Evolving Reservoir-Enhanced Neural Architectures for Biomedical Signal Classification

Bárbara Milagres[1], **Guilherme Silva**[1], **Arthur Negrão**[1],
Ederson N. F. G. Júnior[1], **Matheus Vieira**[1], **Pedro Silva**[2]

[1] Postgraduate Program in Computer Science – Federal University of Ouro Preto – Brazil
`barbara.leticia@aluno.ufop.edu.br`

[2]Computing Department – Federal University of Ouro Preto – Brazil
`silvap@ufop.edu.br`

Abstract. *Neural Architecture Search (NAS) has become an effective approach for automating neural network design, including applications to health data. In this work, we extend the GeneticNAS framework by incorporating Reservoir Computing operations into the search space, enabling the evolution of architectures that combine convolutional layers with reservoir-based dynamics to better capture temporal dependencies in biomedical signals. We evaluate the proposed approach on two benchmark datasets: MIT-BIH Arrhythmia (ECG) and PhysioNet EEG for biometric identification. Experimental results show consistent improvements over the baseline, with gains of 2% and 5.5% in accuracy and F1-Score respectively for arrhythmia detection for EEG biometrics.*

1. Introduction

Deep learning models have advanced feature extraction from unstructured data, enabling significant progress across domains such as biomedical signal processing, speech analysis, and image recognition [Singh et al. 2017, Amberkar et al. 2018, Pak and Kim 2017, Wistuba et al. 2019]. However, designing effective neural network architectures remains a complex and time-consuming task that typically relies on expert knowledge and extensive manual experimentation. To address this challenge, *Neural Architecture Search* (NAS) has emerged as a promising paradigm for automating the design of neural network topologies [Ren et al. 2021].

Early NAS approaches demonstrated remarkable performance but required extremely high computational resources, often relying on reinforcement learning or evolutionary strategies combined with large-scale training procedures [Zoph et al. 2018, Real et al. 2019]. Subsequent research focused on improving the efficiency of architecture search. Methods such as *Differentiable Architecture Search* (DAS) [Shin et al. 2018] and *Efficient Neural Architecture Search* (ENAS) [Pham et al. 2018] introduced weight-sharing mechanisms and differentiable optimization strategies that significantly reduced the computational cost of exploring large architectural spaces. Building on these ideas, *Genetic Network Architecture Search* (GeneticNAS) [Habi and Rafalovich 2019] combines evolutionary algorithms with parameter sharing, enabling efficient exploration of candidate architectures while maintaining competitive performance, particularly in image classification tasks.

In parallel, one-dimensional Convolutional Neural Networks (1D CNNs) and evolutionary search strategies have demonstrated promising results in modeling sequential and temporal data [Rala Cordeiro et al. 2021]. Nevertheless, physiological signals such as electrocardiograms (ECG) and electroencephalograms (EEG) exhibit complex temporal dependencies and long-range dynamical patterns that are not always effectively captured by purely convolutional architectures [Freitas et al. 2022]. Accurately modeling these dynamics is particularly important in healthcare applications, where subtle temporal variations may correspond to clinically relevant events.

To address these limitations, *Reservoir Computing* (RC) has gained attention as a computational framework specifically designed to model temporal dynamics in sequential data [Zhang and Vargas 2023]. In particular, *Echo State Networks* (ESNs) provide an efficient recurrent architecture in which only the readout layer is trained, while the recurrent reservoir remains fixed. This design allows ESNs to capture rich temporal dynamics with significantly lower training complexity compared to fully trained recurrent neural networks.

Despite these advantages, reservoir-based models are rarely considered within NAS search spaces, which are typically dominated by convolutional or transformer-based operations. As a result, NAS frameworks may overlook architectural configurations capable of effectively modeling the temporal dynamics inherent to biomedical signals. Integrating reservoir computing modules into the NAS search space may therefore enable the automated discovery of hybrid architectures that combine the representation power of deep learning with the temporal modeling capabilities of dynamical systems. While the CNN is responsible for capturing local morphology, the RC is used to capture temporal dynamics.

