# A Comparative Analysis of Deep Learning Models for Speech Recognition

SK Sameer[1], Dr. Surender Kalyan[2]
*[1]Research Scholar, NIILM University, Kaithal, Haryana*
*[2]Research Supervisor, NIILM University, Kaithal, Haryana*

***Abstract*—Deep learning has changed the game in speech recognition, enhancing the accuracy, robustness and adaptability of speech recognition systems that are utilised in a plethora of areas such as virtual assistants, self-driving vehicles and health. The current paper would offer a comparison of the most imminent deep learning models used in speech recognition, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Deep Neural Networks (DNNs), and the models based on Transformers. Through the critical textual review of performance-based parameters such as Word Error Rate (WER), real-time factor, and model complexity and scalability, this study highlights the strength and weakness of each model and gives us the knowledge of how appropriate they are to various real-life applications. Furthermore, mention the main obstacles of these models, including the data need, generalization, tolerance to noise, computational needs. The comparative study offers an extensive review of the current state of the art in speech recognition nowadays and help researchers and practitioners to select the most appropriate methods of the deep learning to fit the specific application.**

***Index Terms*—Speech Recognition, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Deep Neural Networks (DNNs), Transformer Models.**

## I. INTRODUCTION

Speech recognition technology is also important to upgrade the way humans communicate with machines, giving them an easy and natural language of communication [1]. Its first advantage is being able to provide hands-free control, making it easier and more efficient for people to use technology. Voice-controlled systems such as Siri, Alexa, and Google Assistant have transformed daily activities into simple voice commands for using their machines [2]. This simplicity of use makes it a choice for many, most importantly those looking for instant information, device operation, or task management [3]. The technology's capability to identify and react to voice commands is a major difference from conventional input methods, like typing or button pressing [4]. Aside from user convenience, speech recognition is incredibly vital in enhancing accessibility, especially among people with disabilities. Individuals with mobility or visual impairments are able to use voice-enabled systems to travel through digital spaces, manage devices, and interact with material without the need to rely on physical input mechanisms. This inclusiveness leads to increased autonomy for these users, enabling them to participate more actively in personal and business sides of life. In addition, speech recognition supports productivity across industries such as healthcare and law, as well as customer service [5]. Professionals are able to use voice-to-text technology to dictate reports, documents, and notes, which saves time and is more accurate, particularly in demanding situations such as hospitals or law firms where documentation needs to happen in real time. Speech recognition also has the benefit of more natural human-computer interaction [6]. Voice is one of the most natural means of communication, and its application in technology does away with the learning curve of typing or utilizing other input methods. This simplicity in interaction is especially useful in designing customized user experiences, since voice can carry emotions, intent, and context in a way that text can't. Furthermore, the technology itself is becoming increasingly capable of understanding and processing different languages and dialects, which is a requirement in a globalized world [7]. Multilingual support enables different linguistic groups to access technology more effortlessly, promoting inclusivity and erasing language barriers. The use of speech recognition is also extending to fields of artificial intelligence (AI) and the Internet of Things (IoT). As smart devices become increasingly integral to everyday life, speech recognition makes voice controls possible for everything from home appliances to autonomous vehicles. These systems are made to react to voice commands, delivering more natural and effective interactions with the environment to users. Moreover, in AI-based systems, speech recognition enables continuous learning, adjusting to various accents, languages, and manners of speaking, making it more usable and accurate. Last but not least, speech recognition is increasingly utilized for extracting valuable information from spoken data. In customer service, for instance, voice interaction analysis can enable firms to know about customers' concerns, tastes, and satisfaction levels. This could be utilized to enhance services or products [8]. In medical care, the analysis of physician-patient conversations could reveal what is wrong with patients or results of the treatment, enhancing the quality of care. As technology keeps on improving, speech recognition will continue to be a key instrument in enhancing both accessibility and efficiency in many areas, revolutionizing the way engage with the world around [9].

Deep learning has completely transformed the function of speech recognition systems right at the bottom to facilitate them to operate to the levels that were previously absent through the traditional techniques [10]. With the introduction of deep learning, more precisely neural networks, the accuracy, robustness, and efficiency of speech recognition systems have been highly amplified due to their ability to learn raw audio information without going through huge hand designed features [11]. Possibly, the ability of deep learning to learn the hierarchical nature of speech signals is one of the most important contributions that deep learning has had on speech recognition. Classical paradigms were more inclined to divide the speech into small, discrete aspects such as the phonemes that are individually modelled. From very large datasets, deep algorithms specifically convolutional neural networks (CNNs) and the recurrent net networks (RNNs) can be taught to recognise high-level features. CNNs have the advantage of learning spatial relations in speech spectrograms so well, but RNNs, and in particular Long Short-term Memory (LSTM) networks, are superior at learning temporal dependencies in speech and are therefore suitable when a sequential data modality is available, such as audio [12]. These models learn automatically to transform raw audio into semantic representations, enhancing the system's proficiency in recognizing and transcribing speech with impressive accuracy. Deep learning has also been crucial in the creation of end-to-end speech recognition systems. Before that, speech recognition pipelines were separated into several stages, e.g., feature extraction, acoustic modelling, language modelling, and decoding. Each of them was usually treated with separate models, thereby increasing the complexity and liability of the system[13]. Deep learning allowed unified, end-to-end models to be created that learn from the audio input and output transcriptions independently without distinct modules. Innovations such as Connectionist Temporal Classification (CTC) loss and attention have further optimized the systems' efficiency in that explicit alignment of input and output is not required, and overall speech recognition system performance has improved.

