

Fig 13. Translation Using Proposed LETCNN-P Transformer Model

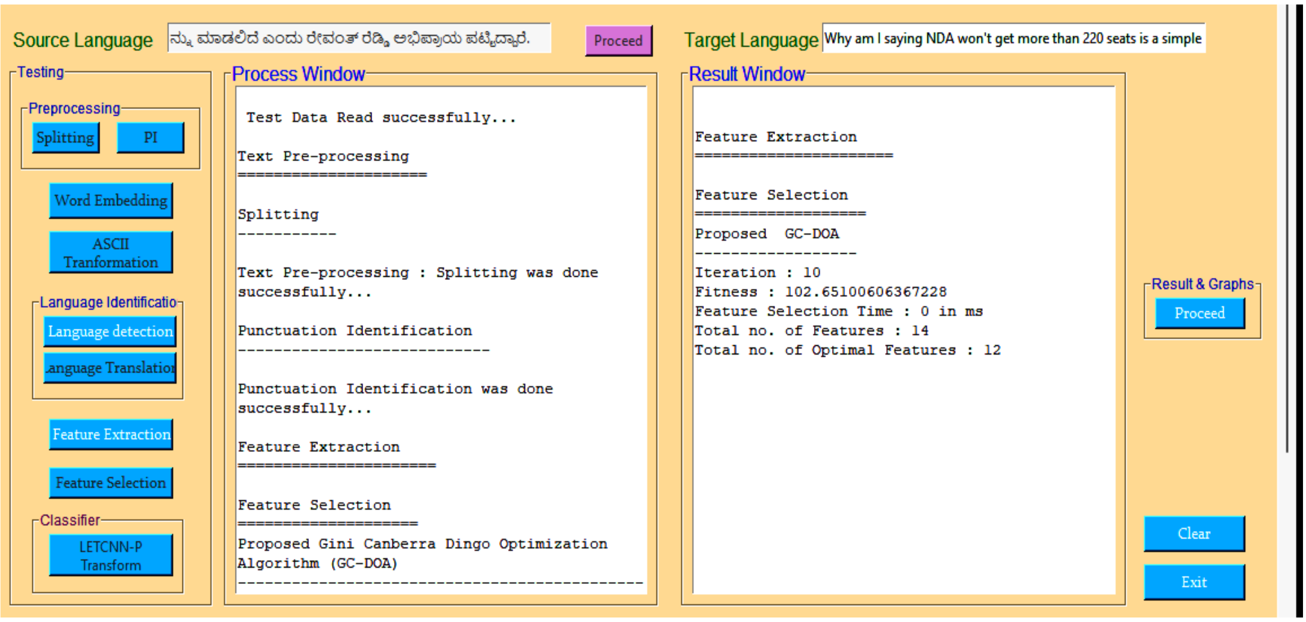

Fig 14. Kannada to English Long Sentence Translation Example

The two translations are compared to show how well our algorithm captures the complexities and nuances of the input text. Especially, our LETCNN-P Transformer model produces a translation that is fairly accurate to the original content and structure of the input text. This finding demonstrates the accuracy and resilience of our methodology in managing multilingual translation tasks and its applicability in practical situations where accurate and trustworthy translations are crucial.

The finding emphasizes the difficulties current MT systems face, especially when processing long texts in languages like Kannada that have intricate grammatical structures. The realization of these drawbacks underscores the need for further development in MT technology to address these issues and enhance the translation quality of longer texts.

Fig. 14 shows how well the proposed model can capture word dependencies and contextual information while accurately translating lengthy Kannada sentences to English. Examining the translation output reveals that the suggested model clearly performs well in maintaining the coherence and fidelity of the source text during the translation process. The model performs admirably, as evidenced by its high BLEU score, in large part because of its ability to maintain word dependencies and background. The suggested method produces translations that are quite like human-generated equivalents, indicating a sophisticated grasp of Kannada sentence structure and semantics. This emphasizes how well the model captures the intricacies of the original language and produces natural and contextually suitable English translations.