Motivated by this perspective, this work investigates whether incorporating a reservoir computing operation (specifically an Echo State Network) into the search space of GeneticNAS can improve automated architecture design for one-dimensional biomedical signal classification. We extend GeneticNAS to operate on architectures tailored to sequential signals and introduce ESN-based reservoir modules as candidate operations within the search process. This raises the following research question: *Can the integration of a Reservoir Computing (specific ESN) operator within a Neural Architecture Search framework improve the representation of temporal dynamics in biomedical signals, thereby leading to improved classification performance measured by accuracy and F1-score?*

The proposed framework is evaluated on two biomedical datasets with distinct statistical characteristics. The first is the *MIT-BIH Arrhythmia* dataset [Moody and Mark 2001], a widely used benchmark for ECG-based arrhythmia detection, characterized by severe class imbalance. The second is an EEG-based biometric identification dataset from *PhysioNet EEG* [Goldberger et al. 2000], which exhibits a more balanced class distribution. These complementary scenarios enable the evaluation of the proposed approach under different signal characteristics and classification regimes. Experimental results demonstrate that architectures discovered using the proposed hybrid search space achieve consistent improvements across multiple evaluation metrics, including accuracy, precision, recall, and F1-score.

The main contributions of this work are summarized as follows: **(i)** we extend GeneticNAS to support architecture search for one-dimensional biomedical signals; **(ii)** we incorporate reservoir computing modules, specifically Echo State Networks, as candidate operations within the NAS search space. **(iii)** we enable the automated discovery of hybrid deep–reservoir architectures that capture both local signal morphology and temporal dynamics; and **(iv)** we evaluate the proposed framework on ECG arrhythmia detection and EEG-based biometric identification, demonstrating improvements over purely convolutional architectures.

The remainder of this paper is organized as follows. Section 2 reviews the theoretical background and related work. Section 3 describes the proposed methodology and implementation details of the hybrid NAS framework. Section 4 presents the experimental setup and discusses the obtained results. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Related Work

Evolutionary strategies have become a prominent approach for Neural Architecture Search. Methods such as *Evolutionary Neural Architecture Search-Rep* (EvoNAS-Rep) [Wen et al. 2022] employ genetic algorithms to jointly optimize architectural configurations and training dynamics, achieving competitive performance while reducing computational cost. Multi-objective evolutionary frameworks further extend this paradigm by balancing predictive performance and architectural complexity. For instance, *NSGA-Net* [Lu et al. 2019] uses the NSGA-II algorithm to jointly optimize classification error and model size, while hardware-aware approaches such as THNAS-GA [Thanh et al. 2024] incorporate training-free performance estimation to discover efficient architectures under computational constraints.

To improve the scalability of NAS, weight-sharing strategies have been widely adopted. *Efficient Neural Architecture Search* (ENAS) [Pham et al. 2018] significantly reduces search cost by forcing candidate architectures to share parameters within a supernet trained through reinforcement learning. Similarly, *GeneticNAS* [Habi and Rafalovich 2019] integrates parameter sharing into an evolutionary framework, enabling efficient exploration of large architectural spaces. To address biases introduced by supernet training, approaches such as *FairNAS* [Chu et al. 2021] introduce fairness constraints that ensure candidate architectures receive balanced optimization during the search process.

Although NAS research has primarily focused on computer vision tasks, several studies have explored its application to time-series modeling. For example, the *Evolutionary Multivariate Time Series Forecasting* (EMTSF) framework [Liang and Sun 2024] automatically evolves spatial and temporal modules for forecasting tasks, outperforming manually designed architectures. Similarly, NAS-T [Rakhshani et al. 2020] proposes a NAS framework for time-series classification using ResNet-based models combined with cross-validation and weight sharing to improve robustness in limited-data settings.

Within biomedical signal processing, NAS has shown promise for tasks involving ECG and EEG signals. Cordeiro *et al.* [Rala Cordeiro et al. 2021] evaluated several NAS strategies for 1D convolutional networks applied to ECG classification and found genetic algorithms to be among the most consistent search methods. In EEG-based applications, Li *et al.* [Li et al. 2022] proposed *Transformer Neural Architecture Search* (TNAS), which combines multi-objective evolution and weight sharing to optimize Transformer-based architectures for EEG biometrics and emotion recognition.

