Computer Engineering and Applications Vol. 14, No. 1, February 2025

Deep Neural Networks for Intelligent Voice Authentication Systems in Large-Scale Electronic Voting

Olayemi Mikail Olaniyi[1*], Bello Kontagora Nuhu[2], Oluwasogo Adekunle Okunade[1], Uchenna Christiana Ezeanya[3], Chimdiebube Emmanuel Eke[5]

[1]*Department of Computer Science, National Open University of Nigeria, Abuja, Nigeria*
[2]*Department of Computer Engineering, Federal University of Technology, Minna, Nigeria.*
[3]*Department of Information Systems and Technology, National Open University of Nigeria, Abuja, Nigeria*
**omolaniyi@noun.edu.ng*

ABSTRACT

The authentication of eligible voters is an area of concern that needs further exploration of the prospects of electronic voting systems. The integration of voice authentication in electronic voting systems for varying numbers of disabled and prospective voters should be secure, scalable and suitable in both federal and state elections. Machine learning (ML) is an evolving field of computing that presents prospects in electronic voting. Applying ML algorithms to electronic voting provides optimal solutions to a wide range of biometric authentication challenges. This paper presents the design of an effective voice classification algorithm from a narrower perspective that can be used in developing prototype electronic voting systems in large-scale voting scenarios, particularly for disabled voters. Applying the knowledge of deep neural networks, a three hidden layer network using a feed-forward architecture is designed for classifying voice data acquired from prospective voters. The proposed design is tested on two different datasets and is adapted to handle small and vast amounts of voters' voice information. Results indicated average training and average validation accuracies of 92% and 97% respectively for both deep learning models for inclusivity and accountability of disabled voters in secure electronic voting systems.

Keywords: Authentication; Biometrics; E-Voting; PWD; DNN; MLP; Machine Learning; Voice Authentication**.**

1. INTRODUCTION

Providing an effective and reliable means for authenticating people living with disabilities (PWDs) during democratic elections has been an area with less research focus when compared to research efforts placed in primary voter's authentication ballot confidentiality [1], and votes integrity [2]. A myriad of factors could contribute to research laxity in this field. However, most of these factors could be attributed to socio-technical policies designed to exclude persons with disabilities. However, statistics indicate that about 25 million Nigerians represent part of the PWD community [3][4]. This subset of the populace is too large a number to be ignored. Finding optimized solutions to this authentication problem would ensure that the right of all citizens to vote is satisfied.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

Technology advancements have brought about Machine Learning (ML), whose methods are a subset of the field of Artificial Intelligence (AI) that improves the performance of systems based on given data [5]. The prospects of AI are enormous and require insatiable amounts of data to learn from. ML algorithms harness data during the learning process for solving various problems such as classification and regression. In solving a classification problem, ML algorithms predict to a degree of correctness which class an observation belongs to certain decision boundaries [6].

Machine learning techniques have been applied to solve varieties of electronic voting problems. For instance, a two-factor biometric authentication system [7] used an eigenfaces-based algorithm that detected each voter's face before facial recognition [8]. Haar cascade classifier was used for face detection in e-voting systems that implemented face recognition for voter authentication. In [9] and [10] machine learning models were respectively proposed to detect and mitigate DoS and DDoS attacks on data centers where voting records were stored. in a similar vein, [11] proposed a deep learning algorithm to analyze voting patterns and behaviors. More also, to ensure a robust authentication, biometric based methods were considered more appropriate in e-voting [12]. In [13] the combination of facial recognition algorithm and Radio Frequency Identification (RFID) was used to authenticate and authorize voters alongside Blockchain technology to protect the integrity of the votes and ensures the verifiability of the cast votes. 14] and [15] further demonstrated the efficiency of Blockchain technology in ensuring security of e-voting platforms.

Deep learning is a subset of machine learning that has emerged as a tool in solving classification problems such as text recognition and speaker verification systems [11]. This is attributed to the ability of deep learning models to discriminate between features in a dataset in several layers of a neural net [16]. Neural networks are non-linear data-driven methods that offer flexibility based on provided data without the underlying model explicitly specifying its functional or distributional form for the underlying model [17]. The need for large amounts of data for a deep learning model to quickly learn and easily discriminate between test features presents a new problem. In designing a speaker authentication system implemented in an electronic voting system, deep learning models must be flexible to classify varying amounts of voters' voice data. Researches by [16][18] and [19] showed the feasibility of designing classification deep learning models that can optimize learning with small amounts of data respectively.

This research aims to design a supervised learning model using deep neural networks that can correctly identify whom the voter is using voice data for many voters (disabled persons inclusive) as required in large-scale electronic voting systems. All correctly identified voters are authenticated, and then the prototype can be implemented by e-voting systems in voting scenarios. The remaining sections of this paper are organized as follows: Section 2 presents the methods used in the design, Section 3, the performance of the prototype design's performance and results are discussed, and Section 4 concludes the paper.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

2. METHODS

2.1 SYSTEM ARCHITECTURE

The system architecture used by [4] was adapted with changes made to only the voice authentication unit as suggested in [20]. Figure 1 shows an overview of the intelligent voice authentication electronic voting system. It further describes how voters' voiceprints are collated in FLAC format and features extracted using the MFCC algorithm. The database comprises trained speaker samples to be authenticated during the voting phase. The DNN serves as the classifier and determines whether voters are whom they claim to be. The flowchart of the entire system is shown in Figure 2, further describing how the authentication process depends on the DNN classifier.