The strength of deep learning is that it can process huge quantities of data and learn from it, and the more data that are pumped into the system, the better it becomes. This aspect is especially useful in speech recognition, where big data with diverse accents, languages, and speaking conditions are important to train the model. Deep learning systems are capable of generalizing more across various speakers and noisy conditions and provide more robustness compared to conventional systems. Therefore, contemporary speech recognition systems based on deep learning are capable of transcribing speech in real-world situations like dictation, call centre automation, voice assistants, and even noisy conditions like public areas or driving. In addition, the capacity of deep learning to be coupled with other AI technologies has brought about tremendous breakthroughs in multimodal applications. For instance, speech recognition models are now increasingly being paired with natural language processing (NLP) systems, enabling them not only to transcribe speech but also to comprehend context, intent, and sentiment. It has been especially helpful in voice assistants, where the system not only needs to recognize words but also interpret the user's command and respond accordingly. The paper is framed so that it first presents the importance of speech recognition and how deep learning transformed the industry. It then discusses classical speech recognition approaches and investigates how deep learning architectures such as CNNs, RNNs, and transformers have contributed to better accuracy and efficiency. The paper contrasts these models on the basis of important performance indicators such as accuracy, latency, and computational complexity and also mentions their constraints and challenges, such as data availability and generalization. It emphasizes the practical uses of speech recognition in voice assistants, healthcare, and customer service, and delves into emerging trends like multimodal systems and cross-lingual recognition. The conclusion encapsulates the findings and recommends areas for future work, with references cited throughout the paper.

## II. TRADITIONAL SPEECH RECOGNITION VS DEEP LEARNING IN SPEECH RECOGNITION

### 2.1 Traditional Speech Recognition

At the initial levels of speech recognition technology, the widely accepted models were statistical ones such as Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs). These models served as the framework for most speech recognition systems between the 1980s and early 2000s and played a key role in shifting from rule-based systems to statistical systems. Hidden Markov Models (HMMs) are stochastic models for describing a system whose underlying states are not necessarily directly observable (and thus "hidden") but can be determined from observable outputs. In speech recognition, the "states" within an HMM are the various speech sounds or phonemes, and the "observations" are the acoustic features extracted from the speech signal. HMMs capture the time dynamics of speech in that they capture the sequential behaviour of phonemes, which change to other phonemes with some probability. This made HMMs extremely efficient at dealing with the time-varying aspect of speech, allowing systems to identify words and sentences even if there was variability in speech patterns or accent. Gaussian Mixture Models (GMMs) were employed along with HMMs to capture the distribution of acoustic features. GMMs postulate that the feature vectors at a particular time step in a speech signal are randomly sampled from a mixture of Gaussians, each corresponding to a distinct group of acoustic features. By describing the probability distribution of the features, GMMs enabled the accurate classification of speech sounds so that the system could identify phonemes or words as a function of the likelihood of different feature vectors. In combination, HMMs and GMMs composed the cornerstone of classical speech recognition systems. HMMs and GMMs were quite successful in controlled settings with clean speech but performed poorly with noise, accent variability, and other hallmarks of real-world speech. Even with these limitations, HMMs and GMMs laid the ground for more sophisticated, deep learning-based solutions that could better deal with the complex patterns and nuances of speech.

### 2.2 Deep Learning in Speech Recognition

The highly efficient artificial neural networks are Deep Neural Networks (DNNs), which have more than one layer and endow the model with the capability to learn hierarchical representations of the input data. DNNs have been used in speaker recognition in the modelling of the correlation between acoustic features and the phonetic unit. The depth of the networks makes them have the capacity to learn subtle patterns in speech, providing better recognition performance than the conventional models such as HMMs and GMMs. DNNs are particularly good at working with large datasets and perform especially well when working on large-scale speech recognition tasks. Convolutional Neural Networks (CNNs) are designed specifically for spatial data and are commonly utilized for image processing but have also been highly promising within speech recognition. CNNs use convolutional filters over input data (e.g., spectrogram of speech) to extract useful features, diminishing dimensionality and emphasizing significant acoustic patterns. In speech recognition, CNNs come in accessible when extracting hierarchical features from Mel-frequency cepstral coefficients (MFCCs) or spectrograms, allowing improved management of noisy or difficult acoustic signals. In contrast to DNNs, RNNs are capable of capturing temporal relationships, i.e., learning the association between consecutive speech frames. This is essential in speech since the meaning of a word tends to be based on the context established by preceding and subsequent words. LSTMs and GRUs, specifically, are capable of preventing such problems as vanishing gradients, and thus they can learn long-term dependencies within speech data better than regular

RNNs. Transformers, a newer and very efficient architecture, have become the core of most leading-edge speech recognition systems. Transformers use self-attention mechanisms that permit the model to attend to various portions of the input sequence at each step of time, as opposed to processing the sequence sequentially. This attention factor allows transformers to learn long-distance dependencies in speech more effectively than RNNs. Transformers have shown better performance in applications such as automatic speech recognition (ASR) because they can process long and complicated sequences of audio without the drawback of sequential processing, unlike RNNs. Such a model architecture lies at the heart of most contemporary systems such as those utilized in large-scale language models and ASR systems. With the combination of these deep learning models, the domain of speech recognition has come a long way. They have made it possible for systems to recognize complex and noisy speech data with great accuracy, support large vocabularies, and run under real-time conditions. Consequently, deep learning-based systems for recognizing speech are now common across applications from voice assistants to real-time transcription and more.

