Classification of Phonetic Syllables Using Stacked Autoencoder and Characterization via Centroid

1st Francisco dos Santos Viana
Doctor's Program in Computer Science
Federal University of Maranhão
São Luis, Brazil
francisco.santos@discente.ufma.br

2nd Carlos Eduardo Nascimento Cajado
Master's Program in Computer Science
Federal University of Maranhão
São Luis, Brazil
carlos.cajado@discente.ufma.br

3rd Samuel Magalhães Pereira
Program in Computer Science
Federal University of Maranhão
São Luis, Brazil
samuel.mp@discente.ufma.br

4th Alexandre Cesar Muniz de Oliveira
Department of Computer Engineering
Federal University of Maranhão
São Luis, Brazil
alexandre.cesar@ufma.br

5th Carlos Soares
Faculty of Engineering of Porto
University of Porto, Portugal
Porto, Portugal
csoares@fe.up.pt

6thAreolino de Almeida Neto
Department of Computer Engineering
Federal University of Maranhão
São Luis, Brazil
areolino.neto@ufma.br

***Abstract*—This work presents a new method for layer insertion in stacked autoencoder neural networks. In this approach, a branch of layers is inserted on the side of the last hidden layer, and the output layer after the training of an existing layer has stabilized. Later, the new branch is merged with the previous layers. This insertion type is called collaborative, as it introduces new knowledge to the network without reducing the knowledge already acquired by the previous layers. This approach enables a neural network to learn with reduced design time. It overcomes the typical problem of defining the number of layers and the number of neurons in each layer. This technique was applied in phonetic syllable classification, where the Fast Fourier Transform obtains the audio data. These audio data were processed using a vertical bar plot to compress the audio data by using centroids. This procedure provided data compression without losing characteristics. Thus, collaborative insertions were evaluated in terms of the degree of growth for a multi-class classification problem.**

***Index Terms*—Artificial Neural Networks, Deep Learning, Stacked Autoencoder Networks, and Centroids.**

I. Introduction

Pattern recognition is an interdisciplinary area that combines machine learning and artificial intelligence to detect and classify patterns in complex data [1], [2]. Speech recognition is a specific branch focused on extracting, processing, and categorizing information from speech signals [3], [4]. Despite its global relevance, Portuguese still lacks extensive Automatic speech recognition (ASR) research and resources [5], motivating efforts toward specialized methods for this language.

ASR systems operate through preprocessing, feature extraction, and classification [6]. Preprocessing reduces noise and speaker variability, while feature extraction generates compact representations that strongly affect classification performance [7]. Common representations include LPC, PLP, GFCC, DWT, FBANKs, and MFCC [8], [9].

Artificial neural networks are widely adopted in ASR, but their performance is highly dependent on architectural design choices, particularly network depth [10], [11]. Since no principled rule exists to define the optimal number of layers, architectures are often selected through trial and error. This work addresses this limitation by proposing a collaborative layer-insertion strategy that incrementally defines network depth, adding parallel hidden-layer branches and retaining them only when measurable learning improvements are achieved.

II. Related Works

Several studies address automated neural network design. Bayesian optimization has been applied to tune learning rate, depth, and activation functions, outperforming manual and grid search [12]. Comparisons with genetic algorithms highlight tradeoffs between exploration capability and computational cost [13]. Particle swarm optimization has also been used to design stacked autoencoders with competitive accuracy and reduced complexity [14], while optimized recurrent models show strong performance in speech recognition [15].

Constructive neural models originate from the CCNN framework. FCCN incrementally adds neurons while training only output weights, reducing training cost [16]. Other variants analytically compute weights [17], dynamically insert neurons [18], or integrate evolutionary strategies to obtain compact and stable architectures [19].

In speech recognition, Portuguese phonetic syllable classification using constructive stacked autoencoders was investigated in [20]. Classical MLP-based methods remain effective for small vocabularies [21], whereas recent models such as SincNet [22] and modular expert networks [23] achieve higher accuracy at the cost of increased computational complexity. In contrast to these approaches, the present work proposes a collaborative layer-insertion strategy that incrementally defines network depth through parallel branches, reducing architectural design effort without relying on extensive hyperparameter optimization.

III. Methodology

The algorithms were implemented in Python using Google Colab. The experiments employed the *Male/Female for Forced Phonetic Alignment* dataset, which consists of clean studio recordings from one male and one female Brazilian Portuguese speaker and is originally highly unbalanced across phonetic classes [24]. To mitigate classification bias, avoid dominance of frequent classes, and allow a fair evaluation of the learning capacity of the proposed model, a balanced subset was constructed by selecting 43 phonetic syllable classes with 30 samples each, resulting in 1,290 syllable instances. Audio preprocessing and phonetic segmentation were performed using Praat, UFPAlign, and Kaldi, producing syllable-level signals used for feature extraction [25]–[27].

