2024 27th International Conference on Computer and Information Technology (ICCIT)
20-22 December 2024, Cox's Bazar, Bangladesh

BDConvSGNet: A Lightweight Neural Network for Isolated Bengali Digit Sound Classification

Ahsanul Islam
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering & Technology
Dhaka, Bangladesh
ahsanulislamshuvo@gmail.com

Nafiz Imtiaz Rafin
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
nafiz.imtiaz@bracu.ac.bd

Md Rakibul Hasan
School of Electrical Eng, Computing and Math Sciences (EECMS)
Curtin University
Bentley, WA 6102, Australia
Rakibul.Hasan@curtin.edu.au

Md. Golam Rabiul Alam
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
rabiul.alam@bracu.ac.bd

2024 27th International Conference on Computer and Information Technology (ICCIT) | 979-8-3315-1909-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCIT64611.2024.11022527

***Abstract*—Speech recognition systems in low-resourced languages are gaining popularity, as command-based systems in such languages enable users to interact with technology without requiring extensive technical knowledge. This paper proposes a lightweight neural network architecture, *BDConvSGNet*, to classify Bengali isolated digit sounds. The proposed model combines a Convolutional Neural Network (CNN), a Bidirectional Gated Recurrent Unit (Bi-GRU), and a Squeeze-and-Excite (SE) mechanism. The model uses Mel-Frequency Cepstral Coefficients (MFCCs) as features from the isolated digit sounds to achieve efficient classification. With a total of only 97,408 parameters, BDConvSGNet is compact and consumes less memory, making it suitable for resource-constrained environments. The model was evaluated on three different datasets, consisting of 984, 2,252, and 13,554 samples, respectively. It achieved high classification accuracy with mean scores of 94.00%, 98.76%, and 99.20% in 10-fold cross-validation on these datasets. The results demonstrate that BDConvSGNet is a robust, efficient, and reliable solution for Bengali digit sound classification.**

***Index Terms*—Speech recognition, Bengali digit classification, CNN, Bi-GRU, Squeeze-and-Excite, MFCC.**

I. Introduction

In this modern era, smart devices and command-based automation gadgets have become significantly popular. Speech recognition-based functionality is one of the key features of those devices. This key feature makes these devices usable for a vast population, even with less technical knowledge. Many people like smart gadgets with speech recognition features because they save time and work more efficiently. The extensive popularity of Google Assistant, Amazon Alexa, and Microsoft Cortana shows the wide acceptance of speech-processing-based systems among common people. However, a major issue with these devices is that most of them can only process English speech, which makes it challenging for people who do not know the English language. The introduction of low-resourced language-based speech recognition systems would significantly enhance the usability and accessibility of those devices among large groups of people. The potential impact of a low-resourced language-based system has gained the attention of many researchers. As a result, many speech recognition-based studies have been done on several low-resourced languages, such as Arabic [1], Malaylam [2], Urdu [3], and many more. At present, Bengali is the fifth most spoken language based on the total number of speakers all around the whole world [4]. So, a Bengali language-based speech recognition system would greatly improve the lifestyles of this large community.

Isolated spoken-digit recognition systems have many potential applications, such as voice-based call forwarding, smart wheelchair control, and command interpretation for smart devices [5]. Deep learning approaches are becoming more popular day by day for implementing these systems due to their ability to learn complex patterns from data. Research has shown that Mel-Frequency Cepstrum Coefficients (MFCC) are effective acoustic features and are widely used in speech recognition systems [6], including those in low-resourced languages like Bengali [7]. The combination of MFCC and deep learning has shown great performance in speech recognition-based systems [1], [8]. Researchers used different variants of deep neural networks for digit recognition tasks, such as Convolutional Neural Networks (CNN) [5], Recurrent Neural Networks (RNN) [9], etc. However, implementing a deep learning-based system can be complex and memory-consuming as it requires a substantial number of parameters for training and prediction, which makes it less suitable for implementation on edge devices [10].