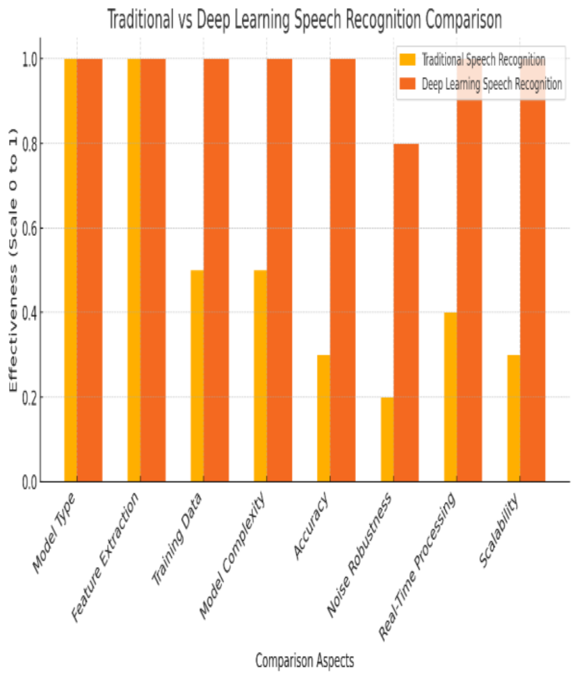


Fig: Traditional Speech Recognition Vs Deep Learning

## III. DEEP LEARNING MODELS FOR SPEECH RECOGNITION

### 3.1 Convolutional Neural Networks (CNNs)in speech recognition

Originally designed to solve image classification and computer vision tasks, Convolutional Neural Networks (CNNs) have been highly effective at recognizing speech, as well. That they get high success rates with the latter can be attributed primarily to their ability to automatically learn spatial hierarchies of features in input data by sequential layers. Speech recognition makes use of audio signals; these signals are usually translated into 2D representations. One important property of these representations is that they maintain the time-frequency structure of speech and are organized in the form of images and as such CNNs are able to process them similarly to the manner in which they process visual data. Each filter is designed to detect some local trend in the spectrogram or MFCC such as edge or transition of frequencies, or energy distribution in time. As the filters convolve across an input, they generate the feature maps that specify that such local patterns occur. Speech-wise, it allows the model to identify phonetics and transitions of sounds, which are essential in speech recognition of words. The convolutional layers will always be followed by an activation function, usually the Rectified Linear Unit (ReLU). This non-linear model introduces more complexity to the model thereby enabling it to learn more complex models. Absence of activation functions would make the entire network a linear model regardless of its depth, which significantly limits the learning capabilities. Instead of keeping the negative activations in the feature map, ReLU sets it to zero in all places; however, the positive activity is maintained and is most likely the representation of meaningful features. Then the feature maps are reduced in spatial size by the pooling layers. This will shrink the complexity of computation and overfitting, and retain the most influential features. The most common is max pooling in which the season of the greatest value within that window of the feature map is studied. Alternatively, one can also use average pooling in which one calculates the average of the data values in the window. The pooling layers help the model in becoming invariant in minor distortion and shifts of the input, which is specifically useful when considering speech variability, such as talking speed or slight differences in pronunciation. After a few convolution, activation and pooling, the result is fed to fully connected (dense) layers. These layers flatten the multi-dimensional feature maps into a one-dimensional representation and perform the higher-level decision making using the feature extraction. In speech recognition this is the point after which the network is trained to change the learned patterns into specified outputs, e.g., phonemes, characters or entire words. These layers are pretty much the same as old neural networks with the notable difference being that they are typically preceded by a SoftMax output layer to carry out the classification tasks.

Algorithm: CNN-Based Speech Recognition System

Input: Raw audio signal

Output: Predicted transcription (e.g., word or phoneme sequence)

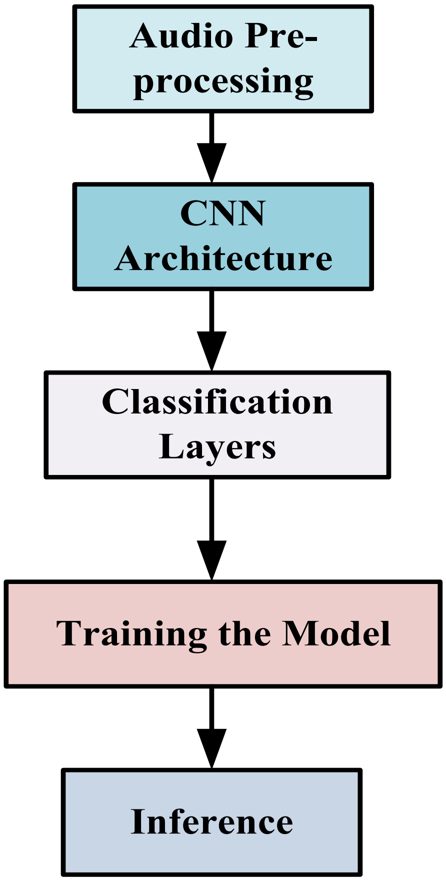


Fig: Flowchart Of CNN-Based Speech Recognition System

Step 1: Audio Preprocessing
Acquire raw audio input (e.g., WAV file). Normalize the audio signal (optional: remove noise or silence). Convert the signal into a time-frequency representation: Use Short-Time Fourier Transform (STFT), Log-Mel Spectrogram, or MFCC.

Result: 2D feature map representing time (horizontal) and frequency (vertical) dimensions.

Step 2: CNN Architecture Setup
Input Layer: Accept the 2D feature map as input to the CNN.

Convolutional Layers: Apply multiple convolutional filters (kernels) across the input feature map. Each filter detects different local patterns in the time-frequency domain.

Activation Function: Apply the ReLU activation function to introduce non-linearity.

Pooling Layers: Apply max or average pooling to down sample the feature map. Helps reduce dimensionality and retain important features. Repeat steps for multiple layers to learn hierarchical features.

Step 3: Classification Layers
Flattening: Convert the final 2D feature map into a 1D vector.

Fully Connected Layers: Pass the vector through one or more dense layers to learn feature combinations.

Output Layer: Apply SoftMax or CTC (Connectionist Temporal Classification) for sequence prediction. Output is a probability distribution over character/phoneme/word classes.

Step 4: Training the Model
Define loss function: Use categorical cross-entropy (for classification) or CTC loss (for sequence alignment).
Select optimizer (e.g., Adam, SGD). Train the model on labelled speech datasets (e.g., LibriSpeech, TIMIT). Validate model performance using metrics such as Word Error Rate (WER).

Step 5: Inference

Feed new audio into the model. Convert audio to feature map (same as step 1). Pass through trained CNN model. Decode the output to generate predicted transcription.