A. Collaborative layer insertion

The main novelty of this work is a collaborative layer-insertion strategy that incrementally defines network depth by adding parallel hidden-layer branches, which are retained only when they provide measurable learning improvement, avoiding exhaustive hyperparameter search. To mitigate error degradation and vanishing gradients [28], [29], new layers are inserted sequentially while preserving prior learning. As illustrated in Fig. 1, the initial network (solid black lines) is first trained and then frozen. A new branch is added using three types of connections: random intra-branch weights (blue dashed), inter-branch connections to previous layers (red dashed), and zero-initialized weights (black dashed).

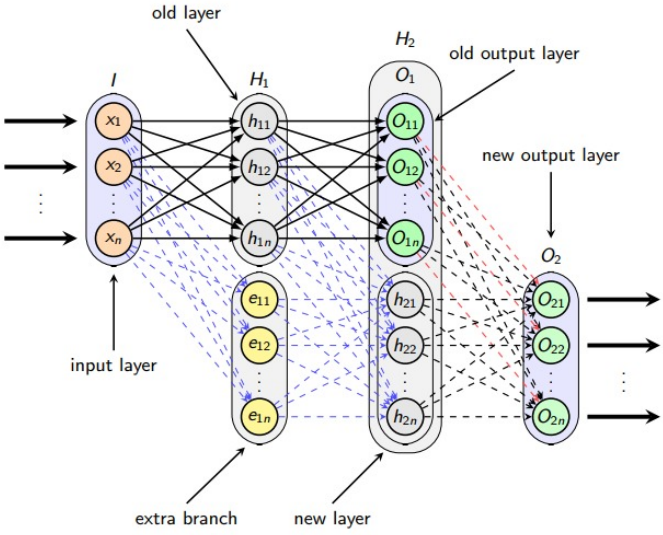


Fig. 1: Collaborative process of inserting a new hidden layer

The network input is $X = (x_1, x_2, \ldots, x_n)$, where n is the size of the input and output layers. The vector $H_1 = (h_{11}, h_{12}, \ldots, h_{1m})$ represents the hidden-layer activation of branch 1 after applying f (Eq. 1), using weight matrix W_1^1 and bias vector b_1^1. Eq. 2 defines the output layer. W_2^1 are the weights from the hidden layer to the output layer, and b_2^1 is the output bias. Branch 2 consists of a hidden layer and an output layer. Its input comes from the last hidden layer of Branch 1, and the output of Branch 1 is concatenated with the hidden layer of Branch 2. To expand the search space and reduce the overhead caused by adding new hidden layers, an extra branch can be added. This branch connects the input layer directly to the new hidden layer and is represented as $E = (e_1, e_2, \ldots, e_n)$. Thus, the output of the hidden layer H_2 of Branch 2 is defined by Eq. 3, where W_2^2 is the weight matrix between the hidden and output layers.

$$H_1 = f(W_1^1 X + b_1^1) \tag{1}$$

$$O_1 = f(W_2^1 H_1 + b_2^1) \tag{2}$$

$$H_2 = [f(W_1^2 H_1 + W_2^e E + b_1^2) \;\; O_1]^T \tag{3}$$

The output layer of the new branch has connections with the elements of its hidden layer (w_2^2) and with the old branch through the weight matrix $I_k(n)$. Together, these components form the weight matrix W_2^2, defined in Eq. 4. The weights w_2^2 are initialized to zero, ensuring that the new neurons do not affect the knowledge already learned by the network at the beginning of training. The matrix $I_k(n)$ is diagonal, with its main diagonal elements set to the value k. The network output is provided by branch 2, defined by Eq. 5. Training branch 2 aims to compensate for the output of branch 1. Since the connections k have values equal to one and the activation function of the output layer is linear, the neural network will have only one output layer (branch 2 output layer), formed by combining the outputs of the two branches.

$$W_2^2 = [I_k(n) \;\; w_2^2] = \begin{bmatrix} k & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & k & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{4}$$

$$O_2 = f(W_2^2 H_2 + b_2^2) \tag{5}$$

B. Feature Extraction Using Centroids

After preprocessing [20], the FFT (first 8 kHz) converts audio to frequency rectangles. Adjacent frequencies form blocks summarized by centroids $C_m = (X_m, Y_m)$ (Eq. 6, 7). Fig. 2 shows grouping: blue points are bar centers, red points are centroids. Sixteen blocks of 50 frequencies yield 16 centroids per syllable, normalized to $[0, 1]$ for input.

