Synthetic voice detection using quantum convolutional neural network with attention mechanism model for authentication applications

Mr. Sukumar B S[1], Dr. G N Kodanda Ramaiah[2], Dr. Santhosh B Panjagal[3],
Mrs. Kavya S[4], Dr. Lakshmipathy M[5]
[1]Research scholar, Department of ECE, Visvesvaraya Technological University, Belagavi, India
[2]Professor and HOD, Department of ECE, Kuppam College of Engineering, Kuppam, AP, India
[3,5] Associate Professor, Department of ECE, Kuppam College of Engineering, Kuppam, AP, India
[4] Assistant Professor, Department of ECE, C. Byregowda Institute of Technology, Karnataka, India

Abstract: This research focuses on the real-time identification of Synthetic voices (Deep fake voices) based on their discrimination between real human voices and the digitally synthesized ones based on the binary classification approach. The methodology is based on a combination of feature extraction via Mel-Frequency Cepstral Coefficients (MFCC), combined with a quantum convolutional neural network (QCNN) enhanced through an attention mechanism (AM). This integration significantly improves the model's ability to identify synthetic voices with high precision and accuracy. The study utilized the Kaggle dataset, containing 320 real and synthetic audio samples, which were pre-processed to remove duplicates and zero-byte files and normalized for consistent analysis. The QCNN model with the attention mechanism was seen to have fantastic results, that is, with 95% accuracy, precision of 96%, recall of 98%, and an F1 score of 97%. Such performance metrics represent a huge increase over the baseline models, namely CNN-LSTM, CNN-BiLSTM, ResNet18 + KNN, and even custom architectures for CNNs. This work demonstrates the efficiency of quantum-enhanced deep learning techniques and places the QCNN with an attention mechanism as a robust and scalable solution for deep fake voice (DFV) detection, which provides reliable performance across diverse scenarios. It opens the doors to further improvement in synthetic media detection and authentication systems.

Keywords: Deepfake Voice Detection, Quantum Convolutional Neural Networks (QCNN), Mel-frequency Cepstral Coefficients (MFCC), Attention Mechanism, Binary Classification.

I. INTRODUCTION

Deepfake refers to a kind of digitally produced media in which computer-generated characters stand in for real people in an image, video, or audio file [1] . The term "deepfakes" initially appeared on Reddit in 2017 when someone using by that username uploaded a fake film to the platform that included a different actor's face. A plethora of legal concerns are certain to arise with this new technology that violate people's rights to privacy, reputation, and portraiture, as well as affect companies' bottom lines and reputations [2]. A media crisis, social turmoil, and political instability may also result from the publishing of a government-affiliated or politician-endorsed bogus film [3-4]. concerning issues of validity, social security, and privacy, deep fakes are becoming an increasingly serious problem. Deepfakes in voice are artificial intelligence produced or altered sounds that seem authentic. The capacity to identify voice deepfakes is essential since, in recent times, they have been used in a number of illegal acts. The field of study on deepfake detection in text, video, and audio is large and dynamic. The number of publications regarding deepfake increased significantly (from 60 to 309) between 2018 and 2019 [5].

These days, individuals live with smart voice assistants. Smart voice assistants like Siri from Apple, Cortana from Microsoft, and Bixby from Samsung are now integrated into a lot of smart products. By identifying the sounds that users make, these intelligent voice assistants may provide tailored services. The primary method for speaker voice recognition is Automatic Speaker Verification (ASV) [6]. The biometric person authentication system is a handy system that detects might be determined by listening to recordings of their voice. Technological advancements in deep neural networks have allowed the automatic speaker verification system to achieve faultless effects. Its many practical and theoretical applications include, but are not limited to, speech-based emotion identification, smart voice assistants, safe building entry, online purchasing, still there are a lot of security holes in ASV systems [7].

Voice conversion (VC), text-to-speech (TTS), and replay are the three most prevalent forms of assaults. As a simple assault method, VC alters speech without influencing the speaker's unique characteristics. Transcribing text to speech (TTS) makes text seem more natural and less robotic than synthetic speech. A replay attack involves the criminal sending a previously received message back to the intended host in order to trick the system. In most cases, it renders identity verification inaccurate. The goal of a replay attack in an ASV system is to impersonate the intended speaker by capturing their voice and then trying to pass the authentication procedure off as them. Deep learning technology has advanced to mimic natural speech, making these assaults more realistic. This is a significant obstacle for the ASV system [8].

To address these issues, it requires a very effective system to discern between actual and fake voice. From the start, various assessment criteria are applied for distinct datasets, preventing comparison of outcomes. A community with standardized datasets and evaluation criteria was established via the creation of many anti-spoofing tournaments. Of them, the ASV spoof challenge, which stands for automated speaker verification spoofing and countermeasures—is the most famous. A first dataset addressing mechanisms for and against automated speaker verification spoofing that encourages research in this area is ASVspoof 2015 [9]. Less than 1.5% is removed from the equivalent error rate (EER). A few assaults have an EER of even 50%. Unknown assaults however, may have five times higher EER. Additionally, the limitations of ASVspoof2017 concerns itself with identifying replay spoofing assaults [10]. The effectiveness of countermeasures is significantly increased by the Instantaneous frequency cosine coefficients (IFCC) and the EER of 6.73%. Afterwards, ASVspoof2019 [11] focused more on countermeasures for automated detecting audio faking and verifying speakers for low-quality audio spectrograms, it is also used in computer vision methods like convolutional neural networks (CNN). to recognize fake speech [12]. With CNN-based models, the temporal information may be lost. Therefore, to improve automated speaker verification and spoof audio recognition, temporal convolutional neural networks are applied in probabilistic forecasting [13].

The limitations of synthetic voice recognition algorithms, as described in earlier research, draw attention to a number of difficulties. For example, when applied to unknown assaults using the ASVspoof 2019 dataset, [14-15] found that overfitting with synthetic data limited the generalizability of Residual CNN and Anti-Spoofing with Squeeze-Excitation and Residual neTworks (ASSERT) (SENet + ResNet), respectively. The ResNet-34 model by [16] requires a modification in the 2-D feature map, which increases the calculation time and reduces the speed. Moreover, the DL-based methods and traditional classifiers of [17] failed to capture significant characteristics properly and needed a lot of human effort to extract features from the Arabic diversified audio dataset (AR-DAD) dataset. Scaling problems also resulted from manually derived features by [18-19]. The profound effect of real-world sounds made the Deep-Sonar model of [20] suffer.