This paper has the following contributions:

1) Proposal of BDConvSGNet, a lightweight deep neural network for Bengali isolated spoken digit classification, integrating CNN, Squeeze and Excite (SE), and Bi-GRU.
2) Design with only 97,408 parameters, ensuring suitability for edge devices while achieving competitive accuracy.
3) Robust evaluation on three datasets confirming the model's effectiveness in classification tasks.

979-8-3315-1909-4/24/$31.00 ©2024 IEEE

II. Related Works

In recent times, many researchers have started to work on digit-sound speech recognition-based systems. In [11], researchers utilized an artificial neural network (ANN) to try to classify the sounds of English digits (0-9). They employed linear predictive coding (LPC) and principal component analysis (PCA) for feature extraction, achieving an overall accuracy of 82%. In the context of the Bengali language, the authors of [12] proposed a Gaussian Mixture Model (GMM) that utilizes MFCC features for classifying Bengali isolated digit sounds. Their proposed model achieved 91.7% accuracy on a dataset of 1000 samples of 10 classes containing (0-9) digits in Bengali. The researchers of [8] showed an improved approach applying 2D-CNN to the MFCC feature to perform the same task, which got 96.7% accuracy on a dataset comprising 4,000 samples. The study in [13] also performed 2D-CNN with MFCC features, and they got 98.37% accuracy with a dataset of 1,230 samples. Most recently, the authors of [5] combined features from Spectral Sub-band Energy (SSE), Mel-Frequency Cepstral Coefficients (MFCC), and Log Spectral Sub-band Energy (LSSE), which they used as input to a CNN model with approximately 182k parameters. Their process achieved 98.52% accuracy in classifying the Bengali digit sounds with a dataset of 14,000 samples.

Apart from CNN-based models, researchers are also using RNN-based models for automatic speech recognition-based tasks [9]. For instance, in [14], the authors proposed a variant of the RNN-based model called Long-Short Term Memory (LSTM) for classifying Indonesian digit sounds. They presented a comparison of two features, MFCC and LPC, for the classification task. It appears that MFCC achieves 96.58% accuracy, while LPC achieves 93.79% accuracy. In [15], the authors attempted to classify (0-99) Bengali digits with CNN, where they achieved 89.61% accuracy. Previous studies clearly depict the performance superiority of CNN models in the classification of digit sounds. While the performance of RNN-based models is also noteworthy. In most cases, the MFCC feature has been utilized because of its effective feature extraction capability.

III. Methodology

Before extracting features, all the sound data are resized to a uniform length so that all the data's resultant feature matrices are the same. In this study, all sound data has been resized to 60,000 samples to ensure getting sound data for more than a second, even with a sampling rate of 44 kHz. Data longer than this has been cropped, while shorter data has been zero-padded to match the required length. After this pre-processing step, all the data have been utilized for feature extraction and normalization. Finally, the normalized values of the extracted features are fed to the proposed deep-learning model.

A. Dataset Description

In order to evaluate our model, we utilized three datasets with varying sample sizes, referred to as Dataset-1, Dataset-2, and Dataset-3. Fig. 1 illustrates a count comparison among the three selected datasets. A detailed description of each dataset is provided below:

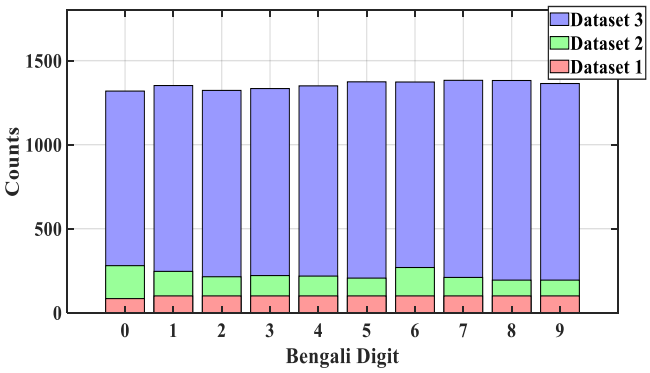


Fig. 1. Statistics of the three datasets of Bengali digits (0-9) used in this study.