$$X_m = \frac{\sum_{f=1}^{n} x_f \times a_f}{\sum_{f=1}^{n} a_f} \tag{6}$$

$$Y_m = \frac{\sum_{f=1}^{n} \frac{y_f}{2} \times a_f}{\sum_{f=1}^{n} a_f} \tag{7}$$

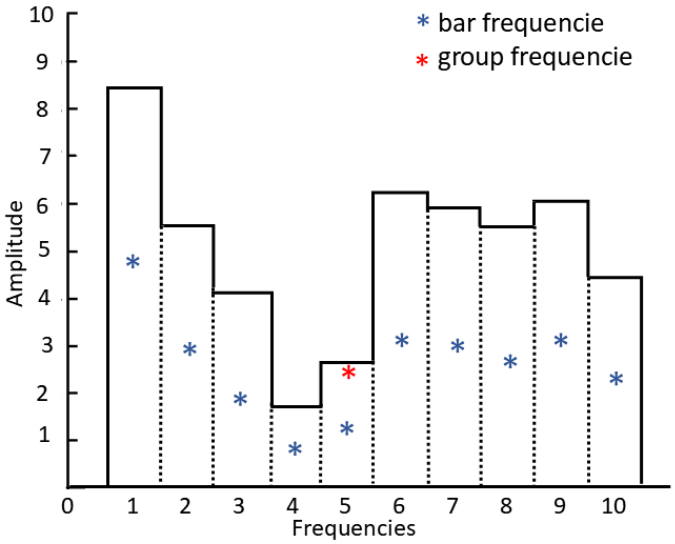


Fig. 2: Representation of a group of 10 frequency intervals.

C. Evaluation of Experiments

MSE was used for training; accuracy, precision, sensitivity, and F1-score validated performance. Growth strategies were constant, increasing, decreasing, and random. Weight k variations: M1 – constant without extra branch; M2 – adjustable without extra branch; M3 – constant with extra branch; M4 – adjustable with extra branch.

Network depth was defined incrementally using the proposed collaborative insertion strategy. Four hidden layers were added sequentially, resulting in one input layer, five hidden layers, and one output layer. The input consisted of centroid coordinates (X_m, Y_m) (32 features), with 25 neurons in the first hidden layer (empirically defined) and 43 outputs. Learning rates were 1×10^{-5} (input–hidden) and 5×10^{-5} (hidden–output). Hidden layers used $\tanh$, the output layer was linear, and performance was evaluated using 5-fold cross-validation.

IV. Results

The following figures analyze the effect of different layer-insertion strategies and collaboration mechanisms on training error and generalization. Fig. 3 shows errors for constant-type insertions; dashed black lines indicate the scenario without new layers, allowing direct comparison. Table I presents training, validation, and test MSE, along with the training error of each inserted layer. For the constant type, all methods achieved similar final errors. Extra depth reduced MSE in all methods (M1–M4), with the largest drops in the first three insertions, while still outperforming the baseline. Methods M2 and M4 achieved the lowest errors, especially in deeper networks.

Figure 4 shows the results of incremental insertion. The largest gains occur in the first added layers, with a clear error drop between layers 1 and 3. Methods M2 and M4 reach the lowest errors, especially in deeper configurations, indicating that adjusting the weight k improves learning. These results confirm that progressive layer insertion enhances modeling and that the branch-connection mechanism gives the new layers greater learning flexibility.

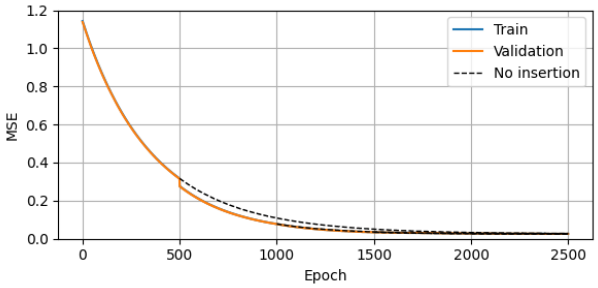
In the descending insertion method, Fig. 5, methods M2 and M4 achieve greater error reductions than M1 and M3. Table I shows that M2 and M4 consistently outperform the others for any depth, indicating greater robustness and generalizability.

In random insertion, each new layer has a dynamic number of neurons. Fig. 6 shows that M1 and M3 do not consistently reduce the training error, except in the third layer, while M2 and M4, with the outer branch and adjustable *k*, reduce the error. Table I shows that varying the number of neurons offers limited gains, but M2 and M4 achieve smaller errors (0.034 and 0.032) compared to M1 and M3 (0.066 and 0.058), reflecting better adaptation.