C. Error Analysis

Conducting a comprehensive error analysis of the LETCNN-P Transformer model for NMT is crucial to understand its limitations and pinpoint areas for improvement. Handling intricate syntactic patterns is a significant obstacle encountered in the translations of the LETCNN-P Transformer model. Although the suggested model translation is generally precise, it occasionally struggles to preserve the appropriate word order in longer and more syntactically intricate lines, resulting in errors such as word misplacement. Another prevalent error category pertains to the translation of idiomatic expressions and words that are distinctive to a particular culture. Google's textual translation and the proposed model both fail to accurately convey the idiomatic term. This emphasizes the necessity of a broader and more varied training dataset that encompasses idiomatic and culturally nuanced expressions. The model sometimes encounters challenges when it comes to rare or infrequently used terms, resulting in inaccurate translations. Data augmentation approaches, such as back-translation, enhance the ability to manage infrequent terms. Fig. 15 shows the distribution of errors.

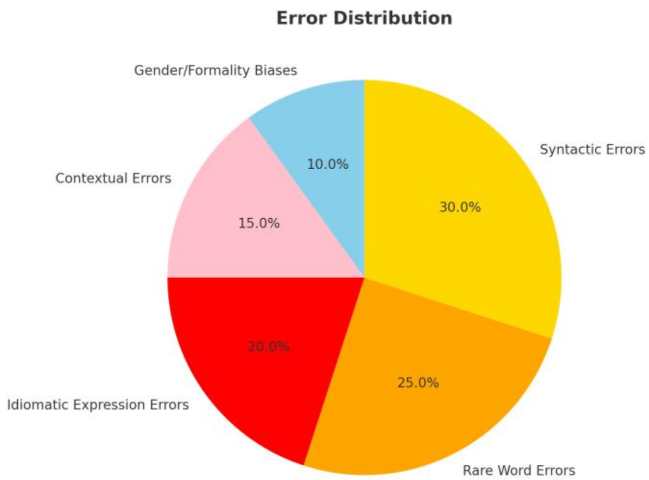

Fig 15. Distribution of Translation Errors

V. Ablation Studies

To further understand the contributions of individual components in the LETCNN-P Transformer model, we conducted a series of ablation studies. Ablation studies involve systematically removing or modifying specific elements of the model to assess their impact on overall performance. This approach helps isolate the effects of each component and determine their significance in achieving the observed results.

A. Impact of Log Exponential Tanh Activation

One of the pivotal innovations in the LETCNN-P Transformer model is the integration of the Log Exponential Tanh (LET) activation function within the CNN layers. This specialized activation function is designed to enhance the model's capability to capture intricate patterns and dependencies within the data, which are crucial for high-quality MT. To rigorously assess the contribution of the LET activation function, we conducted a series of experiments where we systematically replaced LET with more conventional activation functions, such as ReLU (Rectified Linear Unit) and sigmoid.

The results from these experiments were quite revealing. When the LET activation function was replaced with ReLU or sigmoid, there was a significant drop in the model's performance. Specifically, the BLEU scores, which measure the accuracy of the translated text compared to a set of reference translations, decreased by approximately 2-3 points, as shown in Fig. 16. Similarly, the F-measure, which is a harmonic mean of precision and recall, also dropped by a similar margin. These declines in performance metrics suggest that the LET activation function is integral to the model's ability to process and translate text accurately. The superior performance of the LETCNN-P Transformer model with the LET activation function can be attributed to its enhanced ability to capture non-linear relationships and dependencies in the data, which are essential for understanding and translating complex linguistic structures.

