information

Article

Evaluating Feature Impact Prior to Phylogenetic Analysis Using Machine Learning Techniques

Osama A. Salman and Gábor Hosszú *

Department of Electron Devices, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3, 1111 Budapest, Hungary; osamaalisalman.khafajy@edu.bme.hu
* Correspondence: hosszu.gabor@vik.bme.hu

Abstract: The purpose of this paper is to describe a feature selection algorithm and its application to enhance the accuracy of the reconstruction of phylogenetic trees by improving the efficiency of tree construction. Applying machine learning models for Arabic and Aramaic scripts, such as deep neural networks (DNNs), support vector machines (SVMs), and random forests (RFs), each model was used to compare the phylogenies. The methodology was applied to a dataset containing Arabic and Aramaic scripts, demonstrating its relevance in a range of phylogenetic analyses. The results emphasize that feature selection by DNNs, their essential role, outperforms other models in terms of area under the curve (AUC) and equal error rate (EER) across various datasets and fold sizes. Furthermore, both SVM and RF models are valuable for understanding the strengths and limitations of these approaches in the context of phylogenetic analysis This method not only simplifies the tree structures but also enhances their Consistency Index values. Therefore, they offer a robust framework for evolutionary studies. The findings highlight the application of machine learning in phylogenetics, suggesting a path toward accurate and efficient evolutionary analyses and enabling a deeper understanding of evolutionary relationships.

Keywords: feature selection; hyperparameters; machine learning; phylogenetics; scriptinformatics; consistency index (CI); false rejection rate (FRR); false acceptance rate (FAR); classification

Citation: Salman, O.A.; Hosszú, G. Evaluating Feature Impact Prior to Phylogenetic Analysis Using Machine Learning Techniques. *Information* **2024**, *15*, 696. https://doi.org/10.3390/info15110696

Academic Editors: Francesco Fontanella and Heming Jia

Received: 9 August 2024
Revised: 5 September 2024
Accepted: 23 October 2024
Published: 4 November 2024

1. Introduction

Phylogenetics, the study of evolutionary relationships among species, has traditionally relied on models like maximum likelihood (ML) and Bayesian inference. While effective, these methods require substantial computational power, especially with the increasing amount of data involved in phylogenetic studies. Recently, integrating machine learning with phylogenetic analysis has shown promise in addressing these challenges, enhancing both the efficiency and accuracy of phylogenetic tree construction and inference.

In phylogenetic analysis, features are categorized as either homologies or homoplasies. A homology refers to traits inherited from a common ancestor, while a homoplasy describes traits that develop independently, often due to convergent or parallel evolution. Distinguishing between these features is crucial for accurate phylogenetic reconstruction, as homologous features indicate shared lineage, whereas homoplasies can obscure these connections if not properly identified. Accurately differentiating between these features improves the precision of phylogenetic analyses, leading to deeper insights into evolutionary history.

The principal objective of our research is to improve the accuracy and efficiency of phylogenetic tree reconstruction by optimizing feature selection with the help of machine learning models, including deep neural networks (DNNs), support vector machines (SVMs) and random forests (RFs). By identifying the most informative features prior to the phylogenetic analysis, we aim to simplify the tree structure and enhance its consistency.

To achieve our research objectives, we propose the following steps:

1. Perform maximum parsimony to extract a phylogenetic tree [1];
2. Identify all branches and their ancestral values;
3. Determine the number of mutations (or changes) for each feature as represented on the phylogenetic tree;
4. Develop a model to predict feature quality before performing phylogenetic analysis.

Feature selection is inherently challenging, particularly when identifying subgroups of features with a high Consistency Index (CI), which significantly impacts the efficacy of phylogenetic analysis. To address these challenges, we employed a feature selection algorithm, performing experiments to explore the complexities involved in this process.

1.1. Identify Subgroups and Their Quality

Our research focuses on creating and utilizing a feature selection algorithm to enhance phylogenetic tree reconstruction. Feature selection is inherently challenging, particularly when identifying subgroups of features with a high Consistency Index (CI) value, which significantly impacts the efficacy of phylogenetic analysis. Our goal is to address these challenges and demonstrate the robustness of our methodology.

In recent experiments, we highlighted the complexities involved in feature selection. While identifying the most parsimonious tree is crucial, understanding the variability in and computational complexity of feature selection is equally important. To explore these challenges, we randomly selected subgroups of features and used the PAUP* program to perform a branch and bound search. We recorded search times, feature counts, and CI values, as shown in Figure 1.

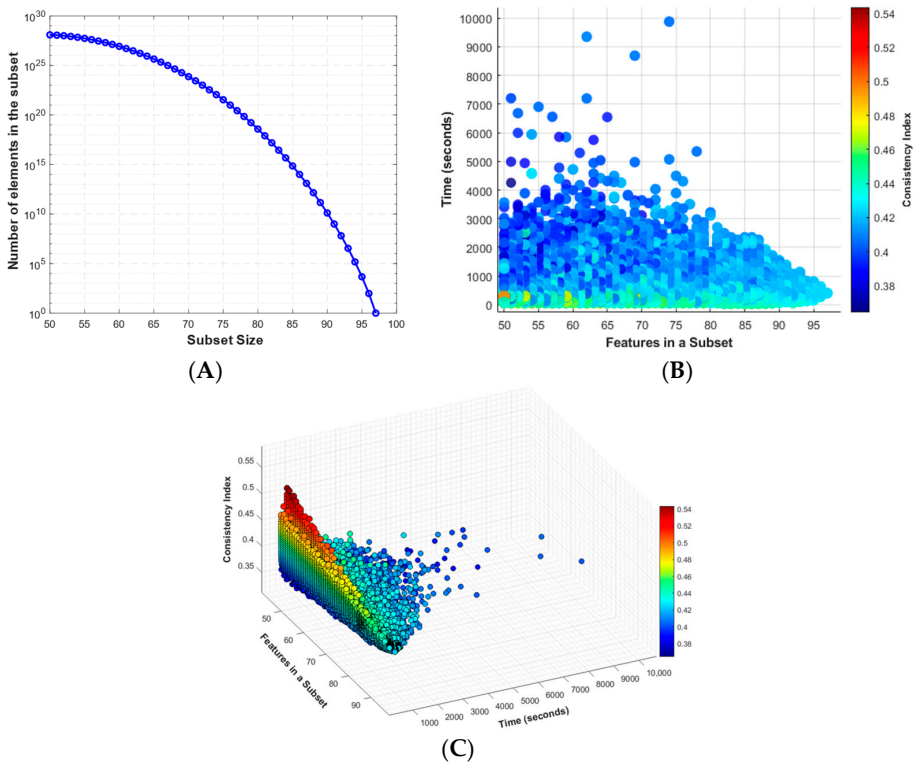

Figure 1. (**A**) Exponential increase in combinations as subset size decreases. (**B**) Processing time vs. number of features, with CI values showing variability. (**C**) 3D plot of CI values, features, and processing time, illustrating the trade-off between computation and CI.

In Figure 1A shows the relationship between the number of elements in a subset and the subset size. As the subset size decreases, the number of possible combinations increases exponentially.

On the other hand, Figure 1B presents a scatter plot of the time taken for the branch and bound search relative to the number of features in each subset. The color-coded Consistency Index (CI) values illustrate the variability in the data.

Finally, Figure 1C shows a 3D plot depicting the relationship between the Consistency Index (CI) values, the number of features, and the processing time, highlighting the challenge of finding subgroups that balance computational feasibility with high CI values.

These figures collectively underscore the delicate balance required in feature selection for phylogenetic analysis, emphasizing the combinatorial explosion and the variability in search times and CI values as key challenges. This analysis demonstrates the importance of developing robust feature selection methods to improve the accuracy and efficiency of phylogenetic studies.

1.2. Leveraging Machine Learning for Phylogenetic Analysis of Historical Scripts

Systematics, traditionally focused on biological evolution, has expanded into scriptinformatics, a field that applies evolutionary modeling and computer science to understand the historical evolution of scripts [2–9]. Advances in computational methods, particularly in feature selection and machine learning, have significantly enhanced the accuracy of phylogenetic inference, especially when dealing with high-dimensional data [3–5,7–9].