Other models, such as CRNN-Spoof by [21] and WIRE-Net-Spoof, linked poor performance combined with computational expenses to other models. Low robustness to varied datasets affected CNN-based methods [22-23] and models like Deep4SNet [24], which requires deepfake detection systems to have better generalization and adaptability, finally affecting scalability issues, high data transformation needs, and low This research offers a novel method to restrict the drawbacks of previous deep fake voice recognition models and help overcome these obstacles by improving generalization and scalability. Therefore, in order to achieve this objective, this research have put forward a methodology that relies for feature extraction, a mix of Mel-Frequency Cepstral Coefficients (MFCC) and Quantum

Convolutional Neural Networks (QCNN) is used, one of several machine learning techniques. The QCNN model, augmented with an attention mechanism, is specifically designed to boost the accuracy of binary classification tasks for synthetic audio detection. The assessment is performed on the Fake-or-Real dataset, with a specific emphasis on recognition of deepfake audio using this novel methodology. The goal of this research project is to find a way to detect deepfake voices in real or non-synthetic audio. The research has made the following primary contributions:

- Employing binary classification, build a QCNN-based deepfake speech recognition system with an eye on synthetic voices.
- Perform thorough tests using QCNN and attention methods on the dataset and its subsets to extend research using the Fake-or-Real dataset for deepfake audio detection.
- Apply a better MFCC feature extraction technique to handle audio data thereby improving deepfake voice recognition.
- Analyse and contrast the performance of the QCNN model with attention mechanisms against other machine learning models across many dataset subsets; note that the QCNN model shows rather excellent results, especially for the for-original dataset.

The following is the paper's framework. Section II lays out the suggested approach and algorithms. Section III presents the assessment and experimental outcomes. The suggested technique is discussed in length in Section IV. The whole study is summarized in Section V.

II. BACKGROUND

A. Mel-frequency Cepstral Coefficient (MFCC)

A nonlinear Mel frequency scale that depicts the short-term power spectrum of a sound using a linear cosine transform of a log power spectrum is the Mel-frequency cepstrum (MFC). Since its inception by Mel-frequency Cepstral Coefficients (MFCCs), one of the most prominent and widely used parts of SR systems has been [25]. Many speech recognition applications and studies employ MFCC as their foundational model for parameterizing speech. The fact that MFCCs enhance the representation of sound by mimicking how people hear frequencies might explain this. A Mel-frequency cepstrum is the source of these (minimize-of-spectrum), which is more in line with how the human auditory system works than the conventional cepstrum, which uses frequency bands that are linearly spaced. The MFCC's block diagram is shown in Figure 1.

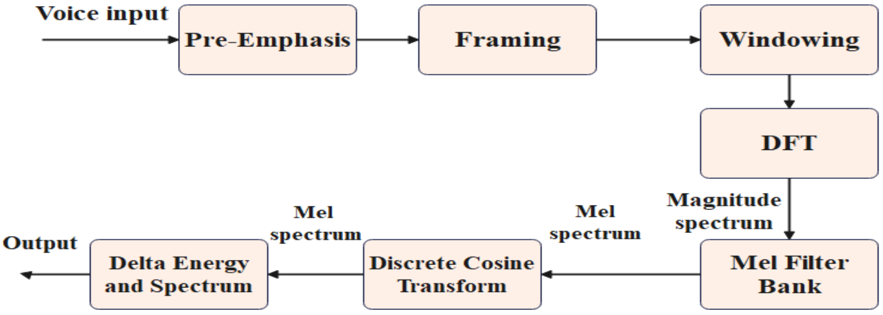

***Figure 1** MFCC Block Diagram.*

Step 1: Pre-emphasis:

The first step in using speech processing for applications like speech recognition and linear prediction (LP) analysis-synthesis is to boost the signal at higher frequencies. For most pre-processing tasks, this is now standard procedure. At this stage, the signal is processed after passing through a filter that prioritizes higher frequencies. If you follow these instructions, you can boost the signal's strength and frequency.

$$Y\left[n\right] = X\left[n\right] - a\,X\left[n-1\right] \qquad \text{Eq. (1)}$$

Step 2: Framing:

Due to the unstable nature of speech transmissions, framing is used to divide the long-term voice signal into the shorter-term speech signal in order to achieve somewhat constant frequency characteristics. Feature extraction occurs at regular intervals. Data gathered within a certain time span, or "window," is what is processed in relation to the signal.

Step 3: windowing:

In order to reduce the frequency domain aliasing effect of shortening a long signal, windowing is often used.

There are several kinds of windows, including:

- Window that is rectangular
- The Bartlett window
- The Hamming window

Among them, the Hamming window is the most frequently used. With all surrounding frequency lines integrated and the next step in the feature extraction processing chain in mind, the hamming window becomes an efficient window form.

The equation for the Hamming window is provided as:

If the window is defined as W(n), $0 \leq n \leq N-1$ where N denotes number of samples in each frame, Y(m) represents Output signal, X(m) indicates input signal and W(m) represents Hamming window, Then the result of windowing signal is shown below:

$$Y(m) = X\ (m) * W\ (m) \qquad \text{Eq. (2)}$$

Step 4: Fast Fourier Transform:

In order to translate all data frames from the time domain into the frequency domain. By using the Fourier Transform in the time domain, the glottal pulse and the vocal tract's impulse response are converted.

$$Y(w) = FFT\left[h(t) \times X(t)\right] = H(w) \times X(w) \qquad \text{Eq. (3)}$$

Here X(w), H(w) and Y(w) respectively represents the Fourier Transform of X(t), h(t) and Y(t).