### 3.2 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a robust family of neural structures specially used to deal with sequential data. In contrast, conventional feedforward neural networks utilize inputs independently, while RNNs involve a memory mechanism that holds information from earlier inputs in a sequence. This feature allows them to be very efficient for temporal dependency modelling, which is a central feature of speech signals. This means that in speech recognition, the audio signal is a form of time-series data that exhibits patterns over time. These signals are usually converted to some form of feature vector, such as MFCCs or log-mel spectrogram; these are compact and perceptually meaningful representations of an audio piece. These features are fed into the RNN frame by frame, and thus the network processes both the current audio context and the information retained from prior frames. The most important strength of RNNs is their recurrent connection, where the t-th hidden state depends not only on the current input but also on the t-1-th hidden state. With this architecture, RNNs are able to learn temporal patterns, e.g., duration of syllables, rhythm, and co-articulation between phonemes. Yet, vanilla RNNs face training difficulties, especially for long input sequences. These are the vanishing gradient issue, in which gradients become too tiny to make a contribution to learning, and the exploding gradient issue, in which gradients get larger and larger. Consequently, RNNs tend to have great difficulty learning long-term dependencies, but these are essential in speech recognition, where contextual interpretation over a few seconds or even entire sentences can be necessary for correct interpretation.

Algorithm: RNN-Based Speech Recognition System
Input: Raw audio waveform

Output: Predicted transcription (phonemes, characters, or words)

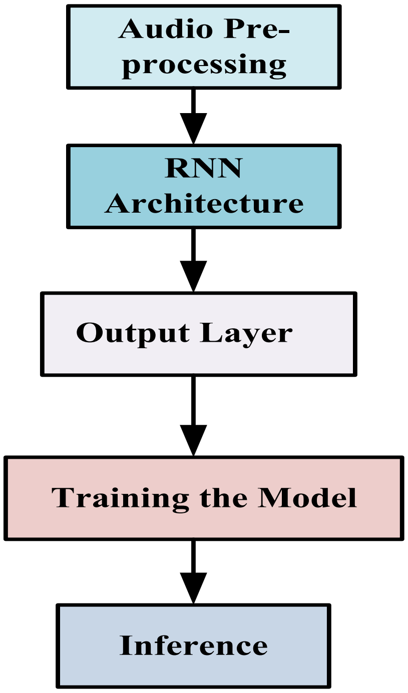


Fig: Flowchart Of RNN-Based Speech Recognition System

Step 1: Preprocessing
Load the raw audio file (e.g., .wav format). Normalize the waveform and remove silence or noise (optional).
Convert the signal into feature vectors:
Extract MFCCs or log-mel spectrogram.

Result: A 2D time-series matrix (frames × features).

Step 2: Define RNN Architecture
Input Layer: Accepts time-series input shape (e.g., [T, F] where T = time steps, F = features).
Recurrent Layers: Add one or more RNN/LSTM/GRU layers. Each layer maintains a hidden state h_t that evolves over time.

Optional Bidirectional RNN:
Use forward and backward passes to capture both past and future context. Dropout Regularization (optional): To prevent overfitting.

Step 3: Output Layer
Apply one of the following:
SoftMax layer for direct classification (if labels align with time steps). CTC (Connectionist Temporal Classification) for unaligned sequence labelling.

Step 4: Model Training
Use a labelled dataset (e.g., TIMIT, LibriSpeech).

Define loss function:
Categorical Cross-Entropy (softmax) or CTC Loss.Choose an optimizer (e.g., Adam, RMSProp). Train the model over multiple epochs using mini-batches. Validate on unseen data using metrics like Word Error Rate (WER).

Step 5: Inference
Input new audio sample. Extract features (same as in training). Pass the sequence through the trained RNN.Decode the output sequence to generate the transcription.

### 3.3 Deep Neural Networks (DNNs)

The Deep Neural Networks (DNNs) can be thought of as a subclass of feedforward neural network (FNN), with more than one hidden layer between the input and output. Connected neurons (nodes) at each layer of DNNs executes a linear transformation of the information fed to them and then passed through a non-linear activation function. The layering (the depth of DNN) allows it to acquire more abstract, complex representations on raw or processed data. DNNs can be used to model the relationship between acoustic features (e.g. MFCCs, PLPs) to linguistic units such as phonemes or words in speech recognition. Each hidden layer in a DNN converts its input into more abstract, higher-level representations, successively capturing more informative and abstract patterns. Lower layers learn local spectral or temporal relationships, while higher layers learn linguistic structures or phonetic distributions. DNNs find most applications in acoustic modelling in conventional hybrid Automatic Speech Recognition (ASR) systems, where they predict posterior probabilities of the phoneme states given acoustic observations. Unlike CNNs or RNNs, typical DNNs lack sequence modelling or spatial awareness mechanisms but are still capable of working extremely well when they are applied in combination with context windows, wherein input frames contain previous and subsequent audio segments in order to offer temporal context. In spite of their relative simplicity relative to RNNs or Transformers, DNNs are popular because of their good generalization properties, trainability using large datasets, and augment ability with other modules within hybrid systems.
Algorithm: DNN-Based Speech Recognition System

Input: Raw audio waveform

Output: Predicted transcription (phoneme or word class)

Step 1: Preprocessing
Load the audio file. Normalize audio and remove silence or background noise. Extract frame-wise acoustic features:
MFCCs, log-mel spectrogram, or PLP coefficients.
Apply context windowing to form input vectors.

Step 2: DNN Architecture Design
Input Layer: Accepts fixed-length feature vectors (e.g., [n_frames × features]).
Hidden Layers: Stack multiple fully connected (dense) layers. Apply activation functions (typically ReLU or tanh). Apply batch normalization and dropout for regularization.

Output Layer: Use SoftMax to produce class probabilities for phonemes or words.

Step 3: Training
Use supervised learning with labelled audio data (e.g., phoneme alignments).
Define loss function: Use categorical cross-entropy.
Choose optimizer (e.g., Adam, SGD). Train the network using mini-batch gradient descent. Monitor accuracy and loss on validation data.