Despite these advances, the integration of NAS with Reservoir Computing (RC) remains relatively underexplored. Existing efforts include automated architecture design for Liquid State Machines [Tian et al. 2021] and surrogate-assisted reservoir search methods [Zhou et al. 2023]. However, most NAS studies still focus on purely convolutional or LSTM architectures. In this work, we address this gap by incorporating Echo State Network (ESN) operations into an evolutionary NAS framework based on GeneticNAS, combining 1D convolutional feature extraction with efficient temporal modeling for biomedical signals such as ECG and EEG.

3. Proposed Method

This section describes the proposed NAS framework for automatically discovering hybrid deep–reservoir architectures for biomedical signal classification. The framework combines evolutionary NAS with Reservoir Computing operators, enabling the search process to jointly optimize convolutional feature extraction and dynamical temporal modeling.

3.1. Datasets and Preprocessing

Experiments were conducted on two biomedical datasets composed of one-dimensional physiological signals with distinct statistical properties: *MIT-BIH Arrhythmia Database* [Moody and Mark 2001] and *PhysioNet EEG Database* [Goldberger et al. 2000].

The *MIT-BIH Arrhythmia Database* [Moody and Mark 2001] is a widely used benchmark for electrocardiogram (ECG) analysis, containing 48 half-hour recordings sampled at 360 Hz. Following the AAMI2 standard [Llamedo and Martínez 2011], heartbeat annotations were grouped into clinically relevant categories. In this study, the unknown beat class (Q) was excluded, and the remaining four classes were considered: normal beat (N), supraventricular ectopic beat (SVEB or S), ventricular ectopic beat (VEB or V), and fusion beat (F), representing a fusion of ventricular and normal beats. The dataset is highly imbalanced, with the N class accounting for 89.46% of all heartbeats.

To prevent patient-level data leakage, we adopted the patient-oriented split proposed in [De Chazal et al. 2004]. The DS1 subset was used for training (with 10% reserved for validation), while DS2 was used exclusively for testing. Each sample corresponds to a 360-point window (1 second) centered on the R-peak of the heartbeat signal.

The *PhysioNet EEG Database* [Goldberger et al. 2000] is used for EEG-based biometric identification. The experiments focus on eyes-open and eyes-closed sessions within the gamma frequency band (30–50 Hz), which has been shown to contain discriminative neural patterns associated with the identity of 109 subjects.

Following the preprocessing strategy proposed by Freitas *et al.* [Freitas et al. 2022], EEG recordings were segmented into temporal windows of 1920 samples (corresponding to 12 seconds). For the training set, a sliding-window segmentation strategy with a stride of 30 samples was adopted. In this configuration, the first segment starts at the first sample of the signal, the second at the 31st sample, and subsequent segments follow the same pattern. As a result, consecutive windows exhibit a high degree of overlap (1890 samples between adjacent windows), which substantially increases the number of training samples and acts as a form of data augmentation. For the test set, non-overlapping segmentation was employed to ensure an unbiased evaluation. Specifically, the first segment begins at the first sample of the signal, while the second segment starts at sample 1921, corresponding to the next 12-second interval.

Signals from both datasets were standardized using z-score normalization based on the mean (μ) and standard deviation (σ) estimated from the training set. This normalization improves numerical stability during training and prevents information leakage from validation and test partitions. It is calculated as:

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

3.2. Neural Architecture Search Framework

The structural optimization of neural networks was conducted using *GeneticNAS* [Habi and Rafalovich 2019], adapted for one-dimensional signal processing. Candidate architectures are represented as directed acyclic graphs (DAGs), where each node corresponds to a computational operation and edges represent information flow between nodes.

Let x_i denote the output of node i. The output of node j is defined as:

$$x_j = \sum_{i<j} o_{i,j}(x_i) \tag{2}$$

where $o_{i,j} \in \mathcal{O}$ represents an operation selected from the search space $\mathcal{O}$.

The search space includes standard operations commonly used in temporal deep learning architectures, as well as reservoir-based operator introduced in this work:

$$\mathcal{O} = \begin{cases} 1 & \text{, Conv1D with } 3 \times 3 \text{ kernel} \\ 2 & , Identity \\ 3 & \text{, Conv1D with } 5 \times 5 \text{ kernel} \\ 4 & \text{, AveragePool} \\ 5 & \text{, MaxPool} \\ 6 & \text{, Reservoir} \end{cases}. \tag{3}$$

The Reservoir operator can be instantiated either as a standalone reservoir module or as part of a hybrid operator combining the reservoir with additional neural layers.

