IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Classifying Maqams of Qur'anic Recitations using Deep Learning

SAKIB SHAHRIAR[1], USMAN TARIQ[2], (Member, IEEE)
[1]Department of Computer Science and Engineering, American University of Sharjah, Sharjah, United Arab Emirates
[2]Department of Electrical Engineering, American University of Sharjah, Sharjah, United Arab Emirates

Corresponding author: Usman Tariq (utariq@aus.edu).

ABSTRACT The Holy Qur'an is among the most recited and memorized books in the world. For beautification of Qur'anic recitation, almost all reciters around the globe perform their recitations using a specific melody, known as *maqām* in Arabic. However, it is more difficult for students to learn this art compared to other techniques of Qur'anic recitation such as *Tajwīd* due to limited resources. Technological advancement can be utilized for automatic classification of these melodies which can then be used by students for self-learning. Using state-of-the-art deep learning algorithms, this research focuses on the classification of the eight popular *maqāmāt* (plural of maqam). Various audio features including Mel-frequency cepstral coefficients, spectral, energy and chroma features are obtained for model training. Several deep learning architectures including CNN, LSTM, and deep ANN are trained to classify audio samples from one of the eight *maqāmāt*. An accuracy of 95.7% on the test set is obtained using a 5-layer deep ANN which was trained using 26 input features. To the best of our knowledge, this is the first ever work that addresses maqam classification of Holy Qur'an recitations. We also introduce the "Maqam-478" dataset that can be used for further improvements on this work.

INDEX TERMS Audio classification, Qur'anic *maqāmāt*, deep learning, CNN, LSTM, supervised learning

I. INTRODUCTION

The Holy Qur'an is considered the highest authority and primary source in Islamic creed and law across all sects of the religion. Muslims around the globe recite from the Qur'an during their 5 daily prayers. With almost 25% of the world's population identifying themselves as Muslims [1], the Qur'an is among the most recited books in the world. High religious importance is also placed by Muslim communities on memorization of the Qur'anic text in its original Arabic form. A lot of Muslims around the globe, regardless of their native language, emphasize reciting the Qur'an in its original Arabic language cover to cover, particularly during Ramadan (Islamic month of fasting). The key to perfecting the recitation in Arabic is linked to *Tajwīd* [2], where the focus is on perfecting the pronunciation of each letter and adhering to the grammatical rules of recitations. Although the first step in perfecting the recitation is mastering *Tajwīd*, there is an added emphasis on beautifying the recitation. This is primarily achieved by reciting in one of the eight *maqāmāt* (plural form of *maqām* in Arabic), also referred to as Qur'anic musical modes [3]. These are patterns of melody that beautify the recitation, often expressing the mood and emotions in the set of verses being recited. The maqam technique denotes the pitches, patterns, and construction of a piece of music, unique to Arabic art music [4]. The word *maqām*, from here on referred to as maqam, which literally means a place or position in Arabic. In our context, maqam is the place where the melody occurs [5].

Both *Tajwīd* and maqams are traditionally learned by direct interaction with a qualified teacher. Most reciters learn a specific maqam by simply imitating a famous reciter or his teacher. In the traditional approach, students would take years to study this art from teachers as it is not easy for beginners to distinguish between the maqams, partly because a reciter may mix between the maqams depending on the meaning of the verses being recited. It is well known that different maqams convey different feelings of happiness, sadness, and love among other emotions [6]. The Islamic call for prayer, *Adhan*, is usually chanted using one of the maqams [7]. Studies have also shown that listeners respond more emotionally to hearing recitations in a maqam as opposed to recitations without melody [8]. The 8 primary maqams used for Qur'anic recitations are *Ajam, Bayat, Hijaz, Kurd, Nahawand, Rast, Saba* and *Seka.* These are captured in the popular Arabic phrase used by students, "*ṣuni`a bisiḥrika*"

(meaning "*it was made with your magic"*), where the letters in Arabic represents the initials of the 8 maqams [5].

Since *Tajwīd* is considered more important for a student in Qur'anic recitation, there are numerous resources for students to study the *Tajwīd* rules in both online platforms and Islamic community centers. In contrast, maqam classes are rare in both cases. Due to the growing success of machine learning algorithms in recent years, we find a lot of work dedicated to the use of machine learning in *Tajwīd* education [9], [10], [11]. A full review of automated *Tajwīd* verification can be found in [12]. Other related works include the use of various machine learning algorithms for the identification of the different authorized styles of recitation, *Qira'ah* [13] [14]. In [13], Mel-frequency cepstral coefficients (MFCCs) were first extracted for the 10 styles of recitation which were then used by machine learning algorithms for classification. Support vector machine provided the best performance with 96% accuracy. The authors in [14], however, utilized binary classification to predict the 2 most popular styles of recitations using MFCCs and Hidden Markov Models. Other notable applications include using speech recognition techniques for tutoring in Qur'an memorization [15] as well as classifying the recitations of popular reciters of the Qur'an using machine learning [16], [17].

Despite the various research articles focusing on machine learning with regards to Qur'anic recitation applications, we only find a single work [18] related to maqam classification with learning. It provides a comparison of audio feature extractions of 7 maqams using Cepstral coefficient, MFCCs and discrete-Fourier transform direct warping (W-DFT). The W-DFT method was deemed the most suitable of the 3 due to having fewer fluctuations and hence it was used for further analysis of the 21 recordings (7 from each maqam for verses from 3 different chapters). Graphical plots show that some maqams have more overlapping sections than others across all 3 chapters, whereas some show the opposite. Although this study provides a basic analysis of audio signal processing concerning maqams, it does not provide it with scope for practical applications, due to limitations in number of recordings.

Deep learning and machine learning-based models have been successfully utilized in related applications, including music genre classification [19], [20], [21] and melody extraction [22], [23], [24]. Therefore, the aim of this paper is to present deep learning architectures that can accurately classify maqams of Qur'anic recitations. Audio features including MFCCs, energy and spectral features are used to train various state-of-the-art deep learning models including CNN, LSTM and CNN+LSTM. Results from this study can be utilized to provide maqam tutoring-based systems for students. In such an application, the proposed deep learning models can provide students with real-time feedback by detecting whether the student is able to recite in a specific maqam for a given set of verses. Following are the key contributions of this paper:

- It introduces the "Maqam-478" dataset containing 478 audio files belonging to the 8 maqams. The dataset can be requested from the authors under a licensing agreement.
- It explores suitable audio feature extraction including spectral and temporal features for maqam classification.
- It utilizes various deep learning architectures such as CNN, LSTM, and deep ANN for successful maqam classification.