Step 5: Mel Filter Bank Processing:

The Fast Fourier Transform (FFT) spectrum covers a large range of frequencies since spoken sounds are not linear. Here filters arranged in a triangle format, when combined, provide an output that is approximately Mel-scale-compatible by virtue of the weighted sum of the filters represents the spectral components. There is a linear drop Since human speech is not linear, the FFT spectrum encompasses a wide frequency range. triangular array of filters and a unity-shaped magnitude-frequency response at the centre frequency of each filter. An individual filter's final product is the total of all spectral parts that have been filtered.

Step 6: Discrete Cosine Transform:

Here we see the Discrete Cosine Transform (DCT) in action, transforming the log Mel spectrum into the temporal domain. Time of My Mel The Cepstrum Coefficient is the product of the two variables after conversion. The term "acoustic vector" describes sets of coefficients. Consequently, an aural vector sequence is generated from each input syllable.

Step 7: Delta Energy and Delta Spectrum:

A formant's slope during a transition is one example of how frame and speech signal changes. As a result, aspects pertaining to how cepstral features evolve throughout time must be included. Consequently, characteristics are introduced that are double delta or acceleration features and delta or velocity features (cepstral features + energy).

B. Quantum Convolutional Neural Networks (QCNN)

Virtual Neural Network A recent proposal, QCNN [26], aims to bring Convolutional Neural Networks 's quantum domain properties to processing images on a computer model. During Quantum Machine Learning (QML) development, it employs Variational Quantum Circuit (VQC) as a quantum kernel convolution filter and employs less qubits to build a convolution kernel. The use of qubits enhances the method in the Quantum Convolutional Neural Network. The use of qubits makes QCNN different from CNN. it makes the model more scalable and allows for better utilization of quantum computing.

The input to the Quantum Convolutional Neural Network (QCNN) system is the voice, and the dataset is utilized to determine whether the input matches the dataset. Then, the quantum filters are applied. Once it has converted the input to qubits, it does the extra operations. The picture illustrates the process of converting voice input into text. To improve the model's scalability and accuracy, add additional layers. A direct correlation between the classical input data and its corresponding quantum state is provided by the architecture for quantum encoding. Using a quantum input vector to make a classical
embedding is what the procedure is all about, to rephrase. In quantum encoding, the letter U is used. A mixture of Controlled-NOT (CNOT) gates and movable rotation gates SX, SY, and SZ may be used to represent the quantum state.

Qubit entanglement is accomplished via the CNOT gates' obligatory entanglement of any two atomic wires. You may change the rotation angles II and I using the trainable parameters SX, SY, and SZ, respectively. In addition, SX, SY, and SZ associated with the rotation gates indicate a direct connection between the inputs and outputs, forming a straight line. One of the primary benefits of the Quantum Neural Network (QNN) model is the use of fewer approaches. The visual depicts a QNN model with 72 trainable parameters, four quantum wires, six VQC layers, and more.

$$S_x(a)=\begin{bmatrix}\cos\frac{a}{2} & -i\sin\frac{a}{2}\\ -i\sin\frac{a}{2} & \cos\frac{a}{2}\end{bmatrix} \quad \text{Eq. (4)} \qquad S_y(b)=\begin{bmatrix}\cos\frac{b}{2} & -\sin\frac{b}{2}\\ \sin\frac{b}{2} & \cos\frac{b}{2}\end{bmatrix} \quad \text{Eq. (5)} \qquad S_z(c)=\begin{bmatrix}\exp\left(-i\frac{c}{2}\right) & 0\\ 0 & \exp\left(i\frac{c}{2}\right)\end{bmatrix} \quad \text{Eq. (6)}$$

The QNN model based on the down method may be trained using the stochastic gradient descent (SGD) technique, which improves the effectiveness of Adam, RMSprop, and Adadelta.

Quantum-to-Classical Parameter Mapping

First, it constructs a traditional neural network (NN) using the parameter vector $\vec{\theta}=(\theta_1,\theta_2,\ldots,\theta_M)$. $N=[\log_2 M]$qubits are used to encode a quantum state $|\psi\rangle$, resulting in a Hilbert space of length $2^{[\log_2 M]}$ to represent the NN parameters.
For $i\in\{1,2,\ldots,2^N\}$, the quantum state's measurement probabilities, $|\langle i|\psi\rangle|^2$, vary from 0 to 1. Mapping these probabilities to the NN parameters θ_i, which typically range from $-\infty$ to ∞, is the aim.
Based on a second NN with parameters$\vec{\gamma}$, a mapping model $G_{\vec{\gamma}}$ is presented. In order to integrate basis information and measurement probabilities, it requires input vectors $\vec{x_i}$, such as $[0,\ 1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0.023]$ for a 7-qubit system, as seen in $|\langle 0100100|\psi\rangle|^2=0.023$. In contrast to earlier models, the mapping function $G_{\vec{\gamma}}(\vec{x_i})=\theta_i$ dynamically determines, allowing sign flexibility. The model's applicability to a wider range of machine learning (ML) tasks is increased by this modification.

Construction of QNNs

The Ry gate, which is represented by the following, is one of the parameterized rotational gates used to produce the quantum state$\langle\psi|$:

$$Ry(\mu)=\begin{bmatrix}\cos(\mu/2) & -\mathrm{s}in(\mu/2)\\ \mathrm{s}in(\mu/2) & \cos(\mu/2)\end{bmatrix} \qquad \text{Eq. (7)}$$

In a linear arrangement, CNOT gates are used to induce entanglement, guaranteeing parameter scalability as $O(poly\log(M))$. With parameters $\vec{\theta}$ establishing the traditional NN weights via the mapping model $G_{\vec{\gamma}}$ this configuration creates the QNN.