The NAS procedure follows a two-stage optimization strategy:

1. **Architecture Search.** An evolutionary process explores the architectural search space for 100 epochs. Genetic operators such as mutation and crossover generate new candidate architectures, while selection is performed according to validation performance. To reduce computational cost, candidate architectures share weights through a supernet structure, allowing multiple architectures to reuse parameters during training.
2. **Architecture Retraining.** After the search stage, the best-performing architecture is selected and trained from scratch for 60 epochs using the full training dataset. This retraining stage removes potential biases introduced by weight sharing and ensures that the final architecture reaches its full representational capacity.

3.3. Reservoir Computing Integration

To explicitly model temporal dynamics inherent to physiological signals, the NAS search space was extended to include operators based on the RC paradigm. In particular, we introduce an *Echo State Network* (ESN) operator that can be selected by the evolutionary search.

An ESN consists of a fixed recurrent reservoir that projects the input signal into a high-dimensional dynamical space. The reservoir state is updated according to:

$$\mathbf{x}(t+1) = (1-\alpha)\mathbf{x}(t) + \alpha \tanh(W_{in}u(t+1) + W\mathbf{x}(t)), \tag{4}$$

where $u(t)$ represents the input signal, $\mathbf{x}(t)$ denotes the reservoir state, W_{in} is the input weight matrix, W is the recurrent reservoir matrix, and α is the leaking rate that controls the update dynamics.

The recurrent matrix W is initialized as a sparse random matrix and scaled to satisfy the *Echo State Property* (ESP), which ensures stable reservoir dynamics. This is typically enforced by constraining the spectral radius $\rho(W)$:

$$\rho(W) < 1 \tag{5}$$

The reservoir produces a high-dimensional state representation that captures temporal dependencies without requiring gradient-based training of the recurrent weights. A trainable linear readout layer then maps the reservoir states to the task-specific output:

$$y(t) = W_{out}\mathbf{x}(t). \tag{6}$$

As the reservoir state size is an important aspect, we empirically evaluate different sizes.

3.3.1. Hybrid Reservoir Operations

Based on the ESN operator, three variants were incorporated into the NAS search space: a single Reservoir, a Reservoir followed by a CNN layer (Reservoir-CNN), and a Reservoir followed by an LSTM layer (Reservoir-LSTM). The objective of these configurations is to investigate whether enforcing an additional temporal or spatial processing stage after the reservoir can improve the overall model performance. They can be described as:

- **Reservoir:** A standalone ESN module that directly processes the input signal and produces temporal representations.
- **Reservoir-CNN:** A hybrid operation where the reservoir module is followed by a single convolutional layer. This configuration allows the model to capture temporal dynamics while simultaneously extracting local morphological patterns in the signal. The convolutional layer uses a 3×3 kernel with padding, preserving the dimensionality of the reservoir representation.
- **Reservoir-LSTM.** A hybrid operation that feeds reservoir states into an LSTM layer, enabling the modeling of long-range dependencies over the high-dimensional reservoir state space. The LSTM layer is defined with the same input and output dimensionalities as the reservoir state vector.

By incorporating these operations into the NAS search space, the evolutionary process can automatically discover architectures that combine convolutional feature extraction with dynamical temporal modeling.

3.4. Evaluation Protocol

The final architectures discovered by the NAS procedure were evaluated strictly on previously unseen test sets. Due to the severe class imbalance present in the MIT-BIH Arrhythmia dataset, model performance was assessed using multiple evaluation metrics rather than relying solely on accuracy. Specifically, the following metrics were used: (i) **Accuracy**: Proportion of correctly classified samples among all predictions; (ii) **Precision**: Proportion of true positive predictions among all predicted positives; (iii) **Recall**: Proportion of actual positives correctly identified by the model; and (iv) **F1-score**: Harmonic mean of precision and recall, providing a balanced evaluation for imbalanced datasets.

These metrics provide a more reliable assessment of model performance in imbalanced clinical datasets, where accurate detection of minority arrhythmia classes is particularly important for practical medical applications. The source code is available on the GitHub[1] platform.