1) Dataset-1: In [16], a dataset of 984 audio samples of spoken Bengali digits (0-9) is utilized. The dataset comprises 84 samples for the digit '0' and 100 samples each for the digits '1' through '9'. All audio recordings are in WAV format with a sampling rate of 44 kHz. For a more comprehensive description of this dataset, please refer to [15].

2) Dataset-2: This dataset had been collected in a classroom setting using a laptop and a generic headset microphone. Native Bengali-speaking undergraduate students recorded the digit sound from '0' to '9' in Bengali. The dataset includes recordings from 40 individuals (35 male, 5 female) with a sampling rate of 16 kHz and 24-bit depth. This dataset has a total of 2,252 utterances across 10 digits [17].

3) Dataset-3: This dataset consists of 13,554 audio signals of '0' to '9' Bengali digit sounds with a 16 kHz sampling rate. All the files are in WAV format and 32-bit floating-point representation. The dataset includes recordings from 28 speakers (18 male, 10 female) from various locations in West Bengal, India. The audio recording was taken using the "Easy Voice Recorder" android app [5].

B. MFCC Feature Extraction

In MFCC feature extraction, a discrete Fourier transform (DFT) is initially applied to windowed segments to convert the signal from the time domain into the frequency domain. Next, the logarithmic values of the magnitude spectrum are taken to simulate the human ear's perception of sound. Furthermore, the results are mapped onto the Mel scale to reflect the non-linear frequency response of the human auditory system. Finally, the Mel spectrum is transformed utilizing the discrete cosine transform (DCT) to produce MFCC [6]. Our study uses the 0.10.1 version of the Python librosa package to extract the MFCC feature. The length of the Fast Fourier Transform (FFT) and hop length are kept at 256 and 128, respectively. The number of utilized MFCCs is 25, following the recommendation of [18].