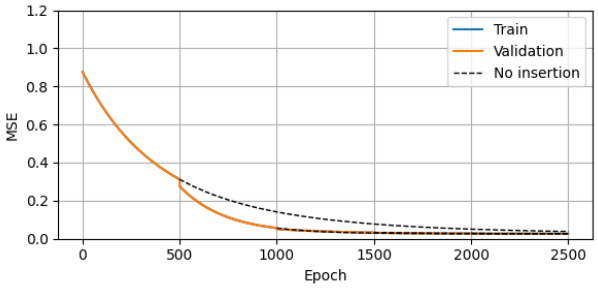
Table II compares the best results of this work (collaborative insertion) with the method proposed by [20]. The collaborative insertion model outperformed the literature, demonstrating consistent classification capabilities across all insertion types. In terms of accuracy, the proposed approach exceeded 76% with only two layers inserted, and the lowest value achieved was 87% for methods M1 and M3 under random insertion.

TABLE I: Mean squared error (MSE) obtained for each layer-insertion strategy.

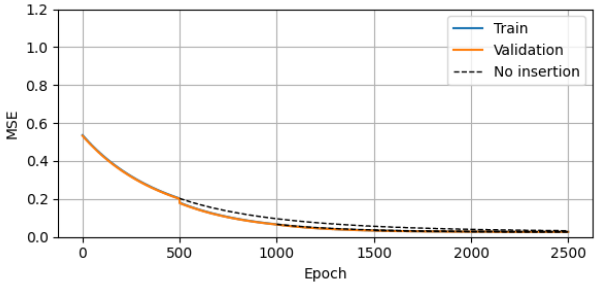
Type	Method	Layer 1			Layer 2			Layer 3			Layer 4			Layer 5		
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test
Constant	M1	0.317	0.315	0.320	0.078	0.077	0.079	0.034	0.034	0.035	0.028	0.029	0.026	0.026	0.026	0.027
	M2	0.312	0.312	0.307	0.056	0.056	0.055	0.033	0.032	0.031	0.028	0.028	0.027	0.027	0.026	0.026
	M3	0.203	0.201	0.203	0.067	0.066	0.067	0.034	0.034	0.034	0.028	0.028	0.028	0.027	0.027	0.027
	M4	0.170	0.168	0.171	0.048	0.047	0.049	0.031	0.030	0.031	0.027	0.027	0.027	0.026	0.026	0.026
Increasing	M1	0.267	0.266	0.266	0.078	0.077	0.077	0.036	0.036	0.036	0.034	0.034	0.034	0.034	0.034	0.034
	M2	0.286	0.284	0.289	0.051	0.051	0.052	0.029	0.029	0.029	0.028	0.028	0.028	0.028	0.028	0.028
	M3	0.295	0.296	0.294	0.101	0.101	0.100	0.034	0.034	0.033	0.033	0.033	0.032	0.032	0.032	0.032
	M4	0.246	0.246	0.245	0.045	0.045	0.044	0.027	0.028	0.027	0.027	0.027	0.027	0.027	0.027	0.027
Decreasing	M1	0.324	0.321	0.325	0.132	0.130	0.132	0.065	0.064	0.065	0.042	0.042	0.042	0.037	0.037	0.037
	M2	0.270	0.272	0.266	0.064	0.065	0.062	0.038	0.039	0.037	0.032	0.033	0.031	0.030	0.030	0.029
	M3	0.352	0.350	0.346	0.135	0.134	0.131	0.064	0.063	0.061	0.046	0.045	0.043	0.038	0.038	0.037
	M4	0.322	0.319	0.321	0.054	0.053	0.054	0.033	0.033	0.033	0.030	0.029	0.029	0.029	0.028	0.028
Random	M1	0.250	0.248	0.249	0.180	0.178	0.179	0.101	0.103	0.103	0.085	0.084	0.084	0.066	0.065	0.065
	M2	0.270	0.271	0.269	0.090	0.091	0.090	0.042	0.043	0.042	0.038	0.038	0.038	0.034	0.034	0.034
	M3	0.296	0.300	0.298	0.205	0.208	0.207	0.099	0.101	0.101	0.085	0.087	0.087	0.058	0.059	0.059
	M4	0.309	0.308	0.310	0.082	0.082	0.083	0.041	0.041	0.041	0.037	0.037	0.037	0.032	0.032	0.032