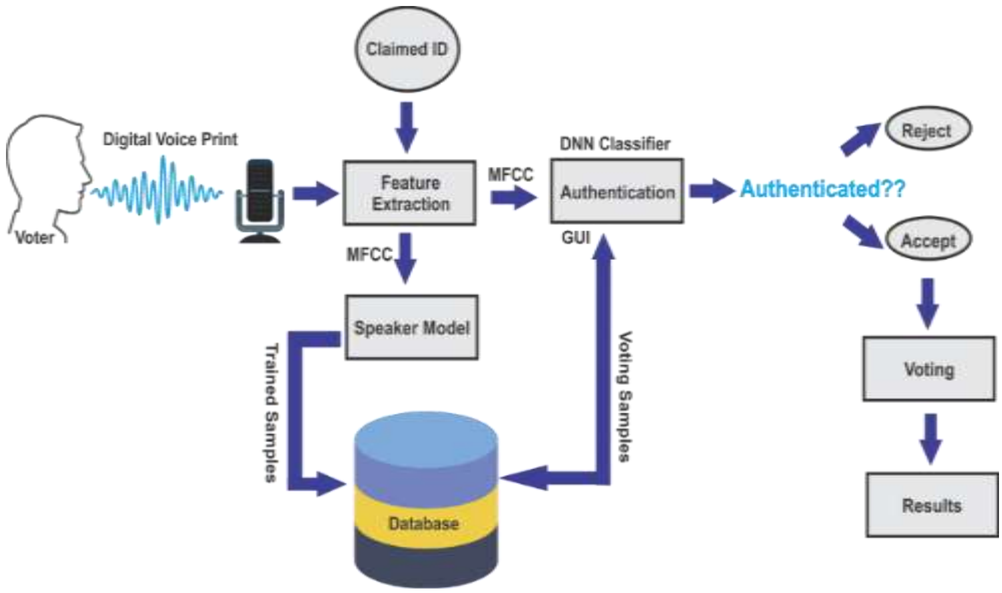

FIGURE 1. Overview of intelligent voice authentication system

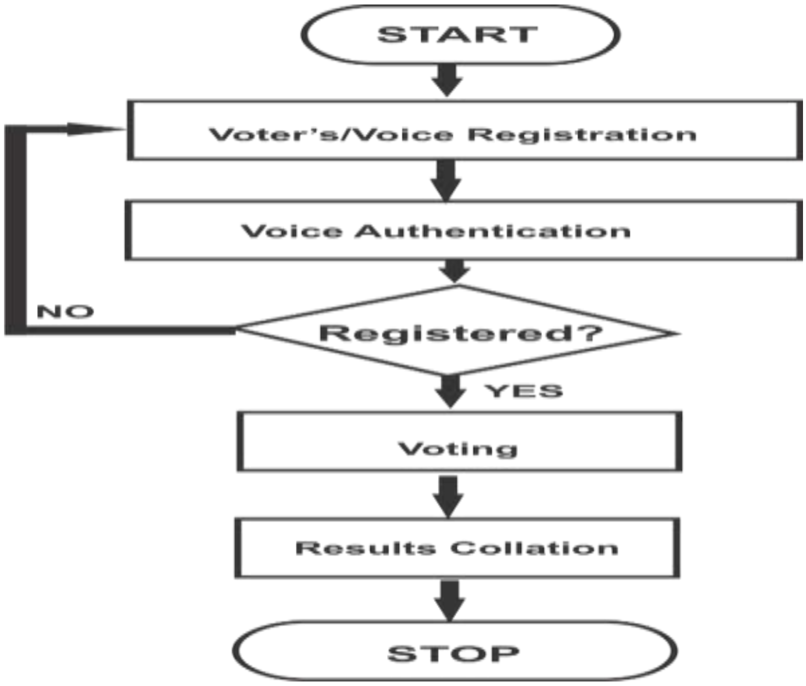

FIGURE 2. The system flowchart

2.2 CLASSIFICATION METHOD

A deep neural network (DNN) consists of multiple hidden layers between the input and output layers. Deep neural networks provide more complex functions in the number of layers and units within a single layer. A feed-forward architecture is implemented for this study, as shown in Figure 3. This design consideration applies to real-life election scenarios as many voters' voice samples can be collated and processed on higher computational devices.

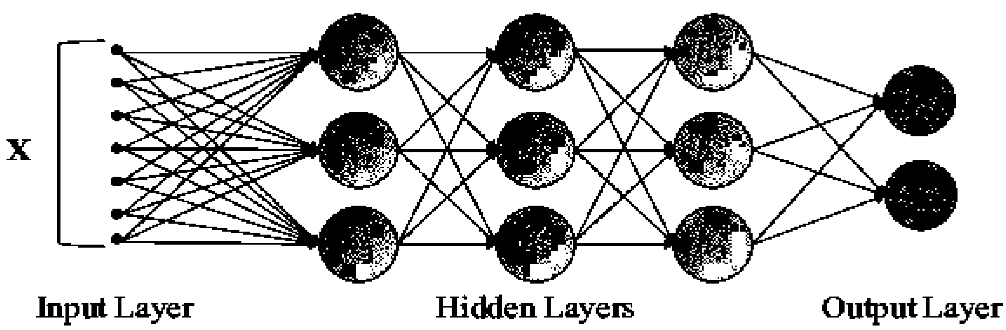

FIGURE 3. Feed Forward Neural Architecture with Three Hidden Layers [21]

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

2.2.1 ALGORITHM FOR THE FEED FORWARD NEURAL ARCHITECTURE WITH THREE HIDDEN LAYERS

Step 1: Begin
- Start the algorithm

Step 2: Initialize weights and biases
- Initialize all weights Wa, Wb, ..., Wn with a random number from a normal distribution.
- Set all bias nodes Ba, Bb, ..., Bn to 1.