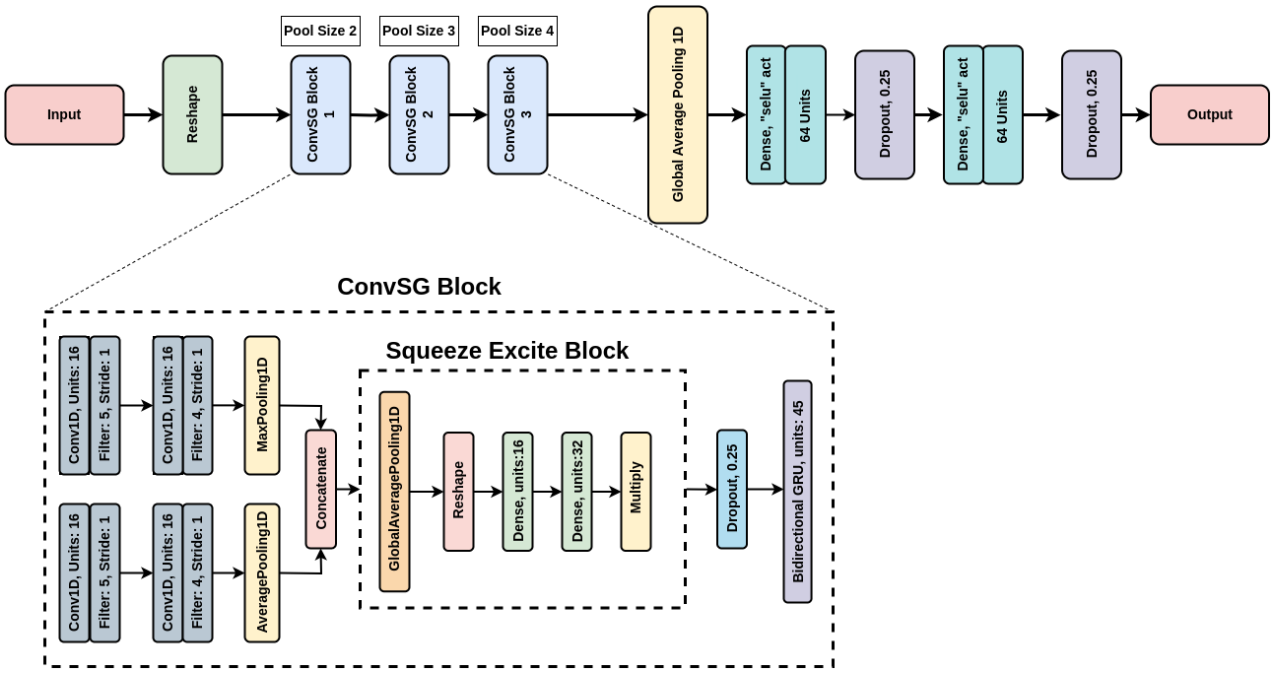


Fig. 2. Proposed deep neural network model, BDConvSGNet for classifying digits from MFCC features, featuring ConvSG blocks with 1D-CNN, max and average pooling, squeeze-and-excite, Bidirectional-GRU with global average pooling, and fully connected layers.

C. Normalization

Standard normalization, commonly known as Z-score normalization, is widely used in deep learning. The purpose of this is to scale input features such that they have a mean value of 0 with a standard deviation value of 1. The process improves the convergence rate, which results in improved performance. The formula for standard normalization is given in (1):

$$x_{\text{norm}} = \frac{x - \mu}{\sigma} \tag{1}$$

where x_{norm} is the normalized value, x is the original feature value, μ is the mean, and σ is the standard deviation of the feature.

D. Proposed Model Architecture

In this section, we present the architecture of the proposed model designed for time-series classification tasks, particularly focused on audio feature extraction using Mel Frequency Cepstral Coefficients (MFCC). The model integrates convolutional layers, pooling operations, a squeeze-and-excite mechanism, and recurrent layers to effectively capture both local and sequential patterns in the data. Fig. 2 shows the graphical representation of the proposed model. The overall model architecture contains 97,408 parameters.

1) Input Layer: The model takes MFCC features as input, where each sample consists of a sequence of time steps and corresponding feature values. The input shape is defined as (T, F), where T represents the number of time steps, and F is the number of extracted features per time step.

2) ConvSG Block: The model is comprised of blocks made up of 1D convolutional layers with squeeze-and-excite and bidirectional Gated Recurrent Unit (GRU), which gives it the name ConvSG block. The model utilizes three ConvSG blocks, each comprising two 1D convolution layers. These layers are designed to extract local patterns within the input sequences. Specifically, the first convolution layer applies filters of size 5, followed by another convolution layer with filters of size 4. Both layers use the SeLU (Scaled Exponential Linear Unit) activation function, which helps in maintaining the self-normalization property of the network.

Additionally, each ConvSG block has max-pooling and average-pooling layers, both with varying pool sizes. The outputs of these pooling layers are concatenated to capture diverse feature representations, enhancing the robustness of feature extraction.

3) Squeeze-and-Excite Mechanism: To further improve the quality of the learned feature maps, we integrate a squeeze-and-excite block [19] after the concatenation operation in each ConvSG block. This mechanism applies global average pooling, followed by two fully connected layers. The first layer reduces the dimensionality of the feature maps, while the second layer scales them through a Sigmoid activation function. This reweights the feature maps based on their importance, allowing the model to focus on more relevant features.

4) Bidirectional GRU Layer: Once the features are extracted, the output from each convolutional block is fed into a bidirectional GRU (Gated Recurrent Unit) layer consisting of 45 units. The bidirectional architecture allows the model to capture sequence dependencies from both forward and

TABLE I
DETAILS OF HYPERPARAMETERS.

Hyperparameter	Value
Learning Rate	0.001
Batch Size	128
Number of Epochs	200
Optimizer	AdamW
Activation Function	SeLU

backward directions, which is essential for time-series data, as understanding both past and future contexts enhances prediction accuracy.

5) Global Average Pooling: To reduce the complexity of the learned feature space before classification, a global average pooling layer is employed. This layer calculates the average value of each feature map across the time dimension, thereby summarizing the information extracted by the preceding layers.

6) Fully Connected Layers: After the global average pooling, the network employs two fully connected layers, each with 64 units, to further process the feature representation. Both layers use the SeLU activation function, and a dropout rate of 0.25 is applied to prevent overfitting during training.