II. BACKGROUND

In this section, we describe the relevant background information including the various deep learning architectures used as well as the audio feature extraction process.

A. DEEP LEARNING

In machine learning (ML), the objective is to provide a given system with the ability to learn from experience or data without requiring explicit programming. Upon successful training, the system can learn the mapping between input and output features (supervised learning) and make intelligent predictions. Deep learning is considered a subset of ML that primarily makes use of variations of artificial neural networks (ANN). Organized in a layered hierarchy of concepts (the term 'deep' coming from the depth of the layers), complex concepts are defined in terms of simpler ones and more abstract representations are gathered using less abstract ones [25]. Some of the remarkable successes of deep learning-based models include state-of-the-art results in natural language processing and computer vision applications. In [25], the authors attribute the recent success of deep learning to the increasing amount of training data sets as well as the availability of powerful hardware such as graphical processing units (GPUs). In this work, 3 popular deep learning models will be considered, namely, deep ANNs, convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. We will now briefly describe each of them.

1) DEEP ARTIFICIAL NEURAL NETWORKS

Often known as ANNs or multilayer perceptrons, these types of architectures can make use of non-linear approximation to provide a suitable mapping between a given set of input and output features. The structure of a deep ANN, shown in Figure 1, consists of 3 primary layers.

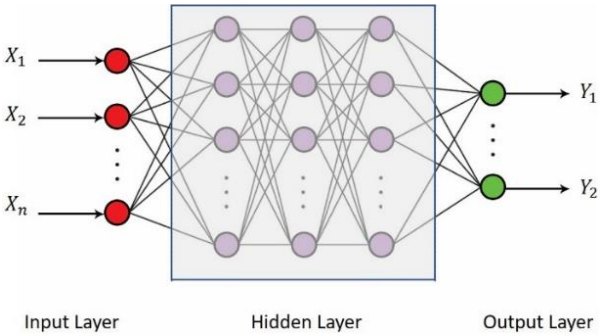

FIGURE 1. An example of a deep ANN with 3 hidden layers [26]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098415, IEEE Access

The input layer accepts a set of input features (the audio features in this case). This is followed by a set of hidden layers (at least 2 hidden layers constitute a deep architecture) that learn the various representations. The output layer is responsible for making the final predictions (the maqam classification in our case).

2) CONVOLUTIONAL NEURAL NETWORKS

A CNN architecture, shown in Figure 2, is generally composed of various layers including convolutional layers, pooling layer, and fully connected layers. The benefit of using convolution operations instead of having fully connected neurons is that the number of total connections and hence calculations are significantly reduced. This makes the training of larger data sets more realistic. Moreover, features can be detected and recognized regardless of their position in the frame [27]. The most common type of pooling operation is max pooling and it significantly reduces the complexity for the deeper layers by selecting the maximum value inside a sub-region after partitioning. A fully connected or dense layer has every node connected to every other node both in the previous and next layer. Following the last pooling layer, the remaining values are flattened into a vector and connected to one or multiple fully connected layers. The fully connected layers generally have the most weights to train and therefore require the most computations. The last fully connected layer contains *k* nodes for *k* classes to be predicted (in classification ML) and usually, a softmax activation function is used to obtain probabilities for each class.

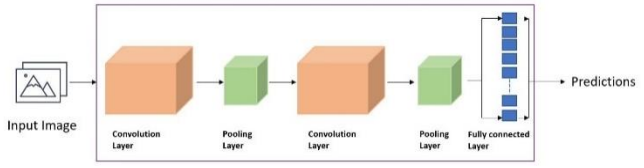

FIGURE 2. **Illustration of a CNN model with 2 convolution layers**

3) LONG SHORT-TERM MEMORY

LSTM [28] is a class of recurrent neural networks that generally contain cyclic connections, unlike feedforward neural networks. This makes recurrent networks suitable for applications that make use of sequential data such as machine translation and text generation. In LSTM-based RNN architecture, specialized memory blocks containing memory cells are present. These are responsible for storing the temporal state of the network along with multiplicative units known as gates for controlling the flow of information [29]. Additionally, they also contain forget gates that are responsible for resetting the internal state values that can grow indefinitely and cause the model to break down. This has an overall effect of resetting or forgetting the memory at a suitable time.

B. AUDIO FEATURES-MFCCs

Amongst various DFT-based spectral feature extraction methods, MFCCs are the most popular and are widely used in speaker recognition applications [30]. The process of computing MFCCs is described in detail in [31]. Figure 3 provides a visual representation of the process. First, the input audio signal is divided into frames by applying the window function at fixed intervals. Then, DFT is applied to each frame to convert the time domain signal into its frequency decomposition. By taking the logarithm of the amplitude spectrum, the phase information is discarded as it is not that significant compared to the amplitude. Next, Mel-scaling and smoothing are performed which emphasizes meaningful frequencies. For instance, it is well known that lower frequencies are perceptually more significant than the higher ones. Since the Mel-spectral values are highly correlated for each frame, applying a suitable transformation, such as Discrete Cosine Transforms (DCT), ensures the components are decorrelated. Following this transform, the output is generally 13-20 cepstral features for each frame, which we can refer to as the MFCCs.

C. OTHER AUDIO FEATURES

Although MFCCs are considered dominant audio feature for various speech related applications, we explored other features as well. In the following, we will visualize some of these features using an audio sample belonging to one of the maqams, arbitrarily selected to be maqam *Seka*. Figure 4 illustrates the waveplot of the signal for the 30-second sample audio. Let us discuss some of the other features.

1) ZERO-CROSSING RATE

We can define it as the number of time-domain zero-crossings within a specific region of a signal divided by the number of samples in that region [32]. In simpler terms, this measures the rate at which the audio signal changes sign, i.e., positive to negative or vice versa. In Figure 5, we illustrate an example of a zero-crossing rate using a small section of the sample audio signal. In this example, a total of 5 instances can be observed where the signal changed signs.

2) CHROMA FEATURE

This computes a chromogram from a given waveform or power spectrogram. The complete spectrum is transformed onto 12 bins that represent the 12 distinct semitones (or chroma) of the musical octave and this could potentially reveal

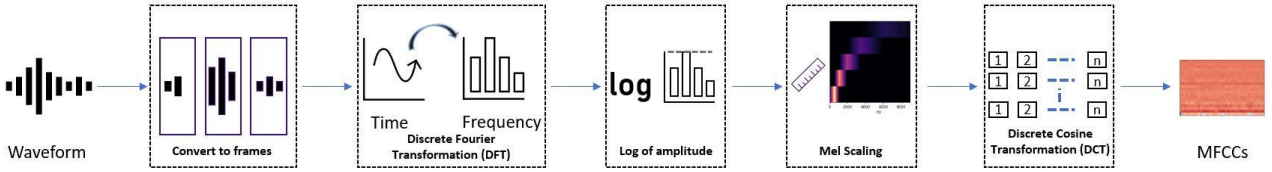

FIGURE 3. **MFCC extraction process**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098415, IEEE Access

perceived musical similarity that is not captured in the original spectra [33]. The chroma feature for the sample audio signal is plotted in Figure 6.