Step 3: Input values
- Set the values of all input nodes.

Step 4: Forward propagation to hidden layers
- Calculate the hidden node values for each hidden layer by multiplying input nodes by their respective weight connections.
- Apply the ReLU activation function for each hidden layer: f(x) = max (0, x)

Step 5: Calculate hidden node activation
- Compute the activation values of hidden nodes after applying the ReLU function.

Step 6: Forward propagation to output layer
- Calculate the output node values.

Step 7: Apply SoftMax activation for output layer
- Use the SoftMax function for each output node: SoftMax(αj) = e^(αj) / ∑k e^(αk)

Step 8: Calculate output node activation
- Compute the activation values of the output nodes.

Step 9: Error calculation
- Calculate the total error: If OAi is the obtained output value for node i, and yi is the desired output, the total error is:

$$e = \frac{1}{n}\sum_{i-1}^{2}(y_i - OA_i)^2 \quad (1)$$

Step 10: Termination
- Stop the algorithm.

2.3 SYSTEM'S MATHEMATICAL MODEL

The mathematical system model depicts each voter's voice MFCC features extracted and fed into a 3-hidden layer feed-forward neural net. A total of 40 MFCC features are extracted for each voter and are represented in the form:

$$C_n = \sum_{m=1}^{M} [\log (Y(m) \cos[\frac{\mu n}{M}(m - \frac{1}{2}]] \tag{2}$$

Where n = index of the cepstral coefficient

The DNN classifier mimics the architecture of the forward feed network with three hidden layers asides from the input and output layer. The input and the three hidden layers use the Rectified Linear Unit (ReLU) activation function, while the output layer utilizes the softmax function. The MLP architecture is represented as MLP-m-n-o. Where m is the total number of inputs in the input layer, n is the total number of hidden layers, and o is the total number of outputs in the output layer. MLP-m-n-o can then be given as MLP-40-3-30, as 30 represents the number of speaker classes.

Taking $f_1, f_2, \ldots, f_h, \ldots, f_{n+1}$ as vectors of real value smooth functions, c_h as the number of neurons in the layer h and c_0, c_{n+1} are respectively the number of input and output neurons. Note that, $c_0 = 40$, and C_{n+1} =number of labels representing each speaker class (30).

If F, is defined from $x \subseteq R^p$ into $y \subseteq R^q$ corresponding to a MLP-NN, such that $p = c_0 \geq 2$, $q = c_{n+1} \geq 1$, then it can be expressed thus:

$$\mathbf{F(x) = f_{n+1}(w^{(n+1)}f_n(w^{(n)}f_{n-1}(w^{(n-1)}\ldots f_h(w^{(h)}\ldots\ldots f_2(w^{(2)}f_1(w^{(1)}x + b^{(1)}) + b^{(2)})\ldots + b^{(h)})\ldots. + b^{(n-1)}) + b^{(n)}) + b^{(n+1)}} \tag{3}$$

$\mathbf{F(x) = (F_1, \Theta_1(x), \ldots, F_{c_{n+1}}, \Theta_{c_{n+1}}(x))}$ is a vector of C_{n+1} outputs and the number of the MLP neural network parameters $\mathbf{\lessgtr = w^{(h)}, b^{(h)}, 1 \leq h \leq n+1}$ is given by:

$$n_\theta = \sum_{\substack{1<i<n \\ 2<j<n+1 \\ i \leq j}} C_i C_j + \sum_{j=1}^{n+1} C_j \tag{4}$$

The mathematical model of this system is classified into three steps.

Step 1

To derive the mathematical expression of a basic neuron and analyse its behaviour in any of the layers asides from the input layer. A neuron i of the first hidden layer realizes the random vector X with z components ($z \subseteq N$, $i \leq j \leq z$); w_{ij} are the synaptic weights and define the relationship between a neuron i and neurons in the preceding layer. The activation of the neuron i, $\alpha_i \subseteq R$ is obtained by the inner product of the vector $x \subseteq R^{z+1}$ and the vector of weights $w_i = (w_{i0}, \ldots, w_{ij}, \ldots, w_{ip}) \subseteq R^{z+1}$:

$$\alpha_i = \sum_{j=0}^{p} w_{ij} x_j = \sum_{j=0}^{p} w_{ij} x_j + b_{i0} \tag{5}$$

with $b_{i0} = w_{io}$ called bias and $x_0 \equiv 1$. The activation function, f is a real value function, continuous, not necessarily non-linear and class C^∞ defined from R to R and applied to the above equation gives:

$$y_i = f(\alpha_i) = f(w_i, X) \quad (6)$$

$f(\alpha_i)$ is called the state of the neuron i, which is the output of the neuron i can be used as input for the other neurons of the next layer or the label y.

Step 2

To derive the state of the neuron i for all other hidden layers asides, the first hidden layer is written as:

$$f_h(\alpha_i^{(h)}) = \sum_{j=0}^{Ch} w_{ij}^{(h)} f_{h-1}(\alpha_i^{(h-1)}) \quad (7)$$

Where f_h and f_{h-1} are the ReLU activation functions obeying the rule; c_h is the number of neurons of that hidden layer h; h-1 is the number of the layer that precedes h; $\alpha_i^{(h)}$ is the activation of the neuron i.

From the previous equations, the expression for a layer of c_h neurons is given as:

and where f_h is the activation function of the h^{th} layer of the MLP.