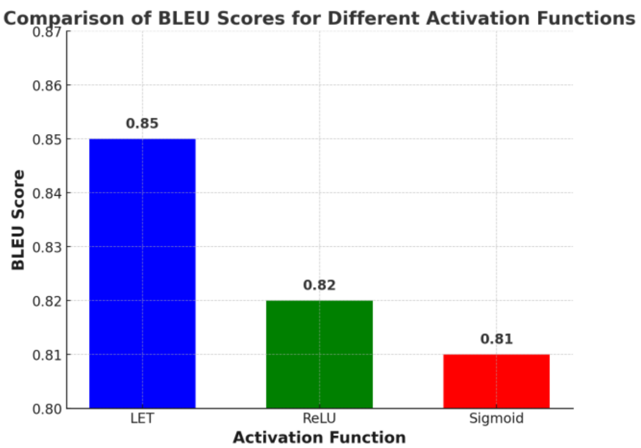

Fig 16. BLEU Score for Different Activation Function

B. Role of Subword Tokenization

Subword tokenization plays a pivotal role in the LETCNN-P Transformer model, particularly in handling LRLs with complex morphological structures. In languages like Kannada, where words can be highly inflected and composed of multiple morphemes, subword tokenization effectively breaks down words into smaller, more manageable units. One of the primary advantages of subword tokenization is its ability to mitigate the issue of rare words. In traditional word-level tokenization, rare words that do not appear frequently in the training corpus can be problematic for the model. By breaking down words into subword units, the model can learn representations that are applicable to a broader range of words. To evaluate the impact of subword tokenization, we conducted experiments comparing the LETCNN-P Transformer model's performance with and without subword tokenization.

The results demonstrated a noticeable decline in translation quality, with BLEU scores decreasing by 3-4 points. This suggests that subword tokenization effectively mitigates the issue of rare words and improves the model's ability to generalize across different linguistic contexts.

C. Effect of Data Augmentation Techniques

Data augmentation (DA) is a crucial technique used to enhance the performance of ML models, particularly in scenarios where the availability of training data is limited. One of the most effective DA methods in the context of MT is back translation. Back translation involves translating target language sentences back into the source language, and then using these newly generated source sentences to augment the training dataset. These back-translated sentences were then paired with their original target language counterparts to form new parallel sentence pairs. By incorporating these pairs into the training dataset, we aimed to provide the model with more diverse linguistic constructs and enhance its ability to handle various translation scenarios.

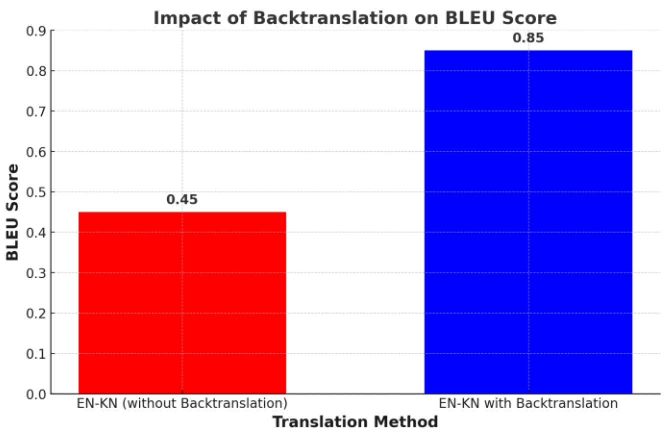

Fig 17. BLEU Score with and without Backtranslation

Fig. 17 illustrates a significant enhancement in BLEU scores for the EN-KN translation model when back translation is utilized. Without back translation, the model achieves a BLEU score of 0.45. However, with the implementation of back translation, the BLEU score dramatically increases to 0.85. This substantial improvement underscores the effectiveness of back translation in enriching the training dataset, thereby enabling the model to produce more accurate and contextually appropriate translations.

VI. Conclusion

The paper introduces the LETCNN-P Transformer model, a novel approach for multilingual NMT. This model demonstrates substantial advancements over existing techniques, as evidenced by superior performance metrics such as BLEU scores, F-measures, and efficient training times. Qualitative analysis further validates its capability to accurately capture word dependencies and contextual nuances, ensuring fluent and contextually appropriate translations. Ablation studies conducted as part of this research highlight the critical contributions of individual components within the LETCNN-P Transformer model. The LET activation function significantly enhances the model's ability to capture complex patterns, while subword tokenization effectively handles the linguistic variability in low-resource languages. Additionally, the P-Transformer architecture in the decoder phase and the application of data augmentation techniques, particularly back translation, were shown to play vital roles in achieving high translation quality. These findings underscore the LETCNN-P Transformer's significance in overcoming challenges in multilingual NMT and transforming translation technology. Future research could focus on enhancing the model's proficiency with diverse language pairs, optimizing computational efficiency, and validating its practical utility in real-world scenarios. The LETCNN-P Transformer represents a notable stride forward in MT, poised to expand technological boundaries and meet evolving global communication demands.

REFERENCES

[1] M. I. Devi and B. S. Purkayastha, "An Exploratory Study of SMT Versus NMT for the Resource Constraint English to Manipuri Translation," *IOT with Smart Systems*, pp. 329–338, 2023, doi: 10.1007/978-981-99-3761-5_31.