3) ROOT MEAN SQUARE (RMS) ENERGY

This indicates the perceptual concept of the loudness of a given signal, x(n) and is calculated by Equation 1 [34]:

$$E(n) = \sqrt{\frac{1}{N}\sum_{j=0}^{N-1} x\,(n+j)^2} \qquad (1)$$

where n is a discrete-time index and N is the size of the frame. An illustration of the RMS energy of the sample signal is shown in Figure 7.

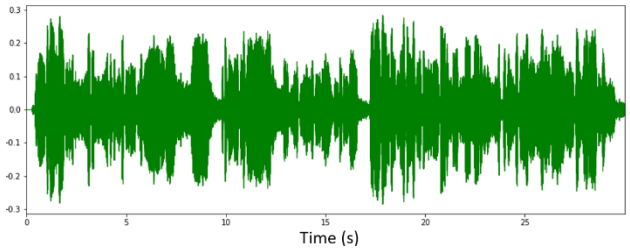

FIGURE 4. **A sample audio file from maqam *Seka***

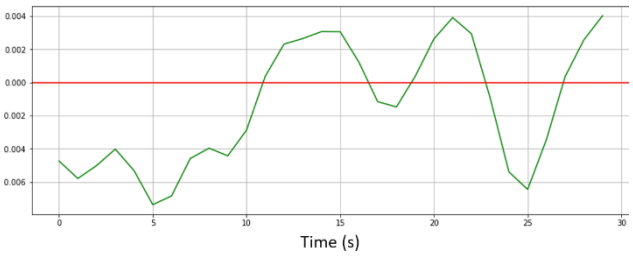

FIGURE 5. **Zero-crossing rate example from a maqam *Seka* sample**

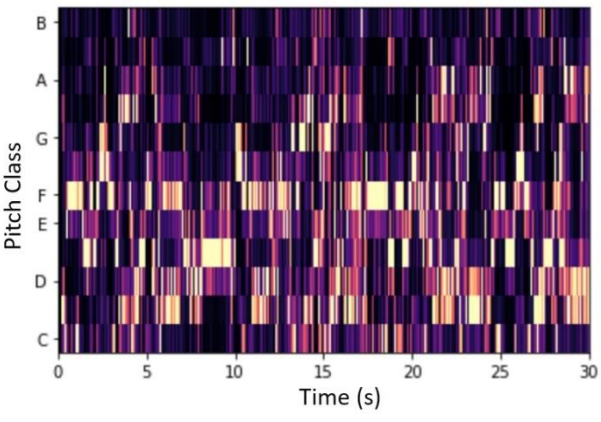

FIGURE 6. **Visualization of chroma features from a maqam *Seka* sample**

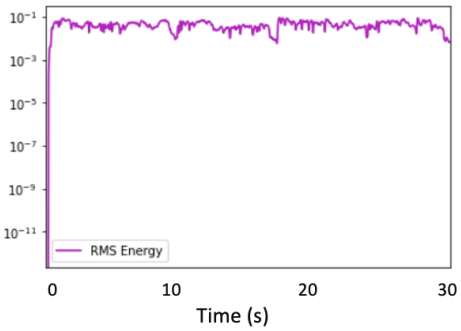

FIGURE 7. **RMS Energy plot from the sample audio signal**

4) SPECTRAL CENTROID

This provides an indication of the location of the center of mass of the magnitude spectrum. Essentially, it is the frequency band where the majority of the energy is concentrated in and it is a measure of the 'brightness' of the sound [35]. The spectral centroid s_c is defined as the weighted mean of the frequencies as follows:

$$s_c = \frac{\sum_{n=0}^{N-1} W_n f_n}{\sum_{n=0}^{N-1} W_n} \qquad (2)$$

where W_n represents the weight of the frequency or the magnitude for the n^{th} bin frequency and f_n represents the center frequency of that bin. In Figure 8, we take sample audio from 3 different classes of maqams and compare their spectral centroid. Despite all the 3 sample recitations being from the same reciter, we can clearly observe the difference in spectral frequencies across the maqams.

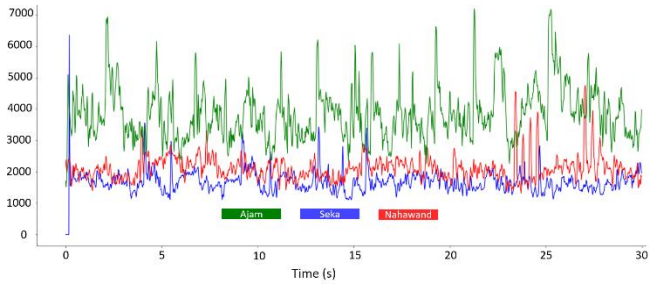

FIGURE 8. **Comparison of spectral frequencies between 3 maqams**

5) SPECTRAL BANDWIDTH

This spectral bandwidth is measured as the weighted average of the differences between the spectral components and the centroid [36]. It can be computed using Equation 3:

$$s_{Bw} = \frac{\sum_{n=0}^{N-1} W_n |n - s_c|}{\sum_{n=0}^{N-1} W_n} \qquad (3)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098415, IEEE Access

Here, W_n and n are defined in the same way as in the previous section. We plot the spectral bandwidth in Figure 9 using the same 3 audio files. Since the spectral bandwidth is derived from the spectral centroid, we get similar trends between the 3 classes. A higher spread of energy across various frequency bands will result in higher spectral bandwidth and vice versa.

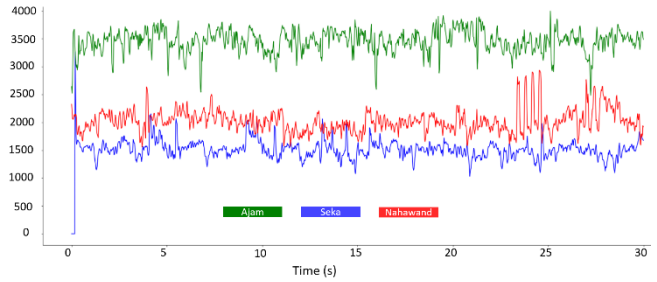

FIGURE 9. Comparison of spectral bandwidth between 3 maqams

5) SPECTRAL ROLL-OFF

Spectral roll-off is defined as the frequency below which a certain percentage of the total spectral energy lies [34]. Typically, the percentage is selected to be 85% or 95%. Figure 10 shows an example of spectral roll-off, for 85% and 95% thresholds respectively, alongside the actual waveplot for the maqam *Seka* sample used in earlier examples.