Step 3

To derive the softmax function of the output layer. The basic form of a softmax function for a transform on n numbers $x_1 \ldots x_n$ is given as:

$$\alpha(x)i \frac{e^{xi}}{\sum_{j=1}^{n} e^{xj}} \quad (8)$$

From equation 2, for the MLP-40-3-30, p = $c_0(\geq 2)$, n = 0 and q =c_4, then F(x) = $f_1(w^{(1)}.x + b^{(1)})$. Since, f_1 is a vector of c_1 softmax functions and q = c_4, then the softmax function in equation 3.50 can be expressed in the form:

$$\forall\, i \subseteq \| 1, c_4 \|, F_{i,\theta_i}(x) = \frac{\exp\ (w_i^{(1)}.x + b_i^{(1)})}{\sum_{i=1}^{c\,n+1} (w_i^{(1)}.x + b_i^{(1)})} \quad (9)$$

3. SYSTEM IMPLEMENTATION

3.1 VOICE AUTHENTICATION PROCESS

Figure 4 shows the implementation methods involved in the voice authentication process. Figure 5 describes the dataset acquisition process. The voice authentication method is text-dependent as each voter is assigned phrases used during enrolment and authentication. Changes are made only to the classification model with two variants of the deep learning model optimized for both small and large datasets. The same feedforward architecture with three hidden layers was implemented for both datasets consisting of 30 selected voters and a large Librispeech ASR corpus labeled SLR12 on openslr.org/resources, respectively. Changes were made to the activation functions of the input and the hidden layer from linear to rectified linear for both small and large datasets, respectively.

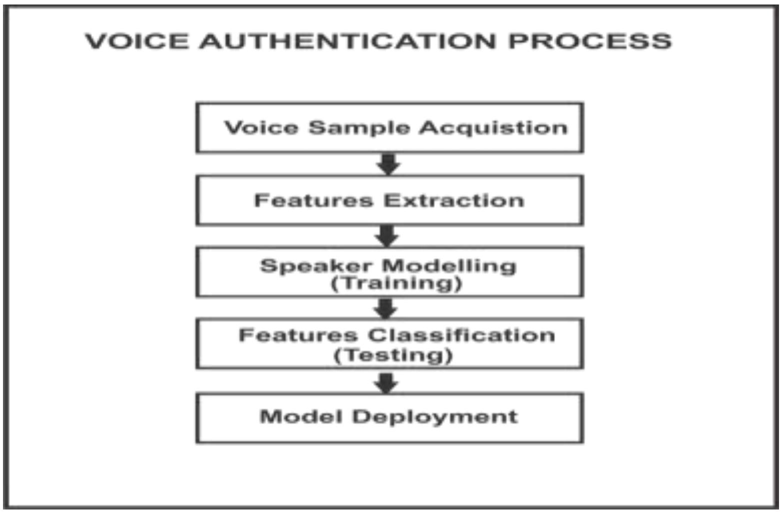

FIGURE 4. Voice authentication process.

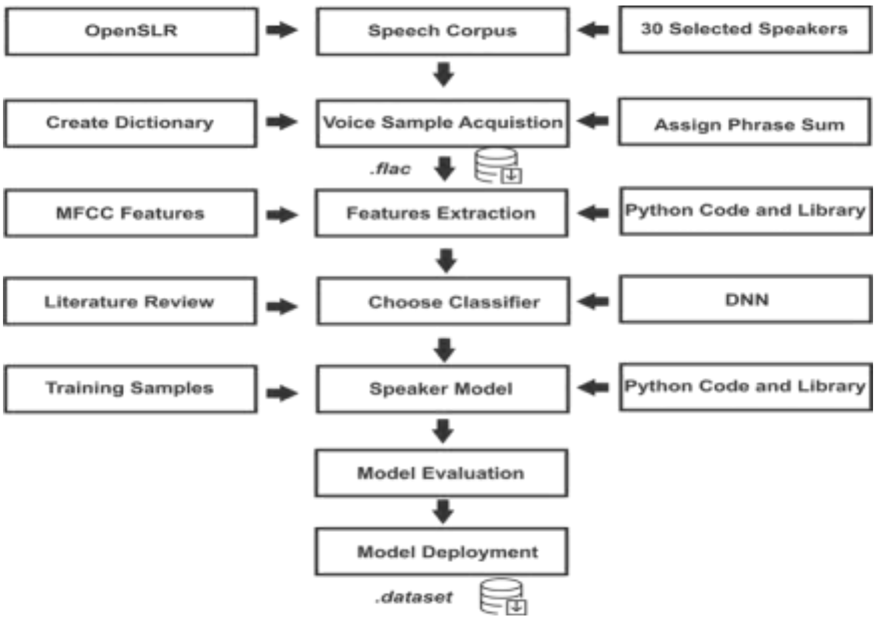

FIGURE 5. Dataset generation process.

3.2 DEEP NEURAL NETWORK TRAINING AND TESTING.

Using the voice samples from both datasets, data frames were created with the name of all the audio files and their corresponding labels. All the MFCC features for

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

each audio file were extracted by passing through a function that iterates through every row of the data frame accessing the file by reading the file's path. An array of 40 features was obtained after extraction with their respective labels set as X and Y. These were further split into validation, training, and test data. A feed-forward neural network with two dense hidden layers was built using linear (30 speakers), relu (Librispeech ASR corpus), and softmax for the outputs. The model was then compiled using adam optimizer and categorical cross-entropy for loss. The performance evaluation was recorded, and the matplotlib library was used to visualize the results graphically.

