ISSN: 0067-2904

A Novel Estimation of Tree Length Using Neural Network Approaches in Phylogenetic Analysis

Osama A. Salman*, Gábor Hosszú
Department of Electron Devices, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, 1111 Budapest, Műegyetem rkp. 3., Hungary

Received: 4/1/2025 Accepted: 7/ 4/2025 Published: 30/4/2026

Abstract

The article studies the integration of artificial neural networks (ANNs) in the phylogenetic analysis of scriptinformatics, focusing on the historical evolution of Arabic, Aramaic, and Middle Iranian scripts. By treating scripts as taxa, pattern systems are exploited to delineate evolutionary trajectories, utilizing an optimised feature selection method and neural network architectures fitnet and feedforwardnet. The presented approach leverages publicly accessible genetic sequence datasets and applies comprehensive preprocessing, including a novel feature extraction process, normalisation, and cross-validation, to predict phylogenetic tree lengths. The results indicate a superior performance of the feedforward neural network, particularly with an architecture of 16 nodes in the first layer and 6 in the second. In machine learning hyperparameters such as learning rate and network size control the training process and affect model performance. Reduced computational time is significantly achieved through meticulous optimisation of hyperparameters without exhaustive phylogenetic analysis. The cross-validation process underlines the model's robust predictive capacity, paving the way for advanced computational tools in scriptinformatics and evolutionary studies.

Keywords: Neural network, Feature selection, Pattern system, Phylogenetic, Scriptinformatics, Machine learning, Hyperparameter.

تقدير جديد لطول الشجرة باستخدام منهجيات الشبكات العصبية في التحليل التطوري

اسامه علي سلمان*, غابور هوسزو
قسم أجهزة الإلكترونيات، كلية الهندسة الكهربائية والمعلوماتية، جامعة بودابست للتكنولوجيا والاقتصاد

الخلاصة

تتناول هذه الدراسة دمج الشبكات العصبية الاصطناعية (ANNs) في التحليل الوراثي التطوري ضمن مجال المعلوماتية الخطية، مع التركيز على التطور التاريخي للخطوط العربية والآرامية والإيرانية الوسطى. من خلال معاملة الخطوط كنظم تصنيفية (Taxa)، يتم استغلال الأنماط الخطية لتحديد المسارات التطورية باستخدام طريقة محسّنة لاختيار الميزات وهندسات شبكية تعتمد على FitNet وFeedforwardNet. تعتمد المنهجية المقترحة على مجموعات بيانات وراثية متاحة للجمهور، مع تطبيق معالجة مسبقة شاملة تتضمن استخراج الميزات، التطبيع، والتحقق المتقاطع بهدف تقدير أطوال الأشجار التطورية بدقة. أظهرت النتائج تفوق الشبكة العصبية Feedforward، لا سيما عند استخدام بنية تحتوي على 16 عقدة في الطبقة الأولى

*Email: osamaalisalman.khafajy@edu.bme.hu

و6 عقد في الطبقة الثانية. في التعلم الآلي، تعد المعلمات الفائقة (Hyperparameters)، مثل معدل التعلم وحجم الشبكة، عناصر حاسمة تتحكم في عملية التدريب وتؤثر على أداء النموذج. تم تحقيق تقليل كبير في زمن الحساب من خلال تحسين دقيق للمعلمات الفائقة، دون الحاجة إلى تحليل فيلوجيني شامل. يؤكد التحقق المتقاطع على قوة القدرة التنبؤية للنموذج، مما يمهد الطريق لتطوير أدوات حاسوبية متقدمة في دراسة تطور الخطوط والمجالات التطورية الأخرى.

1. Introduction

Systematics has expanded from biology to scriptinformatics, which uses evolutionary modelling and computer science to unravel the historical evolution of scripts. This field views scripts as analogous to living taxa, facilitating a systematic study of their development and cultural spread [1-7].

Advancements in computational methods, mainly feature selection and machine learning, have notably improved phylogenetic inference accuracy. The complexity of high-dimensional data in phylogenetics calls for advanced algorithms to select informative features that enhance model efficacy and deepen evolutionary insights [2-3, 5-7].

Despite advances in computational phylogenetic analysis, the time-consuming and costly process of tree construction, particularly for calculating Maximum Parsimony scores, remains a challenge. Traditional methods struggle with large and complex datasets, leading to bottlenecks in evolutionary biology research [8, 9]. Moreover, the difficulty of finding informative features from large datasets is further compounded by the frequent need to reconstruct trees in order to evaluate different feature subsets, making the process increasingly impractical as data sizes grow [5, 8].

The use of complete or nearly complete DNA, RNA, or even any genetic sequences can potentially make a more robust and thorough representation of the evolutionary dynamics that have shaped species diversity, as noted by [10]. However, such massive datasets may compromise the reliability of these evolutionary inferences. The effects of minor errors can be drastically amplified due to the greater complexity of the analysis, and noise and error in accumulation will distort the results and misrepresent evolutionary relationships. The most evident source of concern arises from unaccounted variability in how evolutionary dynamics change over time, be it across different genetic sites, specific genes, or unique lineages as pointed out in [11]. Neural networks can transform the direct prediction of tree lengths from datasets, thereby bypassing extensive phylogenetic analysis. The presented approach reduces computational time by removing the repetitive steps of tree construction and scoring, increasing the efficiency of phylogenetic studies [1,2].

Equally, Neural networks obviate feature selection and allow for fast assessment of the impact of features on predicted tree lengths. However, its capacity in handling complicated events such as duplication, loss, and gene flow in large-scale phylogenetic studies remains underexploited. Our application tackles these issues [12].

Furthermore, in scriptinformatics, scripts are analysed as pattern systems with binary features for the exploration of their developmental path. This approach has shown the importance of systems like Morse code and early writing systems in the development of written communication [2]. A very good example is the review article, which demonstrates the usefulness of taxonomy on graph neural networks and provides a broad overview of research in this area [13].

The field of scriptinformatics applies computational advances to the study of the evolution of writing systems, casting them in terms of frameworks defined by symbols, syntax, and structural rules. This approach allows for the analysis of characteristics and evolutionary paths of the different scripts, hence offering insights into how writing systems evolve alongside the cultures that create them [2-3, 5-7].