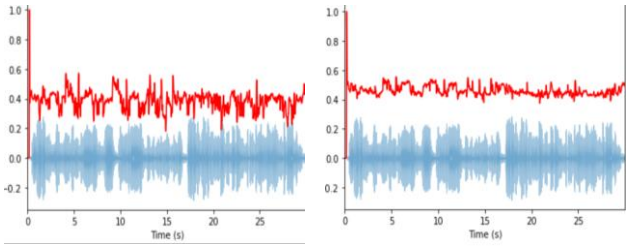

FIGURE 10. Spectral roll-off with 85% (left) and 95% (right) thresholds from a maqam *Seka* sample

III. METHODOLOGY

This section elaborates on the methodology and the steps taken for implementing the various algorithms and performing the classification.

A. DATA COLLECTION

Not all the popular reciters of the Holy Qur'an usually recite in all the maqams. For this research, recordings from two famous reciters were selected, namely, Sheikh Bandar Baleela, Imam (the person who leads the prayer) of the Holy Mosque in Makkah and the late Sheikh Muhammad Ayyub who was the Imam of the Holy Mosque in Medina. We collected a total of 478 audio files consisting of recordings from the 2 reciters across all the 8 maqam classes. Each audio file length was anywhere between 30 seconds and 60 seconds. The dataset is publicly available for research and can be accessed using [37].

Since the used algorithms require a consistent length of audio files, all the recordings in each class were merged and split into 30 seconds .wav files. The total length of recordings for each maqam class is presented in Table 1.

TABLE I
DATASET CLASS DISTRIBUTION

Maqam	Length (s)	Distribution (%)
Ajam	2370	11.3
Bayat	2610	12.4
Hijaz	3150	15.0
Kurd	2220	10.6
Nahawand	2460	11.7
Rast	3090	14.7
Saba	2640	12.6
Seka	2430	11.6
Total: 20970s (~ 6hrs)		

B. TRAINING APPROACH 1

Two main approaches have been considered in this work. The first includes the use of MFCC spectrograms as input features, which is shown to achieve state-of-the-art results in applications such as music genre classification [38]. *Librosa* [39] library in the python programming language was used to compute the MFCC spectrograms. 10-20 MFCC coefficients are typically extracted in audio classification and speech recognition applications. For instance, [40] showed that the speaker identification rate increased as the number of coefficients increased from 13 to 20, but the identification rate started to decrease as the number of coefficients increased further. In our work, we experimented with different coefficient values between 10 and 20 and determined that 20 MFCCs were the most optimum based on a subset of the training data. Therefore, a total of 20 MFCCs were computed for each training example. Figure 11 illustrates the MFCCs computed for the entire 30 seconds using 1 training example from each of the 8 maqam classes. As shown in Table 1, the data for all classes were quite evenly spread out and hence methods such as oversampling a minority class were not required. We then used the *train_test_split* method from the *Scikit-learn* [41] library to split the dataset so that 80% of the examples were used for model training and 20% were used for evaluation. We then performed a 5-fold cross validation in the training phase, where the algorithms are repeatedly trained 5 times with a fraction of 1/5 training examples left out for testing [42]. We adopted this procedure to select the best parameters for our models. The selected models for experimentation were the 3 different state-of-the-art deep learning architectures, namely, CNN, LSTM, and CNN+LSTM (both CNN and LSTM together). After selecting the best parameters, we then trained our models on the entire training set and tested them on the test set. The details of each implementation are presented in the next section along with the results.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098415, IEEE Access

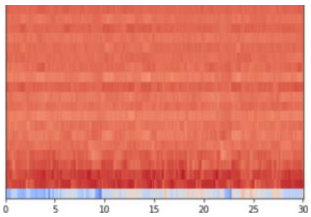

Figure 11 a. *Ajam* MFCC

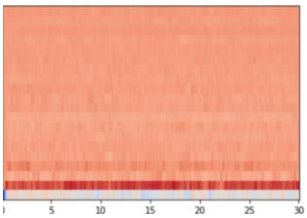

Figure 11 b. *Bayat* MFCC

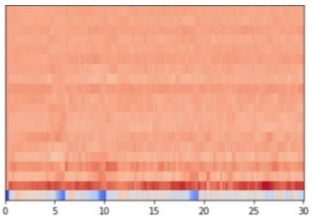

Figure 11 c. *Hijaz* MFCC

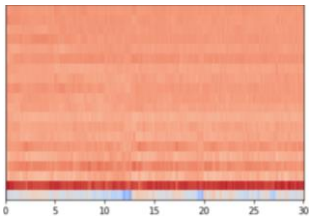

Figure 11 d. *Kurd* MFCC

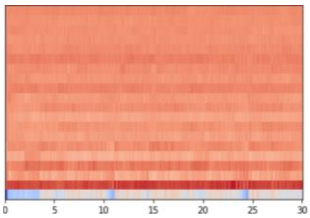

Figure 11 e. *Nahawand* MFCC

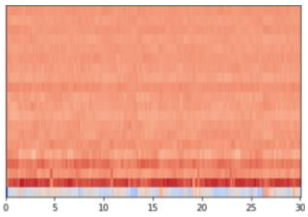

Figure 11 f. *Rast* MFCC

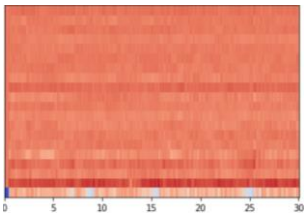

Figure 11 g. *Saba* MFCC

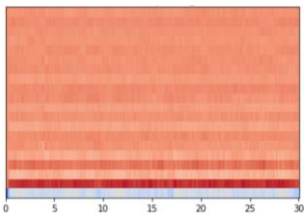

Figure 11 h. *Seka* MFCC

C. TRAINING APPROACH 2

Besides the MFCC features, we also experimented with other spectral and temporal features that were discussed in Section II-C. However, for all the spectral and temporal features, we computed their mean values. We expect that on average, samples from a given class will have similar values in these features and they will differ on these values with other classes. The utility of these features can be seen with the improvement in overall performance. Similarly, for all the 20 MFCC spectrograms, we averaged out the values for each MFCC. As a result, we obtained a total of 26 features for every training example, i.e., the 6 average values of each of the 6 spectral and temporal features in addition to the average of each of the 20 MFCCs. All these features were also extracted using the *Librosa* library. As a result, our final processed dataset containing the features and the labels in this approach is 2-dimensional instead of the 3-dimensional dataset, in the aforementioned approach. Therefore, the natural choice for a deep learning architecture, in this case, was a deep ANN. We used the same training and testing protocols as in the previous section. The architecture along with the results are presented next.