3.3 PERFORMANCE EVALUATION METRICS

The performance of a classification model is determined based on accuracy, error rate, precision, recall, and f1score. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations, while recall measures how well the classifier could accurately detect positive outcomes. Mathematically, these metrics can be represented as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

$$Precision = \frac{TP}{TP+FP} \tag{10}$$

$$Recall = \frac{TP}{TP+Fn} \tag{11}$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

Where

TP is a true positive (number of items classified correctly)

FP is a false positive (number of negative cases classified correctly)

TN is true negative (number of correct cases classified wrongly)

FN is a false negative (number of incorrect cases classified wrongly)

4 RESULT AND DISCUSSION

4.1 EVALUATION OF DNN MODELS

4.1.1 TRAINING AND TESTING OF DNN CLASSIFIER ON 30 SELECTED PRE-REGISTERED VOTERS.

The DNN model was initially trained and tested using 300 voice samples collected from 30 selected pre-registered voters (10 voice samples per voter). Figure 6, 7, 8, 9, and 10 graphically shows the relationship between training and validation accuracy and the number of epochs for five different trials. As the number of epochs increased, the training accuracy increased, signifying the model's ability to learn as more complete passes were made on the training data.

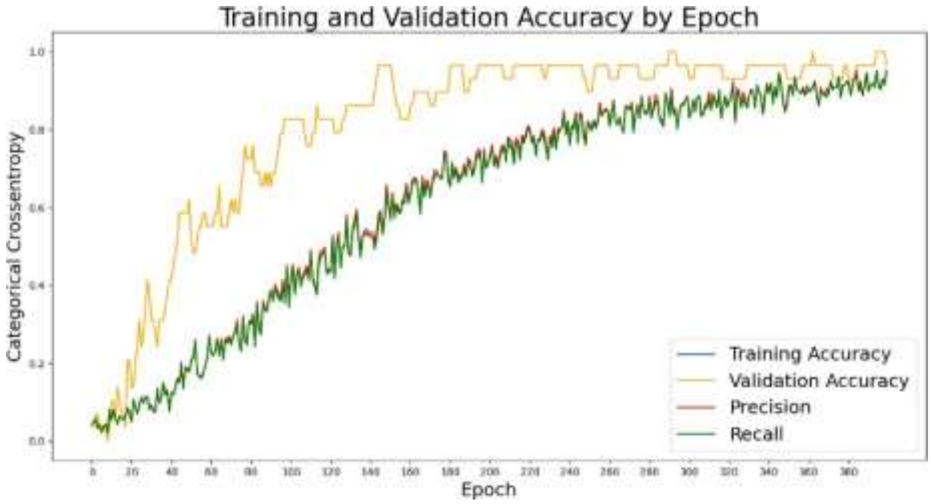

FIGURE 6. Training and Testing of 300 Audio Samples of 30 Pre-Registered Voters at the 400th Epoch (1st Trial)

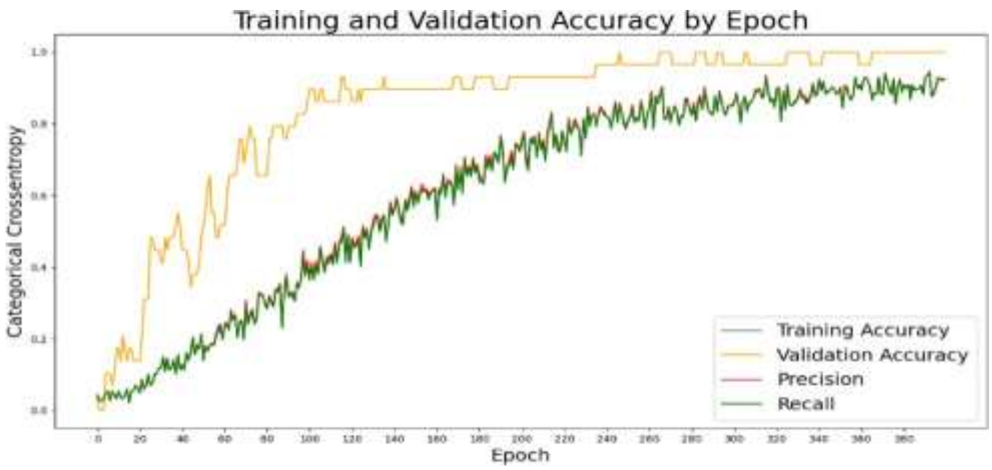

FIGURE 7. Training and Testing of 300 Audio Samples of 30 Pre-Registered Voters at the 400th Epoch (2nd Trial)

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

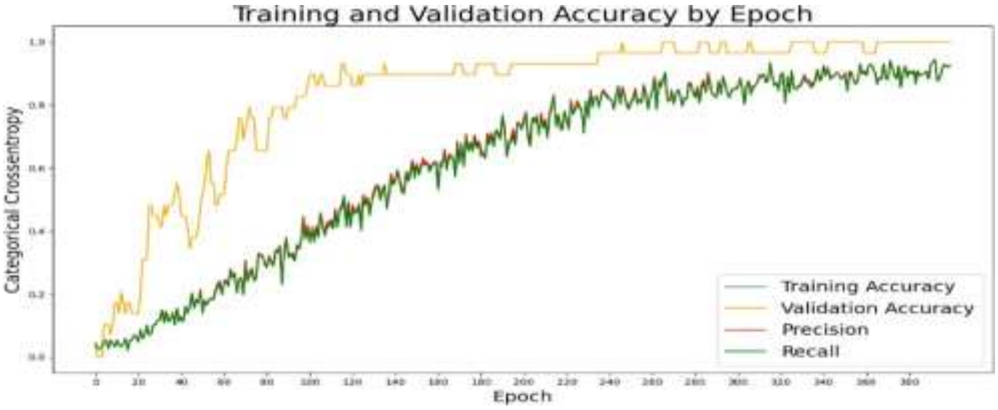

FIGURE 8. Training and Testing of 300 Audio Samples of 30 Pre-Registered Voters at the 400th Epoch (3rd Trial)