Training Procedure for Quantum-Enhanced Learning

To increase capacity, the QT framework entails building an N-qubit QNN with parameterized Ry gates in blocks that may be repeated n block times. For conventional training, the classical NN employs parameters $\vec{\theta}$ produced via quantum-classical mapping, directed by the cross-entropy loss function:

$$\ell_{CE} = -\frac{1}{N_d}\sum_{n=1}^{N_d}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n) \right] \qquad \text{Eq. (8)}$$

where the real and predicted labels are denoted by y_n and $\hat{y}_n$, respectively. For shot-based simulations, gradients for the QNN parameters $\vec{\theta}$ and mapping model $\vec{\gamma}$ are calculated analytically or via the parameter-shift rule. By directing repeated updates, these gradients help to improve the NN parameters. The QT framework, improves practicality under constrained quantum resources by supporting inference on classical hardware and optimizing performance by lowering QNN parameters to $O(poly\log(M))$, as opposed to M for classical NNs.

C. Attention mechanism

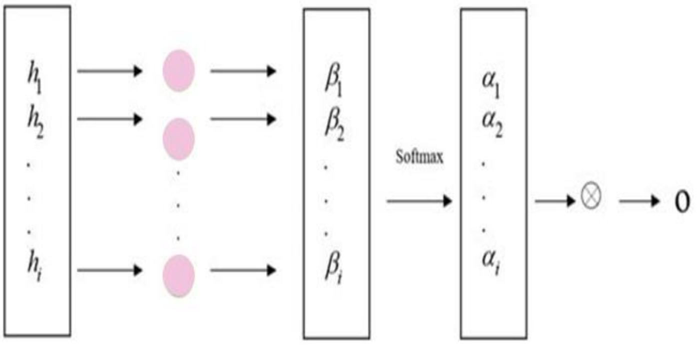

Figure 2 *Attention network structure diagram*

The attention mechanism is based on the human brain's ability to process more important information with less mental effort [27]. After the CNN has recovered features, an attention mechanism may be used to apply different weight coefficients to those features was built as a consequence of this work. This mechanism was established in order to further assign these features. The ability of the model to generate predictions was thus enhanced as a result of this situation.

Figure 2 provides a pictorial picture of the Attention structure that may be seen. The Self-Attention module is responsible for applying time-related feature vectors that the CNN used, together with their respective weights and biases develops via its learning process. Following that, the computer is trained with the help of these feature vectors. In order to carry out deep-level feature mining, the weighted summation method was decided upon as the most appropriate strategy.

Following the completion of the independent calculation of the weight of each characteristic, this was carried out. For the purpose of calculation, the formula that is as follows is utilized:

$$\beta_i = \sigma\left(W_i h_t + b_i\right) \qquad \text{Eq. (9)}$$

$$a_i = soft\max\left(\beta_i\right) = \frac{\exp\left(\beta_i\right)}{\sum_i \exp\left(\beta_i\right)} \qquad \text{Eq.(10)}$$

$$O = H \otimes a_i \qquad \text{Eq.(11)}$$

The activation function is represented by β_i, the feature's relevance is represented by a_i, the attention weight is represented by O, and the output prediction result is represented by W_i, bi denotes the bias vector between neuron nodes, σ represents the weight matrix.

III. METHODOLOGY

A. Data Collection

We collect and process data from the Kaggle dataset on deepfake voice identification. The work Commence by acquiring the dataset from the designated hyperlink and extracting the files to a specific directory on your own computer. After extracting

the files, we analyse the composition of the dataset, which typically consists of folders labelled "real" and "fake," each containing audio recordings that fit into their respective categories. In order to import the data, we use Python modules like OS and Pandas to traverse through the folders and create a comprehensive list of file paths along with their corresponding labels. To handle a more manageable and smaller group, we selected a sample of 320 files. We ensure that this sample has a fair representation of both actual and fake classes. Making sure the sample's class distribution matches the dataset's structure is considered crucial. Finally, for further analysis, we use tools like librosa to import and manipulate the audio recordings.

B. Dataset:

https://www.kaggle.com/datasets/birdy654/deep-voice-deepfake-voice-recognition

C. Data Preprocessing

To get the data ready for our research on deepfake voice recognition using the Kaggle dataset, you would first need to import Standard Scaler from Sklearn along with other necessary libraries like Pandas, Numpy, and Librosa. Next, load the metadata CSV file to understand the labels and structure associated with the audio recordings.

We ensure that we correctly reference the directory containing the audio files. Pre-processing should include a phase to eliminate duplicate files and files with zero bytes in size. Use an iterative process with the audio files to add valid entries to a list of unique files. Librosa can load and save audio at a preset sample rate if required, use it to standardise the bit rate of the audio files.

Next, take each audio file's MFCC characteristics and calculate its mean MFCC values for consistency. Standardise the feature values across the dataset by using Standard Scaler for these extracted features. Create a label array at the end, making sure it matches the audio files based on the information. For model training, we combine the normalised MFCC features and labels into a final Data Frame or NumPy array, then save the processed dataset for later use. Our comprehensive preparation procedure guarantees that what we provide is standardised, clean, and well-suited for efficient model training in deepfake audio recognition.

D. Feature Extraction using MFCC

the accuracy and prediction potential of the model may be significantly impacted by the derived characteristics. According to observations, The feature sets used to create deepfake audio signals often closely resemble those used to create authentic signals that are useful for identifying and categorising deepfake sounds, which is in some situations can fool humans. We use Mel-frequency Cepstral Coefficient (MFCC), a characteristic that is often utilised for voice detection [28-29].

To keep things up-to-date, we conducted this study using the fake or genuine audio dataset. A log function, Mel-filter and triangle band-pass filters are used to simulate human perception while hiding the frequency information, our major emphasis is on MFCC characteristics since they replicate the human hearing system. This is the reason why we are concentrating on them. In spite of this, we did not rely only on MFCC characteristics for our investigation. Additionally, in order to create a featured ensemble, MFCC, the spectrum, the raw signal (with no cross rate), the energy quality of the signal, and the roll-off point, centroid, contrast, and bandwidth. In order to accurately detect deep-fake noises, this study used MFCC. After that, we convert each audio waveform frame into a vector group that represents the MFCC. This is done after we have first processed the audio signals.

In order to simulate the human hearing system, MFCC used the Mel filter and the log function. Also, the MFCC and triangle band-pass filters hide the frequency data so it seems like what the human eye would see to identify deep-fake voices. The proposed framework uses several artificial intelligence (AI) with and without supervision techniques to manage the massive data training set and carry out detection. Figure 3 displays the comprehensive architectural design of the suggested framework, encompassing 1) data preparation, 2) feature extraction, and 3) models for classification.