IV. ARCHITECTURES AND RESULTS

This section will present the deep learning model configurations for each approach and their corresponding results. Additionally, we also define the metrics used for determining the classification performance.

A. CLASSIFICATION METRICS

Given a confusion matrix, as shown in Figure 12, various metrics can be derived. If the model predicts a positive class and the actual value also belongs to a positive class, this case is a true positive (TP). True negative (TN) is when the model correctly predicts a negative class. When the model predicts a positive class when the actual value belongs to the negative class, this is known as false positive (FP). False negative (FN) is when the model incorrectly predicts a negative class when the actual value belongs to a positive class.

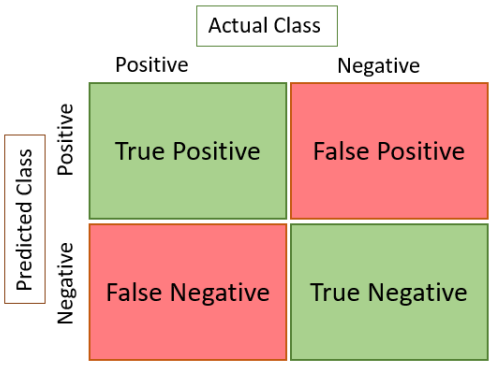

FIGURE 12. A confusion matrix

Although the above confusion matrix is an example for binary classification, we can derive generalized metrics for multi-class classification, by averaging across the classes. The most common metrics include classification accuracy (Equation 4), precision (Equation 5), recall (Equation 6), F1 score or F-measure (Equation 7):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{FN + TP} \tag{6}$$

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098415, IEEE Access

B. CNN WITH MFCC FEATURES

The CNN architecture consisted of 2 convolutional layers along with their corresponding pooling layers. Following the convolutional layers, the model was flattened and connected to 3 fully connected layers. Table 2 summarizes the configurations for each layer. Rectified linear units (ReLu) [43] activation was used in all layers except the output layer where softmax activation was used. Batch normalization [44] technique was also used for all the hidden layers.

The parameters were selected after careful experimentation. A learning rate of 0.0001 and batch size of 64 was used to train the model for a total of 35 epochs. For optimization, the popular Adam [45] algorithm was used. Figure 13 shows the training and validation loss curve. We used dropout [46] technique to prevent overfitting. One can very well see from Figure 13 that we are not having any overfitting problem.

TABLE II
CNN ARCHITECTURE PARAMETERS

Layer	Parameter	Value
Layer 1: Convolutional	Filters	32
	Kernel size	3x3
	Padding	same
	Pooling	Max pool 3x3
	Dropout	0.1
Layer 2: Convolutional	Filters	32
	Kernel size	3x3
	Padding	valid
	Pooling	Max pool 3x3
	Dropout	0.2
Layer 3: Fully Connected	Neurons	512
	Dropout	0.2
Layer 4: Fully Connected	Neurons	265
	Dropout	0.2
Layer 5: Fully Connected	Neurons	100
	Dropout	0.2

The average of the 5-fold validation accuracy was 74.6%. Next, we retrain the model with the chosen parameters, with 5-fold cross validation, on the entire training set and apply it to the test set, which the model has not seen before. The testing accuracy was 72.1%. The respective precision, recall and F1 scores were 0.80, 0.76, and 0.76, respectively. In Figure 14, the confusion matrix obtained from the test set is shown. Maqam Bayat examples were perfectly predicted all the times and maqam Rast examples were predicted right about 90% of the times. This contrasts with Nahawand and Kurd maqams that were only 41% and 59% respectively.

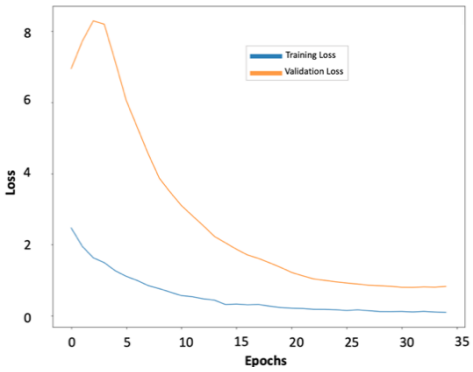

FIGURE 13. Training and validation loss curve for CNN architecture

FIGURE 14. Confusion matrix for the CNN architecture on the test set

C. LSTM WITH MFCC FEATURES

The LSTM-based architecture consisted of 3 LSTM layers followed by 3 fully connected layers. Table 3 summarizes the configurations for each layer, where the parameters were experimentally determined. The activation function in the LSTM layers was tanh meanwhile ReLu and softmax were used for the fully connected and output layers, respectively. Batch normalization was used for all the hidden layers.

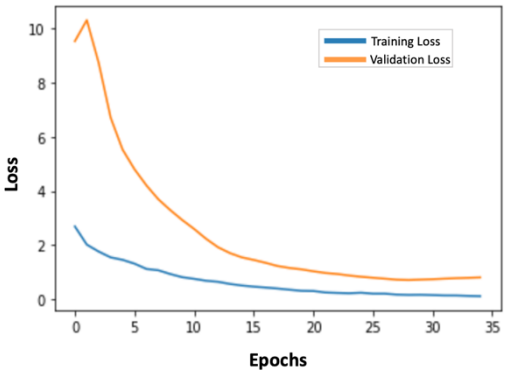

FIGURE 15. Training and validation loss curve for LSTM architecture

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098415, IEEE Access

TABLE III
LSTM ARCHITECTURE PARAMETERS

Layer	Parameter	Value
Layer 1: LSTM	Neurons	1024
	Dropout	0.3
Layer 2: LSTM	Neurons	512
	Dropout	0.3
Layer 3: LSTM	Neurons	256
	Dropout	0.2
Layer 4: Fully Connected	Neurons	512
	Dropout	0.2
Layer 5: Fully Connected	Neurons	265
	Dropout	0.2
Layer 6: Fully Connected	Neurons	100
	Dropout	0.2

The learning rate used was 0.0001 and a batch size of 64 was used to train the model for a total of 50 epochs. Adam optimizer was used. Figure 15 shows the training and validation loss curve. We used dropout and weight regularization. We do not notice any overfitting problem, however it seems the validation set is a bit harder to predict, given the variation in the dataset. We did not notice any appreciable performance gain after 50 epochs.