[2] H. Wang, H. Wu, Z. He, L. Huang, and K. W. Church, "Progress in Machine Translation," *Engineering*, vol. 18, pp. 143–153, Nov. 2022, doi: 10.1016/j.eng.2021.03.023.

[3] X. Wu, Y. Xia, J. Zhu, L. Wu, S. Xie, and T. Qin, "A study of BERT for context-aware neural machine translation," *Machine Learning*, vol. 111, no. 3, pp. 917–935, Jan. 2022, doi: 10.1007/s10994-021-06070-y.

[4] F. Stahlberg, "Neural Machine Translation: A Review," *Journal of Artificial Intelligence Research*, vol. 69, pp. 343–418, Oct. 2020, doi: 10.1613/jair.1.12007.

[5] S. Ranathunga, E.-S. A. Lee, M. Prifti Skenduli, R. Shekhar, M. Alam, and R. Kaur, "Neural Machine Translation for Low-resource Languages: A Survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–37, Feb. 2023, doi: 10.1145/3567592.

[6] O. Ahia, J. Kreutzer, and S. Hooker, "The Low-Resource Double Bind: An Empirical Study of Pruning for Low-Resource Machine Translation," *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, **Published**, doi: 10.18653/v1/2021.findings-emnlp.282.

[7] S. Saxena, S. Chauhan, P. Arora, and P. Daniel, "Unsupervised SMT: an analysis of Indic languages and a low resource language," *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 1–20, Sep. 2022, doi: 10.1080/0952813x.2022.2115142.

[8] R. Dabre, C. Chu, and A. Kunchukuttan, "A Survey of Multilingual Neural Machine Translation," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–38, Sep. 2020, doi: 10.1145/3406095.

[9] S. K. Sheshadri, D. Gupta, and M. R. Costa-Jussà, "A Voyage on Neural Machine Translation for Indic Languages," *Procedia Computer Science*, vol. 218, pp. 2694–2712, 2023, doi: 10.1016/j.procs.2023.01.242.

[10] A. Hegde and H. L. Shashirekha, "KanSan: Kannada-Sanskrit Parallel Corpus Construction for Machine Translation," *Communications in Computer and Information Science*, pp. 3–18, 2023, doi: 10.1007/978-3-031-33231-9_1.

[11] S. B. Das, D. Panda, T. K. Mishra, B. Kr. Patra, and A. Ekbal, "Multilingual Neural Machine Translation for Indic-to-Indic Languages," *ACM Transactions on Asian and Low-Resource Language Information Processing*, Mar. 2024, **Published**, doi: 10.1145/3652026.

[12] M. Maimaiti, Y. Liu, H. Luan, and M. Sun, "Data augmentation for low-resource languages NMT guided by constrained sampling," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 30–51, Aug. 2021, doi: 10.1002/int.22616.

[13] R. Nagai and A. Maeda, "Dataset Augmentation for Grammatical Error Correction Using Markov Chain," Lecture Notes in Engineering and Computer Science, ISSN: 2078-0966, pp. 97-100, 2021.

[14] R. Rubino, B. Marie, R. Dabre, A. Fujita, M. Utiyama, and E. Sumita, "Extremely low-resource neural machine translation for Asian languages," *Machine Translation*, vol. 34, no. 4, pp. 347–382, Dec. 2020, doi: 10.1007/s10590-020-09258-6.

[15] N. Bansal, G. Datta, and A. Singh, "Experimentation with NMT models on low resource Indic languages," *2021 Sixth International Conference on Image Information Processing (ICIIP)*, Nov. 2021, **Published**, doi: 10.1109/iciip53038.2021.9702577.

[16] S. Sen, M. Hasanuzzaman, A. Ekbal, P. Bhattacharyya, and A. Way, "Neural machine translation of low-resource languages using SMT phrase pair injection," *Natural Language Engineering*, vol. 27, no. 3, pp. 271–292, Jun. 2020, doi: 10.1017/s1351324920000303.

[17] A. Hegde, H. L. Shashirekha, A. K. Madasamy, and B. R. Chakravarthi, "A Study of Machine Translation Models for Kannada-Tulu," *Third Congress on Intelligent Systems*, pp. 145–161, 2023, doi: 10.1007/978-981-19-9225-4_12.