Step 4: Inference
Preprocess test audio as in Step 1. Pass feature vectors through the trained DNN.Obtain SoftMax outputs and choose the class with the highest probability. Optionally, use a language model or decoder to form complete transcription.

Step 5: Evaluation
Compare predictions with ground-truth labels. Use performance metrics such asAccuracy, Word Error Rate (WER), Confusion Matrix (for phoneme-level tasks)

### 3.4 Transformer Models

Transformer models are a major breakthrough in sequence modelling, especially for applications like speech recognition that involve capturing long-range dependencies in sequential data. First developed in natural language processing (NLP), Transformers have been used more by speech recognition because they can more economically capture global context compared to recurrent or convolutional models. The self-attention mechanism is at the centre of Transformer architectures. In contrast to RNNs that read data sequentially and are structurally bound to be parallelization-constrained, the self-attention mechanism makes each input element (e.g., a time step in an audio signal) attend directly to all other elements of the sequence at once. This global perspective makes the model efficiently capture both short- and long-range dependencies equally well, which solves one of the principal drawbacks of RNNs inability to model far-away temporal relationships. Self-attention in speech recognition enables the model to assign relative weights to various frames in an utterance while predicting the current output. While recognizing a phoneme or word, for example, the model can automatically decide what past (or even future) frames hold the most relevant context. This results in improved co-articulation effects understanding, speech rate variability, and acoustic ambiguity. A Transformer model generally consists of multi-head self-attention layers, positional encodings (in order to preserve information about sequence order), feedforward networks, and layer normalization. Within speech recognition, these models tend to work over spectrograms or other time-frequency representations of sound. As Transformers have no inherent recurrence, positional encoding is essential, it enables the model to realize the sequence and order of frames within the input data. One of the strongest transformations of Transformers to apply to speech is the Transformer Transducer or similar models such as Wav2Vec 2.0, which merge self-supervised learning with Transformer architecture in order to learn dense acoustic representations. Such models have achieved state-of-the-art results on many speech recognition benchmarks with lower Word Error Rates (WER) than conventional hybrid models or RNN-based systems.

## IV. COMPARISON OF DEEP LEARNING MODELS

The development of deep learning models transformed automatic speech recognition (ASR) and introduced high levels of accuracy and flexibility in representing acoustic, linguistic and contextual data. In this section, these models will be compared comprehensively in several aspects, i.e. accuracy (in terms of WER), processing efficiency (Real-Time Factor and Latency), ability to resist problematic conditions, and computational cost. These differences shall be understood to make informed decisions on the model architecture that fits best on the basis of certain deployment requirements like real time performance, hardware limitations or ambient noise tolerance.

### 4.1 Comparison Based on Accuracy (Word Error Rate – WER)

The traditional measurement of the performance of the speech recognition systems is the Word Error Rate (WER). It also adds up insertions, deletions and replacements of the predicted transcript with a reference transcript. The table also entails a comparative analysis of various models of deep learning architecture used in the tasking of speech recognition on the basis of their Word Error Rate (WER) on clean and noisy speech signal. WER is an important metric of performance measures, and it shows the percentage of incorrectly predicted words found in the output, and a lower value means a more successful recognition process. Deep Neural Networks (DNNs) achieve a high WER of 1822 located in tidy acoustic surroundings, and the WER increases to 2530 when the surrounding is noisy. This implies that DNNs are less effective in coping with noise and do not involve prediction about temporal context, not being very useful in real-world performance. Convolutional Neural Networks (CNNs) present a better result than DNNs with a WER of 1418% clean and 2025% noise due to the localized patterns that they use to extract in spectrograms claiming the usage in the noise reduction. In clean speech, LSTMs bring down the WER to 12 percent to 16 percent, and noisy environments are down to 18 percent to 22 percent. Worse than LSTMs, but still similar, Gated Recurrent Units (GRUs) have a lighter computation variant with WERs of 1317 and 1923 respectively. Transformer-based architectures perform the best of them, reducing

WERs to 6-10 percent in clean conditions and 9-13 percent in noisy ones significantly.

Table: Performance Comparison of Deep Learning Models Based on Word Error Rate (WER)

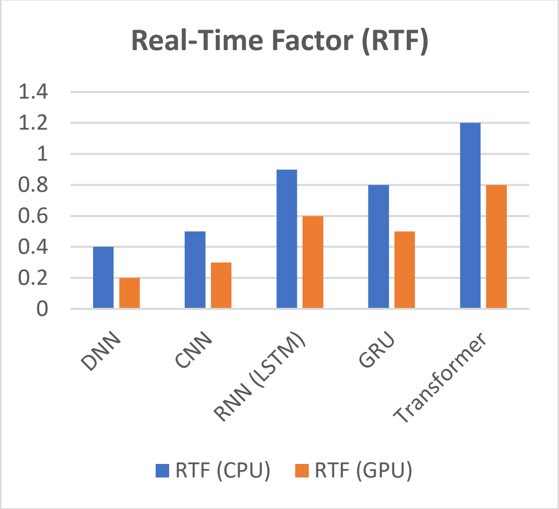
| Model | WER (Clean Speech) | WER (Noisy Speech) |
|---|---|---|
| DNN | 18–22% | 25–30% |
| CNN | 14–18% | 20–25% |
| RNN (LSTM) | 12–16% | 18–22% |
| GRU | 13–17% | 19–23% |
| Transformer | 6–10% | 9–13% |

