International Journal of
**INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING**
**ISSN:2147-6799** **www.ijisae.org** **Original Research Paper**

# Trans-BILSTM Based Speech and Speaker Recognition using Spectral, Cepstral and Deep Features

**Sukumar B. S.[1], Dr. G. N. Kodanda Ramaiah[2], Dr. Sarika Raga[3], Dr. Lalitha Y. S.[4]**

**Submitted:** 05/02/2024 **Revised:** 13/03/2024 **Accepted:** 20/03/2024

**Abstract:** This paper introduces a new Speech and Speaker Recognition System that utilises a Deep Feature Extraction method with a Heuristic Adopted Transformer Bidirectional Long Short Term Memory (LSTM) and Attention Mechanism. The aim of our approach is to address the difficulties associated with effectively recognising speech and identifying speakers in complex audio data. Our system utilises deep learning methods like bidirectional LSTM and attention mechanism in a Transformer framework to extract temporal and long-range relationships in voice input, improving identification accuracy. We compare the proposed model with existing methods like the flow detection algorithm and reptile search algorithm. The experimental findings reveal that our system surpasses existing algorithms in accuracy, precision, recall, and F1-score, demonstrating its value in **Speech** and speaker recognition tasks. Our research enhances the area of deep learning-based audio analysis and provides a reliable solution for real-world applications that need precise voice and speaker recognition.

***Keywords:*** *Long short term memory (LSTM), Speech Recognition, Speaker Recognition.*

## 1. Introduction

Speaker recognition began by drawing inspiration from human communication methods. Just like technical writing, humans can communicate with machines through voice. There are different ways to interact with machines, such as using a keyboard and other devices. One significant disadvantage of these devices is that they are all very slow and cumbersome[1]. Therefore, facilitating the interaction between humans and machines. Speaker recognition involves a constantly evolving biometric challenge. Originating from the broader field of speech processing. Just like other speech-related recognition tasks such as speech recognition and language recognition, speaker recognition is a multidisciplinary issue. Speaker recognition covers speaker verification, identification, speaker segmentation, and indexing[2]. The speaker recognition system aims to confirm a user's identity claim, while speaker identification focuses on identifying the best match from the speaker models in the database. A system for speaker recognition identifies individuals based on their voice. In general, after speaker recognition, the subsequent crucial phase for a natural conversation is identifying whose command or speech is being processed. With the use of speech recognition technology, computers can comprehend spoken language and translate it into text or instructions. It recognizes patterns in voice signals by using algorithms to analyze audio input, then compares those patterns to words or phrases that are already known. Contemporary systems aim for high accuracy across various speakers, accents, and situations, often using deep learning methods. Speech recognition technology is widely utilized in virtual assistants, healthcare, automotive, and customer service to improve efficiency, accessibility, and hands-free contact.

In order to address this challenge, researchers have thoroughly investigated deep learning techniques, which have shown impressive effectiveness in different pattern recognition tasks. Utilizing advanced feature extraction methods like convolutional neural networks (CNNs) or recurrent neural networks (RNNs) in conjunction with robust neural network models for speaker and speech recognition. Moreover, deep learning techniques enable end-to-end learning, enabling the system to automatically extract discriminative features directly from raw speech signals. Several widely used techniques in this field are Gaussian Mixture Model—Universal Background Model (GMM-UBM), Support Vector Machine (SVM) based on GMM super vector compensated using Nuisance Attribute Projection (NAP), Joint Factor Analysis (JFA), and i-vector/PLDA). [3] The study attempts to address several speaker modelling methods, such as Gaussian Mixture Model (GMM), Vector Quantization (VQ), and Neural Networks (NN), among others. Additionally, many methods for obtaining voice characteristics are covered, including Linear Predictive Coding (LPC) and Mel Frequency Cepstral Coefficients (MFCC). Additionally, a thorough examination of these surveyed methods is conducted in order to determine their benefits and drawbacks. [4] This study presents a deep learning-based Persian phoneme identification algorithm. A Deep Bidirectional Long Short-Term Memory (DBLSTM) model with a Connectionist Temporal Classification (CTC) output layer and a deep belief network (DBN) for speech signal feature extraction enhance accuracy. Deep neural networks (DNNs) outperform

[1] *Research Scholar, Department of Electronics and Communication, Visvesvaraya Technological University, Macche, Belagavi, Karnataka, India. sukumar.svm@gmail.com*
[2] *Professor and HOD, Department of Electronics and Communication, Kuppam Engineering College, Kuppam Andhra Pradesh, India. gnk.ramaiah@gmail.com*
[3] *Associated Professor, Department of Electronics and Communication, Visvesvaraya Technological University PG Centre, Muddenahalli, Chikkaballapura, Karnataka, India. raga.sarika@gmail.com*
[4] *Professor, Department of Electronics and Communication, Don Bosco Institute of Technology, Bangalore, Visvesvaraya Technological University Karnataka, India. patil.lalitha12@gmail.com*

shallow networks, and bidirectional construction improves accuracy. [5] study presents modifications to DNN-based extractor systems to enhance speaker identification performance and demonstrated that the discriminatively trained similarity metric learning technique outperforms the traditional LDAPLDA method as an embedding backend. Test results on Speakers in the Wild and NIST SRE 2016 show the proposed systems' resilience in real-life scenarios. [6] The suggested approach uses a unique architecture to enable optimal use of information in brief voice segments for speaker detection. UtterIdNet was trained and evaluated on the newest speaker recognition benchmarks, VoxCeleb. Results indicate steady performance for short portions, with considerable improvement over various durations. [7] SVM was used to develop the acoustic model of a Speech Recognition System employing MFCC and LPC features for Azerbaijani dataset. Analysis of SVM kernel functions like radial basis and polynomial kernels improves recognition performance over neural network models.

In spite of the fact that the vast majority of research in this area has primarily concentrated on either Speaker Identification or Speech Identification, to address these deficiencies we introduced a ground breaking technique for the Deep Feature Extraction-based speech and Speaker Recognition System. Unlike traditional systems that rely exclusively on speaker or speech identification, our system employs the Heuristic Adopted Transformer Bidirectional Long Short Term Memory (TransBiLSTM) architecture with Attention Mechanism. The paper is structured with a background in the 2nd section, delving into the context and previous research in speech and speaker recognition. In the third section, the methodology is discussed, detailing the development of the Deep Feature Extraction-based Speech and Speaker Recognition System, which includes advanced techniques such as the TransBiLSTM model with Attention Mechanism. Finally, the 4th section showcases the experimental results of the study.