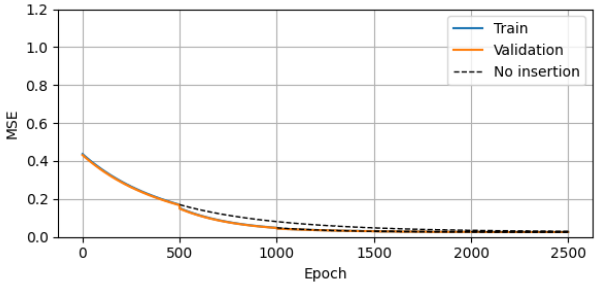
(a) Method M1



(b) Method M2

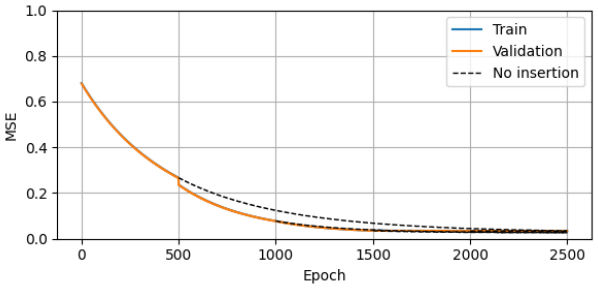


(c) Method M3

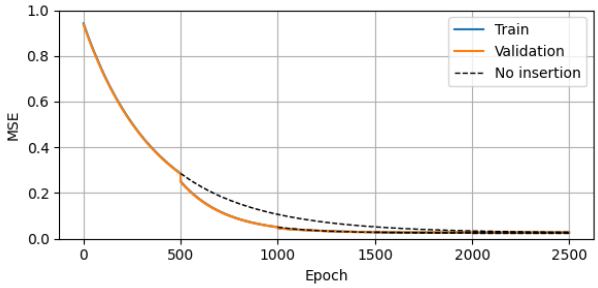


(d) Method M4

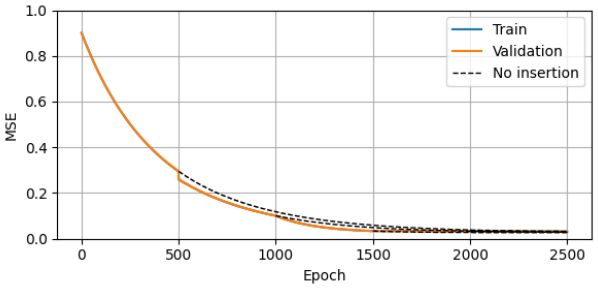
Fig. 3: Mean squared error (MSE) across successive constant layer insertions.



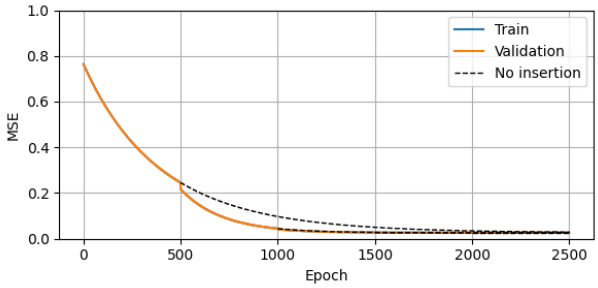
(a) Method M1



(b) Method M2



(c) Method M3



(d) Method M4

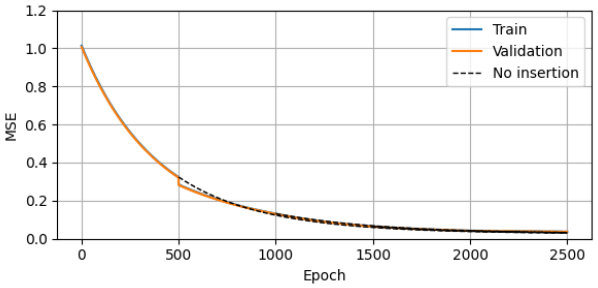
Fig. 4: Mean squared error (MSE) across successive increasing layer insertion.

TABLE II: Performance comparison with related works.