Despite these advancements, constructing phylogenetic trees, and calculating maximum parsimony scores in particular, remains time-consuming and costly. Traditional methods often struggle with large, complex datasets, creating bottlenecks in evolutionary biology research [10]. Additionally, identifying informative features from large datasets is challenging, as it frequently requires the reconstruction of trees to evaluate different feature subsets, making the process increasingly impractical as dataset sizes grow. Neural network approaches, however, offer a groundbreaking solution by predicting tree lengths directly from datasets, thereby reducing the need for exhaustive phylogenetic analysis [2,4].

Although neural network methods simplify feature selection and allow for the quick assessment of feature impacts on predicted tree lengths, their full potential for handling complex evolutionary scenarios—such as feature duplication, loss, and introgression in large-scale phylogenetic analyses—remains to be fully explored. These scenarios introduce significant heterogeneity into datasets, which our application aims to address [11].

In this study, we focus on the evolutionary analysis of historical scripts, including Arabic, Aramaic, and Middle Iranian scripts, using optimized feature selection techniques to reconstruct phylogenetic trees. The datasets used in this study are publicly accessible, promoting transparency and encouraging further research in this growing field [3–5,7–9,12,13]. Beyond contributing to the broader field of scriptinformatics, this research offers practical applications in archeology, with the potential to gradually unravel script evolution and bring us closer to deciphering previously undeciphered inscriptions found by archeologists.

This article is structured as follows: the "Background" section introduces the use of artificial neural networks in phylogenetic reconstruction; the "Methods" section describes our approach; the "Results" section presents our findings; the "Discussion" section interprets the results and their implications; and the "Conclusions" section summarizes the outcomes and suggests future research directions.

2. Background

Phylogenetics, essential to molecular biology and evolutionary studies, uses phylogenetic trees to represent evolutionary relationships, emphasizing the need for accurate the visualization of ancestries and diversification. Traditional methods, such as multiple sequence alignment (MSA) and tree construction algorithms, often struggle with large datasets and biases arising from evolutionary differences. These challenges underscore the need for innovative approaches to effectively manage the complexity and scale of genomic data [14,15].

The reconstruction of phylogenetic trees and the analysis of evolutionary relationships have long been fundamental tasks in evolutionary biology. Traditionally, methods like maximum parsimony and maximum likelihood (ML) have been widely used [1]. However,

recent advancements in machine learning (ML) have opened new avenues for enhancing the accuracy and efficiency of phylogenetic analyses [11,15]. This literature review examines the evolution of these methodologies, focusing on the integration of ML techniques and their potential benefits.

2.1. Established Phylogenetic Methods

The application of machine learning in phylogenetics has gained significant traction in recent years, with several innovative approaches being explored.

Maximum parsimony [1] aims to find the tree topology that requires the fewest evolutionary changes. While this method is straightforward and often effective, it can suffer from issues such as long-branch attraction, which may lead to incorrect tree topologies under certain conditions [16]. In contrast, maximum likelihood (ML) methods are statistically robust, as they evaluate the likelihood of a tree based on a specific model of sequence evolution. However, ML methods are computationally intensive and their use may not be feasible for large datasets [17].

Heuristic tree searches, traditional methods for phylogenetic tree reconstruction, use heuristic searches to manage the computational complexity of evaluating many possible trees. Machine learning has been employed to enhance these heuristic strategies. For instance, Azouri et al. (2020) developed a machine learning algorithm that predicts neighboring trees that increase the likelihood without computing their likelihood, thereby reducing the search space and computational burden [18].

Cherry picking and network construction: A machine learning model introduced by Bernardini et al. [19] that assists in constructing phylogenetic networks using cherry-picking heuristics, ensuring that all input trees are included in the resulting network. This method is particularly useful for large datasets, demonstrating the practical application of machine learning in managing complex evolutionary scenarios.

Neural networks for phylogenetic inference: Zou et al. [20] proposed using deep residual neural networks to infer phylogenetic trees. This method avoids the need for explicit evolutionary models, enabling the neural networks to effectively handle complex substitution heterogeneities. Their results demonstrated improved performances over traditional methods, especially in scenarios involving extensive evolutionary variation.

DendroNet approach: Layne et al. [21] employed supervised learning to create models that incorporate phylogenetic relationships among training data, thereby enhancing the robustness and generalizability of the models in evolutionary studies.

PhyloGAN: Smith and Hahn [22] applied generative adversarial networks (GANs) to phylogenetics by developing PhyloGAN, which infers phylogenetic trees by generating and distinguishing between real and synthetic data. This method shows promise in terms of handling the large model spaces inherent in phylogenetic inference.

ModelTeller: This machine learning-based approach, introduced by Abadi et al. [23], selects the most accurate nucleotide substitution model for phylogenetic reconstruction. It optimizes branch-length estimation, providing more precise tree constructions compared to traditional statistical methods.

Reinforcement learning: Lipták and Kiss [24] investigated the use of reinforcement learning to construct unrooted phylogenetic trees, demonstrating the potential of this approach to improve the efficiency and accuracy with which tree construction tasks are conducted.

2.2. Machine Learning in Phylogenetics

The advent of neural network applications has offered promising solutions to the challenges in phylogenetics, with machine learning (ML), particularly deep learning, transforming tree topology inference, branch length estimation, and model selection [11,15]. Despite their potential, applying supervised ML in phylogenetics presents challenges in certain areas, such as in generating realistic training data and adapting to high-dimensional, heterogeneous biological data [11].

Kalyaanamoorthy et al. introduced ModelFinder, a tool that enhances the accuracy of phylogenetic estimates by incorporating a model of rate heterogeneity across sites. This model selection approach significantly improves the precision of phylogenetic trees, demonstrating the potential of machine learning in model-based phylogenetics [25].

Recent studies have explored the use of deep learning (DL) for phylogenetic inference. For example, one study [15] demonstrated the application of DNNs to predict branch length in phylogenetic trees, showing a superior performance in terms of challenging parameter spaces. Another study introduced Fusang, a DL-based framework for phylogenetic tree inference, which demonstrated a performance comparable to that of ML-based tools and offered potential for optimization through customized training datasets [26].

SVMs have been used to infer phylogenetic relationships by optimizing the hyperplane that separates different evolutionary states. These models are robust in terms of handling high-dimensional data and show promise in various classification tasks [18].

RF algorithms, which build an ensemble of decision trees, have also been applied to phylogenetic inference. Their ability to handle large datasets and provide feature importance metrics makes them valuable for identifying key evolutionary traits [19]. Additionally, combining machine learning with heuristic methods to construct phylogenetic networks efficiently integrates multiple phylogenetic trees into a single network, showcasing the practical application of machine learning in managing complex evolutionary datasets.

2.3. Challenges and Future Directions

While machine learning offers significant benefits for phylogenetic analyses, several challenges persist. Common issues include overfitting, model interpretability, and the substantial need for large training datasets. A recent study [11] discussed these barriers and suggested that careful network design and data encoding could help machine learning to achieve its full potential in phylogenetics.

Tang et al. [27] introduced a neural network model that outperforms traditional methods under long-branch attraction (LBA) conditions. By accounting for tree isomorphisms, this model reduces memory usage and is able to seamlessly extend to larger trees, addressing a critical issue in phylogenetic inference.

The addition of machine learning techniques in phylogenetics represents a significant progression in the field. These approaches enhance model selection (Tree), improve inference accuracy (Tree construction), and offer scalable solutions for large datasets, thus offering powerful tools for evolutionary biologists.

Tadist et al. [28] conducted an extensive review of feature selection methods, particularly in the context of high-dimensional genomic data. These methods share similarities with the challenges faced in phylogenetic analysis. Their work highlights the importance of feature selection in reducing the complexity of data, thereby improving the efficiency and accuracy of machine learning models. This is particularly relevant to our study, where we aim to enhance phylogenetic tree reconstruction by selecting the most informative features of complex datasets. Tadist et al. emphasize that ensemble methods, which combine multiple feature selection techniques, can lead to more robust and accurate models, a principle that underpins our approach to integrating DNN, SVM, and RF models for feature impact assessment.