4. Experiment and Results

This section describes the experiments and results reported in this work. All experiments were conducted on a workstation equipped with an *Intel i9-10900* processor featuring 10 physical cores (20 threads) running at 2.80 GHz, 128 GB of DDR4 RAM, and an NVIDIA *RTX 3090* GPU with 24 GB of GDDR6X memory.

The experimental environment supported GPU acceleration through *CUDA* version 12.4. The models were implemented in *Python* (version 3.10) using the *PyTorch* deep learning framework (version 2.6.0), together with the *TorchVision* library (version 0.21.0). Additionally, the *Scikit-learn* library was employed to compute the evaluation metrics used throughout the experiments.

Each NAS experiment was repeated three times with different random seeds, and the reported results correspond to the mean and standard deviation across runs.

4.1. Hyperparameters used

The same hyperparameter configuration was used for both datasets. During the architecture search phase, models were trained for 100 epochs with a batch size of 128 for training and 1000 for validation. Optimization employed Stochastic Gradient Descent (SGD) with momentum 0.9 and weight decay 10^{-4}. The learning rate was initialized at 0.1 and decayed to a minimum of 0.0001 using a *MultiStepLR* scheduler. Regularization included a dropout rate of 0.2 and a *drop path* mechanism activated from epoch 50 with a probability of 1.0.

Following the parameters used in [Habi and Rafalovich 2019], the evolutionary search procedure used a genetic algorithm with a population size of 20 candidate architectures and two generations evaluated per epoch. The mutation probability was set to 0.02, while crossover was applied with a probability of 1.0 using a *block-based* crossover

---

[1] `https://github.com/BarbaraLeticiaRM/GeneticNAS-Reservoir-.git`

strategy. The architectural search space consisted of two computational cells, each containing five internal nodes, six possible cell types, and convolutional operations with 20 feature channels.

For the final weight optimization phase, selected architectures were trained for 60 epochs using the same batch sizes (128 for training and 1000 for validation). The optimizer remained SGD with momentum 0.9 and weight decay 10^{-4}, while the learning rate started at 0.05 and decayed to 0.0001 via *MultiStepLR*. Regularization included a dropout of 0.2 and a *drop path* probability of 0.9 applied from the beginning of training.

4.2. Reservoir Hyperparameter Calibration on MIT-BIH

Initial experiments focused on calibrating the *Reservoir Computing* operator using the MIT-BIH Arrhythmia dataset. In particular, we investigated the influence of the reservoir dimensionality and the sparsity level of the recurrent matrix on classification performance. As the input signal is a 360-point, Reservoir sizes of $\{180, 360, 540, 1000\}$ neurons were evaluated, combined with sparsity levels of 5% and 10% for the recurrent weight matrix.

Among the evaluated configurations, a reservoir size of 360 neurons with 5% sparsity achieved the best empirical performance, reaching an accuracy of 87.17% and an F1-score of 0.87. This configuration outperformed both the baseline GeneticNAS architecture without reservoir operators and the proposed hybrid configurations that combine the reservoir with additional temporal or spatial modules (Reservoir-LSTM and Reservoir-CNN). The average results across the experimental runs are summarized in Table 1.

Table 1. Average performance obtained across experimental runs on the MIT-BIH dataset. Best results are highlighted in bold.

Layers			Accuracy (%)	Precision (%)	Recall (%)	F1-score
Reservoir	LSTM	CNN				
×	×	×	84.643 ± 2.477	88.557 ± 0.681	84.643 ± 2.477	0.85557 ± 0.01485
✓(180)	×	×	85.630 ± 0.795	**89.207 ± 0.917**	85.630 ± 0.795	0.86873 ± 0.00330
✓(360)	×	×	**87.173 ± 1.684**	89.073 ± 0.871	**87.173 ± 1.684**	**0.87787 ± 0.01157**
✓(540)	×	×	85.050 ± 1.669	88.723 ± 0.090	85.950 ± 3.217	0.86393 ± 0.01066
✓(1000)	×	×	82.303 ± 2.758	87.957 ± 0.549	82.303 ± 2.758	0.84313 ± 0.01894
✓(360)	✓	×	86.030 ± 2.556	89.013 ± 0.876	86.030 ± 2.556	0.87027 ± 0.01708
✓(360)	×	✓	84.027 ± 2.945	88.267 ± 0.647	84.027 ± 2.945	0.85380 ± 0.01936
×	✓	×	84.803 ± 3.314	88.203 ± 0.535	84.803 ± 3.314	0.85777 ± 0.02237

Although the inclusion of the reservoir operator consistently improved the average values of the evaluation metrics and reduced the variance across runs—suggesting improved model stability. The statistical tests performed on MIT-BIH did not reveal significant differences (t-Student and $p > 0.05$). This behavior can be partially attributed to the severe class imbalance inherent in arrhythmia datasets, where the dominance of normal beats can mask improvements in minority classes when aggregated metrics are considered.