E. Hyperparameters

The details of the hyperparameters are shown in Table I. The AdamW optimizer is used because it decouples weight decay from the optimization steps, resulting in improved regularization and better performance during training. Furthermore, the SeLU activation function is used because of its self-normalization property, which helps get faster convergence and improved training performance.

F. Performance Evaluation

The performance of the suggested approach is evaluated using several metrics and techniques. The approaches applied in assessing the performance of the model are fully described in this part.

1) Train-Test Split: Three subsets separate the dataset: test, validation, and training sets. Specifically, 70% of the data is set aside for model training so that it may learn from a sizable fraction of all the data. Designed for hyperparameter adjustment and model selection, a separate 15% serves as the validation set. Respected as the test set to assess the performance of the final model and guarantee it generalizes successfully to unprocessed data is the remaining 15% of the data. This method guarantees correct assessment of fresh, unseen data, efficient training, and suitable tuning of the model.

2) K-Fold Cross-Validation: To further evaluate the model's performance and resilience, 10-fold cross-validation is performed. In this procedure, the dataset is randomly partitioned into 10 equal-sized folds. The model is trained on 9 folds while the remaining fold acts as the validation set. This method is done 10 times, with each fold having a turn as the validation set. The final performance measures are averaged throughout these 10 iterations to offer a full evaluation of the model's performance. K-fold cross-validation assists in minimizing the unpredictability associated with random splits, delivering a more trustworthy assessment of the model's performance.

3) Evaluation Metrics: The performance of the model is assessed using the following metrics:

- **Accuracy:** This metric represents the fraction of instances that are correctly classified out of the total number of instances. It is mathematically expressed as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{2}$$

- **AUC-ROC:** The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) measures the model's capability to separate different classes. A higher AUC score indicates the model is more proficient at distinguishing between positive and negative instances.
- **F1-Score:** Precision refers to the ratio of true positives to the total predicted positives, while recall indicates the proportion of true positives relative to all actual positives. The F1-score serves as the harmonic mean of precision and recall, encapsulating both in a single metric. It is given by:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

- **Kappa Score:** Cohen's Kappa evaluates the agreement between the predicted classifications and the actual observations, adjusting for agreement occurring by chance. It is calculated as:

$$\kappa = \frac{P(O) - P(E)}{1 - P(E)} \tag{4}$$

where $P(O)$ is the observed agreement and $P(E)$ is the chance-expected agreement.

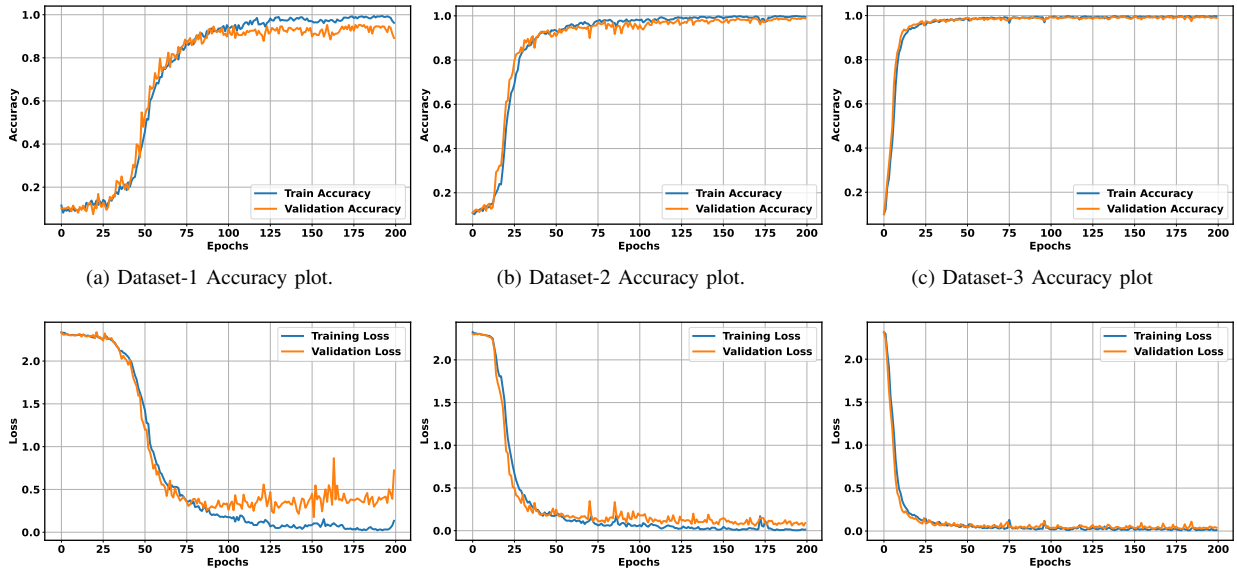
These evaluation metrics comprehensively assess the model's performance, ensuring it performs reliably across various dimensions and generalizes well to unseen data.