4.2 Comparison Based on Real-Time Factor (RTF)

Real-Time Factor (RTF) is the time that a model will require to process 1 second of audio. A time ratio of RTF < 1 means faster than real time processing. This table is the comparison between the Real-Time Factor (RTF) of the various deep learning models implemented in speech recognition when run on CPU and GPU. RTF is a very important performance indicator which evaluates the ratio between the time required to process an audio fragment the fragment itself. An RTF<1.0 implies that the model will process audio at a pace quicker than real-time, which is necessary in live or streaming programs such as voice assistants, transcription services and real-time translation. Deep Neural Network (DNN) has the least RTF values, 0.4 and 0.2 on CPU and GPU, respectively, thus it is the fastest model to execute because of its straightforward feedforward design. CNNs come right after, utilizing the parallel computing of convolutional layers, having RTFs of 0.5 (CPU) and 0.3 (GPU). RNNs (LSTM) which are applied to sequentially treated data is more computationally expensive and consequently RTFs are higher at 0.9 on CPU and 0.6 on GPU. GRUs are a more lightweight version of LSTMs and they have a slight advantage in RTFs of 0.8 (CPU) and 0.5 GPUs. Transformer models represent the most expensive in their computation and are at the same time the most accurate. Their RTF is 1.2 on CPU, and as such they are slower than real time, but on GPU they are better at 0.8, which makes them potential candidates of real-time applications, with sufficient GPU support.

Table: Real-Time Processing Comparison of Deep Learning Models

| Model | RTF (CPU) | RTF (GPU) |
|---|---|---|
| DNN | 0.4 | 0.2 |
| CNN | 0.5 | 0.3 |
| RNN (LSTM) | 0.9 | 0.6 |
| GRU | 0.8 | 0.5 |
| Transformer | 1.2 | 0.8 |



4.3 Comparison Based on Latency

Latency is a delay between an input audio and output transcription used to determine real-time systems (e.g., voice assistants). This table is a cross-sectional comparison of different deep learning models applied in speech recognition wherein two important real time parameters of performance have been considered: average latency and streaming ability. The average latency is characterized by the number of milliseconds that pass when an audio signal is input in a system and when a matching transcription is produced. Time-sensitive application like voice assistants, live caption, and real-time translation requires lower latency. Streaming capability the ability of a model to process streaming audio at its arrival and not in its completion. This is vital in live system to get minimum response time. Based on the table DNNs present the lowest latency of about 100 milliseconds thus they are very desirable in fast and real time answers. They are simple, feed-forward in nature thus they can be used in the streaming environment rather efficiently. CNNs can also be used to stream i.e. they are characterized by slightly longer latencies (~120 ms) caused by more processing during the convolution operation. RNNs (LSTM) and GRUs support streaming since by the nature of their design they consume sequences in the form of timesteps. Nonetheless, they are more latent, with LSTMs at about 200 ms and GRUs at 180 ms because they are sequential, with the requirement to preserve temporal links. Transformer models, in their turn, show the most significant latency (over 300 ms) and lack the inherent streaming support, as they use self-attention over the whole input sequence. This means that they are less appropriate in a case of offline transcription or in a case of batch processing, unless a specific adaptation using streaming mechanisms is provided.

Table: Latency and Streaming Capability Comparison of Deep Learning Models for Speech Recognition

| Model | Average Latency (ms) | Streaming Capability |
|---|---|---|
| DNN | ~100 | Yes |
| CNN | ~120 | Yes |
| RNN (LSTM) | ~200 | Yes |
| GRU | ~180 | Yes |
| Transformer | ~300+ | No |

4.4 Comparison Based on Robustness

Robustness describes how well the model performs in dealing with noise, accents and speaker variability, which are important in uncontrolled conditions. This table is comparing the strength of different deep learning models in speech recognition at three main dimensions that are; noise-resistant, accent-resistant, and speaker variability handling. The models are tested in many challenging conditions by each dimension which makes the results to have a strong correspondence to the speech recognition models in a real-world environment. Noise Robustness is the identification of how a suitable model can embrace noise or clutter in the exchange of speech. Higher noise tolerance models will provide accurate transcriptions even in situations such as in a crowded room or busy roads. Having a better ability to extract local and global patterns present in speech makes CNNs and Transformers less susceptible to noise, and this is not the case with DNNs since they are ineffective and do not consider noise and contextual variations. Transformers, which have a sophisticated self-attention mechanism, offer the best noise robustness and then CNNs. Accent Robustness determines how certain the model will be about multiple accents and regional differences in pronunciation. Transformers and CNN exhibit good results in this respect, with a broader range of accents being more successfully addressed than in the case with such models as DNNs, which are more phonetically sensitive. The RNN (LSTM) and GrU models are flexible to the accents given that they are sequential in nature, and have memory, although it has the drawback of not recognizing the global context as CNNs or Transformers are doing. Speaker Variability Handling measures the ability of a model to handle variability of individual speakers, voiced by different utterances, pitch, speed and speaking style. The application of CNNs and Transformers in speech recognition has been realized by its capacity to utilize the speaker variability through the extraction of the complex features as well as perceiving the speech context over different speech patterns. RNNs (LSTM), and GRUs are slightly less effective but speak variability better than DNNs which are less likely to adapt to speaker specificities.

Table: Robustness Comparison of Deep Learning Models for Speech Recognition

| Model | Noise Robustness | Accent Robustness | Speaker Variability Handling |
|---|---|---|---|
| DNN | Low | Low | Medium |
| CNN | High | Medium–High | High |
| RNN (LSTM) | Medium | Medium | Medium–High |
| GRU | Medium | Medium | Medium |
| Transformer | Very High | High | Very High |