Traditional machine learning methods for phylogenetic analysis often focus on a single representation of the dataset, limiting the flexibility to adapt to different input sizes and complexities. Our approach introduces three distinct forms of the dataset: DS1 (binary form), DS2 (statistical feature extraction), and DS3 (normalized features). This allows for more flexible and comprehensive analysis. This strategy enables the training of models on one dataset form and testing on another, ensuring that the models are robust across different data representations. Unlike Tadist et al. [28], who primarily focus on feature selection methods for high-dimensional data, our method addresses the challenge of varying dataset sizes and transformations through systematic preprocessing steps, offering a more adaptable solution for phylogenetic tree reconstruction. Furthermore, Kaur et al. [29]

provide a comparison of machine learning models, but their approach lacks the multi-form dataset flexibility that is central to our method. By incorporating feature extraction and normalization, our approach enhances the ability of machine learning models to generalize and perform consistently across diverse datasets.

In this study, we use multiple evaluation metrics, including accuracy (Acc), the false acceptance rate (FAR), the false rejection rate (FRR), receiver operating characteristics (ROCs), the area under the curve (AUC), and the equal error rate (EER), to validate and compare the performance of three machine learning models: deep neural networks (DNNs), support vector machines (SVMs), and random forests (RFs). These metrics are commonly used for performance evaluation, enabling us to confirm which model performs best across different datasets and criteria [30–34].

3. Methods

This study aims to improve the precision and efficiency of phylogenetic tree reconstruction by comparing various machine learning models. Specifically, we assess how well these models predict the impact of each feature on the phylogenetic tree before analysis, which can streamline tree construction, reduce computational demands, and improve reliability. We employ three machine learning algorithms—DNN, SVM, and RF—along with three data preprocessing methods to identify the most effective model–technique combination. Phylogenetic tree construction is dependent on the number of taxa involved. The number of rooted, bifurcating trees is calculated using the formula [35] shown in (1).

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

3.1. Data Representation

As shown in Figure 2, we used various approaches to assess the impact of data preprocessing on model performance. In the first approach (DS1), the binary data were directly fed into the classifiers without modification. Each feature (F^b) is represented as either 0 (absence) or 1 (presence) for each taxon. The labels (Ω) represent the contribution of each feature to the phylogenetic tree.

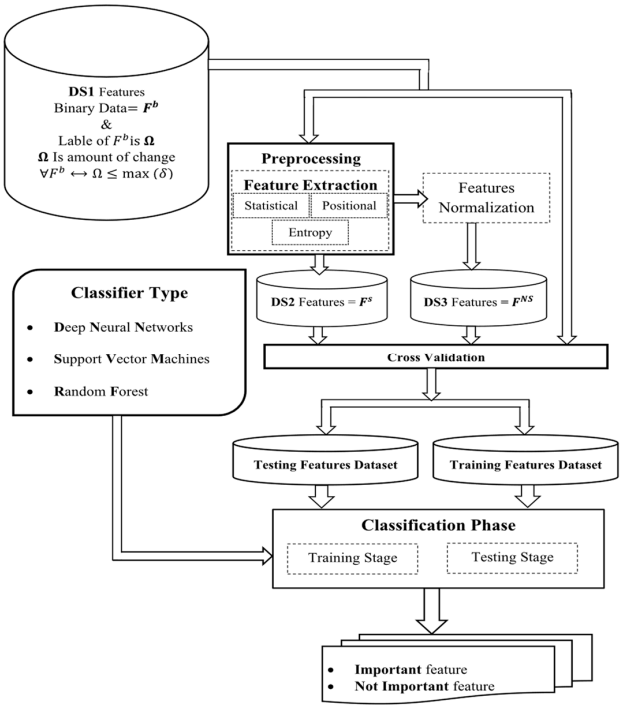

Figure 2. A flowchart of phylogenetic analysis using advanced machine learning algorithms.

3.1.1. DS1 Direct Use of Binary Dataset

The original binary dataset, which consists of historical scripts, was fed directly into the classifiers. These scripts include Arabic, Aramaic, and Middle Iranian, which are treated as unique pattern systems and classified as taxa. Each script is represented as a binary feature vector, where each position corresponds to the presence (1) or absence (0) of a particular phylogenetic feature or trait. This representation allows us to analyze the evolution and historical development of these scripts as symbolic communication systems. The dataset is publicly available on a GitHub repository [13], and it provides a comprehensive collection of binary sequences derived from these historical script variants, capturing key features such as symbols, syntax, and layout rules.

While the use of this binary dataset offers simplicity in terms of processing, its limitation is that the input size varies with the number of taxa, presenting challenges for machine learning classifiers when applied to datasets of different sizes. To address this limitation, the dataset underwent transformations, as described in the subsequent sections, to allow for more flexible analysis across datasets of varying sizes.

3.1.2. DS2: Feature Extraction from DS1

Features were extracted from the binary datasets (DS1) for further analysis. DS1 consists of binary sequences, where each sequence corresponds to a taxon and each position represents a specific phylogenetic feature. The notations used in the feature extraction process and the formulas for each F^S are detailed in Table 1.

Table 1. Mathematical notations for feature extraction DS2.

Notation	Description
m	The total number of taxa in DS1.
n	The total number of features in DS1.
s_{ij}	The value at the j^{th} feature position for the i^{th} taxa in F^b or DS1.
E_j	The Shannon entropy for the j^{th} features in DS1, a measure of randomness in the features.
F_{ij}^S	The j^{th} feature value for the i^{th} taxa in DS2 before normalization.
F_{ij}^{NS}	The normalization of F_{ij}^S and its being saved it in DS3.
ϵ	A small constant added to probabilities to avoid undefined log calculations during entropy computation.
K	A value could be either 0 or 1 to determine if the equation will run for 0's or 1's.

Statistical features are essential for understanding the overall distribution and tendencies within a dataset. These features include total counts, densities, and measures of central tendency and dispersion for both '1's and '0's within each feature in F^b. As shown in the equations in Table 2, these statistical calculations are fundamental in preparing the dataset for subsequent machine learning tasks.

Table 2. Statistical features of DS1 (F^b).

Equation	Description
$C_{K_j} = \sum_{i=1}^{m} \left(s_{ij} = K\right)$	Total count of $0'$s/$1'$s in j^{th} feature of DS1.
$D_{K_j} = \frac{C_{Kj}}{m}$	The density of $0'$s/$1'$s for the j^{th} feature in F^b, where C_{Ki} the total count of K's
$\overline{x}_{Kj} = mean\left(\left\{j \middle\| s_{ij} = K\right\}\right)$	The mean position of K's in F^b for j^{th} feature
$\widetilde{x}_{Kj} = median\left(\left\{j \middle\| s_{ij} = K\right\}\right)$	Median position of K's in F^b
$\sigma_{Kj}^2 = var\left(\left\{j \middle\| s_{ij} = K\right\}\right)$	Variance of positions of K's in F^b

Positional features capture the specific locations of certain values within a F^b, which can be critical for identifying patterns and anomalies. These features include the positions of the first occurrences of '1's and '0's, as shown in Equation (2), and for the last occurrences of '1's and '0's, as shown in Equation (3).

$$P_{K_{Fj}} = \min(\{j|s_{ij} = K\}) \tag{2}$$

$$P_{K_{Lj}} = \max(\{j|s_{ij} = K\}) \tag{3}$$

These positional metrics help to understand the spatial distribution of features across the F^b, which can be especially useful in sequence analysis. As illustrated, these features provide valuable context that complements the statistical measures.

Entropy-based features quantify the randomness or unpredictability within a dataset, providing a measure of its complexity. Shannon entropy (E_i) is calculated for each j^{th} feature in F^b, where higher entropy values indicate greater unpredictability. This measure considers the probabilities of observing '1's and '0's [34].

$$E_j = -\left(p_{1_j}\log_2\left(p_{1_j}+\epsilon\right) + p_{0_j}\log_2\left(p_{0_j}+\epsilon\right)\right),\ p_{K_j} = C_{K_j} \tag{4}$$

Entropy is particularly useful for identifying homogeneity or variability within data, making it a key feature for classification tasks. As illustrated in Equation (4), entropy captures the inherent uncertainty in the dataset, thus informing model development and evaluation.