In consideration of these advancements, the current study focuses on the evolutionary analysis of ancient scripts by using state-of-the-art feature selection methods to rebuild phylogenetic trees that model different script variants. By examining Arabic, Aramaic, and Middle Iranian scripts, we aim to reveal their evolutionary interrelations and to contribute to the development of scriptinformatics. We exploit a dataset available publicly, thus facilitating transparency and promoting further research in this growing field [2-3, 5-7].

Besides advancing scriptinformatics, our research could have practical applications in archaeology. Over time, this method could help decode the evolution of scripts, aiding in the interpretation of previously undeciphered inscriptions discovered by archaeologists.

This study explores the evolution of the Arabic, Aramaic, and Middle Iranian writing systems, taking them as different pattern systems to reconstruct their phylogenetic trees based on data extracted from GitHub [14]. This paper, by attempting to unveil deep evolutionary relationships, furthers the field of scriptinformatics and brings new knowledge of the historical development of these writing systems.

This paper is organized as follows: Background explains the application of artificial neural networks in phylogenetic reconstruction; Method describes our methods; Results presents our findings; and Conclusions discusses the implications of these results.

2. Background

Phylogenetics is a core part of molecular biology and evolutionary studies. However, using phylogenetic trees to depict evolutionary relationships and emphasizing the importance of accurately portraying ancestral lineages and diversification [15]. Traditional methods, including multiple sequence alignment (MSA) and tree-building algorithms, are challenged by dealing with large datasets and biases introduced by evolutionary differences [15,16]. These challenges bring into prominence the need for new methods to handle the complexity and size of genomic data.

Neural networks afford exciting opportunities to solve phylogenetic challenges, such as tree topology inference, branch length estimation, and model selection using machine learning [12,16]. However, several obstacles toward supervised ML in phylogenetics include generating realistic training data and adapting to various biological data [12].

We apply neural networks, such as CNNs and MLPs, to construct phylogenetic trees, emphasizing the potential for ML to surpass traditional methodologies. Despite certain testing limitations, the approach promises increased accuracy and efficiency in the inference of the phylogenies and may also uncover broader insights into evolutionary histories beyond scriptinformatics.

Artificial neural networks (ANNs) are models mimicking biological neurons with interconnected nodes organized into layers. The topology of the nodes determines the performance. The various connection strategies yield different ANNs with different structural properties. A comprehensive review of the methodologies and applications of ANNs is given in [23], with their computational modelling and data analysis usage described in [17].

In addition, the hyperparameters of ANN are model structure and model training parameters with a strong influence on the model's performance and accuracy [18].

$Length_{PAUP}$ is the most parsimonious tree length derived using the PAUP* software. The length indicates the smallest number of mutations needed to explain taxa differences and serves as a point of reference for the evaluation of the ANN4P approach. PAUP* is a program for the computation of evolutionary trees [19].

We conducted a Maximum Parsimony analysis, efficiently finding the optimal tree by discarding non-optimal branches. This method prunes the search space to ensure accurate and thorough results. The exhaustive search allowed us to calculate parsimony scores and generate maximum parsimony trees, comprehensively evaluating our method's performance [20].

3. Method

The research aims to predict the phylogenetic tree length, specifically the Maximum Parsimony score or, in other words, tree length, by utilising Artificial Neural Networks [21] to process binary genetic sequences without traditional tree construction methods. Usually, phylogenetics analysis is finding or constructing the phylogenetic tree. The amount of those trees depends on the number of taxa that have been considered, where the number of rooted, bifurcating trees, according to [22] can be given by formal (1).

$$\frac{(2n-3)!}{2^{n-2}(n-n)!} \tag{1}$$

Figure 1 provides a visual representation of our ANN4P approach. The process begins with the collection of binary genetic sequences representing various taxa. These sequences undergo a comprehensive preprocessing phase, critical for preparing the data for ANN input.

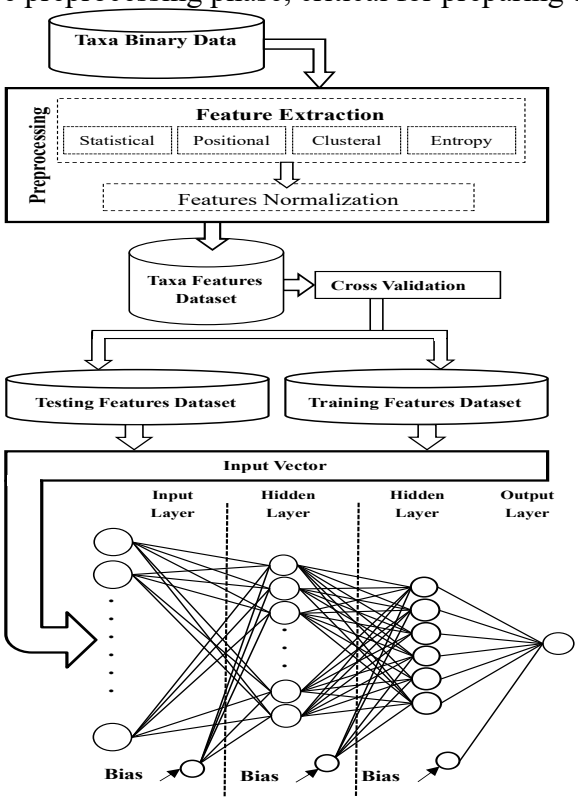

Figure 1: Flowchart of the ANN4P phylogenetic analysis using neural network

Our study extracts features from binary datasets to analyse phylogenetic tree lengths. The datasets are composed of binary sequences of features, where each sequence corresponds to a taxon, and each position in the sequence represents a specific phylogenetic feature. The following notations are used in the feature extraction process.

$\boldsymbol{s_{ij}}$: The value of the j^{th} position (feature) in the i^{th} Taxon's sequence of features.

$\boldsymbol{m}$: The total number of features in a taxon's sequence.