IV. RESULTS AND DISCUSSION

A. Evaluation of Train-Test Data Split

First, each dataset is split into 70% for training, 15% for validation, and 15% for testing. The proposed model is compared with traditional machine learning models, including K-Nearest Neighbor (KNN), Gaussian Naive Bayes (GNB), Decision Tree, Random Forest, and XGBoost. Flattening is applied because traditional ML models cannot handle 2D matrices. In the deep neural network model, 15% of the data is used for validation to monitor overfitting, with training on 70% and testing on 15%. To ensure a fair comparison, traditional ML models have been trained on 70% and tested on 15%, the same as the proposed deep neural network.

Fig. 3 illustrates the epoch-wise training progress of accuracy and loss for each dataset. Dataset-1, with 984 samples, shows the largest performance gap between training and



(a) Dataset-1 Accuracy plot. (b) Dataset-2 Accuracy plot. (c) Dataset-3 Accuracy plot

(d) Dataset-1 Loss plot. (e) Dataset-2 Loss plot. (f) Dataset-3 Loss plot.

Fig. 3. Training and Validation Accuracy (top row) and Loss (bottom row) for Dataset-1, Dataset-2, and Dataset-3.

TABLE II
TEST RESULTS FOR DATASET-1

Model	Accuracy	AUC-ROC	F1-Score	Cohen's κ
KNN	35.81%	74.76%	34.79%	28.67%
GNB	14.19%	53.59%	5.56%	4.84%
Decision Tree	32.43%	62.55%	33.67%	24.87%
Random Forest	54.05%	87.50%	54.11%	48.93%
XGBoost	49.32%	84.97%	48.81%	43.68%
Proposed Model	**93.24%**	**99.72%**	**93.32%**	**92.49%**

TABLE III
TEST RESULTS FOR DATASET-2

Model	Accuracy	AUC-ROC	F1-Score	Cohen's κ
KNN	66.27%	91.93%	65.33%	62.41%
GNB	74.56%	94.00%	74.27%	71.68%
Decision Tree	67.46%	81.54%	67.21%	63.80%
Random Forest	90.24%	98.89%	90.34%	89.13%
XGBoost	89.05%	99.09%	89.13%	87.82%
Proposed Model	**100.00%**	**100.00%**	**100.00%**	**100.00%**

TABLE IV
TEST RESULTS FOR DATASET-3

Model	Accuracy	AUC-ROC	F1-Score	Cohen's κ
KNN	88.64%	98.08%	88.59%	87.38%
GNB	43.12%	82.45%	42.38%	36.72%
Decision Tree	74.24%	85.69%	74.28%	71.37%
Random Forest	94.10%	99.53%	94.04%	93.44%
XGBoost	94.35%	99.77%	94.32%	93.72%
Proposed Model	**98.87%**	**99.99%**	**98.87%**	**98.74%**

TABLE V
10-FOLD CROSS-VALIDATION RESULTS (MEAN $\pm$ STANDARD DEVIATION) FOR DATASETS

Metric	Dataset-1	Dataset-2	Dataset-3
Accuracy	$(94.00 \pm 2.20)\%$	$(98.76 \pm 0.84)\%$	$(99.20 \pm 0.17)\%$
AUC-ROC	$(99.51 \pm 0.39)\%$	$(99.96 \pm 0.04)\%$	$(99.98 \pm 0.03)\%$
F1-Score	$(93.99 \pm 2.25)\%$	$(98.76 \pm 0.84)\%$	$(99.20 \pm 0.17)\%$
Cohen's κ	$(93.33 \pm 2.45)\%$	$(98.61 \pm 0.93)\%$	$(99.11 \pm 0.19)\%$

validation, attributed to its smaller size. This gap decreases with larger datasets, highlighting the model's improved generalization with more samples.