3.1.3. DS3: Normalization DS2

The extracted features (DS2) were normalized using min–max normalization, scaling the values between -1 and 1. Each feature (F_{ij}^{NS}) was normalized according to formula (5), where *min* (F_j^S) and *max* (F_j^S) represent the minimum and maximum values of the j^{th} feature across all taxa [36].

$$F_{ij}^{NS} = \begin{cases} 2\left(\frac{F_{ij}^S - \min\left(F_j^S\right)}{\max\left(F_j^S\right) - \min\left(F_j^S\right)}\right) - 1 & , \ \min(F_j^S) \neq \max(F_j^S) \\ 0 & , \ \min(F_j^S) = \max(F_j^S) \end{cases} \tag{5}$$

If the minimum and maximum values are equal, the normalized feature value is set to zero. An advantage of DS3 is that is ensures all features contribute equally to the analysis, preventing any feature from dominating due to scale differences.

3.2. Cross-Validation and Data Transformation

Cross-validation was employed to validate model robustness. The dataset was split into training and testing sets for each test (DS1, DS2, and DS3). The training dataset was used to train the machine learning models, while the testing dataset evaluated model performance. We applied k-fold cross-validation, where k was set to 2, 3, and 4, to assess the effect of different training and testing sizes on each model.

We focus on the features that cause the least amount of change in the phylogenetic tree. Specifically, we select only the features in bin 1—shown in Figure 3—which cause only one change, where δ is the threshold, as shown in Equation (6).

$$\Omega_j = \begin{cases} 1 & if\ 0 < \Delta F_j^b \leq \delta \\ 0 & Otherwise \end{cases} \tag{6}$$

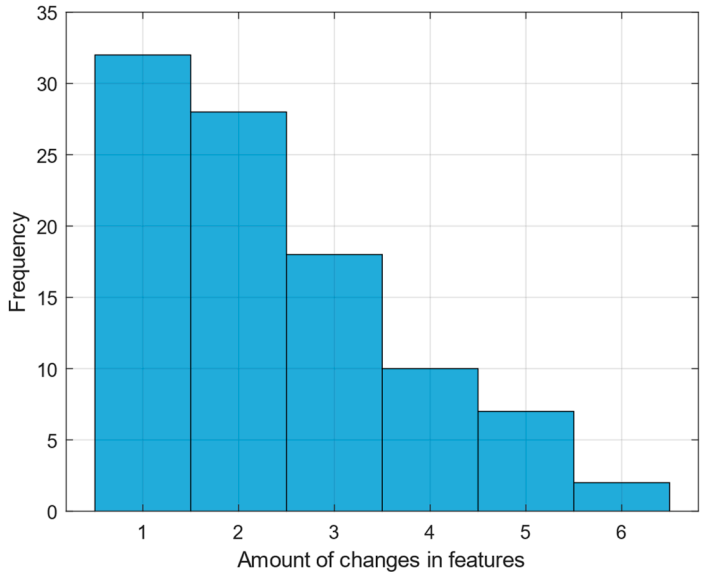

Figure 3. A histogram of features that show a similar number of changes.

3.3. Classification Phase

Three classifiers—DNN, SVM, and RF—were used to analyze the impact of features on the phylogenetic tree. The hyperparameter setups for each model are listed in Table 3.

Table 3. Hyperparameters for each model.

DNN	Three hidden layer there sizes: [15 ׀ 8 ׀ 4] Mean squared error: 0.001 Learning rate: 0.001 Actvation function to hidden layers (tansig) Actvation function for output node is (logsig)
SVM	Kernel function: radial basis function (RBF) Box constraint: 30 Kernel scale: 10
RF	Number of trees: 300 Max number of splits: 50 Number of variables to sample: all Minimum leaf size: 5

The DNN model was trained on the training dataset, with weights optimized to minimize the loss function. We explored various network topologies and parameter settings to optimize performance and prevent overfitting. This process involved iterative adjustments to the network architecture, learning rate, and performance parameters (MSE), as detailed in Table 3. After generating raw output probabilities from the DNN, a thresholding mechanism was applied to convert these probabilities into binary classifications, as shown in Equation (7).

$$Y_{predR} = \begin{cases} 1 & if\ Y_{predR} > 0.5 \\ 0 & if\ Y_{predR} \leq 0.5 \end{cases} \quad (7)$$

This adjustment ensures that the classifier output is given in a binary form, making it easier to evaluate the classification accuracy.

The SVM algorithm identifies the optimal hyperplane that separates features based on their labels (Ω), as illustrated in Figure 2. The trained SVM model was then used to predict the labels of the testing dataset, and performance metrics were computed. The SVM model setup is detailed in Table 3.

An ensemble of decision trees was trained on various subsets of the training dataset. The performance of the random forest model was then evaluated on the testing dataset.

3.4. Experimental Setup

Experimental analyses were conducted using PAUP* version 4.0a (build 168) for Unix/Linux. The server utilized an Intel® Xeon® CPU E5-2640 v2 @ 2.00 GHz with 24 CPU cores. This setup, optimized for Intel® 64 architecture and compiled with GNU C compiler (gcc) version 4.4.7, supported SSE vectorization, SSSE3 instructions, and multithreading via Pthreads. In parallel, our method 'FIPPA' was deployed, using MATLAB R2023b on the same machine to perform neural network estimations.

4. Results

4.1. Model Performance on Original Dataset

This study evaluates the performance of three machine learning models, namely, DNN, SVM, and RF, on three datasets: DS1 (F^b), DS2 (F^S), and DS3 (F^{NS}). The evaluation metrics include accuracy (Acc), the false acceptance rate (FAR), the false rejection rate (FRR), the area under the curve (AUC), and the equal error rate (EER).

Figure 3 illustrates the distribution of features based on the number of changes they induce, showing that most features cause only one change. This distribution emphasizes the importance of focusing on the most impactful features to reduce the complexity of the phylogenetic tree.

Table 4 summarizes the trade-offs between the number of features selected and the resulting phylogenetic tree's consistency and length. Selecting features that cause fewer changes simplifies the tree and enhances its consistency, as indicated by higher Consistency Index (CI) values.

Table 4. The number of bins selected according to Equation (6).

δ	No. Features	Tree Length	Optimal Tree	CI	Time Sec
6	97	229	2	0.424	369.3
5	95	217	3	0.438	184.7
4	88	181	2	0.486	28.8
3	78	140	3	0.557	1.33
2	60	86	2	0.698	0.03
1	32	32	1	1	0.005

Main finding: Focusing on features that cause minimal changes results in a shorter and more consistent phylogenetic tree. For example, only selecting features that induce one change produced a cladogram with a tree length of 32 and a perfect CI score of 1.0.

Figure 4 compares two phylogenetic trees obtained using a maximum parsimony search. Figure 4A includes all 97 features, resulting in a tree length of 229 and a CI score of 0.424. In contrast, Figure 4B features a subset of features that cause only one change, significantly simplifying the tree structure to a length of 32 with a perfect CI score of 1.0.

Moreover, the analysis shows the number of features selected at different δ (threshold) values, along with their corresponding tree lengths and CI values. As δ decreases in (6), fewer features are selected, resulting in shorter tree lengths and higher CI values. This suggests better consistency in the phylogenetic trees.

The table also includes the optimal tree and Time Sec columns. The optimal tree column indicates the number of optimal trees found after performing a branch and bound search, all having similar maximum parsimony scores. Generally, fewer optimal trees are found as δ decreases, reflecting more stable feature selection. The Time Sec column shows that computation time drops significantly with lower δ values, from 369.3 s at $\delta = 6$ to nearly 0 s at $\delta = 1$, highlighting the efficiency gained through feature reduction.