4.3. Evaluation on the PhysioNet EEG Dataset

To further evaluate the robustness of the proposed hybrid NAS framework under a more balanced classification scenario, the discovered architectures were also evaluated on the PhysioNet EEG biometric identification dataset.

In this context, we explored the impact of scaling the reservoir dimensionality to match the signal window size. Specifically, a reservoir with 1920 units—corresponding to the EEG window length—was evaluated in addition to the previously tested configuration with 360 units. This configuration produced a substantial improvement in predictive performance. The classification accuracy increased from 49.29% for the baseline GeneticNAS architecture to 54.68%, representing a statistically significant gain ($p < 0.05$). The detailed results are reported in Table 2. As the Hybrid models did not perform, we did not test them for EEG scenario.

Table 2. Average performance obtained on the PhysioNet EEG dataset. Best results are highlighted in bold.

Reservoir Size	Accuracy (%)	Precision (%)	Recall (%)	F1-score
×	49.297 ± 2.781	44.870 ± 3.191	49.297 ± 2.781	0.42997 ± 0.02837
✓(360)	46.973 ± 0.318	41.723 ± 1.785	46.940 ± 0.260	0.40347 ± 0.00095
✓(1920)	**54.680 ± 1.200**	**50.997 ± 2.769**	**54.680 ± 1.200**	**0.48250 ± 0.01689**

Across all experiments, incorporating the ESN operator into the NAS search space consistently improved model performance. These improvements were statistically significant according to a Student's t-test ($p < 0.05$).

4.4. Architecture Analysis

The architecture corresponding to the best-performing run, with a reservoir size of 360 neurons for the MIT dataset, is illustrated in Figure 1.

An inspection of the directed acyclic graphs generated during the evolutionary search revealed consistent structural patterns. GeneticNAS tends to place reservoir operations in the early stages of both Normal and Reduction cells. This positioning allows the reservoir to capture temporal dynamics directly from the input signal before deeper convolutional processing occurs. In contrast, reservoir operators appearing in the Input cell are typically located toward later stages, where they operate on higher-level feature representations.

4.5. Computational Cost Analysis

The integration of the reservoir operator increased training time compared to the baseline GeneticNAS architecture. This overhead primarily arises because the reservoir projects the input signal into a high-dimensional dynamical space, increasing the number of matrix operations within each neural cell. The effect becomes more pronounced for larger reservoirs, such as the configuration with 1920 units used in the EEG experiments.

In addition, the recurrent dynamics of the reservoir require sequential state updates, which limit GPU parallelism. Unlike convolutional operations that can be executed with high parallel efficiency, reservoir state propagation introduces temporal dependencies that reduce computational throughput. Although sparse recurrent matrices were used to construct the reservoirs, they were implemented using dense tensor representations, preventing the model from benefiting from sparse computation optimizations.

Despite this additional computational cost, experiments on the PhysioNet EEG dataset show that reservoir-enhanced architectures achieve statistically significant im-