$\boldsymbol{C_{1i}, C_{0i}}$: Total amount of '1's and '0's, respectively, in the i^{th} sequence.
$\boldsymbol{D_{1i}, D_{0i}}$: Density of '1's and '0's, respectively, in the i^{th} sequence.
$\boldsymbol{P_{F1i}, P_{L1i}, P_{F0i}, P_{L0i}}$: Position of the first and last occurrence of '1's and '0's, respectively, in the i^{th} Taxon's sequence.
$\boldsymbol{K_i}$: The number of contiguous clusters of '1's in the i^{th} Taxon's sequence.
$\boldsymbol{\tilde{x}_{1i}, \tilde{x}_{0i}}$: The median position of '1's and '0's, respectively, in the i^{th} Taxon's sequence.
$\boldsymbol{\sigma^2_{1i}, \sigma^2_{0i}}$: The variance of positions of '1's and '0's, respectively, in the i^{th} Taxon's sequence.
$\boldsymbol{E_i}$: The Shannon entropy for the i^{th} Taxon is a measure of randomness in the sequence.
$\boldsymbol{p_{1i}, p_{0i}}$: The probability of observing '1's and '0's, respectively, in the i^{th} Taxon's sequence.
$\boldsymbol{F_{ij}}$: The j^{th} feature value for the i^{th} Taxon before normalisation.
$\boldsymbol{F'_{ij}}$: The normalised j^{th} feature value for the i^{th} Taxon.
$\mathbf{min}(\boldsymbol{F_i})$, $\mathbf{max}(\boldsymbol{F_i})$: Minimum and maximum values of the i^{th} feature through all taxa.
I: Indicator function, where $I(condition) = 1$ If the condition is accurate, and 0 otherwise.
$\boldsymbol{\epsilon}$: During entropy computation A small constant added to it to prevent undefined of it.

3.1. Phylogenetic analysis using NN

3.1.1 Preprocessing

The dataset, comprising binary genetic sequences from a diverse array of taxa, is sourced from a publicly available GitHub repository [14]. and depicted in Figure 1. The sequences undergo a crucial preprocessing phase, converting categorical data into a neural network-compatible format. This preparatory step is essential for feature extraction, which is rigorously defined in Table 1. The table provides the mathematical formulations to refine the binary datasets for input into the neural network.

Table 1: The mathematical representations for the feature extraction process from binary datasets

Feature type	Equation	Description
Statistical Feature	$C_{1i} = \sum_{j=1}^{m} (s_{ij} = 1)$	Total count of '1's for the i^{th} taxon
	$C_{0i} = \sum_{j=1}^{m} (s_{ij} = 0)$	Total count of '0's for the i^{th} taxon
	$D_{1i} = \frac{C_{1i}}{m}$	The density of '1's for the i^{th} taxon
	$D_{0i} = \frac{C_{0i}}{m}$	The density of '0's for the i^{th} taxon
	$\bar{x}_{1i} = mean(\{j \mid s_{ij} = 1\})$	The mean position of '1's
	$\tilde{x}_{1i} = median(\{j \mid s_{ij} = 1\})$	Median position of '1's
	$\sigma^2_{1i} = var(\{j \mid s_{ij} = 1\})$	Variance of positions of '1's
	$\bar{x}_{0i} = mean(\{j \mid s_{ij} = 0\})$	The mean position of '0's
	$\tilde{x}_{0i} = median(\{j \mid s_{ij} = 0\})$	Median position of '0's
	$\sigma^2_{0i} = var(\{j \mid s_{ij} = 0\})$	Variance of positions of '0's
Positional Feature	$P_{F1i} = \min(\{j \mid s_{ij} = 1\})$	Position of the first '1'
	$P_{L1i} = \max(\{j \mid s_{ij} = 1\})$	Position of the last **'1'**
	$P_{F0i} = \min(\{j \mid s_{ij} = 0\})$	Position of the first **'0'**
	$P_{L0i} = \max(\{j \mid s_{ij} = 0\})$	Position of the last **'0'**
Clustering Feature	$K_i = \sum(diff([0, s_i, 0] == 1) == 1)$	Total number of contiguous clusters of '1's.
Entropy-Based Feature	$E_i = -(P_{1i} \log_2(p_{1i} + \epsilon) + P_{0i} \log_2(p_{0i} + \epsilon))$	Shannon entropy for the i^{th} taxon, epsilon as small constant to avoid log of zero

3.1.2 Dataset Subsampling and Training Process

For the purpose of obtaining diverse training data and avoiding overfitting to a single taxon set, we employed a random subsampling approach. We had 19 taxa with a binary sequence in our original data. We randomly selected 12 taxa for each subset such that the subsets were different and no taxa were repeated within subsets.

We used this approach to generate 100 different subsets based on different data variations. We ran a Maximum Parsimony analysis for each subset and employed the resulting tree length as the target value for our neural network.

Following the computation of the dataset and the corresponding tree lengths, the data were split into a testing set and a training set. The model was trained with a very diverse array of phylogenetic topologies so that it would generalize well across taxonomic groups and not be biased towards any group of taxa.

This diversity-based data construction process allowed the model to learn patterns under various evolutionary conditions rather than memorize one structure alone, thus making it robust and reducing the chances of overfitting.

3.1.3 Feature Normalization

After extraction, the features were normalized to a consistent scale to ensure they contributed equally during model training. Each feature F_{ij} was normalized using formula (2), where $\min(F_i)$ and $\max(F_i)$ represent the minimum and maximum values of the i^{th} feature across all taxa.

$$F'_{ij} = \begin{cases} 2\left(\dfrac{F_{ij} - \min(F_i)}{\max(F_i) - \min(F_i)}\right) - 1 & , \quad \min(F_i) \neq \max(F_i) \\ 0 & , \quad \min(F_i) = \max(F_i) \end{cases} \tag{2}$$

If the minimum and maximum values of a feature are the same, the normalized value is set to zero. While the direct binary representation of the dataset provides a comprehensive view, it does not offer flexibility in adjusting dataset size during model evaluation.

To overcome this, we extracted key phylogenetic features such as mutation density, entropy, clustering, and positional variation. These features highlight evolutionary trends, including mutation hotspots and conserved regions, which directly influence phylogenetic tree length.

Comparative testing demonstrated that our chosen feature set produced more stable and accurate tree length predictions than raw binary sequences alone. This approach improves generalization while allowing for adaptable dataset sizes in performance evaluations.