FIGURE 9**.** Training and Testing of 300 Audio Samples of 30 Pre-Registered Voters at the 400th Epoch (4th Trial)

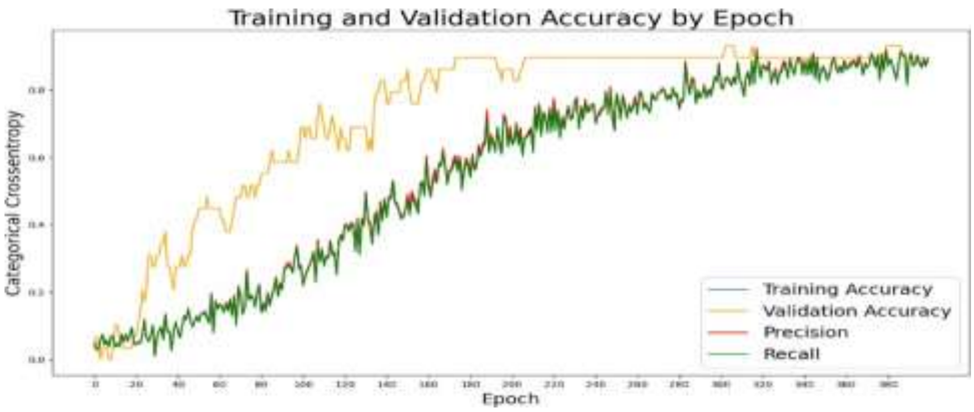

FIGURE 10**.** Training and Testing of 300 Audio Samples of 30 Pre-Registered Voters at the 400th Epoch (5th Trial)

Examining further the graphs in Figure 6, it can be observed that the validation accuracy is slightly higher than the training accuracy for each of the five trials. This could be attributed to the data size and the similarities in the phrases recited by each voter. The results of the precision, accuracy, recall, f1score, and the training time for each of the five trials are shown in Table 1. After five trials, the average training accuracy is about 6.07% lower than the validation accuracy, indicating a slight overfitting condition. Also stated in Table 1 are the average precision, recall, and f1scores of 90.76%, 90.37%, and 90.56%, respectively. The high results obtained in these parameters indicate that the slight memorization of the training data does not yield a gain in the generalization of unlearned data.

TABLE 1
Summary of Results on Training and Testing the DNN Classifier on 300 Audio Recordings Collected From 30 Pre-Registered Voters.

No of Trials	1st Trial	2nd Trial	3rd Trial	4th Trial	5th Trial	Average
Metrics						
Precision	0.9519	0.9251	0.8973	0.8710	0.8925	0.9076
Training Accuracy	0.9519	0.9251	0.8877	0.8663	0.8930	0.9048
Validation Accuracy	0.9655	1.0000	1.0000	0.9655	0.8966	0.9655
Recall	0.9519	0.9251	0.8877	0.8663	0.8877	0.9037
f1score	0.9519	0.9251	0.8925	0.8686	0.8901	0.9056
Training Time (seconds)	33.3863	32.8661	32.7673	38.0815	35.5068	34.5216

The DNN model was further tweaked by changing the activation functions of the hidden layers from linear to reLU using the same architecture to adapt for more significant amounts of data. After fine-tuning the model parameters, the optimal number of epochs was set at 250, and the model was trained using 360 hours of audio recordings from a Librispeech corpus labeled SLR12. Figure 11, 12, 13, and 14 shows the results obtained for training at five different trials, and Table 2 summarizes the graphical results displayed in Figure 11.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

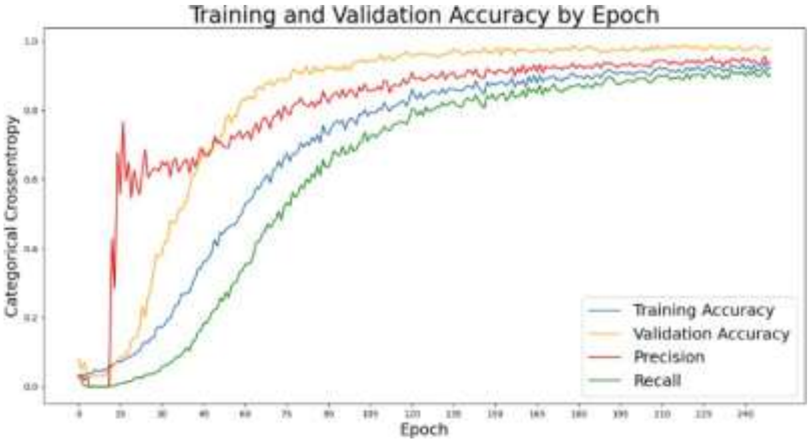

FIGURE 11. Results of Training and Testing on 360 Hours of Audio Recordings from Librispeech Corpus Slr12 at the 250th Epoch (1st Trial).