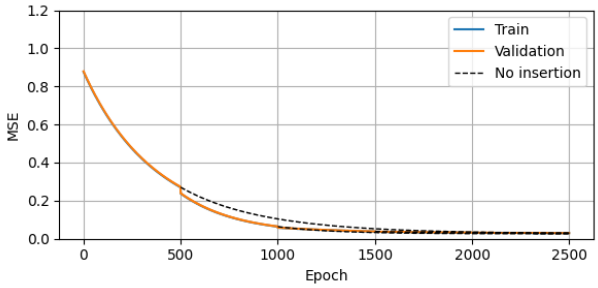
Metric	Collabnet	Collaborative insertion
accuracy	0.76	0.98
sensitivity	0.76	0.89
precision	0.82	0.93
f1-score	0.77	0.91

V. Conclusion

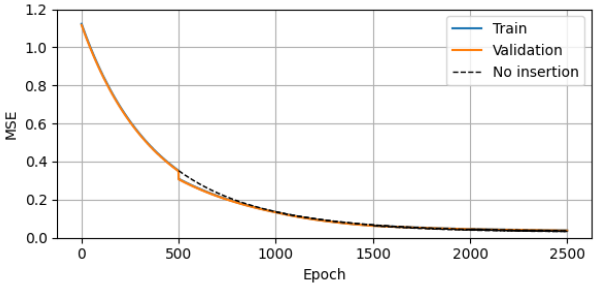
The insertion types and methods show that a collaborative branching strategy for adding hidden layers to stacked autoencoders improves performance, overfitting control, and reduces hyperparameter tuning effort, while enhancing scalability and



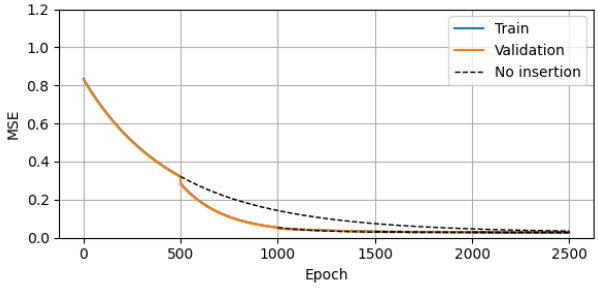
(a) Method M1



(b) Method M2

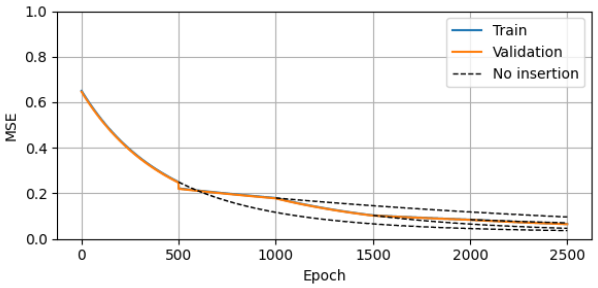


(c) Method M3

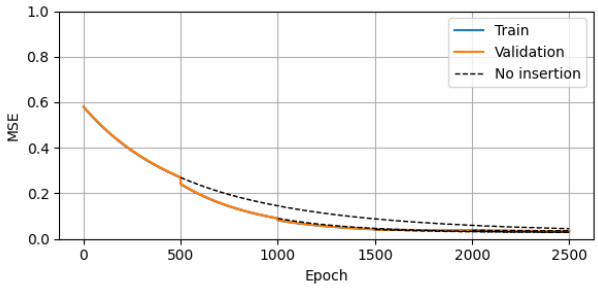


(d) Method M4

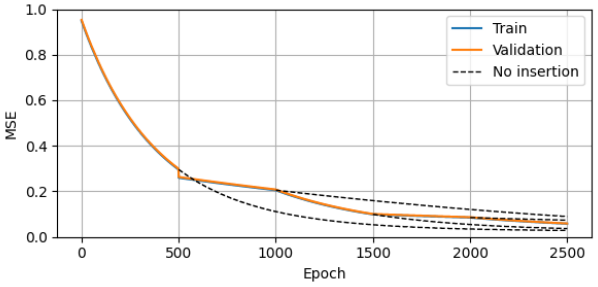
Fig. 5: Mean squared error (MSE) across successive decreasing layer insertion.



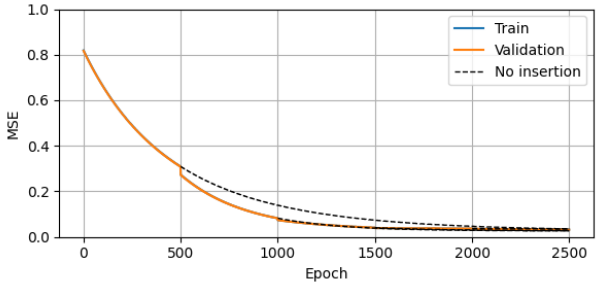
(a) Method M1



(b) Method M2



(c) Method M3



(d) Method M4

Fig. 6: Mean squared error (MSE) across successive random layer insertion.

design efficiency.