3.2 Cross-Validation and Data Transformation

To evaluate the model and optimize its parameters, we employed k-fold cross-validation. The multi-dimensional feature set was transformed into a flattened vector to fit the input layer of the artificial neural network (ANN). This approach allowed for a comprehensive assessment of the model's predictive performance, ensuring that each data subset was utilized for both training and validation. In our study, we set $k = 4$.

3.3 Neural Network Training

Configured and trained an ANN with pre-processed and normalised feature vectors, adjusting learning rates and performance goals to find the optimal setup.

3.3.1 Performance Metrics

Actual tree lengths from conventional phylogenetic studies were compared with Maximum Parsimony scores that were predicted by the ANN. Consequently, the accuracy of this evaluation depended largely on the comparison of the phylogenetic connection inference that was made by the ANN.

Our research unites biology and computer science, providing a method in the intersection of these disciplines that employs statistical and machine-learning methodologies for the new phylogenetic analysis. As a result, it is likely to increase the power and accuracy of studies on evolution and of genetic sequences.

We seriously considered a few topologies of neural networks for phylogenetic analysis; among them, we elaborated on two varieties, which are the feedforward network (feedforwardnet) and the fit network (fitnet). We selected these due to their structural differences and learning potentials, so that we could check their working conditions and suitability for our dataset.

This was achieved by thoroughly exploring different neural network conflagration and parameter settings, trying to maximize performance without the issue of overfitting, again, of most importance within machine learning. We did this through careful adjustment of the learning rate and performance objective parameters, together with small incremental improvements to the design of the network. We then took a close look at several neural network topologies and cycled them through until it was determined which design would do the best job of capturing the complex connections in our evolutionary data. We varied the number of layers and nodes in a systematic way to find the effect of network complexity on model accuracy and generalization.

3.3.2 Learning Rate

The learning rate is a crucial hyperparameter that influences model convergence. In our approach, we systematically set it within a range of 0.1 to 0.7. This broad range allowed for a balance between conservative updates, which help prevent missing optimal solutions, and more aggressive updates, which accelerate convergence toward the desired performance goal.

3.3.3 Performance Goal

To prevent overfitting and ensure the model generalizes well to unseen data, we set a performance goal on a logarithmic scale ranging from 10^{-6} to 10^{-2}. This wide range was instrumental in fine-tuning the model's sensitivity to training errors, balancing the need for precise data fitting while maintaining robustness against new inputs.

By employing this rigorous methodology, we underscore our commitment to precision in optimizing neural networks for phylogenetic analysis. We carefully navigate the complexities of model architecture and parameter selection to enhance both predictive accuracy and generalizability.

Algorithm 1: Iterative Exploration and Evaluation of Neural Network Configurations

Input:

- **Data** : X, Y : Feature vector and their corresponding tree length as shown in Figure 1
- hiddenLayerSize : Explore various neural network configurations by adjusting the number and size of hidden layers
- K : Employ k-fold cross-validation, partitioning the dataset into k=4 folds, to robustly assess the model's performance
- η : Learning rate
- ϵ : Performance goals
- $epoch$: Maximum number of iteration

Output:

- net : Optimized Neural Network Model Configuration
- $Kmse$: The mean square error for each k-fold

Data Preparation: $cv = partition(size(Y, 1), 'KFold', K)$
Divide data to training and testing subsets according to the current fold's partitioning.

For i = 1 to K
trainIndices = cv.training(i)
testIndices = cv.test(i)
X_train = X(trainIndices, :)
Y_train = Y(trainIndices, :)
X_test = X(testIndices, :)
Y_test = Y(testIndices, :)

foreach netType in {fitnet, feedforwardnet}
net = configureNet(netType, hiddenLayerSize, η, ϵ)
[net, tr] = train(net, X_train, Y_train)
Y_pred = net(X_test')

MSE of training Neural network
$Kmse_i$ = simulate(net, Y_test, Y_pred)

End *// network type*
End *//* of k-flod loop

This cumulative metric shows the model's overall predictive power, considering data diversity and the randomness of the training process. Figure 2 visualises the detailed effects of network configurations topology and training parameters on model accuracy, informing the optimal design for neural network-based phylogenetic analysis. To represent the calculation of the cumulative MSE for a specific neural network configuration, we need to calculate the average MSE over all folds, learning rates, and performance goals as in equation (3).

$$Cumulative\ \mathrm{M}^{Type} = \frac{1}{N_G} \sum_{G=10^{-6}}^{10^{-2}} \left(\frac{1}{N_L} \sum_{L=0.1}^{0.7} \left(\frac{1}{N_k} \sum_{k=1}^{4} M_{G,L,K}^{Type} \right) \right) \quad (3)$$

In equation (3)
N_G: The number of performance goals tested.
N_L: The number of learning rates tested.
N_k: The number of k-folds, which is 4 in your case.
$M_{G,L,K}^{Type}$: Represents the value (MSE, number of epochs, or time in milliseconds) for a given performance goal G, learning rate L and fold k.

3.4 Experimental Setup

In our experiments, PAUP* version 4.0a (build 168) for Unix/Linux was used. 24 CPU cores were housed in a dual-socket Intel® Xeon® CPU E5-2640 v2 @ 2.00GHz on the server. This configuration supports SSE vectorization and SSSE3 instructions, as well as multithreading using Pthreads, and is optimized for the Intel® 64 architecture. It was created using the GNU C compiler (gcc) version 4.4.7. Our new ANN4P approach was implemented concurrently with the PAUP* runs. This method used the processing power of the same computer to do neural network estimations and was programmed in MATLAB R2023b.

4. Results

In our systematic exploration of neural network architectures for predicting phylogenetic tree lengths, we examined various configurations, from single-layered networks to more complex multi-layered structures. Our approach involved the following steps:

1- For single-layer networks, the number of nodes varied from 9 to 20.

2- For two-layer networks, all permutations of nodes ranging from 9|6 to 20|31 were tested.

Each specific architecture was then subjected to a performance goal ranging from 10^{-6} to 10^{-2}. A sweep of learning rates from 0.1 to 0.7 was performed for each goal. Utilising a 4-fold cross-validation method ($k = 4$) we collected four MSE values for each configuration corresponding to each fold, providing a robust estimate of model performance across different data subsets.