E. Train-test split

We may use Python and other tools such as Pandas and Scikit-learn to do a train-test split on the dataset that was provided via the Kaggle link for our study on deepfake voice recognition. To begin, we will be loading the dataset into a data frame first. The data will next undergo any required preparation, which may include the removal of duplicates and the standardisation of formats, among other things.

The data will be divided between separating the training and testing sets using Scikit-Test Split. after it has been prepared. "Your-File-Path" will be replaced with the actual path to our dataset file and "label-column" will be updated with the name of the column that contains our labels, such as "real" or "fake." This method will be carried out and completed. We are able to change this ratio as required nevertheless, the test size=0.2 option will utilise twenty percent of the data for testing purposes. For the purpose of preserving the proportion of classes in the dataset, we guarantee that our divide is maintained by setting stratify=y. The train-test split for our deepfake voice recognition study may be generated in an effective manner with the help of this strategy.

F. Model building

The suggested approach uses Quantum Convolutional Neural Networks (QCNN) for binary classification in the field of deepfake voice recognition, with a focus on distinguishing between artificial and authentic human voices. The first step in this creative method is the extraction of Mel-frequency Cepstral Coefficients (MFCC), a crucial feature set that captures the main qualities of audio signals. The model improves its emphasis on the most important aspects by including an attention mechanism into the QCNN framework, which enables better differentiation between real and fake voices.

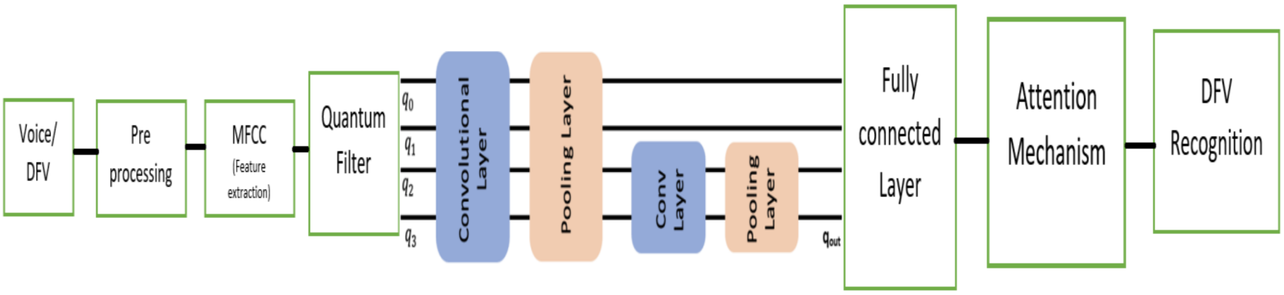

Figure 3: Graphical Representation of Proposed Approach for recognition of deepfake voice

The QCNN is able to prioritize the most relevant portions of the audio input, increasing classification accuracy, thanks to the attention mechanism, which dynamically weights various MFCC properties. The integration of QCNN and attention not only enhances the model's efficiency but also tackles the inherent difficulties brought about by the intricacies of deepfake voice data. This opens the door for improving detecting methods to make them more robust and reliable in the dynamic field of synthetic audio material.

G. Model Evaluation: The confusion matrix's performance measures, such as the F1-score, accuracy, sensitivity, and precision, are used to rate algorithms.

Accuracy: A subject detection rate is the number of correctly named subjects as a percentage of all the subjects.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{Eq. (12)}$$

Precision: You can get an idea of how accurate a view is by keeping track of how many right statements it makes. The idea behind this could also be called "predictive value."

$$Precision = \frac{TP}{TP + FP} \qquad \text{Eq. (13)}$$

Recall: False negatives, on the other hand, are different from True Negatives.

$$Recall = \frac{TP}{TP + FN} \qquad \text{Eq. (14)}$$

F1-Score: The F1-score is a measure of both accuracy and memory.

$$F_1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \qquad \text{Eq. (15)}$$

IV. RESULTS

Our results are shown below Quantum Convolutional Neural Network (QCNN) with attention mechanism model when applied to the problem of deep fake voice categorization. We first downloaded and analysed audio data, transforming it into spectrogram pictures to capture key information. The dataset was then updated to improve model resilience and separated into training and testing sets. The QCNN with attention mechanism model was trained to separate the spectrograms into two categories: Real (0) and Fake (1). The two confusion matrices that have been presented highlight the difference in performance between two distinct models.

It can be seen from the first confusion matrix, which is derived from an Custom CNN model, that 66 occurrences were accurately categorized as class 0, while 54 examples were identified as class 1. Nevertheless, it also incorrectly classed six occurrences of class 0 being mistaken for class 1, and two occurrences of class 1 being mistaken for class 0. The second confusion matrix, on the other hand, indicates that the QCNN with attention mechanism model fared better, with 109 cases properly placed in the "class 0" category and thirteen in the "class 1" category. While four cases of class 1 being mistakenly identified as class 0 occurred, only two cases of class 0 being improperly labelled as class 1 occurred. The Custom CNN model is compared to, the QCNN with attention mechanism model shows superior accuracy and recall, especially when it comes to categorizing instances as belonging to class 0. This indicates that the QCNN model achieved better overall performance.

Evaluation metrics:

A comparative study with the state-of-the-art models in terms of the standard evaluation metrics: Accuracy, Precision, Recall, and F1-Score is provided below. The Table 1 contains the comparative results. Compared across all the critical criteria, such as accuracy, precision, recall, and F1-score, it has been seen that deepfake speech recognition models exhibit differences in performance (see Figure 7). Islam Altalahin et al.'s CNN-LSTM model had 88% accuracy, 89% precision, 88% recall, and 88% F1-score. With an accuracy of 95%, precision of 96%, recall of 98%, and F1-score of 97%, the QCNN with Attention Mechanism outperforms it by far, even if these metrics seem to point at a very strong performance.