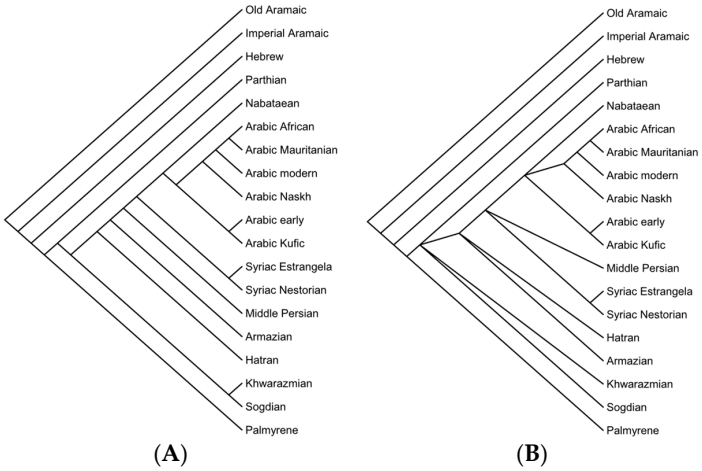

Figure 4. Phylogenetic trees after performing a maximum parsimony search including all features, as in (**A**), and a subset of features, as in (**B**).

Given the nondeterministic nature of DNNs, SVMs, and RFs, we performed each test 50 times to ensure stability and reliability. By averaging the outcomes, we mitigated random variations, ensuring the results reflected the true performance of each model across different datasets and folds

Figures 5–7 present ROC curves for DNNs, SVMs, and RFs models across different k-fold sizes (k = 4, k = 3, and k = 2). Each figure shows the ROC curves, AUC values, and EER locations for use in comparative performance analysis.

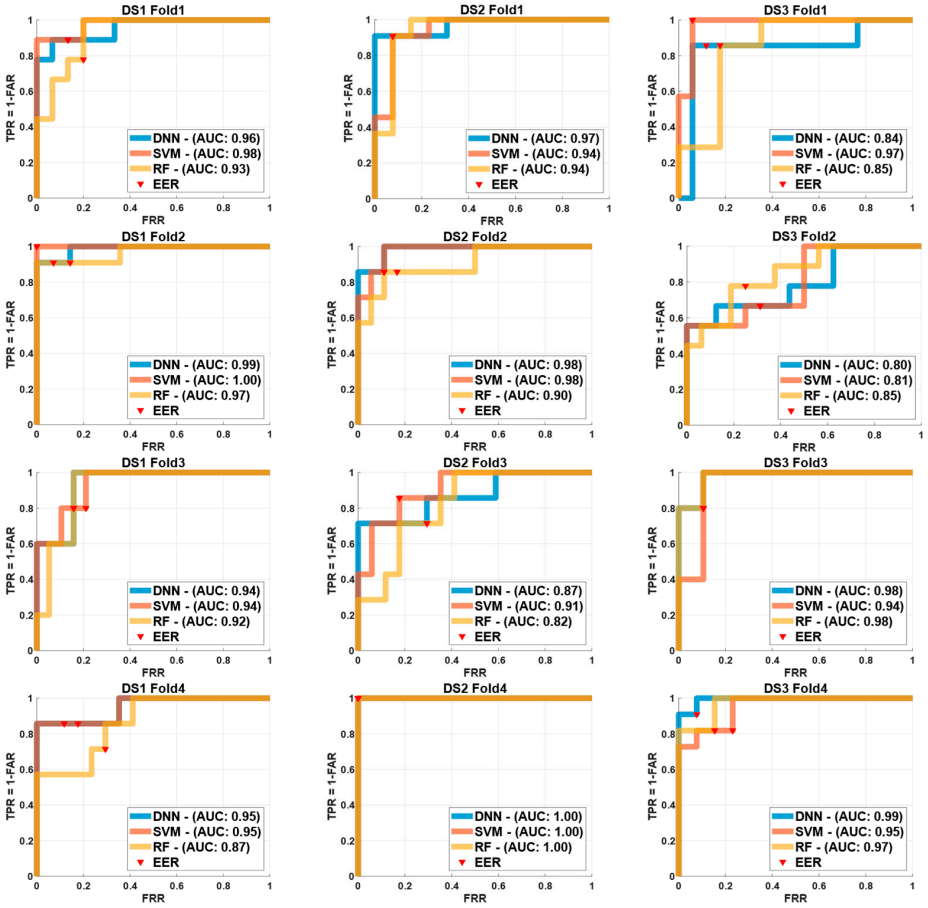

Figure 5. ROC curves of DNN, SVM, and RF models across 4 folds for $DS1 \equiv F^b$, $DS2 \equiv F^S$ and $DS3 \equiv F^{NS}$.

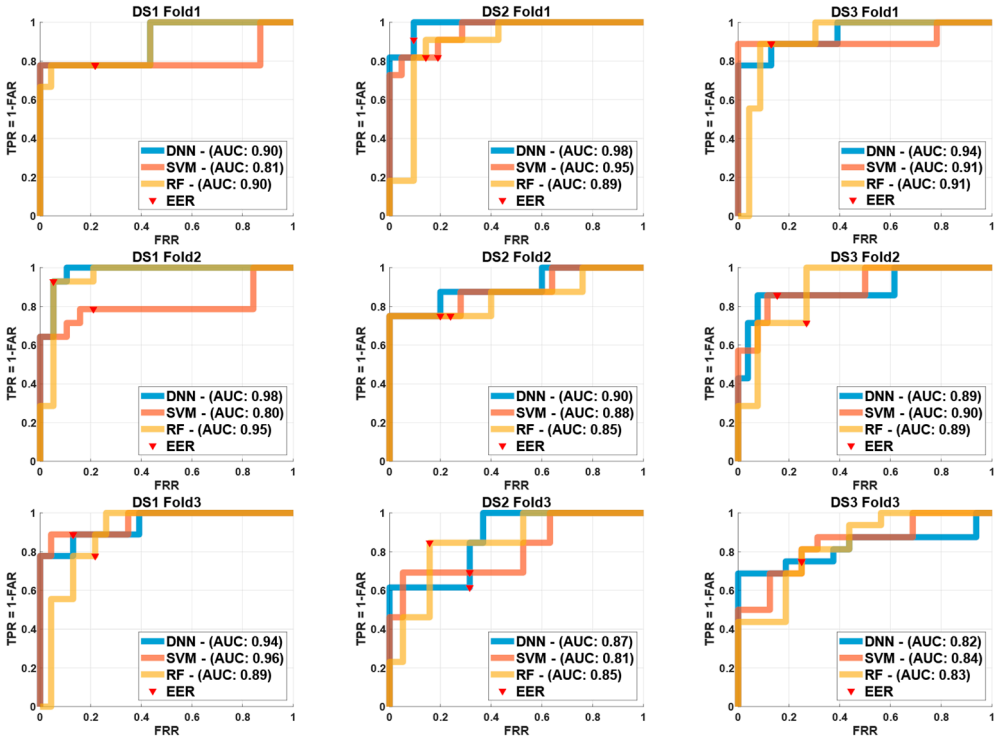

Figure 6. ROC curves of DNN, SVM, and RF models across 3 folds for $DS1 \equiv F^b$, $DS2 \equiv F^S$ and $DS3 \equiv F^{NS}$.

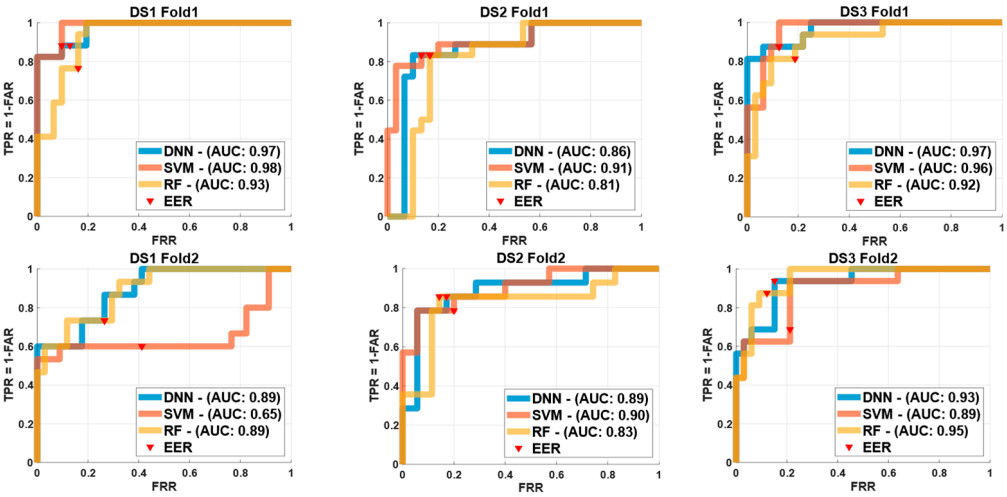

Figure 7. ROC curves of DNN, SVM, and RF models across 2 folds for $DS1 \equiv F^b$, $DS2 \equiv F^S$ and $DS3 \equiv F^{NS}$.