The cumulative MSE for each configuration was computed by summing the MSEs from all folds, offering an aggregated error measure across the entire cross-validation process. This will give us the mean MSE over all combinations of performance goals and learning rates for a specific neural network configuration averaged over the k-folds.

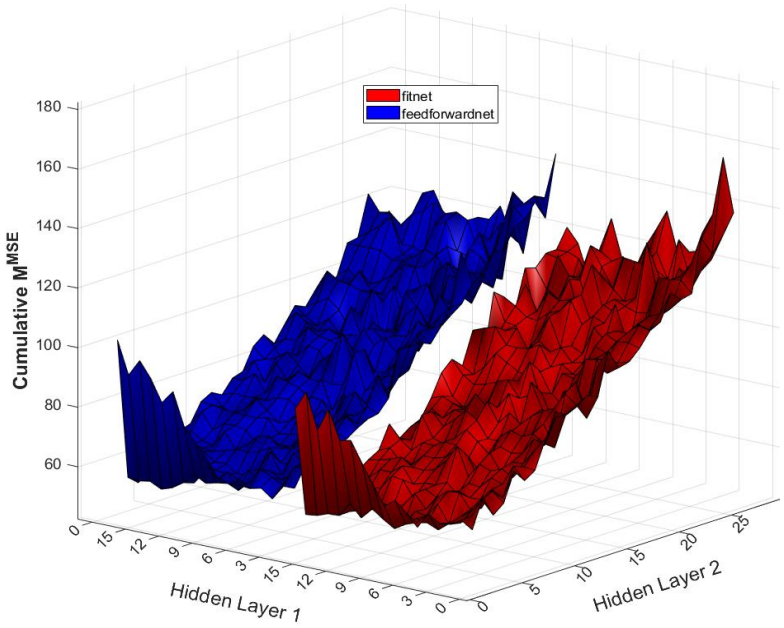

Figure 2: Presented illustrate the Cumulative MSE for Fitnet vs feedforwardnet

In Figures 3 and 4, we explore the training dynamics of neural network models, where Figure 3 correlates the number of epochs to various hidden layer configurations, and Figure 4 contrasts this with the training times measured in milliseconds.

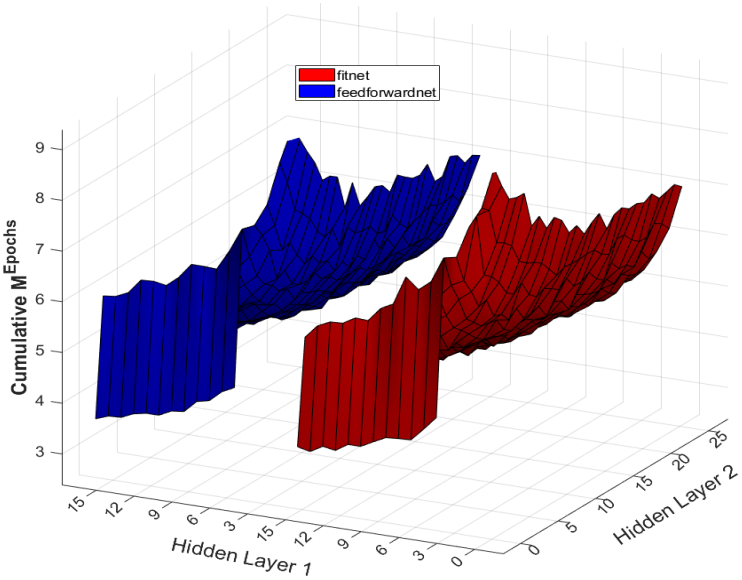

Figure 3: Presented the Epochs performance for Fitnet vs feedforwardnet.

These visualizations elucidate the efficiency and computational demands of different network architectures, indicating how the model's complexity impacts the training process. This comprehensive view assists in identifying optimal configurations that offer a balance between training depth and expedience, essential for practical phylogenetic tree length estimation. In both cases, equation (3) was applied to find the cumulative Epoch and Time.

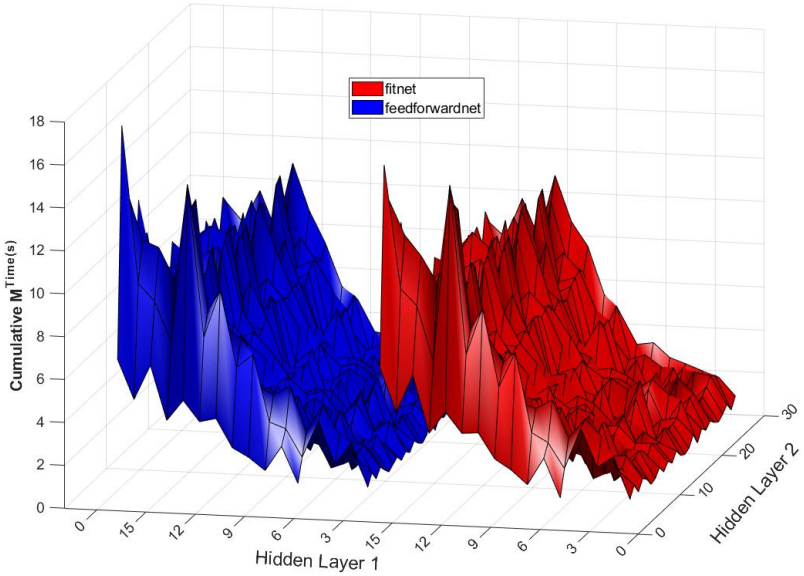

Figure 4: Shows the time performance comparison between Fitnet and Feedforwardnet

Through our empirical investigations reveal how different neural network configurations perform in phylogenetic analysis. These findings, therefore, will help us select a model that balances accuracy and computational efficiency. This advancement will further the use of machine learning in phylogenetics.

4.1 Optimizing Neural Network Architectures for Phylogenetic Estimation

By analyzing Figures 2, 3, and 4, which compare different fitnet and feedforward neural network configurations, our study found that feedforward architectures excel in estimating phylogenetic tree lengths. The optimal feedforward network, with 20 nodes in the first hidden layer and 6 in the second, achieved the lowest cumulative Mean ($M_{G,L,K}^{MSE}$) of 52.67 as shown in (3). This highlights the superior performance of feedforward networks over fitnet for this task and the crucial role of network architecture in model effectiveness. The optimal feedforward configuration enhances neural network models for phylogenetic analysis, improving accuracy and efficiency. This comparison underscores the importance of network design in creating practical computational tools for phylogenetics.