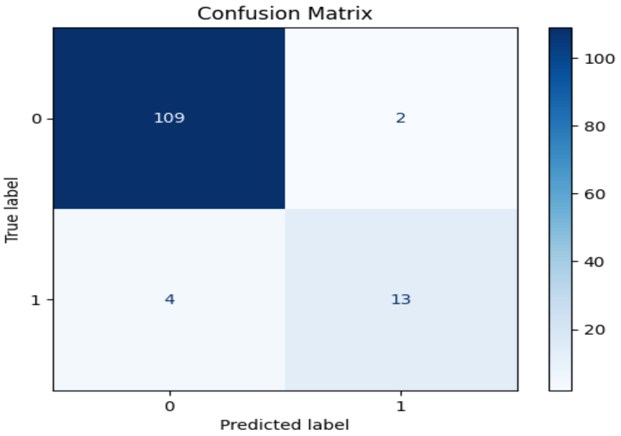

Figure 5 *QCNN with Attention mechanism Model of Confusion Matrix*

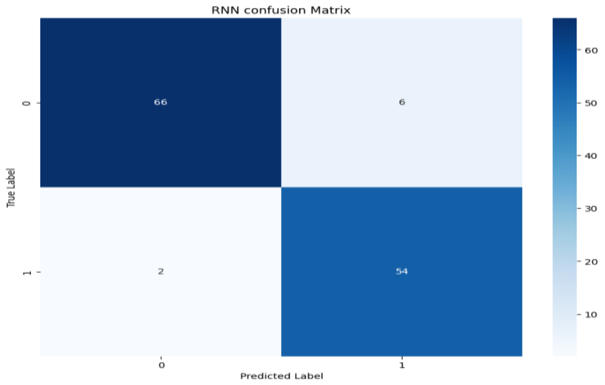

Figure 6: *Custom CNN Model of Confusion Matrix*

Table 1 Evaluation Metrics

Source	Model	Accuracy	Precision	Recall	F1-Score
Islam Altalahin et al [30]	CNN-LSTM	0.88	0.89	0.88	0.88
Lui et al. [31]	CNN and BiLSTM based neural network	0.82	0.85	0.81	0.83
Rafique et al. [32]	ResNet18 + KNN	0.89	0.89	0.89	0.89
Camacho et al. [33]	Custom Convolutional Neural Network (CNN)	0.88	0.90	0.87	0.95
Proposed Model	QCNN with Attention Mechanism	0.95	0.96	0.98	0.97

Besides, it performs better in lowering false negatives, resulting in a better precision-recall trade-off. Similarly, Lui et al.'s CNN + BiLSTM neural network achieves 82% accuracy, 85% precision, 81% recall, and 83% F1-score. Even while it can process data sequentially, it is not as successful as the QCNN, which has a 13% higher accuracy and a much better recall, demonstrating its improved capacity to identify deepfake voices.

The ResNet18 + KNN model, by Rafique et al., had an accuracy, precision, recall, and F1-score of 89%. So, the performance level is fairly good. As such, it can be observed that the QCNN surpasses this model by showing a notable increase in the accuracy level at 6%, with notable increases in precision and recall. Furthermore, Camacho et al.'s proprietary CNN model obtains an F1 score of 95%, accuracy of 88%, precision of 90%, and recall of 87%. By obtaining 7% greater accuracy, 6% better precision, and 11% higher recall, the QCNN with Attention Mechanism demonstrates its improved capacity to analyse complicated audio patterns and maintain a better balance of detection metrics, even with its strong performance. In every case of deepfake voice detection, the QCNN with Attention Mechanism outperforms every high-performing model each time.

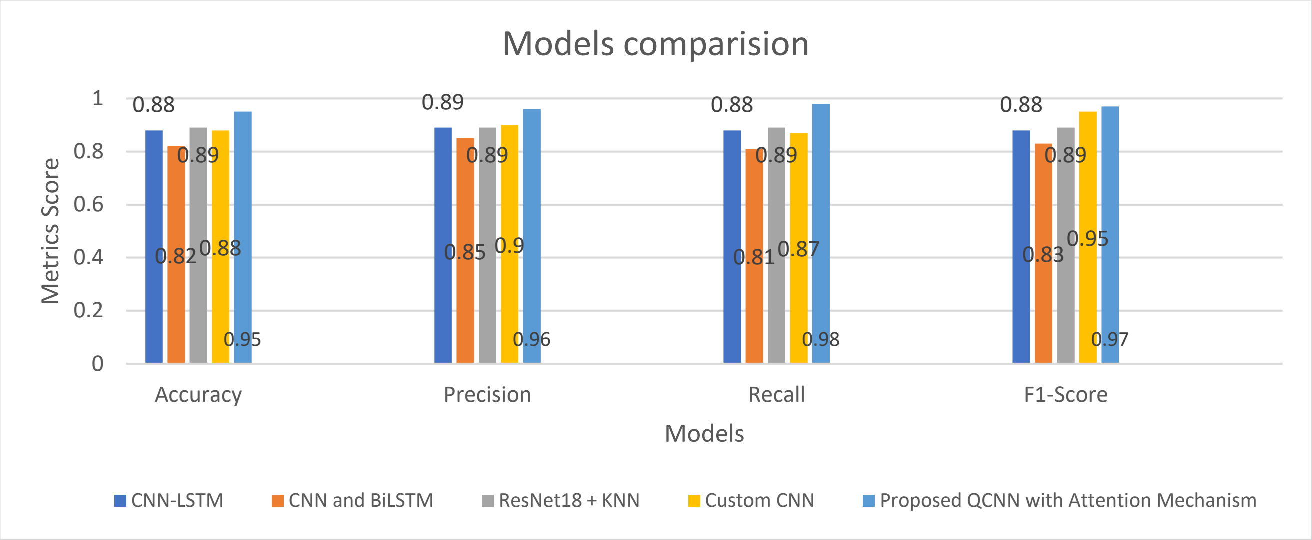

Figure 7 *Models Comparison*

The improved measures reveal how robust the QCNN is in the identification of synthetic audio with higher accuracy and fewer mistakes, mainly in terms of recall and F1-score. The QCNN is an extremely successful model for deepfake audio detection as it applies quantum principles combined with attention processes that produce excellent feature extraction and pattern recognition better than standard and hybrid architectures.