## 2. Background

### 2.1 Bidirectional LSTM

The LSTM network has shown that it is capable of doing strong modeling in sequential learning tasks, such as named item tagging. The purpose of this work is to include bidirectional LSTM layers into the transformer model in order to significantly enhance the performance of the transformer. In the research carried out by [8], an effort is made to reduce a BERT model to a single-layer bidirectional LSTM model structure. Due to the fact that both of us are using bidirectional LSTM, it is pertinent to our work. However, when compared to the baseline models developed by BERT, their work results in a lower level of accuracy. In our tests, we demonstrate that the use of the BLSTM model alone (even with many stacked BLSTM layers) leads to much inferior outcomes compared to the usage of BERT models. This is similar to the discovery that they made. On the other hand, the joint modeling framework that we have suggested, which is called TRANS-BLSTM, has the capability of improving the accuracy of the transformer BERT models.

### A. TRANS and Proposed TRANS-BLSTM Architecture

In this part, we begin by examining the architecture of the transformer, and then we proceed to present the transformer bidirectional LSTM network designs (TRANS-BLSTM), which integrates the BLSTM to either the encoder or the decoder of the transformer.

**I. Transformer architecture (TRANS):** The BERT model is made up of a transformer encoder and for both the encoder and the decoder, the original transformer design makes use of many stacked self-attention layers in addition to point-wise completely linked layers. For sequential classification tasks (such as named entity tagging and question answering), as well as sentence classification tasks (such as sentiment classification), BERT only uses the encoder to generate hidden value representation. The original transformer decoder, which was used for generating text in neural machine translation and other applications, is replaced by a linear layer, which is then followed by a softmax layer. Depending on whether the BERT-base or -large scenario is being considered, the encoder is made up of a stack that has either N = 12 or N = 24 layers. The sublayers that make up each layer are two in number. In the first sub-layer, there is a mechanism for multi-head self-attention, and in the second sub-layer, there is a feed-forward network that is basic, position-wise, and completely linked. Through the use of a residual connection surrounding each of the two sub-layers, which is then followed by the normalizing of the layers. The output of each sublayer is denoted by the expression

*LayerNorm (x + Sublayer(x)),*

where Sublayer (x) refers to the function that is implemented by the sublayer itself. All of the sub-layers in the model, in addition to the embedding layers, generate outputs with dimensions of 768 and 1024 for BERT-base and BERT-large, respectively, in order to make it easier to establish these residual connections. In this study, we used the multi-head self-attention method that was utilized in the initial work [9]. In the BERT study [10], they utilized the same input and output representations, which are the embedding and positional encoding. Additionally, they used the same loss goal, which is the masked LM prediction and the next sentence prediction.

**II. Proposed Transformer Bidirectional LSTM (TRANS-BLSTM) Architectures:** Studies conducted in the past suggested that a bidirectional LSTM model by itself would not be able to achieve the same level of performance as a transformer. An example of this would be the distillation of a transformer model into a single-layer bidirectional LSTM model (Tang et al., 2019), which resulted in a much lower level of accuracy. Moreover, this was shown by the results of their tests. The hypothesis that we provide in this work is that the transformer and the bidirectional LSTM has the potential to be complimentary in sequence modelling. The fact that a bidirectional LSTM has the potential to further increase accuracy in downstream tasks in comparison to a traditional transformer model is what drives us to examine this possibility. Figure 1 illustrates the suggested Transformer with Bidirectional LSTM designs, which are referred to as the TRANS-BLSTM models according to their distinct features:

**(a) TRANS-BLSTM**

By introducing a bidirectional LSTM (Long Short-Term Memory) layer into our model architecture, we are able to improve upon the BERT layer that are already there. In the same way that the original BERT layer processes the input sequence, this bidirectional LSTM layer does the same thing. Prior to the implementation of Layer Normalization, the output of the bidirectional LSTM layer is merged with the output of the first BERT layer. This occurs after the input sequence has been processed. The information that was gathered by the bidirectional LSTM and that which was retrieved by the BERT layer are successfully combined via the process of summation, which is how this combination is accomplished. We hope that by including bidirectional LSTM, we will be able to extract more contextual information from the input sequence. This will allow us to improve the overall performance and efficacy of the model in subsequent tasks, such as the comprehension of natural language or the analysis of sentiment. As we shall see in the experiments that follow, the purpose of including BLSTM is to merge the self-attention and bidirectional LSTM in order to develop a stronger joint model framework. This will be shown in the experiments. We concentrate on the latter (TRANS-BLSTM), and for the sake of convenience, we will refer to this model as TRANS-BLSTM from this point forward. We acquire the BLSTM output with a dimension of 2H for both architectures if we employ the same number of BLSTM hidden units as we did for the BERT model H. As a result, we need a linear layer to project the output of the BLSTM (with dimensions 2H) to H in order to guarantee that it is identical to the output of the transformer. Alternately, if we set the number of hidden units in the BLSTM to H/2 (we will refer to this model as TRANS-BLSTM-SMALL), then we will not be need to incorporate an extra projection layer.