4.2 Optimization of Hyperparameters

Proved the optimal neural network architecture for both the first and second Hidden layers our objective now will be focused on fine tuning the model's hyperparameters to achieve the lowest error among all k-folds. As depicted in Figure 5, we meticulously plotted the Mean of $\left(M_{G,L,K}^{MSE}\right)$ equation (3) as a function of both the performance goal (ϵ) and the learning rate (η), covering a range of values for each parameter. The surface plot reveals the nuanced interplay between these hyperparameters and their collective impact on model accuracy.

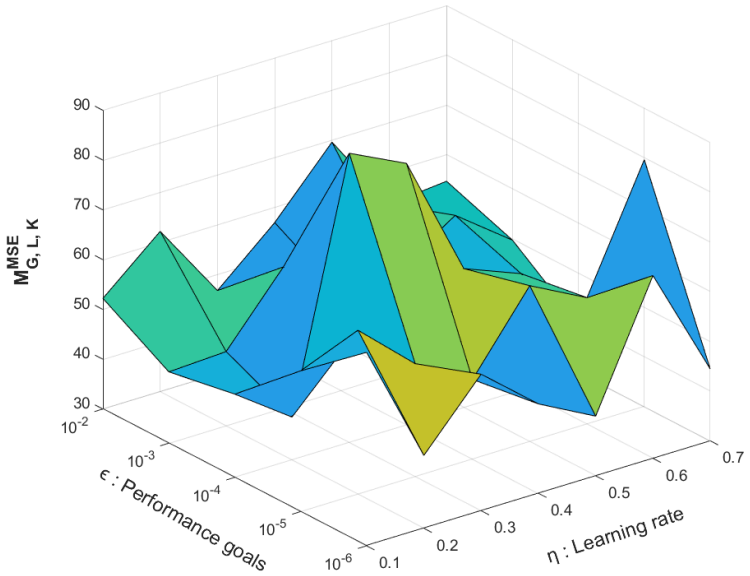

Figure 5: Optimization of Hyperparameters for Phylogenetic Tree Length Estimation: A Surface Plot of mean for MSE across ϵ and η

A performance goal of $\epsilon = 10^{-4}$ and a learning rate of $\eta = 0.2$ $M_{G,L}^{MSE}$ of 38.596. Accurately calibrating ϵ and η is crucial for neural network training and highlights the need for targeted parameter searches to optimise phylogenetic tree length estimation.

4.3 Performance Evaluation using k-Fold Validation

After optimising the neural network architecture and hyperparameters, we evaluated the model's predictive performance using 4-fold cross-validation. This robust method tests the model against multiple data subsets, validating its generalizability and accuracy in estimating phylogenetic tree lengths.

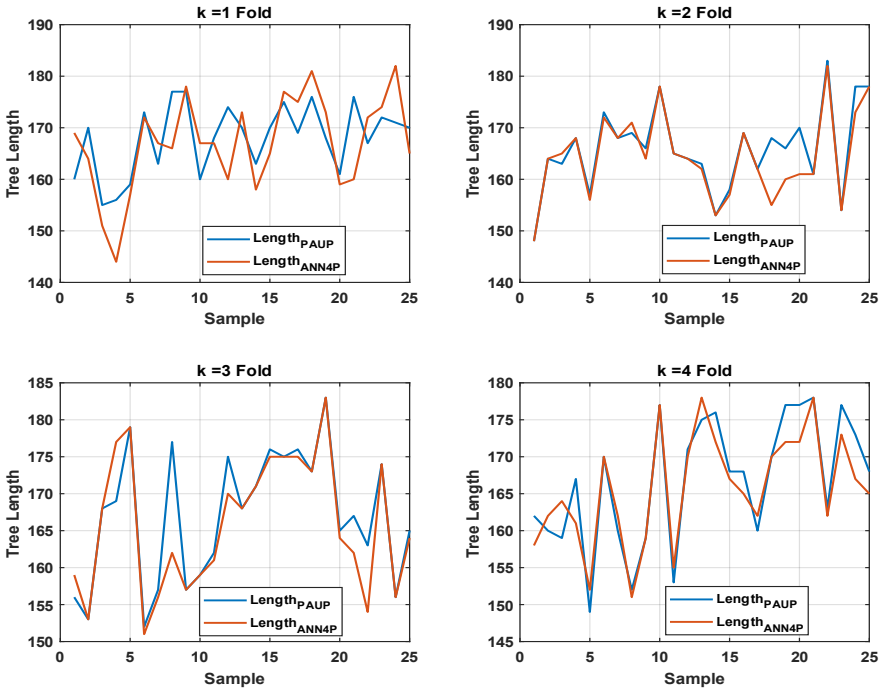

Figure 6: Comparative Analysis of Phylogenetic Tree Length Estimations Across 4-Fold Cross-Validation

The cross-validation results are visualized in Figure 6, where the true lengths of the phylogenetic trees are plotted alongside the lengths estimated by the neural network for each fold. As can be seen, the model demonstrates a consistent estimation pattern across all folds, closely aligning with the true lengths and indicating high precision in its predictions.

Tables 2 to Table 5 compare traditional phylogenetic tree lengths computed with PAUP* program and the estimated tree lengths from our ANN4P method. The $Length_{PAUP}$ columns display tree lengths from PAUP* analysis for each dataset sample across each folds. At the same time the $Length_{ANN4P}$ columns exhibit the lengths estimated by our neural network model ANN4P.

To emphasize on computational efficiency we included $Time^{s}_{PAUP}$ and $Time^{s}_{ANN4P}$ columns documenting the time in seconds taken by PAUP* and ANN4P to compute these tree lengths. The reason why in each Table the $Time^{s}_{ANN4P}$ there is no change because the nature of neural network ANN4P. In order to train a model, we need multiple datasets and after training the testing will be in near-zero time or in the algorithm will be called O(1) so the time in the Tables is simply the training time of ANN4P.