In case of k-4 folds = 4, Figure 5 illustrates the ROC curves for the three different machine learning models—DNN, SVM, and RF—across four folds (k = 4) for each of the three datasets (DS1, DS2, and DS3). Each row in the figure corresponds to a different fold (Fold 1 to Fold 4), and each column corresponds to a different dataset. The ROC curves for each model are plotted, with the area under the curve (AUC) values annotated for comparative performance analysis. Additionally, the equal error rate (EER) locations are marked on each curve.

In Figure 5, the SVM model demonstrated a superior performance for F^b in Fold 1, with an AUC value of 0.98, while all models showed similar performances for F^S, with DNNs slightly outperforming others for F^{NS}. In Fold 2, SVMs again outperformed the others for F^b and F^S, with DNNs leading for F^{NS}. In Fold 3, DNNs achieved the highest

AUC values for F^b and performed similarly well for F^S, and both DNNs and SVMs showed equal AUC values for F^{NS}. Finally, in Fold 4 SVM maintained the highest AUC value for F^b while DNNs outperformed for F^S and F^{NS}. These results highlight the robustness and variable effectiveness of each model across different datasets and folds, underscoring the importance of selecting a model suited to the specific dataset.

Additionally, Figure 6 presents the ROC curves for the DNN, SVM, and RF models across three folds (k = 3) for each of the three datasets (F^b, F^S, and F^{NS}). All rows relate to a different fold and each column relates to a different dataset. The ROC curves are plotted, with AUC values marked for comparison, and EER locations are marked on each curve for reference.

As shown in Fold 1 of Figure 6, DNNs outperformed other models for F^b with $AUC = 0.90$; on other hand, SVMs had the highest AUC value for F^S, equal to 0.95, and RF use on F^{NS} yielded $AUC = 0.91$. In Fold 2, SVM and RF performed similarly well for F^b, with both achieving an AUC value of 0.85. DNNs were best for F^S, with $AUC = 0.89$, while RF models maintained the lead for F^{NS}, with $AUC = 0.89$. In Fold 3, DNNs showed the best performance using F^b, with $AUC = 0.96$. SVMs intended for F^S yielded $AUC = 0.88$ and RFs continued to perform effectively for F^{NS} with $AUC = 0.89$.

Moreover, Figure 7 describes the ROC for DNN, SVM, and RF models across $k = 2$. Applying two folds to each of the three datasets, we obtain F^b, F^S, and F^{NS}. Each row relates to a different fold and each column to a different dataset. The ROC curves are plotted alongside AUC values and EER locations are marked on each curve.

In Figure 7, particularly Fold 1, the DNN showed the highest performance for F^b with $AUC = 0.97$, whereas SVMs displayed better performance in F^S with $AUC = 0.91$. For F^{NS}, RFs had the greatest AUC value of 0.95. In Fold 2, the application of DNNs to F^b yielded $AUC = 0.89$ and SVM showed a good performance for F^S with $AUC = 0.86$. RF continued to perform best for F^{NS}, with $AUC = 0.89$. These results indicate the varying strengths of each model across different datasets and folds, highlighting the importance of selecting the most suitable model based on dataset characteristics in order to achieve an ideal performance.

Table 5 presents the average accuracy (Acc), FRR and FAR for the DNN, SVM and RF models across different folds when applied to datasets F^b, F^S, and F^{NS}. This thorough comparison shows the values of each metric, demonstrating the efficiency of each model.

Table 5. Average accuracy, false rejection rate, and false acceptance rate of different models across various folds for $F^b \equiv DS1$, $F^S \equiv DS2$, and $F^{NS} \equiv DS3$.

	Fold	F^b			F^S			F^{NS}		
		Acc	**FRR**	**FAR**	**Acc**	**FRR**	**FAR**	**Acc**	**FRR**	**FAR**
DNN	2	81.12	0.09	0.32	79.74	0.08	0.35	84.26	0.1	0.24
	3	80.45	0.09	0.3	85.07	0.1	0.21	83.9	0.13	0.25
	4	83.66	0.06	0.31	90.3	0.05	0.18	86.37	0.09	0.22
SVM	2	88.71	0.13	0	88.56	0.1	0.14	83.19	0.17	0.13
	3	87.65	0.15	0	83.23	0.13	0.22	79.56	0.2	0.11
	4	92.75	0.09	0	86.43	0.11	0.17	84.61	0.13	0.17
RF	2	82.65	0.11	0.27	82.26	0.08	0.31	85.66	0.08	0.24
	3	78.64	0.13	0.25	82.4	0.1	0.29	77.01	0.18	0.33
	4	81.8	0.14	0.28	82.79	0.09	0.29	84.16	0.12	0.19

Tables 5 and 6 present a detailed comparison of accuracy (Acc), FRR, FAR, AUC, and EER values across different folds and datasets.

Table 6. Average AUC and EER values of different models across various folds for $F^b \equiv DS1$, $F^S \equiv DS2$, and $F^{NS} \equiv DS3$.

	Fold	F^b		F^S		F^{NS}	
		AUC	**EER**	**AUC**	**EER**	**AUC**	**EER**
DNN	2	0.92	0.19	0.87	0.16	0.94	0.13
	3	0.94	0.13	0.91	0.2	0.88	0.17
	4	0.95	0.13	0.95	0.12	0.9	0.15
SVM	2	0.81	0.25	0.9	0.16	0.92	0.16
	3	0.85	0.18	0.88	0.24	0.88	0.17
	4	0.96	0.11	0.95	0.09	0.91	0.17
RF	2	0.91	0.21	0.82	0.15	0.93	0.15
	3	0.91	0.16	0.86	0.18	0.87	0.21
	4	0.91	0.19	0.91	0.13	0.91	0.17

Main findings:

- DNNs consistently delivered strong performances across all datasets and fold sizes, with AUC values ranging from 0.87 to 0.95 and EER values between 0.12 and 0.19.
- SVMs also demonstrated robust performances, particularly for F^b and F^S, with AUC values as high as 0.96 and 0.95, respectively, at k = 4. However, its performance when applied to F^{NS} was slightly lower, with AUC values between 0.88 and 0.92.
- RFs displayed more variable performances, with AUC values ranging from 0.82 to 0.91 across datasets. Despite this variability, RF maintained relatively low EER values, particularly for F^{NS}.

Lastly, Figure 8 illustrates the number of epochs required for the DNN training across different folds and datasets (F^b, F^S, F^{NS}) for k = 2, k = 3, and k = 4. Each bar represents the number of epochs needed to achieve the final model performance for each fold within the respective k-fold cross-validation setups.