**(b) Adding Bidirectional LSTM to Transformer Decoder**

While the approach described above adds bidirectional LSTM layers to a transformer encoder, we also have the option of replacing the linear layer in the decoder with bidirectional LSTM layers. The bidirectional LSTM layers in the encoder are helpful for the pre-training job for the masked language model and the next sentence prediction task. On the other hand, the bidirectional LSTM layers in the decoder may be helpful for downstream sequential prediction tasks such as question answering

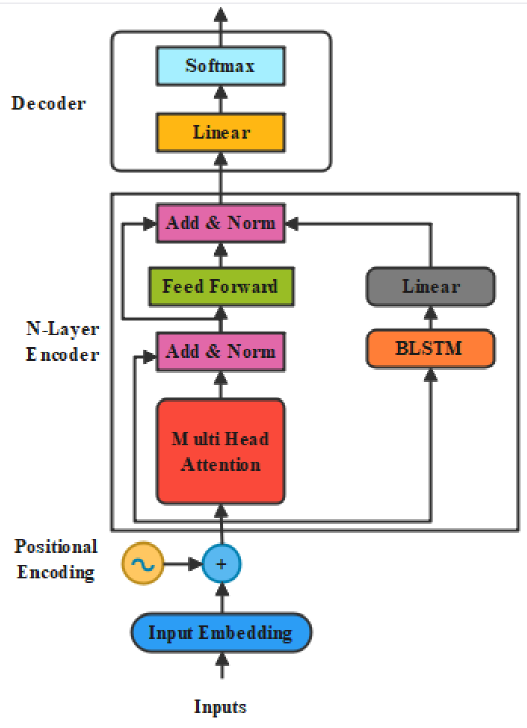


**Fig 1** TRANS-BLSTM-2, adds a BLSTM layer in parallel

## 2.2 Flow Direction Algorithm

The FDA algorithm is based on the D8 approach for predicting flow direction in a drainage basin after converting rainfall to runoff. Initially, this method will generate a population in the drainage basin or issue search space at the beginning of the process. Once this is accomplished, the flows are directed to a site at a lower height, with the objective of reaching the optimal answer or the outflow point at the lowest altitude. In order to function properly, the method relies on the following assumptions:

1. There is a place and a height associated with each flow.
2. A number of β points are located around every flow, and each of these sites has a height or goal function.
3. There is a direct correlation between the slope and the flow movement velocity.
4. A velocity of V is shown by the flows, and they are moving in the direction that has the lowest height.
5. The flow location that has the most objective function that is ideal is the place at which the basin outflow is located.

The starting parameters of the algorithm consist of the population number α, the number of neighbours β, and the neighbourhood radius Δ. With regard to the algorithm used by the FDA, the beginning position of flows is determined by the following relation:

$$Flow\_X(i) = lb + rand*(ub-lb)$$

$$Neighbour\_X(j) = Flow\_X(i) + randn*\Delta \quad (1)$$

where Flow_X(i) depicts the position of the flow ith, lb and ub indicate the lower and upper bounds of the decision variables, and rand displays a random number between zero and one with uniform distribution, respectively. Flow X(i) displays the location of the flow ith. In addition, it is assumed that there are β neighborhoods around every flow, the location of which is determined by the connection that is shown below:

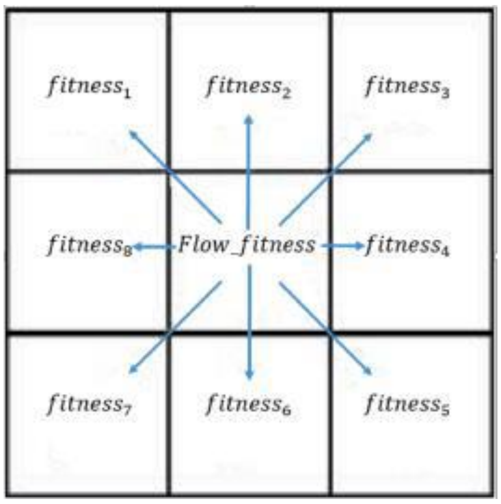


**Fig 2 .** Flow and the eight positions around it

In the above equation, *Neighbour*_X represents the neighbor $j^{th}$ position, whereas *rand**n is a random variable that follows a normal distribution, has a mean of zero, and a standard deviation of 1. Because of the tiny values of Δ, the search is limited to a narrow range, while the big values of this parameter enable the search to be conducted throughout a broad range. As a matter of fact, searching across a broad range yields a bigger diversity of answers and enhances the likelihood of discovering solutions that are close to ideal (global search).

## 2.3 Reptile Search Algorithm

The met heuristic approach known as the reptile search algorithm was developed after observing the hunting habits of crocodiles in their natural environment. The RSA is dependent on two stages in order to function properly: the encircling phase and the hunting phase. The RSA is able to transition between the encircling phase and the hunting search phase. The process of switching between various phases is accomplished by splitting the total amount of iterations into four distinct segments.

**1) Initialization**

In the beginning, the reptile search method will generate a collection of initial solution candidates by utilizing the following equation in a stochastic manner:

$x_{jk} = rand \times (Ub-Lb) + Lb \qquad k=1,2\dots\dots n$ **(1)**

where $x_{jk}$ represent the initialization matrix and j ranges from 1 to n. which are represented by the columns of the initialization matrix. *Lb, Ub*, and rand are the symbols that stand for the lower bound limit, the upper bound limit, and the values that are created at random.

**2) Encircling Phase (Exploration)**

During the surrounding phase, the primary objective is to conduct an investigation of a high-density region. The motions of high walking and belly walking, which are based on crocodile movements, play a very essential function throughout the encircling phase of the process. These motions are not helpful in capturing prey; rather, they are helpful in locating a large available search region**.**

$x_{jk}(\tau+1) = Best_k(\tau) \times (-\mu_{(jk)}(\tau)) \times \beta - (R_{(jk)}(\tau) \times rand), \quad \tau \le \frac{T}{4}$ (2)

$x_{jk}(\tau+1) = Best_k(\tau) \times x_{(r1,k)} \times ES(\tau) \times rand \qquad \tau \le 2\frac{T}{4} \text{ and } \tau > \frac{T}{4}$ (3)

Let us consider the following equation: $Best_k(\tau)$ represents the ideal solution achieved at the kth location, rand is a random integer, τ indicates the current iteration number, and T is the maximum number of iterations. It is the value of the hunting operator of the jth solution at the kth location that is denoted by the symbol μ(j,k). To get the value of μ(j,k), the following formula is utilized:

$\mu(j,k) = Best_k(\tau) \times P_{(j,k)}$ (4)

in which β is a sensitivity parameter, and it provides an explanation for the correctness of the exploration. The following is an example of how to compute a different function called R(j,k), whose objective is to decrease the size of the search space area:

$R(j,k) = \frac{Best_k(\tau) - P_{(r2,k)}}{Best_k(\tau) + \epsilon}$ (5)