On the contrary the time by PAUP $Time^{s}_{PAUP}$ can be masuared for each dataset indvasualy. Although this is unfair compared to our method but this is the only way and on top of that our method overcame the PAUP. The inclusion of time measurements allows for an evaluation of the accuracy of tree length estimation and the speed at which our neural network model operates compared to PAUP*.

The selection of 4-Fold Cross-Validation was based on ensuring a clear and structured evaluation after feature transformation. Each dataset (12 taxa) was initially represented by binary genetic sequences but was then transformed into a fixed-length feature vector containing statistical, positional, clustering, and entropy-based attributes. This transformation allowed for a standardized input structure while preserving phylogenetic information.

With a 4-Fold split (75% training, 25% testing), we maintained a balance between training efficiency and evaluation stability. While other folds (e.g., 2-Fold, 3-Fold) could be tested in future work to analyze their effect on performance, the 4-Fold structure provided consistency in testing different subsets while preventing model bias.

Table 2: Comparison of Estimated and True Phylogenetic Tree Lengths in 1-Fold Cross-Validation

Sample	$Length_{PUAP}$	$Time^{s}_{PAUP}$	$\mathbf{Length_{ANN4P}}$	$Time^{s}_{\mathbf{ANN4P}}$
1	160	201.5	169	12.15
2	170	200.9	164	12.16
3	155	180.9	151	12.17
4	156	186.1	144	12.19
5	159	184.5	157	12.21
6	173	177	172	12.21
7	163	175.9	167	12.21
8	177	183.8	166	12.21
9	177	179.9	178	12.21
10	160	177.6	167	12.21
11	168	177.3	167	12.21
12	174	176.5	160	12.21
13	170	178.1	173	12.22
14	163	178.9	158	12.22
15	170	179.4	165	12.22
16	175	176.9	177	12.22
17	169	175.5	175	12.22
18	176	176.5	181	12.22
19	168	176	173	12.22
20	161	177.7	159	12.22
21	176	190	160	12.22
22	167	176	172	12.22
23	172	184.6	174	12.22
24	171	177	182	12.22
25	170	180.7	165	12.22

Table 3: Comparison of Estimated and True Phylogenetic Tree Lengths in 2-Fold Cross-Validation

Sample	$\mathbf{Length_{PUAP}}$	$Time^{s}_{PAUP}$	$\mathbf{Length_{ANN4P}}$	$Time^{s}_{\mathbf{ANN4P}}$
1	148	183.2	148	8.47
2	164	181.4	164	8.47
3	163	182.4	165	8.47
4	168	179.8	168	8.47
5	157	190	156	8.48
6	173	116.6	172	8.48
7	168	177.3	168	8.48
8	169	178.2	171	8.48
9	166	178.3	164	8.48
10	178	182.3	178	8.48
11	165	178.5	165	8.48
12	164	177.4	164	8.48
13	163	175.9	162	8.48
14	153	176.8	153	8.48
15	158	176.4	157	8.48
16	169	175.7	169	8.48
17	162	178.6	162	8.48
18	168	176	155	8.48
19	166	178.9	160	8.48
20	170	191.4	161	8.48
21	161	193.4	161	8.49
22	183	177.6	182	8.49
23	154	181	154	8.49
24	178	177.6	173	8.49
25	178	175.6	178	8.49

Table 4: Comparison of Estimated and True Phylogenetic Tree Lengths in 3-Fold Cross-Validation

Sample	$\mathbf{Length_{PUAP}}$	$Time^{s}_{PAUP}$	$\mathbf{Length_{ANN4P}}$	$Time^{s}_{ANN4P}$
1	156	180.6	159	17.4
2	153	199.4	153	17.4
3	168	179.5	168	17.4
4	169	184.6	177	17.4
5	179	178.5	179	17.4
6	152	183.1	151	17.4
7	157	185.9	156	17.4
8	177	180.5	162	17.4
9	157	187.5	157	17.4
10	159	177.1	159	17.4
11	162	179.6	161	17.4
12	175	176.2	170	17.4
13	168	176.6	168	17.4
14	171	178.2	171	17.41
15	176	179.3	175	17.41
16	175	180.4	175	17.41
17	176	179.1	175	17.41
18	173	174.3	173	17.41
19	183	175.7	183	17.41
20	165	178.3	164	17.41
21	167	178	162	17.41
22	156	177.1	159	17.41
23	153	199.2	153	17.41
24	168	177.4	168	17.41
25	169	176.1	177	17.41

Table 5: Comparison of Estimated and True Phylogenetic Tree Lengths in 4-Fold Cross-Validation

Sample	$\mathbf{Length_{PUAP}}$	$\boldsymbol{Time^{s}_{PAUP}}$	$\mathbf{Length_{ANN4P}}$	$\boldsymbol{Time^{s}_{ANN4P}}$
1	162	184.3	158	6.13
2	160	201.1	162	6.13
3	159	181	164	6.13
4	167	180.6	161	6.13
5	149	181.5	152	6.13
6	170	180.1	170	6.13
7	160	179.6	162	6.13
8	152	175.8	151	6.13
9	159	177.3	159	6.13
10	177	179.3	177	6.13
11	153	180.6	155	6.13
12	171	178	170	6.13
13	175	177.3	178	6.13
14	176	174.4	172	6.13
15	168	178.9	167	6.13
16	168	183.1	165	6.13
17	160	175.6	162	6.13
18	170	177.8	170	6.13
19	177	181	172	6.13
20	177	187.9	172	6.14
21	178	184.6	178	6.14
22	162	187	158	6.14
23	160	179.2	162	6.14
24	159	177.3	164	6.14
25	167	178.3	161	6.14

The ANN4P model closely approximates PAUP* derived tree lengths and does so much faster highlighting its capability for quick and reliable phylogenetic analysis. In other words, the neural network correctly estimates tree lengths: this is indicated by minimal variance between the estimated and actual lengths. Cross-validation confirm the model's effectiveness, making it a dependable tool for phylogenetic analysis without traditional tree construction methods by bypassing the phylogenetic analysis.