V. CONCLUSION

In conclusion, the investigation clearly indicates that the Quantum All things considered, this research has proved the exceptional deepfake speech recognition capability of the suggested QCNN with Attention Mechanism against various state-of- the-modern models. QCNN always outperformed CNN-LSTM, CNN-BiLSTM, ResNet18 + KNN, and custom CNN architectures on the most crucial evaluation criteria. Its accuracy stood at 95%, while the precision, recall, and F1-score values stood at 96%, 98%, and 97%, respectively.

The QCNN showed significantly enhanced performance compared with models such as CNN-LSTM (88% accuracy), CNN-BiLSTM (82% accuracy), and ResNet18 + KNN with 89% accuracy. However, the precision, recall, and accuracy with the custom CNN model were pretty low compared with the QCNN, even at the point when the F1-score was maximized at 0.95.

These results point out that QCNN is quite capable of synthetic voice detection effectively while reducing false positives and negatives. It can be attributed to the integration of MFCC for feature extraction, the ability of the quantum convolutional architecture to process complex data patterns, and the focus of the attention mechanism on critical features. Such a combination is more accurate and efficient, hence significantly advancing the field of deepfake voice detection.

Overall, QCNN with Attention Mechanism stands as a new benchmark in the detection of synthetic voice, giving a more reliable, scalable, and efficient solution than traditional models. Its real-life use case applications for this include fraud prevention and audio content validation. This advanced methodology not only showcases how much further it is than existing models but also gives a reliable solution toward deepfake audio identification. The QCNN with Attention Mechanism sets the bar for the next wave of synthetic media detection and opens its avenues for its usage in applications where robust, scalable solutions are the need.

References:

[1] A. Abbasi, A. R. R. Javed, A. Yasin, Z. Jalil, N. Kryvinska, and U. Tariq, "A large-scale benchmark dataset for anomaly detection and rare event classification for audio forensics," IEEE Access, vol. 10, pp. 38885–38894, 2022.

[2] A. R. Javed, Z. Jalil, W. Zehra, T. R. Gadekallu, D. Y. Suh, and M. J. Piran, "A comprehensive survey on digital video forensics: Taxonomy, challenges, and future directions," Eng. Appl. Artif. Intell., vol. 106, p. 104456, 2021.

[3] S. Anwar, M. O. Beg, K. Saleem, Z. Ahmed, A. R. Javed, and U. Tariq, "Social relationship analysis using state-of-the-art embeddings," ACM Trans. Asian Low-Resource Lang. Inf. Process., vol. 22, no. 5, pp. 1–21, 2023.

[4]A. Abbasi, A. R. Javed, F. Iqbal, Z. Jalil, T. R. Gadekallu, and N. Kryvinska, "Authorship identification using ensemble learning," Sci. Rep., vol. 12, no. 1, p. 9537, 2022.

[5] C. Stupp, "Fraudsters used AI to mimic CEO's voice in unusual cybercrime case," Wall Str. J., vol. 30, no. 08, 2019.

[6] D. A. Reynolds, "An overview of automatic speaker recognition technology," in 2002 IEEE international conference on acoustics, speech, and signal processing, IEEE, 2002, pp. IV–4072.

[7] M. Todisco et al., "ASVspoof 2019: Future horizons in spoofed and fake audio detection," arXiv Prepr. arXiv1904.05441, 2019.

[8] J. Xue and H. Zhou, "Physiological-physical feature fusion for automatic voice spoofing detection," Front. Comput. Sci., vol. 17, no. 2, p. 172318, 2023.

[9] T. Kinnunen et al., "The 2nd Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof 2017) Database, Version 2," 2018.

[10] T. Kinnunen et al., "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," in Interspeech 2017, International Speech Communication Association, 2017, pp. 2–6.

[11] J. Yamagishi et al., "Asvspoof 2019: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," ASV Spoof, vol. 13, 2019.

[12] S. Ö. Arık, H. Jun, and G. Diamos, "Fast spectrogram inversion using multi-head convolutional neural networks," IEEE Signal Process. Lett., vol. 26, no. 1, pp. 94–98, 2018.

[13] Y. Chen, Y. Kang, Y. Chen, and Z. Wang, "Probabilistic forecasting with temporal convolutional neural network," Neurocomputing, vol. 399, pp. 491–501, 2020.

[14] Alzantot, M.; Wang, Z.; Srivastava, M.B. "Deep residual neural networks for audio spoofing detection". arXiv CoRR 2019, arXiv:1907.00501.

[15] Lai, C.-I.; Chen, N.; Villalba, J.; Dehak, N. "ASSERT: Anti-spoofing with squeeze-excitation and residual networks". arXiv 2019, arXiv:1904.01120.

[16] Aravind, P.R.; Nechiyil, U.; Paramparambath, N. "Audio spoofing verification using deep convolutional neural networks by transfer learning". arXiv 2020, arXiv:2008.03464

[17] Lataifeh, M.; Elnagar, A.; Shahin, I.; Nassif, A.B. "Arabic audio clips: Identification and discrimination of authentic cantillations from imitations". Neurocomputing 2020, 418, 162–177

[18] Rodríguez-Ortega, Y.; Ballesteros, D.M.; Renza, D. A "machine learning model to detect fake voice. In Applied Informatics;" Florez, H., Misra, S., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 3–13.

[19] Singh, A.K.; Singh, P. "Detection of ai-synthesized speech using cepstral & bispectral statistics". In Proceedings of the 2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR), Tokyo, Japan, 8–10 September 2021; pp. 412–417.

[20] Wang, R.; Juefei-Xu, F.; Huang, Y.; Guo, Q.; Xie, X.; Ma, L.; Liu, Y. "Deepsonar: Towards effective and robust detection of ai-synthesized fake voices". In Proceedings of the the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1207–1216

[21] Chintha, A.; Thai, B.; Sohrawardi, S.J.; Bhatt, K.M.; Hickerson, A.; Wright, M.; Ptucha, R. Ptucha "Recurrent convolutional structures for audio spoof and video deepfake detection". IEEE J. Sel. Top. Signal. Process. 2020, 14, 1024–1037.