The model achieved an average accuracy of 77.5% with the 5-fold cross validation, which is an improvement on the previous architecture. After retraining on the entire training set, the model is then evaluated on the test set. It obtained an accuracy of 79.3%, with the corresponding precision, recall and F1 scores being 0.78, 0.79, and 0.78, respectively. In Figure 16, the confusion matrix obtained from the test set is shown. The most correct predictions were obtained from maqams Rast (94%) followed by Seka (93%) whereas the least accurate maqams were Ajam (57%) and Kurd (65%).

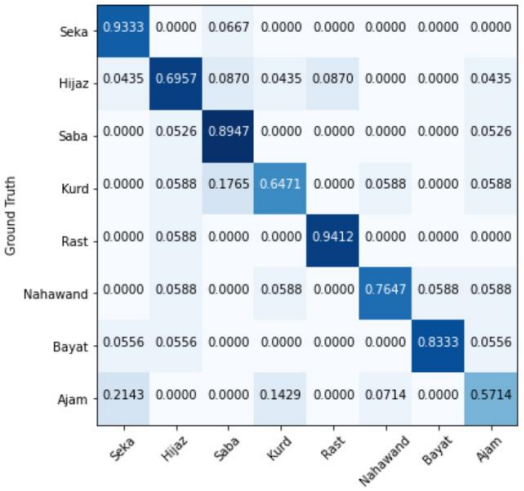

FIGURE 16. **Confusion matrix for the LSTM architecture on the test set**

D. CNN AND LSTM WITH MFCC FEATURES

This architecture consisted of 2 convolutional layers followed by an LSTM layer, which was followed by a fully connected layer. Table 4 summarizes the configurations for each layer. ReLu activation was used for the convolution layers and the fully connected layer whereas tanh activation was used for the LSTM layer. As usual, softmax activation was used for the output layer. A learning rate of 0.0001, Adam optimizer and a batch size of 64 was used to train the model for a total of 37 epochs. Figure 17 shows the training and validation loss curve. Again, we notice minimal overfitting using this architecture.

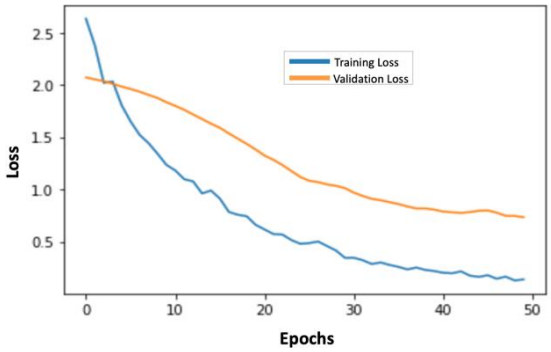

FIGURE 17. **Training and validation loss curve for CNN+LSTM model**

TABLE IV
CNN+LSTM ARCHITECTURE PARAMETERS

Layer	Parameter	Value
Layer 1: Convolutional	Filters	64
	Kernel size	3x3
	Padding	same
	Pooling	Max pool 3x3
	Dropout	0.1
Layer 2: Convolutional	Filters	32
	Kernel size	3x3
	Padding	same
	Pooling	Max pool 3x3
	Dropout	0.1
Layer 3: LSTM	Neurons	512
	Dropout	0.25
Layer 4: Fully Connected	Neurons	100
	Dropout	0.2

The average cross validation accuracy across the 5 folds was 75.8%. The testing accuracy was 76.4%. In addition, the corresponding precision, recall and F1-score was 0.86, 0.78, and 0.79, respectively. In Figure 18, the confusion matrix obtained from the test set is shown. Maqam Saba examples were predicted most successfully with correct prediction rate of 95%, whereas, maqam Nahawand was the least accurate with correct prediction rate of 50%.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098415, IEEE Access

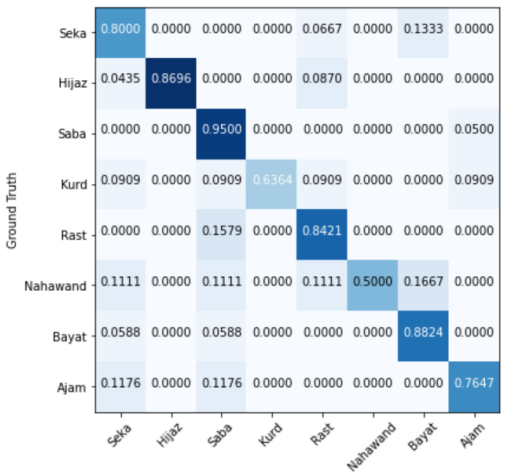

FIGURE 18. **Confusion matrix for the CNN+LSTM architecture on the test set**

E. DEEP ANN WITH SPECTRAL AND TEMPORAL FEATURES

As mentioned in Section III-C, in this approach we consider the average of 26 spectral and temporal features. A deep ANN with a total of 5 hidden layers has been trained, and Table 5 summarizes the configurations for each layer. ReLu activation was used in all layers except the output layer where softmax activation was used. Batch normalization was used for all the hidden layers.

TABLE V
DEEP ANN ARCHITECTURE PARAMETERS

Layer	Parameter	Value
Hidden Layer 1	Neurons	1024
	Dropout	0.2
Hidden Layer 2	Neurons	512
	Dropout	0.1
Hidden Layer 3	Neurons	256
	Dropout	0.1
Hidden Layer 4	Neurons	128
	Dropout	0
Hidden Layer 5	Neurons	64
	Dropout	0

The parameters shown in the table were experimentally determined with cross-validation. A learning rate of 0.0001 and batch size of 64 and Adam optimizer was used to train the model for a total of 45 epochs. Figure 19 shows the training and validation loss curve. As the training iterations increased, we notice the training and validation losses are very close to each other, suggesting gradual improvement in performance without overfitting.