4. Conclusions

ANN in general are proven to be an important tool in phylogenetics particularly in the field of scriptinformatics. This research affirms using feedforward neural networks for estimating phylogenetic tree lengths and highlights the importance of optimised network architectures and hyperparameters in improving model performance. The study shows that artificial neural networks (ANNs) have the potential to simplify phylogenetic analysis by combining computational and biological sciences. ANNs offer a fast, accurate, and computationally efficient alternative to traditional methods. The work paves the way for other research, where applications of ANNs for more diverse types of phylogenetic data can be developed and the feature selection algorithms designed so far can also be improved further in order to better the accuracy in evolutionary inference. While training the model requires

multiple datasets, the testing time becomes negligible (O(1)) after training, with the times in the tables reflecting the training duration of ANN4P . In contrast, the time for PAUP is measured individually for each dataset, which might seem unfavorable compared to our method. However, this approach is necessary, and our method ultimately outperforms PAUP.

While this study demonstrates the effectiveness of neural networks for phylogenetic tree length estimation, integrating classical models such as Maximum Likelihood (ML) and Bayesian inference could further improve biological interpretability. A potential approach is to use ML-based priors to guide the learning process, incorporating established evolutionary models into ANN training.

Such a hybrid framework could leverage the computational efficiency of ANNs while retaining the theoretical robustness of classical phylogenetic methods. Future work will explore these hybrid models, aiming to enhance both prediction accuracy and biological relevance in phylogenetic analysis.

5. Competing interests

The authors declare that they have no competing interests.

Acknowledgement

Authors like to thank their colleague Péter Pálovics from the BME-EET Department, who greatly contributed with the configuration of the server since it was of importance to our research. We would also like to extend our special thanks to the Stipendium Hungaricum scholarship program.

References

[1] O. A. Salman and G. Hosszú, "Cladistic analysis of the evolution of some Aramaic and Arabic script varieties," Int. J. Appl. Evol. Comput., vol. 12, pp. 18–38, 2021.

[2] O. A. Salman, G. Hosszú, and F. Kovács, "A new feature selection algorithm for evolutionary analysis of Aramaic and Arabic script variants," Int. J. Intell. Eng. Inf., vol. 10, pp. 313–331, 2022.

[3] O. A. Salman and G. Hosszú, "Optimised feature dimension reduction method and its impact on the search for optimal trees," in Proc. Workshop Adv. Inf. Technol., 2023, pp. 23–28.

[4] O. A. Salman and G. Hosszú, "A phenetic approach to selected variants of Arabic and Aramaic scripts," Int. J. Data Anal., vol. 3, pp. 1–23, 2022.

[5] O. A. Salman and G. Hosszú, "Phylogenetic inference using advanced feature selection," in Proc. 14th IEEE Int. Conf. Cogn. Infocommun., 2023, pp. 173–178.

[6] O. A. Salman and G. Hosszú, "Phylogenetic modelling scripts for identifying script versions," *Procedia Comput. Sci.*, vol. 239, pp. 1417–1424, 2024.

[7] O. A. Salman and G. Hosszú, "Using distance-based methods to calculate optimal and suboptimal parsimony trees," in *Proc. Workshop Adv. Inf. Technol.*, 2024, pp. 79–84.

[8] Salman, O. A.; Hosszú, G. Evaluating feature impact prior to phylogenetic analysis using machine learning techniques. Information 2024, 15(11), 696. https://doi.org/10.3390/info15110696

[9] C. H. Wu, H. L. Chen, and S. C. Chen, "Gene classification artificial neural system," *Int. J. Artif. Intell. Tools*, vol. 4, pp. 501–510, 1995.

[10] C. Scornavacca, F. Delsuc, and N. Galtier, Eds., Phylogenetics in the Genomic Era. Open Access, 2020, 568 pp., ISBN 978-2-9575069-0-3.

[11] P. Kapli, Z. Yang, and M. J. Telford, "Phylogenetic tree building in the genomic age," Nat. Rev. Genet., vol. 21, pp. 428-444, 2020.

[12] Y. K. Mo, M. Hahn, and M. L. Smith, "Applications of machine learning in phylogenetics," 2023.

[13] Y. Zhou et al., "Graph neural networks: taxonomy, advances, and trends," *ACM Trans. Intell. Syst. Technol.*, vol. 13, pp. 1-54, 2022.

[14] O. A. Salman, "Extended Arabic-Aramaic DataSet," GitHub repository, [Online]. Available: https://github.com/OsamaAliSalman/Extended_Arabic-Aramaic-DataSet.git. [Accessed: Aug. 2, 2024].

[15] T. Halgaswaththa, A. S. Atukorale, M. Jayawardena, and J. Weerasena, "Neural network based phylogenetic analysis," in *Proc. 2012 Int. Conf. Biomed. Eng. (ICoBE)*, 2012, pp. 155-160, doi: 10.1109/ICoBE.2012.6178974.

[16] A. Suvorov and D. R. Schrider, "Reliable estimation of tree branch lengths using deep neural networks," *bioRxiv*, 2022. [Online]. Available: https://doi.org/10.1101/2022.11.07.515518.

[17] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Pearson Education India, 2009. [Online]. Available: dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf.

[18] F. Z. El-Hassani, M. Amri, N.-E. Joudar, and K. Haddouch, "A new optimization model for MLP hyperparameter tuning: Modeling and resolution," 2023.

[19] D. L. Swofford, *PAUP* Phylogenetic Analysis Using Parsimony (*and Other Methods)*, Version 4. [Online]. Available: phylosolutions.com/paup-documentation/paupmanual.pdf.

[20] M. D. Hendy and D. Penny, "Branch and bound algorithms to determine minimal evolutionary trees," *Math. Biosci.*, vol. 59, pp. 277-290, 1982.

[21] L. L. Cavalli-Sforza, A. Piazza, P. Menozzi, and J. Mountain, "Reconstruction of human evolution: bringing together genetic, archaeological, and linguistic data," *Proc. Natl. Acad. Sci. USA*, vol. 85, pp. 6002-6006, 1988.

[22] J. Felsenstein, *Inferring Phylogenies*. Sunderland, MA: Sinauer Associates, 2004.

[23] R. Qamar and B. A. Zardari, "Artificial Neural Networks: An Overview," *Mesopotamian J. Comput. Sci.*, vol. 2023, pp. 124–133, Aug. 2023, doi: 10.58496/MJCSC/2023/015.