Where the value of the random number that falls between 1 and N is denoted by the variable r1. The entire number of potential solutions is denoted by the letter N in this context. A random location for the kth solution is represented by the expression z(r1,l). A value of a tiny magnitude is represented by the symbol e, whereas r2 is likewise an arbitrary (random) integer that may range anywhere from 1 to N. ES (τ), often referred to as Evolutionary Sense, is a measurement that is based on probability. The conceptualization of the evolutionary sense may be expressed mathematically as follows:

$ES(\tau) = 2 \times r_3 \times (1 - \frac{1}{T})$ (6)

In this context, r3 stands for a random number. Compute the value of P(j,k) as follows:

$P(j,k) = \alpha + \frac{x_{(j,k)} - M(x_j)}{Best_k(\tau) x (U_b(k) - L_{b(k)}) + e}$ (7)

where the symbol α represents a sensitivity limit that regulates the precision of the investigation. *M(xj)* corresponds to the average location of the *jth* solution and may be determined using the following formula:

$M(x_j) = \frac{1}{n}\sum_{k=1}^{n} x_{(j,k)}$ (8)

**3) Hunting Phase (Exploitation)**

Hunting coordination and hunting collaboration are the two tactics that are used throughout the hunting phase, just as they were during the surrounding phase. It is possible to explore the search space locally using both of these procedures, which also assist in targeting the prey (finding the best possible solution). Similar to the previous phase, the hunting phase is split into two halves according to the iteration. The hunting coordination approach is implemented for iterations that fall within the range of τ ≤ 3 T/4 and > 2 T/4. On the other hand, the hunting cooperation strategy is implemented for iterations that fall within the range of τ < T and exceed 3 T/4. The local search space is traversed with the help of stochastic coefficients in order to accomplish the generation of optimum solutions. During the period of exploitation, equations (9) and (10) are used respectively:

$x(j,k)(\tau+1) = Best_k(\tau) \times rand, \tau \le 3\frac{T}{4} \text{ and } \tau > 2\frac{T}{4}$ (9)

$x(j,k)(\tau+1) = Best_k(\tau) - \mu_{(jk)}(\tau) \times e - R_{(j,k)}(\tau) \times rand \qquad \tau > 3\frac{T}{4}$
$\tau \le T$ (10)

*Bestk (τ)* represents the k location in the solution that has been achieved with the highest quality in the current iteration. A similar representation of the hunting operator is denoted by the symbol *μ(j,k),* which is derived from Equation (4)

## 3. Methodology

The process of data acquisition involves the collection of audio recordings that comprise speech from a variety of sources. These sources also include a wide range of speakers, locations, and recording equipment. This collection of recordings serves as the basis for the dataset that will be used for further research.

During the pre-processing stage, the audio recordings or speech that have been processed in order to improve their quality and make subsequent analysis more straightforward. In order to do this, noise reduction techniques are used to reduce the amount of background noise, filtering techniques are utilized to enhance clarity, and segmentation algorithms are utilized to divide the recordings into unique spoken words. In order to guarantee that the data is clean and suitable for feature extraction, pre-processing is performed.

### 3.1 Extraction of Features

In order to extract useful information from each speech, a number of different feature extraction approaches are used. These methods include the following:

**1. Spectral Features:** These features provide insights into the spectral properties of the speech signal as well as fluctuations over time. They are responsible for characterizing the frequency content of the speech signal.

**Algorithm 1 Extract Spectral Features**

**Require:** *audio signal*: 1D array, the pre-processed audio signal
1: *sample rate*: int, the sampling rate of the audio signal
2: *frame length*: int, the length of each frame (in samples) for

```
short time Fourier transform (STFT)
3: hop length: int, the hop length (in samples) between adjacent frames
Ensure: spectral features: 2D array, extracted spectral features
4: Apply Short-Time Fourier Transform (STFT):
5: Compute the STFT of the audio signal with a window function of length frame length and a hop length of hop length.
6: Compute Spectral Features:
7: for each frame in the STFT do
8: Compute the magnitude spectrum by taking the absolute value of the complex-valued STFT coefficients.
9: Optionally, apply a mel filter bank to the magnitude spectrum to obtain Mel spectrogram.
10: Compute statistical features over the magnitude spectrum or Mel spectrogram, such as mean, variance, skewness, kurtosis, etc.
11: end for
12: Aggregate Features:
13: Aggregate spectral features over time, e.g., by taking the mean or median across frames.
14: Return Spectral Features:
15: Return the aggregated spectral features as the output of the function.
```

**2. Cepstral Features:** These features emphasize essential temporal and spectral patterns by translating the spectrum of the speech signal into the cepstral domain. Furthermore, they are useful for tasks such as speaker identification and speech synthesis since they highlight these patterns.

**Algorithm 1 Extract Cepstral Features**

```
Require: signal: Pre-processed 1D array, the input speech signal
Require: sample rate: Integer, the sampling rate of the input signal
Require: num cepstral coefficients: Integer, the number of cepstral coefficients to extract
Require: frame length: Float, the length of each analysis frame in seconds.
Require: frame stride: Float, the stride between consecutive frames in seconds
Ensure: cepstral features: 2D array, extracted cepstral features
1: Preemphasis:
2: Apply a preemphasis filter to the input signal to amplify higher frequencies.
3: Frame Blocking:
4: Divide the preemphasized signal into overlapping frames of length frame length with a stride of frame stride.
5: Windowing:
6: Apply a window function (e.g., Hamming window) to each frame to reduce spectral leakage.
7: Compute Power Spectrum:
8: Compute the power spectrum of each windowed frame using the Fourier transform.
9: Mel-Frequency Filter Bank:
10: Apply a Mel-frequency filter bank to the power spectrum to convert it into the Mel-frequency domain.
11: Logarithm:
12: Take the logarithm of the filter bank energies.
13: Discrete Cosine Transform (DCT):

14: Apply a discrete cosine transform (DCT) to the log filter bank energies to obtain cepstral coefficients.
15: Keep Top Coefficients:
16: Retain the first num cepstral coefficients cepstral coefficients as features.
17: Return Cepstral Features:
18: Return the extracted cepstral features as the output of the function.
```

**3. Deep Features:** Using deep learning architectures, deep features are learnt representations that are taken from layers inside neural networks. Deep features are one kind of deep learning architecture. These characteristics are responsible for capturing hierarchical representations of voice input, which provide the model the ability to recognize complicated patterns and subtleties.