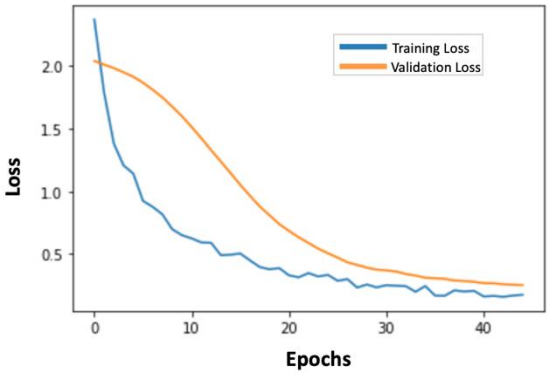

FIGURE 19. **Training and validation loss curve for deep ANN model**

The average 5-fold cross validation accuracy was 93%, a significant improvement from the aforementioned approaches. Next, we retrain the model on the entire training set and applied it to the test set. We obtained an accuracy of 95.7%, with the corresponding precision, recall and F1 scores all being 0.96. In Figure 20, the confusion matrix obtained from the test set is shown. Maqams Seka, Hijaz, Nahawand, and Bayat were predicted correctly for all examples (100%). Other maqams had the prediction rate of above 90%, except for Kurd which was 86%.

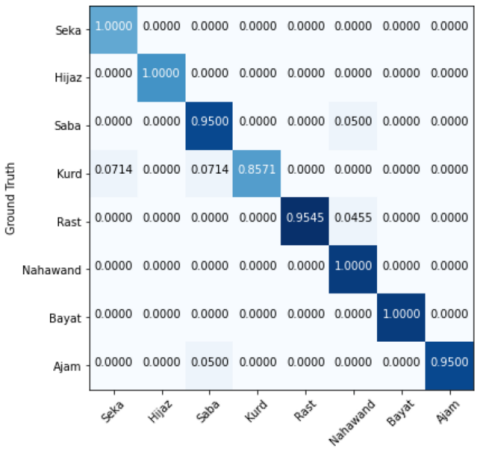

FIGURE 20. **Confusion matrix for the deep ANN architecture on the test set**

Table 6 summarizes the performance on the test set for all 4 deep learning architectures. The best performing model is the deep ANN, as highlighted.

TABLE VI
RESULTS SUMMARY OF ALL MODELS

Model	5-Fold CV Accuracy (%)	Test Accuracy (%)	Test F1-Score
CNN	74.6	72.1	0.76
LSTM	77.5	79.3	0.78
CNN+LSTM	75.8	76.4	0.79
Deep ANN	**93.0**	**95.7**	**0.96**

It is evident that the latter approach of using the additional spectral and temporal features resulted in a significant improvement in model performance. It is likely that since the size of training data is relatively small, MFCCs alone were not enough for a better performance. Perhaps better results can be obtained with a larger dataset. One can notice that Maqam Kurd was classified consistently with the least accuracy across all the models. This is most likely due to this class having the least number of training examples in the entire dataset.

V. CONCLUSION AND FUTURE WORK

This work presented the "Maqam-478" dataset, which to the best of our knowledge is the first of its kind for maqam classification of Qur'anic recitations. We then performed audio feature extractions, including MFCCs, energy, spectral and chroma features. The input features were then used to train 4 different deep learning architectures for classifying one of the 8 maqams. The best performing deep ANN with 5 hidden layers achieved an accuracy of 95.7% and a corresponding F1-score of 0.96 on the test set.

This work provided a platform for maqam classification using deep learning. To improve on generalization, the dataset should be expanded to include recitations from other reciters besides the selected two. Moreover, having a greater amount of training data generally leads to an increase in performance, particularly for deep learning-based models. Additionally, recitations with *mujawwad* style which is slower in speed but more elaborative in melody compared to the *murattal* style [47], which has been utilized in this work, needs to be looked at. It is also worth experimenting with other deep learning architectures, including bidirectional LSTMs [48] and convolutional LSTMs [49] to enhance performance. Finally, results from this study can be used to develop a maqam tutoring application.

REFERENCES

[1] C. Hackett, B. Grim, M. Stonawski, V. Skirbekk, M. Potančoková, and G. Abel, "The global religious landscape," *Washington, DC: Pew Research Center*, 2012.

[2] K. Nelson, *The art of reciting the Qur'an*. American Univ in Cairo Press, 2001.

[3] S. Ḥisāmpūr and A. Jabbāra, "A Study of the Qur'ānic Musical Modes (Maqāmāt) in the Recitation of Several Qārīs (Reciters)," *Quran and Hadith Studies*, vol. 42, no. 2, 2010.

[4] H. H. Touma and H. Touma, *The music of the Arabs*. Hal Leonard Corporation, 2003.

[5] J. Farraj and S. A. Shumays, *Inside Arabic Music: Arabic Maqam Performance and Theory in the 20th Century*. Oxford University Press, 2019.

[6] M. A. Shokouhi and A. Yusof, "The Influence of Islamic culture and holy Quran on performing arts: relating to sacred vocal music (lahn)," presented at the The 3rd Annual International Quranic Conference 2013, Centre of Quranic Research, 2013. [Online]. Available: http://eprints.um.edu.my/id/eprint/9577

[7] T. Al Bakri, M. Mallah, and N. Nuserat, "Al Adhan: documenting historical background, practice rules, and musicological features of the muslim call for prayer in Hashemite Kingdom of Jordan," 2019.

[8] Q. R. R. Zaidah and M. Imaduddin, "Listening to the Quran Recitations:'Does It Affect Psychophysiological Measures of Emotion?'," 2018.

[9] A. Alfaries, M. Albahlal, M. Almazrua, and A. Almazrua, "A Rule-Based Annotation System to Extract Tajweed Rules from Quran," in *2013 Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences*, 2013, pp. 281–286. doi: 10.1109/NOORIC.2013.63.

[10] T. Altalmas *et al.*, "Lips Tracking Identification Of A Correct Quranic Letters Pronunciation For Tajweed Teaching And Learning," *IIUMEJ*, vol. 18, no. 1, pp. 177–191, May 2017, doi: 10.31436/iiumej.v18i1.646.

[11] H. H. AlKhatib, E. I. Mansor, Z. Alsamel, and J. A. AlBarazi, "A Study of Using VR Game in Teaching Tajweed for Teenagers," in *Interactivity and the Future of the Human-Computer Interface*, IGI Global, 2020, pp. 244–260.

[12] N. J. Ibrahim, Z. M. Yusoff, and Z. Razak, "Improve design for automated Tajweed checking rules engine of Quranic verse recitation: a review," *QURANICA-International Journal of Quranic Research*, vol. 1, no. 1, pp. 39–50, 2011.

[13] K. M. Nahar, M. Ra'ed, A. Moy'awiah, and M. Malek, "An efficient holy quran recitation recognizer based on svm learning model," *Jordanian Journal of Computers and Information Technology (JJCIT)*, vol. 6, no. 04, 2020.