### 4.5 Comparison Based on Model Complexity and Resource Usage

This table contrasts memory requirements, computational requirements that influence viability in mobile or edges settings. In this table, there are comparisons of the complexity of the models and the resource requirements of the various deep learning models in the speech recognition line. Parameters, training time, and resource requirements are the primary metrics which are needed to determine whether and how these models can be applied in practice. The parameter Count is a number of parameters or weights in the model that can be trained. The least complex ones are DNNs, having 510 million parameters, and are simpler to train, running faster. The parameter count of CNNs is a bit higher, starting at 10 to 30 million and allowing it to capture more spatial patterns in data about speech. RNNs (LSTM) and GRUs demand much more parameters (15-50 million or more) due to the necessity to keep and operate with temporal dependencies in the speech chains. Transformers are characterized by the most parameters (50 million up to 200 million and above), the so-called self-attention mechanisms, that give the ability to capture more global connections in the data, but are also more computationally demanding. Training Time determines the amount of time required to train every model. Because of its simple architecture, DNNs are the quickest to train, and CNNs take more time to train because of adding the convolutional layers, but are still relatively efficient. RNNs (LSTM) and GRUs are more time-consuming as they are sequential in processing but the slowest are Transformers with their multi-level structure and time-consuming self-action attention. Resource Demand is the requirement of the computation resources to train and deploy the model. The DNNs are the least demanding in terms of resources as they do not demand a lot of computation power and the CNNs are more demanding because they involve the convolution operations. RNNs (LSTM) and GRUs come very abundant in the resources needed as they need substantial memory and computing-power to train, especially long sequences. Transformers have the highest resource demand, both in terms of memory and computational power, especially when trained on large datasets. Transformers with a self-attention mechanism are extremely effective in modelling long-range dependencies, but computationally expensive and must be trained and used on specialized hardware including GPUs.

Table: Model Complexity and Resource Demand Comparison for Deep Learning Speech Recognition Models

| Model | Parameter Count | Training Time | Resource Demand |
|---|---|---|---|
| DNN | 5–10 million | Fast | Low |
| CNN | 10–30 million | Medium | Moderate |
| RNN (LSTM) | 15–50 million | Medium–High | High |
| GRU | 10–40 million | Medium | Moderate–High |
| Transformer | 50–200 million+ | High | Very High |

## V. CHALLENGES IN DEEP LEARNING-BASED SPEECH RECOGNITION

Speech recognition has been quite enhanced through deep learning although there are various barriers to enhancing these systems to be more precise and efficient. These issues cut across technical, practical as well as computational units and they are critical in developing effective real-life applications. Among the key issues, noise and distortions management presents one of them. Background noise may blend with the signal and mask the speech signal hence disadvantage speech recognition systems. This might involve frequent interference such as talking in a facility, automobile sounds or low quality of a microphone. These errors are able to seriously alter the precision of the entire system which in turn makes it difficult to comprehend the words of the speaker. Neural networks (DNNs, in particular) tend to be highly sensible to shifts in input; thus, such models are likely to make errors when faced with noisy environments. Dialect and accent variations is another big challenge. Individuals use various accents and this may make some words be misinterpreted by speech recognition systems. To give an example, a person speaking with a British accent might pronounce words more differently than a person speaking with an American accent, in spite of the fact that both persons speak the same language. Unless speech recognition model has been trained to recognize a wide scope of accents, it can be inaccurate in speech transcribing and the error rates can be increased, particularly in a rather beaten-out environment. A significant obstacle is also speaker variability. Each speaker possesses different voice, speaking style, speed and pitch and that makes the problem of speech recognition even more complicated. One voice-based model trained on a single person voice may not be applicable across other people resulting in a deviation in the performance of a model when it is exposed to new voices. Such a variation as the speed at which the speaker talks or his tone can make this task even more difficult to accomplish so that the speaker-independent recognition becomes harder. The training requirements of deep learning linked speech recognition systems also are very high. A lot of labelled data is needed by these models, hundreds or thousands of hours of audio data with a correct transcription. In the case of languages or dialects that have very few resources, data might not be able to amass adequate data to fill the gaps and hence poor performance when the model is exposed to new words or phrases. The other key issue is real-time processing. Most speech recognition applications, including voice assistants, need almost real-time or prompt transcription. There are deep learning models such as some of the intricate models like the transformers that are computationally difficult and may require time to process an audio, delaying real-time activities. It is necessary that balance be achieved between accuracy and speed of processing so that real-time systems perform well. Also, deep learning models can be a challenge in terms of computational resources requirements. Large parameterized models, whether transformers or deep RNNs, consume a lot of computational resources and especially when training. This requires special hardware like GPUs or TPUs and that this is expensive and might not be applicable to low powered devices such as smart phones or embedded stages. Lastly, there is always a problem of generalizing to reach unseen situations. Deep learning models can be trained on limited layers of data that probably represent all possible variants of speech. As an example, a model trained with clean speech can perform poorly on noisy, accented, or distorted speech in practice. This unpretentiousness may lead to bad performance once the model is applied to new settings that were not included in the training set.

## VI. FUTURE DIRECTIONS

With the evolving current state of speech recognition technology, it has thus been seen that there are a number of captivating future prospects being researched in aid of enhancing quality, diversity and usefulness of the speech recognition technology. Such innovations are an attempt to solve the existing limitations and open up new opportunities in the real world. The development of multimodal models is one of the promising areas. These models combine the recognition of speech with other modalities i.e., vision and touch to offer more context-sensible recognition. One example is to combine speech with some visual input (such as motions of the lips) or some tactile feedback that would increase accuracy in the presence of noise or allow the system to infer more about the context, particularly with context-dependent applications such as virtual assistants or robotics. The other significant responsibility is few-shot learning. Conventionally, speech recognition models need enormous portions of labelled information to teach them. Few-shot learning targets models to generalize on the minimal amount of labelled data, becoming more versatile and applicable in those situations when few annotated speech data are available or hard to record. This may also be of assistance in languages with fewer resources or in specific uses. It is also a persistent problem that model speech recognition is biased and unfair. Most of the models do not perform well in some accents, genders, or dialects and this may cause discrimination or unfairness. It is imperative to mitigate these prejudices and create an even ground where speech recognition tools are equal and can suit any user irrespective of the accent, gender, or location. Another sexy direction is self-supervised learning. This method enables models to be trained using unlabelled data and thus models can be trained on unstructured audio data that is freely available in large volumes. With the help of self-supervised learning, speech recognition systems are able to enhance their performance, and in particular, in the low resource environment, when the labelled data is scarce. Lastly, it is necessary to develop cross-lingual and multi-lingual models in order to create speech recognition systems that will be able to operate in a variety of languages and dialects. These models would enable real-time translation and transcription, allowing users from different linguistic backgrounds to interact with speech recognition systems seamlessly.