**Auto-encoders:** These are unsupervised learning models that try to build efficient representations of input data by compressing it into a lower-dimensional latent space. Auto-encoders are also known as autonomous learning models. They are applied to extract concise but informative representations of spoken utterances, which makes subsequent analysis and classification jobs easier to do.

**Algorithm 1 Extract Deep Features**

```
Require: signal: 1D array, the input signal
Ensure: deep features: 2D array, extracted deep features.
1: Reshape the Input Signal:
2: Reshape the input signal into a 2D array with one column and a number of rows determined by the length of the input signal.
3: Split the Data:
4: Split the data into training and testing sets using an 80-20 split, with 80% of the data used for training and 20% for testing.
5: Build the Autoencoder Model:
6: Define an input layer with the shape of the input signal.
7: Add a dense layer with ReLU activation as the encoder layer.
8:Add another dense layer with sigmoid activation as the decoder layer.
9: Construct the model using the input and output layers.
10: Compile the Autoencoder Model:
11: Compile the model using the Adam optimizer and mean squared error loss function.
12: Train the Autoencoder Model:
13: Train the autoencoder model using the training data. The input and output for training are both the training data.
14: Train for one epoch with a batch size of 32.
15: Extract Deep Features:
16:Separate the encoder part of the trained autoencoder into a new model.
17: Extract deep features by passing the original signal through this encoder model.
18: Return Deep Features:
19: Return the extracted deep features as the output of the function.
```

The method proceeds to feature extraction once the audio recordings have undergone pre-processing to improve their quality and get them ready for analysis. Currently, the objective is to identify important features of the speech signal that can help with speaker identification. Several methods, such as Mel-frequency cepstral coefficients (MFCCs), can be used to extract valuable features from the pre-processed audio data.

After feature extraction, the system moves into a crucial phase called Optimal Weighted Feature Selection and Fusion.

## 3.2 Optimal Weighted Feature Selection and Fusion

One of the most important processes in the field of speech and speaker identification systems is called Optimal Weighted

Feature Selection and Fusion. Its purpose is to enhance the accuracy of recognition by choosing and merging the most useful features retrieved from audio data in a strategic manner. In the first stages of this procedure, a subset of features is carefully selected from the pool of all the characteristics that are accessible. The relevance and discriminative capability of each feature is evaluated in order to determine whether or not it is capable of capturing the distinctive qualities of specific speakers. Weights are applied to each feature when the feature selection process is complete. These weights indicate the significance of each feature in the classification job. The traits that have a larger discriminating value are given higher weights, whereas the ones that are less significant are given lower weights.

The features are then integrated using fusion methods in order to provide a unified representation that contains the information that is gathered from all of the chosen features. This occurs after the features have been weighted. In order to accomplish this fusion process, a variety of strategies may be used, ranging from simple linear combinations to more intricate neural network-based approaches. One of the goals is to produce a complete feature representation that is capable of capturing the distinctive characteristics of various speakers in the most effective manner possible. In addition, the weights that are allocated to the characteristics are further optimized via the use of optimization techniques such as gradient-based optimization methods or heuristic search algorithms by use of optimization techniques.

Enhancing the system's effectiveness in properly recognizing speakers based on the characteristics of their voices is the ultimate objective of the Optimal Weighted Feature Selection and Fusion technique. The system is able to successfully differentiate between types of speakers and achieve greater levels of accuracy in speaker identification tasks. This is accomplished by methodically picking and combining the elements that are most relevant to the job at hand while simultaneously optimizing their weights. Taking this method not only enhances the accuracy of identification, but it also guarantees that the system is able to adjust to a wide range of speaker characteristics and ambient circumstances, which makes it durable and dependable for use in applications that are carried out in the real world.

Here, the system strives to enhance recognition accuracy by choosing the most informative features and merging them in an optimal way. The system utilizes the Flow Direction Algorithm and Reptile Search Algorithm (FDA-RSA) to achieve this. These algorithms enhance the weights assigned to the extracted features, enabling the system to more effectively differentiate between various speakers. The result of this stage acts as the starting point for the next phase of the process.

## 3.3 Implementation of Hybrid Optimized FDA-RSA Algorithm

**Algorithm 1 Hybrid Optimized FDA-RSA**

```
Require: Start Point, Elevation Grid, Max Iterations, Convergence Criteria
1: Initialize flow directions using Flow Direction Algorithm (FDA)
2: Let current point be the starting point
3: Set iteration = 0
4: while iteration < max iterations do
5:Perform reptile search from current point on the elevation grid
6: Update current point based on the search result
7: if convergence criteria met then
8: break
9: end if
10: iteration ← iteration + 1
11: end while
12: return Optimized point.
```

## 3.4 Implementation of TransBiLSTM-AM model

During the implementation stage of the TransBiLSTM-AM model, the refined features are used to train a deep learning model, specifically following the TransBiLSTM-AM architecture. This model combines bidirectional Long Short-Term Memory (LSTM) units with an attention mechanism, allowing it to efficiently capture temporal dependencies and concentrate on important sections of the input sequence. Throughout the model training process, adjustments are made to parameters like the number of hidden neurons and training epochs using the FDA-RSA algorithm. This iterative process guarantees that the model becomes adept at accurately differentiating between speakers using their voice characteristics.