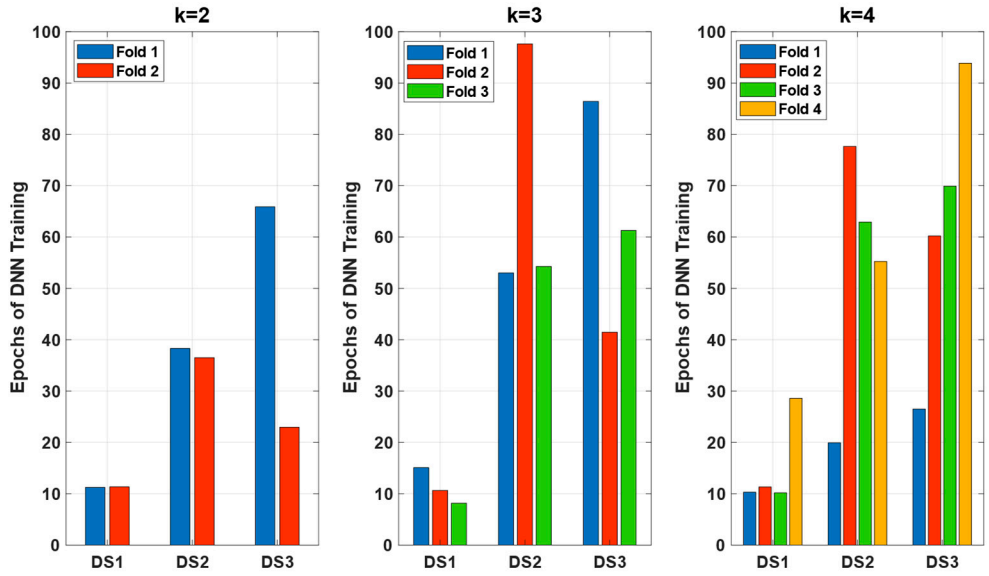

Figure 8. Training epochs required for DNNs across different folds and datasets ($F^b \equiv DS1$, $F^S \equiv DS2$, and $F^{NS} \equiv DS3$) for k = 2, k = 3, and k = 4.

Figure 8 illustrates the number of epochs required for DNN training across different folds and datasets (F^b, F^S, F^{NS}) for k = 2, k = 3, and k = 4. The number of epochs varied significantly across different datasets and folds. F^S consistently required more epochs, reflecting its higher complexity and the need for more iterations to achieve an optimal performance.

When k = 3, the variability in the number of epochs increased, particularly for F^S, which again required the most training epochs across all folds, highlighting its complexity. F^{NS} also showed a substantial increase in epochs needed for Fold 2, suggesting variability in the training process.

For k = 4, F^S continued to demand a high number of epochs, with Fold 1 showing the maximum epochs among all the datasets and folds. F^{NS}, however, showed more consistency across folds, indicating a more stable training process for this dataset under the k = 4 setup.

The analysis shows that F^S consistently required more training epochs across all k values, reflecting its higher complexity and the model's need for more iterations if it is to learn effectively. F^b generally required fewer epochs, suggesting it was less complex and easier for the DNN to train. This variability in training epochs across datasets and folds underscores the importance of considering dataset complexity and ensuring adequate training to achieve optimal model performance.

4.2. Validation Using External Dataset

To validate the generalizability of our feature impact classifiers, we applied them to an external dataset from Hoffmann et al. (2021) [37], which comprised 20 languages and 1359 features. In order to reduce the number of features, we deleted all unknown features, leaving us with 1119 features. The extent of alteration to these features is illustrated in Figure 9. This dataset enabled us to evaluate the models trained on our dataset (19 taxa and 97 features) to ascertain whether the machine learning classifiers, designed to predict feature impact, could accurately generalize to novel linguistic data.

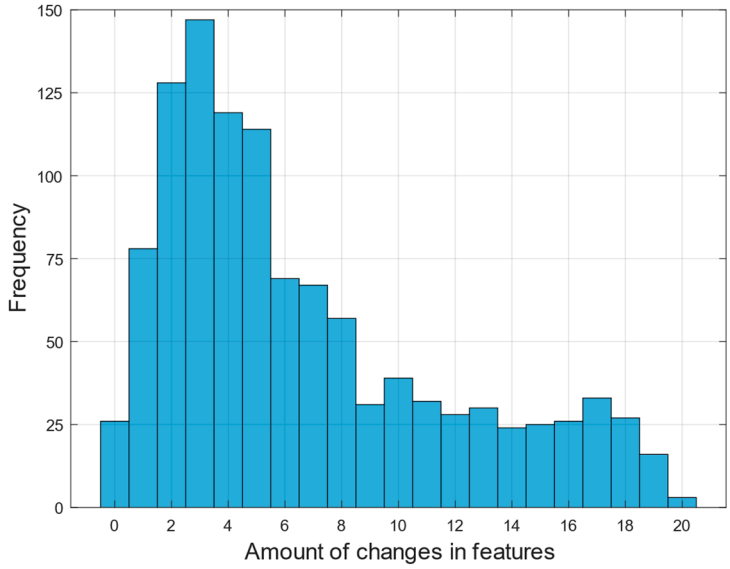

Figure 9. Histogram showing the distribution of the number of mutations (changes) per feature, with the same constraints applied to a dataset from [37].

As illustrated in Figure 9, there are notable differences between this dataset and our previous one, especially when compared to Figure 3. In this dataset, there are some features that exhibit no change, which is why Equation (6) constrains ΔF_j^b between the open and close interlevel $(0,\ \delta]$.

The objective of this subsection is to evaluate the efficacy of training classification algorithms, specifically DNN, SVM and RF algorithms, on a single dataset, and to subsequently test their performances on distinct datasets, namely, F^S and F^{NS}. This ensures that the size of the inputs for any classifier is equal, as illustrated by the ROC curves in Figure 10.

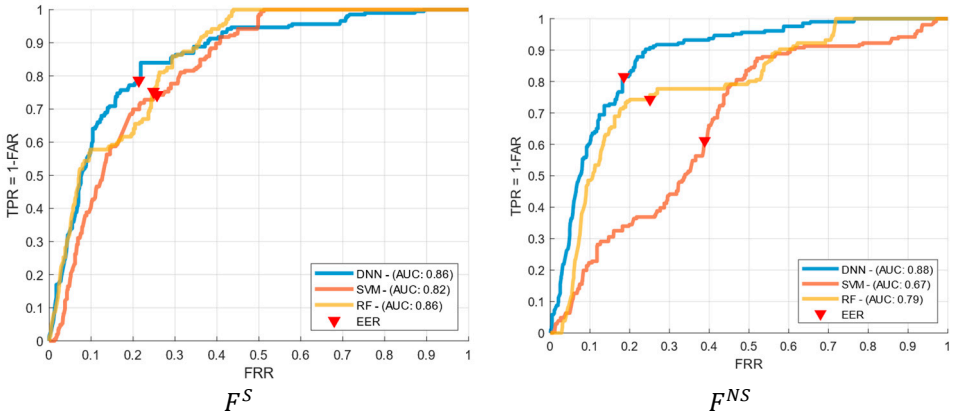

Figure 10. ROC curves of DNN, SVM, and RF models for $DS2 \equiv F^S$ and $DS3 \equiv F^{NS}$.

Figure 10 illustrates the capacity of our classifiers to predict the influence of individual features on phylogenetic analysis. By identifying homology-promoting features at an early stage, these classifiers facilitate the streamlining of phylogenetic tree construction, enabling researchers to concentrate on features that are more likely to yield more reliable trees. Conversely, by identifying features that are susceptible to homoplasy, the models assist in the identification of regions where borrowing or parallel evolution may be occurring, thereby providing insights that would otherwise require extensive post-analysis work.

The successful cross-validation across different datasets demonstrates that our models, particularly those utilizing the F^S and F^{NS} representations, are capable of generalizing beyond the original training data, thereby making them a valuable tool for broader phylogenetic studies across diverse datasets.

5. Discussion

This study explored the impact of feature selection on phylogenetic tree reconstruction using machine learning algorithms. The results underscore the critical role of effective feature selection in enhancing the accuracy and reliability of phylogenetic analyses. We applied DNN, SVM, and RF models to the binary dataset (F^b) and preprocessed datasets (F^S and F^{NS}), comparing their performance across different fold sizes.

The use of the binary dataset F^b offers simplicity, facilitating easier implementation and faster processing times. However, its limitation lies in the varying input sizes required for machine learning algorithms, which restricts its applicability to datasets of different sizes.

In contrast, preprocessed features F^S and normalized features F^{NS} introduce additional complexity due to preprocessing steps. However, they maintain a fixed input size, allowing machine learning models to be applied across datasets of varying sizes. This flexibility is crucial for generalizing models to a broader range of phylogenetic analyses, ensuring robust performance across different scenarios.

Our findings show that DNNs consistently performed well across all datasets and folds, achieving the highest AUC values and maintaining low EER values, particularly for F^b and F^S. This indicates that DNNs effectively capture complex patterns within the data, making it a reliable choice.