Tables II, III, and IV compare the proposed BDConvSGNet model with traditional ML models for each dataset. The results show that the proposed model significantly outperforms traditional models, effectively capturing complex patterns in MFCC features from isolated Bengali digit sounds and providing more robust and reliable performance.

B. Performance of 10-Fold Cross-Validation

Tables V, presents the 10-fold cross-validation performance of our model highlighting it's superiority. The model achieves high accuracy rates with means of 94.00%, 98.76%, and 99.20% for Dataset-1, Dataset-2, and Dataset-3, respectively, accompanied by standard deviations of 2.20%, 0.84%, and 0.17%. Its AUC-ROC scores are notably high, with means of 99.51%, 99.96%, and 99.98% and standard deviations of 0.39%, 0.04%, and 0.03%. The F1-Scores further reflect the model's effectiveness, reaching means of 93.99%, 98.76%, and 99.20% with standard deviations of 2.25%, 0.84%, and 0.16%. Additionally, Cohen's κ values of 93.33%, 98.61%, and 99.11% with standard deviations of 0.84%, 0.93%, and 2.45% demonstrate strong agreement, underscoring the model's robustness and reliability in isolated Bengali digit

TABLE VI
COMPARISON OF PREVIOUS STUDIES ON ISOLATED BENGALI DIGIT SOUND CLASSIFICATION WITH THE PROPOSED MODEL

Reference	Dataset Size	Method	# of Params	Accuracy
[13]	1,230	CNN + MFCC	-	98.37%
[8]	4,000	CNN + MFCC	-	96.70%
[17]	2,252	CNN + Spectrogram	726,474	98.23%
[5]	14,000	CNN + MFCC, SSE, LSSE	182,458	98.52%
Ours	**1. 984**	**BDConvSGNet + MFCC**	**97,408**	**94.00%**
	2. 2,252			**98.76%**
	3. 13,554			**99.20%**

sound classification.

C. Performance Comparison with Previous Studies

Table VI compares the performance of various studies on isolated Bengali digit sound classification with the proposed BDConvSGNet model. Previous studies used datasets ranging from 1,230 to 14,000 samples and applied methods combining CNNs with MFCCs, Spectral Sub-band Energy (SSE), Log Spectral Sub-band Energy (LSSE), and spectrograms. Their reported accuracy varies from 96.70% to 98.52%. In contrast, the proposed model consistently achieves higher accuracy across different datasets, with values of 94.00% for 984 samples, 98.76% for 2,252 samples, and 99.20% for 13,554 samples. It is noteworthy to highlight that the proposed model gives competitive performance despite having only 97,408 parameters lower than all the studies mentioned in the table. There are studies showing models with parameter numbers around $100k$ can produce good results on edge devices [20]. This demonstrates the effectiveness and superiority of the BDConvSGNet model.

V. CONCLUSION

This paper proposes a lightweight deep neural network for classifying Bengali isolated digit sounds. The neural network utilizes CNN, squeeze-and-excite mechanisms, and bidirectional GRU to effectively classify Bengali digit sounds from the MFCC feature. The model comprises 97,408 parameters, which is less than 100k, making it light and less memory-consuming than most of the related research. The model's performance is evaluated on three datasets of different sizes. The proposed model performed greatly on the datasets and showed competitive results on all three datasets, making the model reliable to apply to unseen datasets. The less memory consumption of the model makes it easier to implement on edge devices. The next step of this study would be to implement it on an edge device and evaluate the performance on the edge device with new users to check the robustness of the model in the real world. The successful prototype of an edge device with a Bengali-digit sound classifier will have great application on different command-based systems.

REFERENCES

[1] E. S. Wahyuni, "Arabic speech recognition using mfcc feature extraction and ann classification," in *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 22–25, IEEE, 2017.