The centroid-based representation efficiently compresses phonetic-syllable data, reducing dimensionality and computational cost while preserving relevant spectral information for learning. Although the experiments used a controlled dataset with two speakers, the results validate the approach as a proof of concept for adaptive depth construction in stacked autoencoders. Future work includes evaluation on larger datasets, comparison with modern speech recognition architectures, and investigation of dynamic activation-function switching after layer insertion to further enhance adaptability

and performance.

ACKNOWLEDGMENT

This work was supported by the Coordination for the Improvement of Higher Education Personnel (CAPES) - Financing Code 001. We also thank FAPEMA and CNPq (call no. 26/2021) for funding this research.

REFERENCES

[1] Z. Amiri, A. Heidari, N. J. Navimipour, M. Unal, and A. Mousavi, "Adventures in data analysis: A systematic review of deep learning techniques for pattern recognition in cyber-physical-social systems," *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 22 909–22 973, 2024. [Online]. Available: https://doi.org/10.1007/s11042-023-16382-x

[2] A. F. Alnuaimi and T. H. Albaldawi, "An overview of machine learning classification techniques," in *BIO Web of Conferences*, vol. 97. EDP Sciences, 2024, p. 00133. [Online]. Available: https://doi.org/10.1051/bioconf/20249700133

[3] T. Weise, K. C. Demir, P. A. Pérez-Toro, T. Arias-Vergara, A. Maier, E. Nöth, M. Schuster, B. Heismann, and S. H. Yang, "Towards end-to-end speech articulation and spoken language analysis using deep learning," *Human-Centric Intelligent Systems*, pp. 1–20, 2025. [Online]. Available: https://doi.org/10.1007/s44230-025-00094-6

[4] L. Ganu and B. Arun, "Deep learning and multiwavelet approach for nyishi phoneme recognition: acoustic analysis and model development," *International Journal of Information Technology*, pp. 1–18, 2025. [Online]. Available: https://doi.org/10.1007/s41870-025-02461-9

[5] T. Aguiar de Lima and M. Da Costa-Abreu, "A survey on automatic speech recognition systems for portuguese language and its variations," *Computer Speech & Language*, vol. 62, p. 101055, 2020. [Online]. Available: https://doi.org/10.1016/j.csl.2019.101055

[6] D. Al-Fraihat, Y. Sharrab, F. Alzyoud, A. Qahmash, M. Tarawneh, and A. Maaita, "Speech recognition utilizing deep learning: A systematic review of the latest developments," *Human-centric computing and information sciences*, vol. 14, 2024. [Online]. Available: https://doi.org/10.22967/HCIS.2024.14.015

[7] Y. Li, Y. Wang, L. M. Hoi, D. Yang, and S.-K. Im, "A review on speech recognition approaches and challenges for portuguese: exploring the feasibility of fine-tuning large-scale end-to-end models," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2025, no. 1, p. 3, 2025. [Online]. Available: https://doi.org/10.1186/s13636-024-00388-w

[8] A. Meftah, Y. A. Alotaibi, and S.-A. Selouani, "A comparative study of different speech features for arabic phonemes classification," in *2016 European Modelling Symposium (EMS)*, 2016, pp. 47–52. [Online]. Available: https://doi.org/10.1109/EMS.2016.018

[9] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, "Automatic speech recognition: a survey," *Multimedia Tools and Applications*, vol. 80, pp. 9411–9457, 2021. [Online]. Available: https://doi.org/10.1007/s11042-020-10073-7

[10] H. T. Ünal and F. Başçiftçi, "Evolutionary design of neural network architectures: a review of three decades of research," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 1723–1802, 2022. [Online]. Available: https://doi.org/10.1007/s10462-021-10049-5

[11] M. M. Hammad, "Artificial neural network and deep learning: Fundamentals and theory," *arXiv preprint arXiv:2408.16002*, 2024. [Online]. Available: https://doi.org/10.1007/978-3-031-29642-0

[12] R. Silva and J. Camata, "Hyperparameter optimization of physics-guided neural networks in a convective-diffusive problem," in *Companion Proceedings of the 25th Symposium on High Performance Computing Systems*. Porto Alegre, RS, Brasil: SBC, 2024, pp. 137–144. [Online]. Available: https://sol.sbc.org.br/index.php/sscad_estendido/article/view/30979

[13] H. Alibrahim and S. A. Ludwig, "Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, 2021, pp. 1551–1559. [Online]. Available: https://doi.org/10.1109/CEC45853.2021.9504761

[14] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "An experimental study on hyper-parameter optimization for stacked auto-encoders," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018, pp. 1–8. [Online]. Available: https://doi.org/10.1109/CEC.2018.8477921