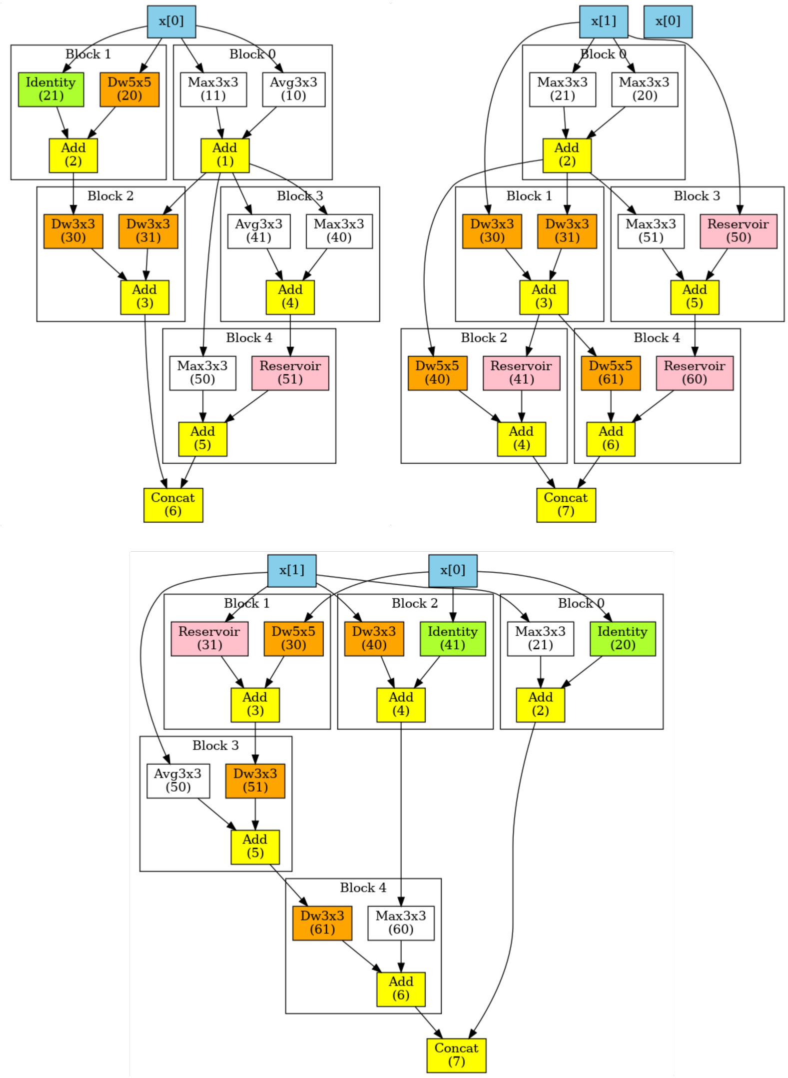

Figure 1. Best architecture discovered across three runs using a reservoir size of 360 neurons. From left to right: Input Cell, Normal Cell, and Reduction Cell.

provements in classification accuracy. Moreover, these models exhibit lower performance variance across runs, suggesting improved robustness to random weight initialization.

5. Conclusion

This work investigated whether incorporating a *Reservoir Computing* (RC) operator into the GeneticNAS search space improves the classification of one-dimensional biomedical signals. The results confirm that RC-enhanced search consistently produces architectures that outperform the standard GeneticNAS baseline, indicating that ESN-style reservoirs constitute an effective inductive bias for physiological time-series modeling.

A key finding is that proper reservoir calibration is critical. For ECG classification, a reservoir with 360 units—aligned with a 1-second analysis window—and high sparsity (95%) provided the best representational trade-off. This configuration outperformed both larger reservoirs and alternative hybrid operators based on LSTM or CNN layers. Across two datasets with distinct characteristics—the highly imbalanced MIT-BIH Arrhythmia dataset and a more balanced PhysioNet-EEG biometric dataset—the resulting RC-based architectures (with reservoir sizes 360 and 1920, respectively) consistently surpassed the vanilla GeneticNAS. These results highlight the potential of RC as a search-space primitive for NAS applied to biomedical time series.

This study focused empirically on reservoir size and sparsity. Future work includes: (i) extending validation to additional medical time-series datasets; (ii) incorporating reservoir-specific parameters, such as spectral radius and *leak rate*, directly into the genetic search space; (iii) exploring multi-objective evolutionary methods (e.g., NSGA-II) to jointly optimize predictive performance and computational cost for deployment-oriented settings; and (iv) exploring the impact of the reservoir on the number of model parameters and, consequently, the FLOPs and inference latency of the searched model.

Acknowledgments

The authors acknowledge the support of the *Fundação de Amparo à Pesquisa do Estado de Minas Gerais* (FAPEMIG, project APQ-01768-24), the *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq), the C*oordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES), the *Universidade Federal de Ouro Preto* (PROPPI/UFOP), and its Graduate Program in Computer Science (PPGCC/UFOP).

References

Amberkar, A., Awasarmol, P., Deshmukh, G., and Dave, P. (2018). Speech recognition using recurrent neural networks. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pages 1–4.