## VII. CONCLUSION

The present paper outlined a thorough comparative analysis of many speech recognition architectures of deep learning: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Deep neural networks (DNNs), Transformer-based models. The evaluation of all models was performed on a number of significant parameters, such as Word Error Rate (WER), one of the real-time factors, models complications, and their scalability, and it offers the detailed representation of advantages and limitations of all models. In the paper, the authors underscore how the deep learning has come to achieve impressive improvements in solving speech recognition especially with the transformer-based models which have produced improved results in terms of accuracy and error resilience to noise and challenging conditions. This does not impair the fact that previously existing models, the DNNs, and RNNs, might still be helpful in specific use-cases, but it is observable that the region is transitioning to other more advanced and performance-demanding models, which produce better accuracy and model generalization. In addition to the most significant drawbacks of these models viz, the need of large amounts of labelled data, the difference in application to many contexts, and vast amounts of computation resources. However, a certain advancement in such development as self-supervised learning, few-shot learning, and multimodal integration will be possible to occur in the future and push forward the efficiency, scalability, and robustness of speech recognition models. On the whole, the article creates a positive picture of the current methods in speech recognition that could assist the researchers and practice in selecting the most appropriate deep learning models in the implementation.

## REFERENCES

[1] T. M. Wani, T. S. Gunawan, S. A. A. Qadri, M. Kartiwi and E. Ambikairajah, "A Comprehensive Review of Speech Emotion Recognition Systems," in IEEE Access, vol. 9, pp. 47795-47814, 2021, doi: 10.1109/ACCESS.2021.3068045.

[2] Aggarwal et al., “Two-Way Feature Extraction for Speech Emotion Recognition Using Deep Learning,” Sensors, vol. 22, no. 6, p. 2378, Mar. 2022, doi: 10.3390/s22062378.

[3] Pratama and S. W. Sihwi, "Speech Emotion Recognition Model using Support Vector Machine Through MFCC Audio Feature," 2022 14th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 2022, pp. 303-307, doi: 10.1109/ICITEE56407.2022.9954111.

[4] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, and T. Alhussain, “Speech Emotion Recognition Using Deep Learning Techniques: A Review,” IEEE Access, vol. 7, pp. 117327–117345, 2019, doi: 10.1109/ACCESS.2019.2936124.

[5] J. W. Mao, Y. He and Z. T. Liu, "Speech Emotion Recognition Based on Linear Discriminant Analysis and Support Vector Machine Decision Tree," 2018 37th Chinese Control Conference (CCC), Wuhan, China, 2018, pp. 5529-5533, doi: 10.23919/ChiCC.2018.8482931.

[6] Li Y, Baidoo C, Cai T, Goodlet A, Kusi (2019) Speech emotion recognition using 1d cnn with no attention. In 2019 23rd international computer science and engineering conference (ICSEC), pp. 351– 356. IEEE

[7] Singh YB, Goel S (2021) 1D CNN based approach for speech emotion recognition using MFCC features. Artificial Intelligence and Speech Technology. CRC, pp 347–354

[8] Swain T, Anand U, Aryan Y, Khanra S, Raj A, Patnaik S (2021) Performance Comparison of LSTM Models for SER. In Proceedings of International Conference on Communication, Circuits, and Systems: IC3S 2020, pp. 427–433. Singapore: Springer Singapore

[9] Bhandari SU, Harshawardhan S, Kumbhar, Varsha K, Harpale, Triveni D, Dhamale (2022) On the Evaluation and Implementation of LSTM Model for Speech Emotion Recognition Using MFCC. In Proceedings of International Conference on Computational Intelligence and Data Engineering: ICCIDE 2021, pp. 421–434. Singapore: Springer Nature Singapore.

[10] Choudhary R, Raj G, Meena, Krishna Kumar M (2022) Speech emotion-based sentiment recognition using deep neural networks. In Journal of Physics: Conference Series, vol. 2236, no. 1, p. 012003. IOP Publishing.

[11] Zhao P, Liu F, Zhuang X (2022) Speech sentiment Anal using hierarchical conformer networks Appl Sci 12:16 10. Ishaq M, Khan M, Kwon S (2023) TC-Net: A Modest & Lightweight Emotion Recognition System Using Temporal Convolution Network. Comput. Syst Sci Eng 46(3):3355–3369.

[12] Ye JX, Wen X-C, Wang X-Z, Xu Y, Luo Y, Wu CL, Chen LY (2022) GM-TCNet: Gated multi-scale temporal convolutional network using emotion causality for speech emotion recognition. Speech Commun 145:21–35.

[13] Patni H, Jagtap A, Bhoyar V, Gupta A (2021) Speech emotion recognition using MFCC, GFCC, chromagram and RMSE features. In 2021 8th International conference on signal processing and integrated networks (SPIN), pp. 892–897. IEEE.

[14] Abbaschian, B. J., Sierra-Sosa, D., &Elmaghraby, A. (2021). Deep learning techniques for speech emotion recognition, from databases to models. Sensors, 21(4), 1249.

[15] Toshniwal, S., Sainath, T. N., Weiss, R. J., Li, B., Moreno, P., Weinstein, E., & Rao, K. (2018, April). Multilingual speech recognition with a single end-to-end model. In 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4904–4908). IEEE.

[16] P. B. Atosha, E. Özbilge and Y. Kırsal, "Comparative Analysis of Deep Recurrent Neural Networks for Speech Recognition," 2024 32nd Signal Processing and Communications Applications Conference (SIU), Mersin, Turkiye, 2024, pp. 1-4, doi: 10.1109/SIU61531.2024.10600944.

[17] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4774–4778. IEEE, 2018.