[15] A. Morteza, A. A. Yahyaeian, M. Mirzaeibonehkhater, S. Sadeghi, A. Mohaimeni, and S. Taheri, "Deep learning hyperparameter optimization: Application to electricity and heat demand prediction for buildings," *Energy and Buildings*, vol. 289, p. 113036, 2023. [Online]. Available: https://doi.org/10.1016/j.enbuild.2023.113036

[16] X. Wu, P. Rozycki, J. Kolbusz, and B. M. Wilamowski, "Constructive cascade learning algorithm for fully connected networks," in *Artificial Intelligence and Soft Computing: 18th International Conference, ICAISC 2019, Zakopane, Poland, June 16–20, 2019, Proceedings, Part I 18*. Springer, 2019, pp. 236–247.

[17] Z. Wang, W. A. Khan, H.-L. Ma, and X. W. and, "Cascade neural network algorithm with analytical connection weights determination for modelling operations and energy applications," *International Journal of Production Research*, vol. 58, no. 23, pp. 7094–7111, 2020. [Online]. Available: https://doi.org/10.1080/00207543.2020.1764656

[18] S. A. E.-M. Mohamed, M. H. Mohamed, and M. F. Farghally, "A new cascade-correlation growing deep learning neural network algorithm," *Algorithms*, vol. 14, no. 5, p. 158, 2021. [Online]. Available: https://doi.org/10.3390/a14050158

[19] J. Deng, Q. Li, and W. Wei, "Improved cascade correlation neural network model based on group intelligence optimization algorithm," *Axioms*, vol. 12, no. 2, 2023. [Online]. Available: https://doi.org/10.3390/axioms12020164

[20] B. V. L. PEREIRA, "Phoneme recognition with frequency compression via centroid and stacked autoencoder networks." Master's thesis, Universidade Federal do Maranhão, São Luis - MA, 2014. [Online]. Available: https://tedebc.ufma.br/jspui/handle/tede/5486

[21] J. F. Valiati, "Voice recognition for driving commands through neural networks," Master's thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000. [Online]. Available: http://hdl.handle.net/10183/2947

[22] K. Radha, M. Bansal, and R. B. Pachori, "Automatic speaker and age identification of children from raw speech using sincnet over erb scale," *Speech Communication*, vol. 159, p. 103069, 2024. [Online]. Available: https://doi.org/10.1016/j.specom.2024.103069

[23] N. Nedjah, A. D. Bonilla, and L. de Macedo Mourelle, "Automatic speech recognition of portuguese phonemes using neural networks ensemble," *Expert Systems with Applications*, vol. 229, p. 120378, 2023. [Online]. Available: https://doi.org/10.1016/j.eswa.2023.120378

[24] C. Batista and N. Neto, "Forced phonetic alignment in brazilian portuguese using time-delay neural networks," in *International Conference on Computational Processing of the Portuguese Language*. Springer, 2022, pp. 323–332. [Online]. Available: https://doi.org/10.1007/978-3-030-98305-5_30

[25] C. Batista, A. L. Dias, and N. Neto, "Free resources for forced phonetic alignment in brazilian portuguese based on kaldi toolkit," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, p. 11, 2022. [Online]. Available: https://doi.org/10.1186/s13634-022-00844-9

[26] G. Souza and N. Neto, "An automatic phonetic aligner for brazilian portuguese with a praat interface," in *Computational Processing of the Portuguese Language: 12th International Conference, PROPOR 2016, Tomar, Portugal, July 13-15, 2016, Proceedings 12*. Springer, 2016, pp. 374–384. [Online]. Available: https://doi.org/10.1007/978-3-319-41552-9_38

[27] A. L. Dias, C. Batista, D. Santana, and N. Neto, "Towards a free, forced phonetic aligner for brazilian portuguese using kaldi tools," in *Brazilian Conference on Intelligent Systems*. Springer, 2020, pp. 621–635. [Online]. Available: https://doi.org/10.1007/978-3-030-61377-8_44

[28] O. A. Montesinos López, A. Montesinos López, and J. Crossa, *Fundamentals of Artificial Neural Networks and Deep Learning*. Cham: Springer International Publishing, 2022, pp. 379–425. [Online]. Available: https://doi.org/10.1007/978-3-030-89010-0_10

[29] J. P. S. Rosa, D. J. D. Guerra, N. C. G. Horta, R. M. F. Martins, and N. C. C. Lourenço, *Overview of Artificial Neural Networks*. Cham: Springer International Publishing, 2020, pp. 21–44. [Online]. Available: https://doi.org/10.1007/978-3-030-35743-6_3