Chu, X., Zhang, B., and Xu, R. (2021). Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. In *Proceedings of the IEEE/CVF International Conference on computer vision*, pages 12239–12248.

De Chazal, P., O'Dwyer, M., and Reilly, R. B. (2004). Automatic classification of heartbeats using ecg morphology and heartbeat interval features. *IEEE transactions on biomedical engineering*, 51(7):1196–1206.

Freitas, C., Silva, P., Moreira, G., and Luz, E. (2022). Rede neural convolucional e lstm para biometria baseada em eeg no modo de identificaçao. In *Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS)*, pages 256–267. SBC.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23):e215–e220.

Habi, H. V. and Rafalovich, G. (2019). Genetic network architecture search. *arXiv preprint arXiv:1907.02871*.

Li, C., Zhang, Z., Zhang, X., Huang, G., Liu, Y., and Chen, X. (2022). Eeg-based emotion recognition via transformer neural architecture search. *IEEE transactions on industrial informatics*, 19(4):6016–6025.

Liang, Z. and Sun, Y. (2024). Evolutionary neural architecture search for multivariate time series forecasting. In *Asian Conference on Machine Learning*, pages 771–786. PMLR.

Llamedo, M. and Martínez, J. P. (2011). Heartbeat classification using feature selection driven by database generalization criteria. *IEEE Transactions on Biomedical Engineering*, 58(3):616–625.

Lu, Z., Whalen, I., Boddeti, V., Dhebar, Y., Deb, K., Goodman, E., and Banzhaf, W. (2019). Nsganet: neural architecture search using multi-objective genetic algorithm. In *Proceedings of the genetic and evolutionary computation conference*, pages 419–427.

Moody, G. B. and Mark, R. G. (2001). The impact of the mit-bih arrhythmia database. *IEEE engineering in medicine and biology magazine*, 20(3):45–50.

Pak, M. and Kim, S. (2017). A review of deep learning in image recognition. In *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, pages 1–3.

Pham, H., Guan, M., Zoph, B., Le, Q., and Dean, J. (2018). Efficient neural architecture search via parameters sharing. In *International conference on machine learning*, pages 4095–4104. PMLR.

Rakhshani, H., Fawaz, H. I., Idoumghar, L., Forestier, G., Lepagnot, J., Weber, J., Brévilliers, M., and Muller, P.-A. (2020). Neural architecture search for time series classification. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Rala Cordeiro, J., Raimundo, A., Postolache, O., and Sebastião, P. (2021). Neural architecture search for 1D CNNs—different approaches tests and measurements. *Sensors*, 21(23):7990.

Real, E., Aggarwal, A., Huang, Y., and Le, Q. V. (2019). Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Chen, X., and Wang, X. (2021). A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys (CSUR)*, 54(4):1–34.

Shin, R., Packer, C., and Song, D. (2018). Differentiable neural network architecture search.

Singh, S. P., Kumar, A., Darbari, H., Singh, L., Rastogi, A., and Jain, S. (2017). Machine translation using deep learning: An overview. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pages 162–167.

Thanh, T. H., Doan, L., Luong, N. H., and Huynh Thi Thanh, B. (2024). Thnas-ga: A genetic algorithm for training-free hardware-aware neural architecture search. In *Proceedings Of The Genetic And Evolutionary Computation Conference*, pages 1128–1136.

Tian, S., Qu, L., Wang, L., Hu, K., Li, N., and Xu, W. (2021). A neural architecture search based framework for liquid state machine design. *Neurocomputing*, 443:174–182.

Wen, L., Gao, L., Li, X., and Li, H. (2022). A new genetic algorithm based evolutionary neural architecture search for image classification. *Swarm and Evolutionary Computation*, 75:101191.

Wistuba, M., Rawat, A., and Pedapati, T. (2019). A survey on neural architecture search. *arXiv preprint arXiv:1905.01392*.

Zhang, H. and Vargas, D. V. (2023). A survey on reservoir computing and its interdisciplinary applications beyond traditional machine learning. *IEEE Access*, 11:81033–81070.

Zhou, Y., Jin, Y., Sun, Y., and Ding, J. (2023). Surrogate-assisted cooperative co-evolutionary reservoir architecture search for liquid state machines. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(5):1484–1498.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.