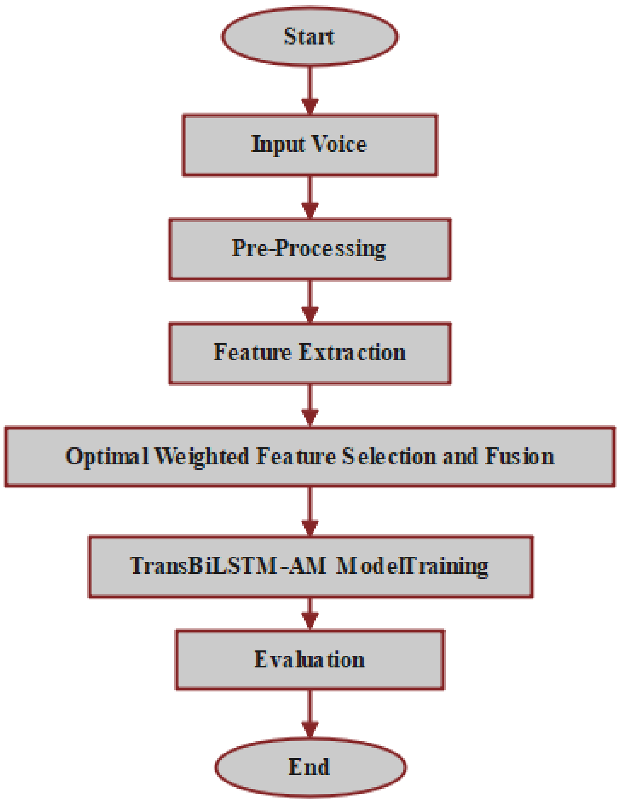

Start
Input Voice
Pre-Processing
Feature Extraction
Optimal Weighted Feature Selection and Fusion
TransBiLSTM-AM ModelTraining
Evaluation
End

**Fig 3** Flow Chart

After the model is trained, it is evaluated to measure its performance using metrics like accuracy and F1-score. This assessment is a critical part of confirming the efficiency of the proposed system in accurately recognizing speakers by their voice. Through a methodical examination of the model's performance, the system can continuously improve its approach and potentially enhance accuracy and reliability in speaker identification tasks.

## 3.5 Performance Metrics

Performance measurements from the confusion matrix, including accuracy, sensitivity, precision, as well as F1-score, are used to evaluate algorithms.

**Accuracy:** A subject recognition rate is the proportion of properly identified subjects to a total number of subjects.

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

**Sensitivity:** For our purposes, recall is synonymous with sensitivity, and refers to the percentage of true positive labels that our system is able to accurately identify.

Sensitivity= $\frac{TP}{TP+FN}$

Precision: The accuracy of an outlook may be estimated by counting the number of correct predictions. One alternative name for this idea is "predictive value."

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

Specificity: Specifically, the method has identified the negative.

Specificity =$\frac{TN}{TN+FP}$

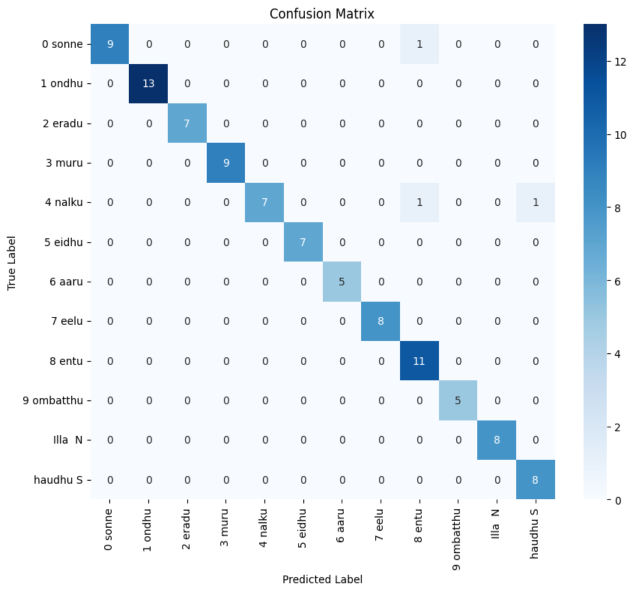
Where, TP= True Positive, TN= True Negative, FP= False Positive and FN= False Negative

## 4. Results

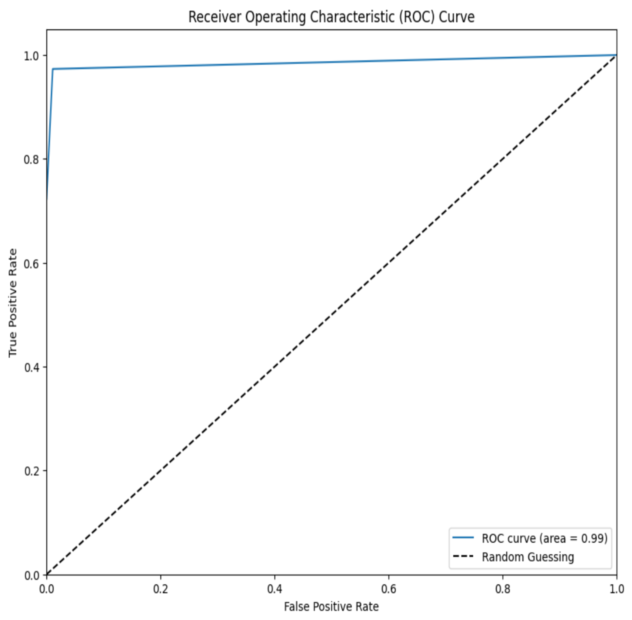
The implementation is done using python of the system windows 10 operating system, 8 GB of RAM, 512 GB of Hard Disk and I3 Processor.
In this study speech recognition and speaker recognition was done using transformer BiLSTM and comparison of the proposed model was done with the existing models flow direction algorithm and reptile search algorithm.

### Speech Recognition

The classification outcomes of the proposed trans BiLSTM model are depicted in the confusion matrices provided, which contrast with the current Flow Directional Algorithm model and Reptile Search Algorithm for the following classes: sonne, ondhu, eradu, muru, nalku, eidhu, aaru, eelu, entu, ombatthu, illa N, and haudhu S. The proposed trans BiLSTM model depicts
accurate classification of classes and precise predictions. The significance of improving models in order to enhance classification precision and reduce misclassifications is highlighted by these results.