[18] A. V. Hujon, T. D. Singh, and K. Amitab, "Transfer Learning Based Neural Machine Translation of English-Khasi on Low-Resource Settings," *Procedia Computer Science*, vol. 218, pp. 1–8, 2023, doi: 10.1016/j.procs.2022.12.396.

[19] M. Nivaashini, G. Priyanka, and S. Aarthi, "Deep Neural Machine Translation (DNMT)," *Automatic Speech Recognition and Translation for Low Resource Languages*, pp. 331–373, Mar. 2024, doi: 10.1002/9781394214624.ch16.

[20] R. Baruah, R. K. Mundotiya, and A. K. Singh, "Low Resource Neural Machine Translation: Assamese to/from Other Indo-Aryan (Indic) Languages," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 1, pp. 1–32, Nov. 2021, doi: 10.1145/3469721.

[21] Y. Li, J. Li, J. Jiang, S. Tao, H. Yang, and M. Zhang, "P-Transformer: Towards Better Document-to-Document Neural Machine Translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 3859–3870, 2023, doi: 10.1109/taslp.2023.3313445.

[22] G.-H. Choi, J.-H. Shin, Y.-H. Lee, and Y.-K. Kim, "Toward Understanding Most of the Context in Document-Level Neural Machine Translation," *Electronics*, vol. 11, no. 15, p. 2390, Jul. 2022, doi: 10.3390/electronics11152390.

[23] H.-I. Liu and W.-L. Chen, "Re-Transformer: A Self-Attention Based Model for Machine Translation," *Procedia Computer Science*, vol. 189, pp. 3–10, 2021, doi: 10.1016/j.procs.2021.05.065.

[24] S. Xie, Y. Xia, L. Wu, Y. Huang, Y. Fan, and T. Qin, "End-to-end entity-aware neural machine translation," *Machine Learning*, vol. 111, no. 3, pp. 1181–1203, Jan. 2022, doi: 10.1007/s10994-021-06073-9.

[25] R. Weng, H. Yu, S. Huang, S. Cheng, and W. Luo, "Acquiring Knowledge from Pre-Trained Model to Neural Machine Translation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 9266–9273, Apr. 2020, doi: 10.1609/aaai.v34i05.6465.

[26] M. Singh, R. Kumar, and I. Chana, "Corpus based Machine Translation System with Deep Neural Network for Sanskrit to Hindi Translation," *Procedia Computer Science*, vol. 167, pp. 2534–2544, 2020, doi: 10.1016/j.procs.2020.03.306.

[27] A. F. Adoma, N.-M. Henry, and W. Chen, "Comparative Analyses of Bert, Roberta, Distilbert, and Xlnet for Text-Based Emotion Recognition," *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Dec. 2020, **Published**, doi: 10.1109/iccwamtip51612.2020.9317379.

[28] J. McNulty, S. Alvarez, and M. Langmayr, "Detecting Research from an Uncurated HTML Archive Using Semi-Supervised Machine Learning," *2021 Systems and Information Engineering Design Symposium (SIEDS)*, Apr. 2021, **Published**, doi: 10.1109/sieds52267.2021.9483725.

[29] P. Chakraborty *et al.*, "Recognize Meaningful Words and Idioms from the Images Based on OCR Tesseract Engine and NLTK," *Lecture Notes in Electrical Engineering*, pp. 297–310, 2022, doi: 10.1007/978-981-19-1520-8_23.

[30] H. Navidan *et al.*, "Generative Adversarial Networks (GANs) in networking: A comprehensive survey & evaluation," *Computer*

Networks, vol. 194, p. 108149, Jul. 2021, doi: 10.1016/j.comnet.2021.108149.

[31] L. Yujian and L. Bo, "A Normalized Levenshtein Distance Metric," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1091–1095, Jun. 2007, doi: 10.1109/tpami.2007.1078.

[32] D. K. Po, "Similarity Based Information Retrieval Using Levenshtein Distance Algorithm," *International Journal of Advances in Scientific Research and Engineering*, vol. 06, no. 04, pp. 06–10, 2020, doi: 10.31695/ijasre.2020.33780.