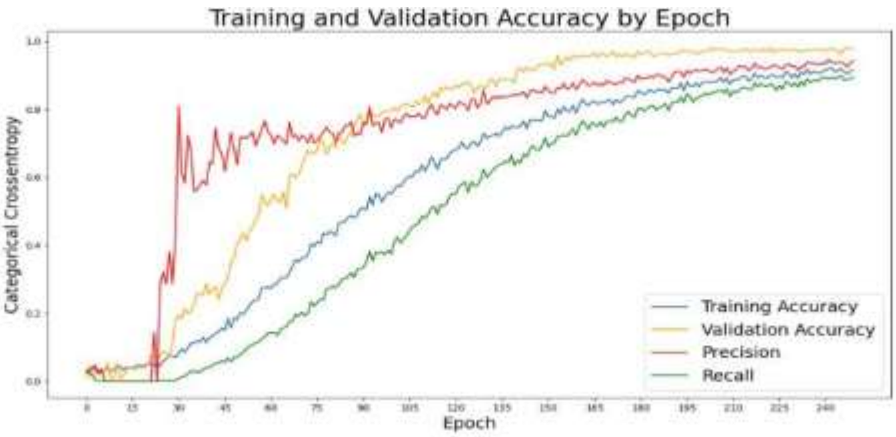

FIGURE 12. Results of Training and Testing on 360 Hours of Audio Recordings from Librispeech Corpus Slr12 at the 250th Epoch (2nd Trial).

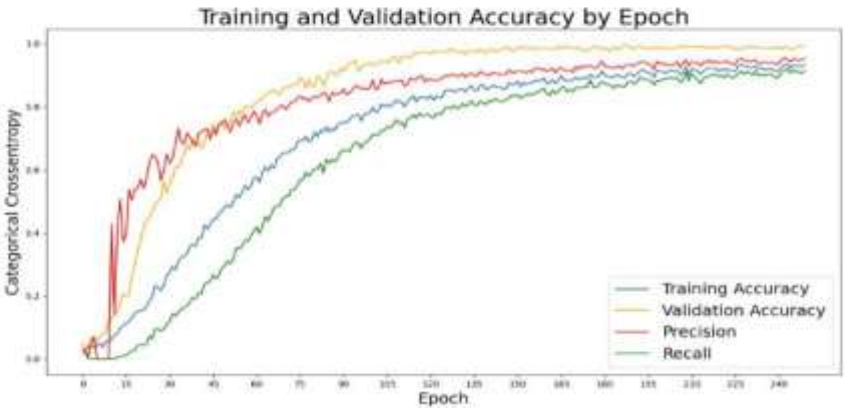

FIGURE 13. Results of Training and Testing on 360 Hours of Audio Recordings from Librispeech Corpus Slr12 at the 250th Epoch (3rd Trial).

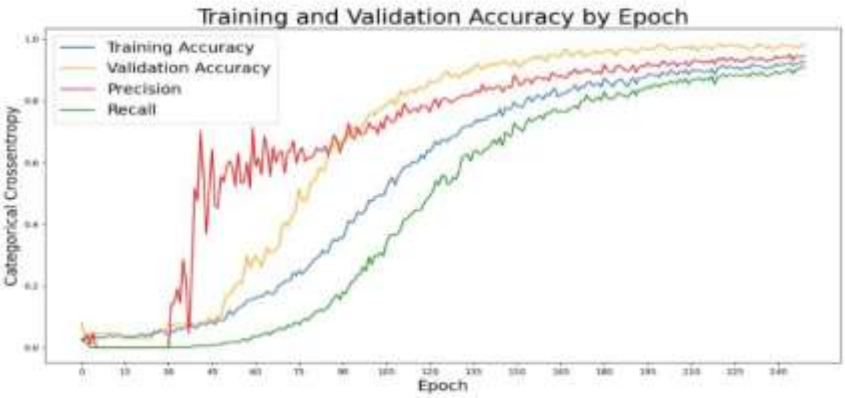

FIGURE 14. Results of Training and Testing on 360 Hours of Audio Recordings from Librispeech Corpus Slr12 at the 250th Epoch (4th Trial).

Generally, the learning model was robust in handling large amounts of data, achieving average validation and training accuracies of 97.71% and 92.39%, respectively, when trained with audio recordings from Librispeech corpus represented in Table 2. A 5.32% gap between the training and validation accuracy indicated a slight chance for overfitting. However, accuracy alone is not enough to determine whether the model memorizes the trained data. The average precision, recall, and f1score obtained from five trials were 94.47%, 90.56%, and 92.47%, respectively. Unlike popular opinion, deep learning algorithms cannot effectively learn on small datasets. The DNN model obtained an average validation accuracy of 96.55% when initially trained with audio recordings from pre-registered voters, as seen in Table 2.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

TABLE 2
Summary of Results on Training and Testing the DNN Classifier for 250 Epochs on 360 Hours of Audio Recordings from Librispeech SLR12

No of Trials	1st Trial	2nd Trial	3rd Trial	4th Trial	5th Trial	Average
Metrics						
Precision	0.9376	0.9375	0.9542	0.9504	0.9437	0.9447
Training Accuracy	0.9216	0.9127	0.9338	0.9287	0.9227	0.9239
Validation Accuracy	0.9771	0.9771	0.9924	0.9847	0.9542	0.9771
f1score	0.9207	0.9123	0.9351	0.9315	0.9237	0.9247
Training Time (seconds)	190.9773	152.7350	165.1565	184.5161	163.6554	171.4081

5 CONCLUSION

This paper proposes a deep neural classifier for classifying disabled voters using their enrolled voice data. Two deep learning models were designed and optimized for small datasets and utilized in large voting scenarios. The use of deep learning algorithms for a large amount of voice data is effective as neural networks learn with more data. E-voting systems can effectively use the designed prototype to authenticate disabled voters and all other voters. Future works can be undertaken to improve the recognition rate of the proposed algorithms.