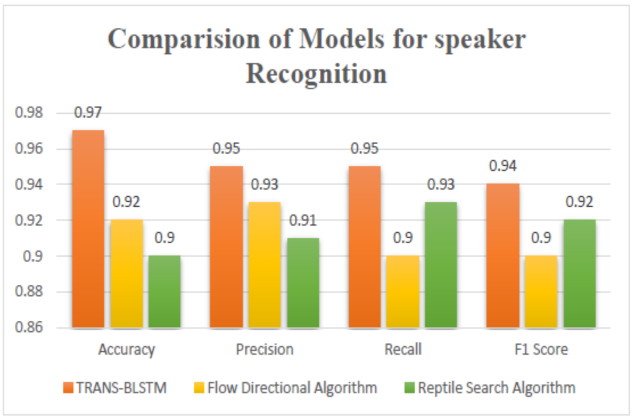


| | 0 sonne | 1 ondhu | 2 eradu | 3 muru | 4 nalku | 5 eidhu | 6 aaru | 7 eelu | 8 entu | 9 ombatthu | illa N | haudhu S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 sonne | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 ondhu | 0 | 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 eradu | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 muru | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 nalku | 0 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 5 eidhu | 0 | 0 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 aaru | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 |
| 7 eelu | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 0 |
| 8 entu | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 0 | 0 | 0 |
| 9 ombatthu | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 |
| illa N | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 |
| haudhu S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 |



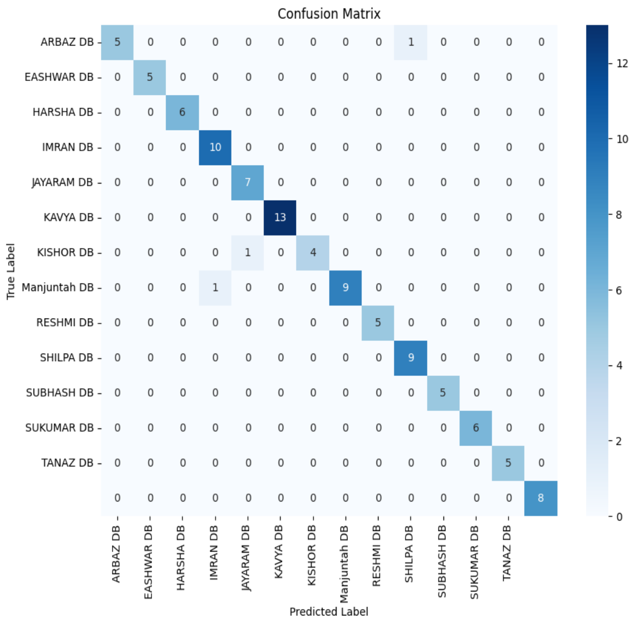
The above image demonstrates a Receiver Operating Characteristic (ROC) curve, which is a metric used to evaluate models for binary classification. The True Positive Rate (TPR), which represents positively categorised instances, is plotted against the False Positive Rate (FPR), which signifies negatively classified instances, in this curve. The ROC curve of an ideal model should approach the upper left corner, signifying its capability to differentiate between positive and negative instances. The diagonal line represents an arbitrary estimation, and any curve that intersects it indicates performance that is merely coincidental.
The ROC curve is precisely positioned in the upper left corner of this image, exhibiting an Area Under the Curve (AUC) value of 0.99. This indicates an impeccable model that is capable of distinguishing positive instances from negative ones without any flaws. When evaluating classification models, it is essential to keep in mind, however, that ROC curves are only one component. Additional variables, including recall and precision, might be essential to contemplate in order to obtain a holistic comprehension of the model's efficacy.

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| TRANS-BLSTM | 0.97 | 0.95 | 0.95 | 0.94 |
| Flow Directional Algorithm | 0.92 | 0.93 | 0.90 | 0.90 |
| Reptile Search Algorithm | 0.90 | 0.91 | 0.93 | 0.92 |

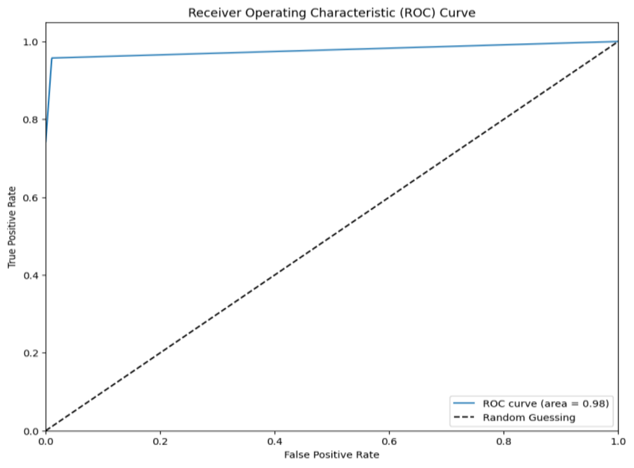


The bar graph shows a comparison between the F1 score, accuracy, precision, and recall of three distinct models: TRANS-BLSTM, Flow Directional Algorithm, and Reptile Search Algorithm. The models are represented on the x-axis of the graph, while the percentage score, which ranges from 0.84 to 1.02, is displayed on the y-axis. Among the three models, the TRANS-BLSTM model exhibits the highest accuracy 0.97, precision of 0.95, recall of 0.95, and F1 score of 0.94, as depicted in the bar graph; these values place the Flow Directional Algorithm in second place across all categories. Accuracy: 0.92; precision: 0.93; recall: 0.90; and F1 score: 0.90; these are the lowest scores of all categories for the Reptile Search Algorithm.

In terms of accuracy, precision, recall, and F1 score, the bar graph indicates that the TRANS-BLSTM model exhibits superior performance compared to the other two models.



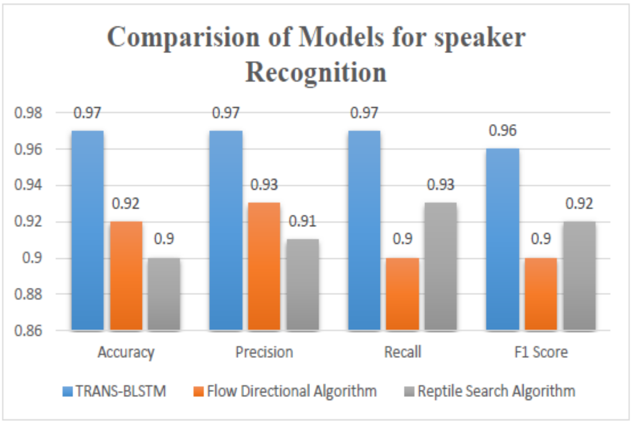
The proposed trans BiLSTM model's classification results are shown in the confusion matrices, comparing with the current Flow Directional Algorithm model and Reptile Search Algorithm for the classes: Arbaz db, Eashwar db, Harsha db, Imran db, Jayaram db, Kavya db, Kishor db, Manjuntah db, Reshmi db, Shilpa db, Subhash db, Sukumar db, Tanaz db. The proposed trans BiLSTM model demonstrates reliable classification of classes and accurate predictions. These findings underscore the need of refining models to increase classification accuracy and minimise misclassifications.