[2] F. K. Mohamed and V. Lajish, "Nonlinear speech analysis and modeling for malayalam vowel recognition," *Procedia Computer Science*, vol. 93, pp. 676–682, 2016.

[3] A. Khalid, M. D. A. Awan, N. I. Kajla, A. Firdous, H. M. S. Badar, and M. M. S. Missen, "Audio-to-text urdu chatbot using deep learning algorithms rnn and wav2vec2," *Journal of Computing & Biomedical Informatics*, 2024.

[4] Wikipedia contributors, "Bengali language — Wikipedia, the free encyclopedia," 2024. [Online; accessed 9-September-2024].

[5] B. Paul and S. Phadikar, "A hybrid feature-extracted deep cnn with reduced parameters substitutes an end-to-end cnn for the recognition of spoken bengali digits," *Multimedia Tools and Applications*, vol. 83, no. 1, pp. 1669–1692, 2024.

[6] Z. K. Abdul and A. K. Al-Talabani, "Mel frequency cepstral coefficient and its applications: A review," *IEEE Access*, vol. 10, pp. 122136–122158, 2022.

[7] M. R. Hasan and M. M. Hasan, "Investigation of the effect of MFCC variation on the convolutional neural network-based speech classification," in *2020 IEEE Region 10 Symposium (TENSYMP)*, pp. 1408–1411, IEEE, 2020.

[8] O. Sen, P. Roy, *et al.*, "A convolutional neural network based approach to recognize bangla spoken digits from speech signal," in *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, pp. 1–4, IEEE, 2021.

[9] J. Oruh, S. Viriri, and A. Adegun, "Long short-term memory recurrent neural network for automatic speech recognition," *IEEE Access*, vol. 10, pp. 30069–30079, 2022.

[10] Z. Q. Lin, A. G. Chung, and A. Wong, "Edgespeechnets: Highly efficient deep neural networks for speech recognition on the edge," *arXiv preprint arXiv:1810.08559*, 2018.

[11] P. Sarma, S. Sarmah, M. Bhuyan, K. Hore, and P. Das, "Automatic spoken digit recognition using artificial neural network," *Int. J. Sci. Technol. Res*, vol. 8, no. 12, pp. 1400–1404, 2019.

[12] B. Paul, S. Bera, R. Paul, and S. Phadikar, "Bengali spoken numerals recognition by mfcc and gmm technique," in *Advances in Electronics, Communication and Computing: Select Proceedings of ETAEERE 2020*, pp. 85–96, Springer, 2021.

[13] R. Sharmin, S. K. Rahut, and M. R. Huq, "Bengali spoken digit classification: A deep learning approach using convolutional neural network," *Procedia Computer Science*, vol. 171, pp. 1381–1388, 2020.

[14] E. R. Swedia, A. B. Mutiara, M. Subali, *et al.*, "Deep learning long-short term memory (lstm) for indonesian speech digit recognition using lpc and mfcc feature," in *2018 Third International Conference on Informatics and Computing (ICIC)*, pp. 1–5, IEEE, 2018.

[15] O. Sen and P. Roy, "A novel bangla spoken numerals recognition system using convolutional neural network," in *International Conference on Machine Intelligence and Emerging Technologies*, pp. 344–357, Springer, 2022.

[16] P. Roy, "Bangla spoken 0-99 numbers," 2021. Accessed: 2024-09-16.

[17] M. S. Mohammad, A. Zahid, and M. A. Iqbal, "Banglanum–a public dataset for bengali digit recognition from speech," *arXiv preprint arXiv:2403.13465*, 2024.

[18] M. R. Hasan, M. M. Hasan, and M. Z. Hossain, "How many Mel-frequency cepstral coefficients to be utilized in speech recognition? a study with the Bengali language," *The Journal of Engineering*, vol. 2021, no. 12, pp. 817–827, 2021.

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[20] Z. Q. Lin, A. G. Chung, and A. Wong, "Edgespeechnets: Highly efficient deep neural networks for speech recognition on the edge," *arXiv preprint arXiv:1810.08559*, 2018.