REFERENCES

[1] Hao, Y., Zeng, Z., & Chang, L. (2021). An Improved Coercion-Resistant E-Voting Scheme. Journal of Security and Communication Networks, Hindawi.

[2] Mochtak, M., Lesschaeve, C., & Glaurdić, J. (2021). Voting and winning: perceptions of electoral integrity in consolidating democracies. Democratization, 1-19.

[3] Haruna, M. (2017). The problems of living with disability in Nigeria. *Journal of Law, Policy and Globalization*, *65*, 103–113. http://www.iiste.org/Journals/index.php/JLPG/article/view/39027/40128

[4] Olaniyi, O. M., Bala, J. A., Ndunagu, J., Abubakar, A., & Is'Haq, A. I. (2019). *V-Authenticate: Voice Authentication System for Electorates Living with Disabilities.* Proceedings of Cyber Secure Nigeria 2019 Conference (CSNC 2019), Abuja, Nigeria, pp 29-38

[5] Bansal, S. (2021). What Is Classification in Machine Learning? Classification Algorithms. https://www.analytixlabs.co.in/blog/classification-in-machine-learning/

[6] Mishra, U. (2021). Binary and Multiclass Classification in Machine Learning | Analytics Steps. https://www.analyticssteps.com/blogs/binary-and-multiclass-classification-machine-learning

[7] Komatineni, S., & Lingala, G. (2020). Secured E-voting System Using Two-factor Biometric Authentication. *Proceedings of the 4th International Conference on Computing Methodologies and Communication, ICCMC 2020*, *Iccmc*, 245–248. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00046

[8] Najam, S., Shaikh, A. Z., & Naqvi, S. (2018). A Novel Hybrid Biometric Electronic Voting System: Integrating Fingerprint and Face Recognition. *Mehran University Research Journal of Engineering and Technology*, *37*(01), 59–68 https://doi.org/10.22581/muet1982.1801.05

[9] Cheema, M. A., Ashraf, N., Aftab, A., Qureshi, H. K., Kazim, M., & Azar, A. T. (2020). Machine Learning with Blockchain for Secure E-voting System. Proceedings - 2020 1st International Conference of Smart Systems and Emerging Technologies, SMART-TECH 2020, 177–182. https://doi.org/10.1109/SMART-TECH49988.2020.00050

[10] Almustapha, J. A., Olaniyi, O. M., Mohammed, I., Ndunagu, J., & Osang, F. B. (2020). Multi-layer Perceptron Model for Mitigating Distributed Denial of Service Flood Attack in Internet Kiosk Based Electronic Voting, Covenant Journal of Informatics and Communication Technology (CJICT) 8(2), 60–68.

[11] Prabhakar, E., Kumar, K. N., Karthikeyan, S., Kumar, A. N., & Kavin, P. (2021). Smart Online Voting And Enhanced Deep Learning to Identify Voting Patterns. 04, 162–165.

[12] Chetty, G., Arunodhayam, T. P. P., & Maheswaran, C. P. (2024, April). Biometric Authentication Voting System. In *2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)* (pp. 1-4). IEEE.

[13] Olaniyi, O. M., Dogo, E. M., Nuhu, B. K., Treiblmaier, H., Abdulsalam, Y. S., & Folawiyo, Z. (2022). A Secure Electronic Voting System Using Multifactor Authentication and Blockchain Technologies. In *Blockchain Applications in the Smart Era* (pp. 41-63). Cham: Springer International Publishing.

[14] Tanwar, S., Gupta, N., Kumar, P., & Hu, Y. C. (2024). Implementation of blockchain-based e-voting system. *Multimedia Tools and Applications*, *83*(1), 1449-1480.

[15] Spanos, A., & Kantzavelou, I. (2023). A Blockchain-based Electronic Voting System: EtherVote. *arXiv preprint arXiv:2307.10726*.

[16] Variani, E., Lei, X., McDermott, E., Moreno, I. L., & Gonzalez-Dominguez, J. (2014). Deep neural networks for small footprint text-dependent speaker verification. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 4052–4056. https://doi.org/10.1109/ICASSP.2014.6854363

[17] Ruparel, N. H., & Shahane, N. M. (2013). Learning from Small Data Set to Build Classification Model: A Survey. International Journal of Computer Applications, 4.

[18] Variani, E., Lei, X., McDermott, E., Moreno, I. L., & Gonzalez-Dominguez, J. (2014). Deep neural networks for small footprint text-dependent speaker verification. ICASSP, IEEE International Conference on Acoustics, Speech and

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

Signal Processing-Proceedings, 4052–4056. https://doi.org/10.1109/ICASSP.2014.6854363

[19] Suzić, S., Delić, T., Jovanović, V., Sečujski, M., Pekar, D., & Delić, V. (2018). A comparison of multi-style DNN-based TTS approaches using small datasets. 03005, 1–6.

[20] Oke, B. A., Olaniyi, O. M., Aboaba, A. A., & Arulogun, O. T. (2021). Multifactor authentication technique for a secure electronic voting system. Electronic Government, an International Journal, *17*(3), pp 312-338.

[21] Chen, G., Zhang, H., Dai, N., & Song, Y. (2020). Topology-free optimal power dispatch for distribution network considering security constraints and flexible building thermal inertia. 1–5. http://arxiv.org/abs/2011.09630