[22] Subramani, N.; Rao, D. "Learning efficient representations for fake speech detection". In Proceedings of the The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, 7–12 February 2020; pp. 5859–5866.

[23] Bartusiak, E.R.; Delp, E.J. "Frequency domain-based detection of generated audio". In Proceedings of the Electronic Imaging; Society for Imaging Science and Technology, New York, NY, USA, 11–15 January 2021; Volume 2021, pp. 273–281.

[24] Ballesteros, D.M.; Rodriguez-Ortega, Y.; Renza, D.; Arce, G. "Deep4SNet: Deep learning for fake speech classification". Expert Syst. Appl. 2021, 184, 115465.

[25] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," IEEE Trans. Acoust., vol. 28, no. 4, pp. 357–366, 1980.

[26] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, "Quanvolutional neural networks: powering image recognition with quantum circuits," Quantum Mach. Intell., vol. 2, no. 1, p. 2, 2020.

[27] 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.

[28] F. M. Rammo and M. N. Al-Hamdani, "Detecting the speaker language using CNN deep learning algorithm," Iraqi J. Comput. Sci. Math., vol. 3, no. 1, pp. 43–52, 2022.

[29] Z. A. Abbood, B. T. Yasen, M. R. Ahmed, and A. D. Duru, "Speaker identification model based on deep neural networks," Iraqi J. Comput. Sci. Math., vol. 3, no. 1, pp. 108–114, 2022.

[30] Altalahin, I., AlZu'bi, S., Alqudah, A., & Mughaid, A. (2023, August). "Unmasking the truth: A deep learning approach to detecting deepfake audio through mfcc features". In *2023 International Conference on Information Technology (ICIT)* (pp. 511-518). IEEE.

[31] Liu, Z., Guo, Z., Ling, Z., & Li, Y. (2021, June). "Detecting Alzheimer's disease from speech using neural networks with bottleneck features and data augmentation". In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 7323-7327). IEEE.

[32] Rafique, R., Gantassi, R., Amin, R., Frnda, J., Mustapha, A., & Alshehri, A. H. (2023). "Deep fake detection and classification using error-level analysis and deep learning". *Scientific Reports*, *13*(1), 7422.

[33] Steven Camacho, Dora Maria Ballesteros, Diego Renza, "Fake Speech Recognition Using Deep Learning", Communications in Computer and Information Science Applied Computer Sciences in Engineering, 2021, p. 38-48

	Mr. Sukumar B S received the BE degree in Electronics and Communication Engineering (ECE) in Channabasaveshwara Institute of Technology Gubbi, Tumakuru and M. Tech degree in Digital Electronics and Communication Systems (DECS) from Malnad College of Engineering Hassan, Visvesvaraya Technological University (VTU). Presently pursuing Ph.D degree in AI based Signal processing in VTU, India. Currently he is working as Assistant Professor in the Department of ECE at C Byregowda Institute of Technology (CBIT), Kolar, India. His research interests include Signal processing, Artificial Intelligence, Embedded Systems, Electronic Instrumentation, Power Electronics and IoT. He is a member of the IEEE, LMISTE, IEAE. He can be contacted at email: sukumar.svm@gmail.com.
	Dr. G N Kodanda Ramaiah received the BE degree in Instrumentation & Technology at Sri J C College of Engineering, Mysore from Mysore University in 1997 and M. Tech degree in Bio Medical Instrumentation at Sri J C College of Engineering, Mysore from Mysore University in 2001, Ph. D degree in speech signal processing from JNTU India. Currently he is working as Professor and Head of the Department of Electronics and Communication Engineering at Kuppam Engineering College, Kuppam, India. His research interests includes Signal processing, Artificial Intelligence, Bio Medical Instrumentation, IOT and Embedded Systems. He is a member of the IETE, LMISTE, MIE. He can be contacted at email: gnk.ramaiah@gmail.com
	Dr. Santhosh B Panjagal Currently working as an Associate Professor, ECE-Department, KEC-Kuppam, A.P, INDIA. He pursued his B.E in Electronics & Communication Engineering (2006) from PDACE-Gulbarga affiliated to VTU-Belagavi. M.Tech in Embedded Systems 2014 from SVCET-Chittoor, JNTU-Anantapur and Doctorate Degree Ph.D (2022) in Embedded & Wireless Networks from VTU-Belagavi. Published more than 16 research papers in many international journals, presented papers in a national & international conference. He is a life member of IEI, ISRD & IAENG. Received many Research grants from DST, UGC, MSME, IEI etc., and also, published 4 patents for innovative project ideas. His research interests are embedded systems, Wireless Sensor Networks, Energy Storage Systems, AI- Machine Learning, IoT. He can be contacted at email: gnk.ramaiah@gmail.com
	Mrs. Kavya S received her M.Tech in VLSI and Embedded System in C. Byregowda Institute of Technology, Kolar, Visvesvaraya Technological University (VTU), India, in the year 2015. Working as Assistant professor in the department of ECE in C. Byregowda Institute of Technology Kolar from the year 2012 She has guided many students in UG She has attended more than 50 worships and Conferences. Her area of interest is VLSI, Embedded System, and Internet of Things.
	Dr. M. Lakshmipathy is an Associate Professor in the Department of Electronics and Communication Engineering at Kuppam Engineering College, Andhra Pradesh, India. With over 15 years of academic and research experience, he has significantly contributed to teaching, research, and institutional development. He earned his Ph.D. from Visvesvaraya Technological University (VTU), Belagavi, Karnataka, and holds an M.Tech in Electronic Design and Technology from NIT Calicut. His research interests include IoT, AI-driven applications, embedded systems, and air quality prediction models. He has published SCI-indexed research papers on air quality prediction and environmental impact analysis and actively participates in institution-level research initiatives. As a member of the R&D Cell and Entrepreneurship Cell, he fosters a research-driven academic culture. He can be contacted at email: lakshmipathiece@gmail.com