The image displayed above illustrates a Receiver Operating Characteristic (ROC) curve, a metric utilised to assess binary classification models. This curve illustrates the relationship between the True Positive Rate (TPR), which represents instances that were correctly classified, and the False Positive Rate (FPR), which represents instances that were incorrectly classified. In an ideal model, the ROC curve would converge towards the upper left quadrant, indicating the model's ability to distinguish between positive and negative instances. The diagonal line symbolises a conjecture, and the intersection of any curve with it signifies performance that is purely fortuitous.

The ROC curve is accurately located in the upper left quadrant of the image and demonstrates an AUC (Area Under the Curve) value of 0.99. This signifies an impeccable model that possesses the ability to differentiate positive instances from negative ones without exhibiting any imperfections. However, when assessing classification models, it is critical to remember that ROC curves constitute only a single component. Further consideration of additional variables, such as recall and precision, may be necessary to achieve a comprehensive understanding of the effectiveness of the model.

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|---|
| TRANS-BLSTM | 0.97 | 0.97 | 0.97 | 0.96 |
| Flow Directional Algorithm | 0.92 | 0.93 | 0.90 | 0.90 |
| Reptile Search Algorithm | 0.90 | 0.91 | 0.93 | 0.92 |



The bar graph displays a comparison of the F1 score, accuracy, precision, and recall of three different models: TRANS-BLSTM, Flow Directional Algorithm, and Reptile Search Algorithm. The x-axis of the graph represents the models, while the y-axis displays the percentage score, which runs from 0.86 to 0.98. The TRANS-BLSTM model outperforms the other two models with an accuracy, precision, recall, and F1 score of 0.97 each, as seen in the bar graph. This places the Flow Directional Algorithm in second position in all categories. The Reptile Search Algorithm achieved the lowest ratings in all categories: Accuracy (0.92), Precision (0.93), Recall (0.93), and F1 score (0.90). The bar graph shows that the TRANS-BLSTM model outperforms the other two models in terms of accuracy, precision, recall, and F1 score.

## 5. Conclusion

Our research presents a novel approach for recognising speakers and speech based on deep feature extraction. This system integrates a heuristic adopted transformer bidirectional long short-term memory (LSTM) with an attention mechanism. By means of thorough experimentation and evaluation, we have successfully displayed the efficacy of our proposed model in addressing the complexities associated with speech and speaker recognition in complex audio data. Through the utilisation of deep learning methodologies, including bidirectional LSTM and an attention mechanism integrated into a Transformer architecture, our system surpasses the performance of established approaches like the reptile search algorithm and flow detection algorithm. The findings suggest that our model demonstrates enhanced precision and resilience, rendering it a potentially viable resolution for a multitude of practical scenarios that demand precise speaker and speech identification. By incorporating deep learning techniques, our system is capable of approximating long-range and temporal dependencies in speech data, thereby facilitating the identification of nuanced patterns and characteristics that are vital for precise recognition. our research makes a valuable contribution to the progression of audio analysis utilizing deep learning techniques. Moreover, it highlights the profound influence that our suggested system could have on a multitude of fields, such as telecommunications, human-computer interaction, and security. Additional research and development in this domain may result in models that are even more sophisticated, possess improved efficacy, and have a wider range of practical applications.

## References:

[1] J. Santoso, T. Yamada, and S. Makino, "Classification of causes of speech recognition errors using attention-based bidirectional long short-term memory and modulation spectrum," *2019 Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. APSIPA ASC 2019*, pp. 302–306, 2019, doi: 10.1109/APSIPAASC47483.2019.9023288.

[2] M. A. Laskar and R. H. Laskar, "HiLAM-aligned kernel discriminant analysis for text-dependent speaker verification," *Expert Syst. Appl.*, vol. 182, no. September 2019, p. 115281, 2021, doi: 10.1016/j.eswa.2021.115281.

[3] S. P. Todkar, S. S. Babar, R. U. Ambike, P. B. Suryakar, and J. R. Prasad, "Speaker Recognition Techniques: A Review," *2018 3rd Int. Conf. Converg. Technol. I2CT 2018*, pp. 1–5, 2018, doi: 10.1109/I2CT.2018.8529519

[4] R. Mohd Hanifa, K. Isa, and S. Mohamad, "A review on speaker recognition: Technology and challenges," *Comput. Electr. Eng.*, vol. 90, no. April 2020, p. 107005, 2021, doi: 10.1016/j.
compeleceng.2021.107005.

[5] S. Novoselov, A. Shulipa, I. Kremnev, A. Kozlov, and V. Schemelinin, "On deep speaker embeddings for text-independent speaker recognition," *Speak. Lang. Recognit. Work. ODYSSEY 2018*, pp. 378–385, 2018, doi: 10.21437/Odyssey.2018-53.

[6] A. Hajavi and A. Etemad, "A deep neural network for short-segment speaker recognition," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2019-Septe, pp. 2878–2882, 2019, doi: 10.21437/Interspeech.2019-2240.

[7] K. Aida-Zade, A. Xocayev, and S. Rustamov, "Speech recognition using Support Vector Machines," *Appl. Inf. Commun. Technol. AICT 2016 - Conf. Proc.*, vol. 1, 2017, doi: 10.1109/ICAICT.
2016.7991664.

[8] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," *Adv. Neural Inf. Process. Syst.*, vol. 32, no. NeurIPS, pp. 1–11, 2019.

[9] K. Mohiuddin *et al.*, "Retention Is All You Need," *Int. Conf. Inf. Knowl. Manag. Proc.*, no. Nips, pp. 4752–4758, 2023, doi: 10.1145/3583780.3615497.

[10] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.