[14] B. Yousfi and A. M. Zeki, "Holy Qur'an speech recognition system distinguishing the type of recitation," in *2016 7th International Conference on Computer Science and Information Technology (CSIT)*, 2016, pp. 1–6. doi: 10.1109/CSIT.2016.7549454.

[15] B. Abro, A. B. Naqvi, and A. Hussain, "Qur'an recognition for the purpose of memorisation using Speech Recognition technique," in *2012 15th International Multitopic Conference (INMIC)*, 2012, pp. 30–34. doi: 10.1109/INMIC.2012.6511440.

[16] A. Elnagar and M. Lataifeh, "Predicting Quranic Audio Clips Reciters Using Classical Machine Learning Algorithms: A Comparative Study," in *Recent Advances in NLP: The Case of Arabic Language*, M. Abd Elaziz, M. A. A. Al-qaness, A. A. Ewees, and A. Dahou, Eds. Cham: Springer International Publishing, 2020, pp. 187–209. doi: 10.1007/978-3-030-34614-0_10.

[17] M. Lataifeh, A. Elnagar, I. Shahin, and A. B. Nassif, "Arabic audio clips: Identification and discrimination of authentic Cantillations from imitations," *Neurocomputing*, vol. 418, pp. 162–177, 2020, doi: https://doi.org/10.1016/j.neucom.2020.07.099.

[18] F. H. Jabar *et al.*, "Preliminary Results of Quranic Maqamat Profil-Ing Using W-DFT Technique", (2018) 25th International Congress on Sound and Vibration 2018, ICSV 2018: Hiroshima Calling, 8, pp. 4572-4579.

[19] J. Ramírez and M. J. Flores, "Machine learning for music genre: multifaceted review and experimentation with audioset," *Journal of Intelligent Information Systems*, vol. 55, no. 3, pp. 469–499, Dec. 2020, doi: 10.1007/s10844-019-00582-9.

[20] S. Oramas, F. Barbieri, O. Nieto, and X. Serra, "Multimodal deep learning for music genre classification," *Transactions of the International Society for Music Information Retrieval. 2018; 1 (1): 4-21.*, 2018.

[21] C. Senac, T. Pellegrini, F. Mouret, and J. Pinquier, "Music Feature Maps with Convolutional Neural Networks for Music Genre Classification," New York, NY, USA, 2017. doi: 10.1145/3095713.3095733.

[22] L. Su, "Vocal Melody Extraction Using Patch-Based CNN," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 371–375. doi: 10.1109/ICASSP.2018.8462420.

[23] D. Basaran, S. Essid, and G. Peeters, "MAIN MELODY EXTRACTION WITH SOURCE-FILTER NMF AND CRNN," Paris, France, Sep. 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02019103

[24] S. Kum, C. Oh, and J. Nam, "Melody Extraction on Vocal Segments Using Multi-Column Deep Neural Networks.," in *ISMIR*, 2016, pp. 819–825.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[26] S. Shahriar, A. R. Al-Ali, A. H. Osman, S. Dhou, and M. Nijim, "Machine Learning Approaches for EV Charging Behavior: A Review," *IEEE Access*, vol. 8, pp. 168980–168993, 2020, doi: 10.1109/ACCESS.2020.3023388.

[27] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," 2014.

[30] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition," *Speech Communication*, vol. 54, no. 4, pp. 543–565, 2012, doi: https://doi.org/10.1016/j.specom.2011.11.004.

[31] B. Logan and others, "Mel frequency cepstral coefficients for music modeling.," in *Ismir*, 2000, vol. 270, pp. 1–11.

[32] F. Gouyon, F. Pachet, O. Delerue, and others, "On the use of zero-crossing rate for an application of classification of percussive sounds," in *Proceedings of the COST G-6 conference on Digital Audio Effects (DAFX-00), Verona, Italy*, 2000, vol. 5.

[33] D. Ellis, "Chroma feature analysis and synthesis," *Resources of Laboratory for the Recognition and Organization of Speech and Audio-LabROSA*, 2007.

[34] A. Klapuri and M. Davy, *Signal processing methods for music transcription*. Springer Science & Business Media, 2007.

[35] J. M. Grey and J. W. Gordon, "Perceptual effects of spectral modifications on musical timbres," *The Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1493–1500, 1978.

[36] E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval of audio," *IEEE MultiMedia*, vol. 3, no. 3, pp. 27–36, 1996, doi: 10.1109/93.556537.

[37] S. Shahriar and U. Tariq, *Maqam478 Dataset: Qur'anic Recitations in 8 different Maqams*. figshare, 2021. doi: 10.6084/m9.figshare.13489359.

[38] A. Shakya, B. Gurung, M. S. Thapa, M. Rai, and B. Joshi, "Music classification based on genre and mood," in *International conference on computational intelligence, communications, and business analytics*, 2017, pp. 168–183.

[39] B. McFee *et al.*, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015, vol. 8, pp. 18–25.

[40] C. Kumar, F. ur Rehman, S. Kumar, A. Mehmood, and G. Shabir, "Analysis of MFCC and BFCC in a speaker identification system," in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2018, pp. 1–5. doi: 10.1109/ICOMET.2018.8346330.

[41] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[42] Y. Bengio and Y. Grandvalet, "No unbiased estimator of the variance of k-fold cross-validation," *Journal of machine learning research*, vol. 5, no. Sep, pp. 1089–1105, 2004.

[43] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," 2010.

[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[47] M. Frishkopf, "Mediated Qur'anic recitation and the contestation of Islam in contemporary Egypt," *Music and the play of power in the Middle East, North Africa, and central Asia*, pp. 75–114, 2009.

[48] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," in *International Conference on Artificial Neural Networks*, 2005, pp. 799–804.

[49] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, pp. 802–810, 2015.

SAKIB SHAHRIAR received his B.S. (2018) in Computer Engineering from American University of Sharjah, Sharjah, U.A.E in 2018. He is currently pursuing his M.S. degree in Computer Engineering at the American University of Sharjah, Sharjah, U.A.E. He is also a research assistant at the same university. His research interests include the applications of Machine Learning, Big Data Analytics and Deep Learning.

USMAN TARIQ (Member, IEEE) received the M.S. and Ph.D. degrees from the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign (UIUC), in 2009 and 2013, respectively. He worked as a Research Scientist with the Xerox Research Center, Computer Vision Group, Europe, France. He is currently a Faculty Member with the Department of Electrical Engineering, American University of Sharjah (AUS), United Arab Emirates. His research interests include computer vision, image processing, and machine learning, in general while facial expression recognition and face biometrics, in particular.