SVM also demonstrated strong performance, particularly for F^b and F^S, with high AUC values. However, its performance was slightly lower on F^{NS}, aligning with the known strengths of SVMs, which excel with well-defined boundaries but may struggle with more complex or noisy data.

RF models showed more variable performances, with AUC values ranging from 0.82 to 0.91 across datasets. Despite this variability, RF models maintained relatively low EER values, particularly for F^{NS}. The ability of RF models ability to handle large datasets makes them valuable for identifying key evolutionary traits, even though their overall performance is somewhat less consistent compared to that of DNN and SVM models.

This study also highlights the importance of considering dataset complexity. F^S consistently required more training epochs for DNNs across all k values, reflecting its higher complexity and the need for more iterations to achieve effective learning. This underscores the necessity of adequate training to optimize model performance.

The successful validation across different datasets demonstrates that our models, especially those employing the F^S and F^{NS} representations, are capable of generalization beyond the initial training data. This underscores their potential as valuable tools for comprehensive phylogenetic analyses across diverse datasets. It is noteworthy that the F^S representation yielded more consistent and stable results, shown in Figure 10, thereby further reinforcing its reliability for evaluating the impact of features.

Finally, cross-validation provided a comprehensive assessment of each model's predictive capability. Given that DNNs, SVMs, and RFs are nondeterministic algorithms, running each with the same data and settings can yield slightly different results. To account for this variability, we conducted each test 50 times and averaged the outcomes. This approach reduced the effects of random variation and ensured the stability and reliability of our results, offering a more accurate evaluation of each model's performance.

6. Conclusions

This research presents a feature selection method designed to enhance phylogenetic reconstruction using machine learning techniques such as DNNs, SVMs, and RFs. Our results demonstrate that DNNs consistently outperformed other models in terms of AUC and EER values, showcasing a strong performance across various preprocessed datasets and folds. SVMs and RFs also performed well, although with some variability.

These machine learning techniques significantly enhance the accuracy and efficiency of phylogenetic analyses, providing powerful tools for evolutionary studies. This approach not only simplifies tree structure but also improves the Consistency Index (CI) values, providing deeper insights into evolutionary relationships.

However, there are some limitations to this study. The binary dataset (DS1) has limitations due to its requirement for datasets of the same size, which can restrict its applicability to more diverse datasets. While the transformations applied to create DS2 and DS3 mitigate these issues by standardizing input sizes, the preprocessing steps add complexity and may introduce challenges when dealing with extremely large datasets or those with high levels of noise. Additionally, the models were tested on specific datasets, and their performance on significantly different types of data (e.g., with more varied or complex evolutionary histories) needs to be fully explored.

Future research should aim to integrate these models to further improve the robustness of phylogenetic inference. Additionally, applying these techniques to more complex evolutionary scenarios, such as feature duplication, loss, and introgression, could offer even greater insights into evolutionary processes.

Author Contributions: Conceptualization, O.A.S. and G.H.; data curation, O.A.S. and G.H.; formal analysis, O.A.S.; funding acquisition, O.A.S. and G.H.; investigation, O.A.S. and G.H.; methodology, O.A.S.; project administration, O.A.S. and G.H.; resources, O.A.S. and G.H.; software, O.A.S.; supervision, O.A.S. and G.H.; validation, O.A.S.; visualization, O.A.S.; writing—original draft, O.A.S. and G.H.; writing—review and editing, O.A.S. and G.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The applied datasets can be found on GitHub [13].

Acknowledgments: The authors extend their sincere appreciation to their colleague Péter Pálovics from the Department of Electron Devices, Budapest University of Technology and Economics, for his invaluable assistance in configuring the server that played a critical role in our research. Additionally, we wish to express our gratitude to the Stipendium Hungaricum scholarship program for its vital support. The scholarship program has been instrumental in facilitating our research endeavors, contributing substantially to our academic development and the successful realization of our project.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. Semple, C.; Steel, M. *Phylogenetics*; Oxford University Press on Demand: Oxford, UK, 2003.
2. Salman, O.A.; Hosszú, G. Cladistic Analysis of the Evolution of Some Aramaic and Arabic Script Varieties. *Int. J. Appl. Evol. Comput. (IJAEC)* **2021**, *12*, 18–38. [CrossRef]
3. Salman, O.A.; Hosszú, G. Enhanced Phylogenetic Inference through Optimized Feature Selection and Computational Efficiency Analysis. *Acta Polytech. Hung.* **2024**. *under review*.
4. Salman, O.A.; Hosszú, G.; Kovács, F. A new feature selection algorithm for evolutionary analysis of Aramaic and Arabic script variants. *Int. J. Intell. Eng. Inform.* **2022**, *10*, 313–331. [CrossRef]
5. Salman, O.A.; Hosszú, G. Optimised feature dimension reduction method and its impact on the search for optimal trees. In Proceedings of the Workshop on the Advances of Information Technology, Budapest, Hungary, 6–7 February 2023; BME Department of Control Engineering and Information Technology: Budapest, Hungary, 2023.
6. Salman, O.A.; Hosszú, G. A Phenetic Approach to Selected Variants of Arabic and Aramaic Scripts. *Int. J. Data Anal.* **2022**, *3*, 1–23. [CrossRef]
7. Salman, O.A.; Hosszú, G. Phylogenetic Inference Using Advanced Feature Selection. In Proceedings of the 2023 14th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Budapest, Hungary, 22–23 September 2023; pp. 000173–000178.
8. Salman, O.A.; Hosszú, G. Phylogenetic modelling scripts for identifying script versions. *Procedia Comput. Sci.* **2024**, *239*, 1417–1424. [CrossRef]
9. Salman, O.A.; Hosszú, G. Using distance-based methods to calculate optimal and suboptimal parsimony trees. In Proceedings of the Workshop on the Advances of Information Technology, WAIT 2024, Budapest, Hungary, 6–7 February 2023; BME Department of Control Engineering and Information Technology: Budapest, Hungary, 2024.
10. Wu, C.H.; Chen, H.-L.; Chen, S.-C. Gene classification artificial neural system. *Int. J. Artif. Intell. Tools* **1995**, *4*, 501–510. [CrossRef]
11. Mo, Y.K.; Hahn, M.; Smith, M.L. Applications of Machine Learning in Phylogenetics. *Mol. Phylogenetics Evol.* **2024**, *196*, 108066. [CrossRef]
12. Zhou, Y.; Zheng, H.; Huang, X.; Hao, S.; Li, D.; Zhao, J. Graph neural networks: Taxonomy, advances, and trends. *ACM Trans. Intell. Syst. Technol. (TIST)* **2022**, *13*, 1–54. [CrossRef]
13. Available online: https://github.com/OsamaAliSalman/Extended_Arabic-Aramaic-DataSet.git (accessed on 2 August 2024).
14. Halgaswaththa, T.; Atukorale, A.S.; Jayawardena, M.; Weerasena, J. Neural network based phylogenetic analysis. In Proceedings of the 2012 International Conference on Biomedical Engineering (ICoBE), Penang, Malaysia, 27–28 February 2012; pp. 155–160.
15. Suvorov, A.; Schrider, D.R. Reliable estimation of tree branch lengths using deep neural networks. *bioRxiv* **2022**. [CrossRef]
16. Philippe, H.; Zhou, Y.; Brinkmann, H.; Rodrigue, N.; Delsuc, F. Heterotachy and long-branch attraction in phylogenetics. *BMC Evol. Biol.* **2005**, *5*, 1–8. [CrossRef]
17. Sullivan, J.; Swofford, D.L. Should we use model-based methods for phylogenetic inference when we know that assumptions about among-site rate variation and nucleotide substitution pattern are violated? *Syst. Biol.* **2001**, *50*, 723–729. [CrossRef] [PubMed]
18. Azouri, D.; Abadi, S.; Mansour, Y.; Mayrose, I.; Pupko, T. Harnessing machine learning to boost heuristic strategies for phylogenetic-tree search. *Prepr. Res. Sq.* **2020**. [CrossRef]
19. Bernardini, G.; van Iersel, L.; Julien, E.; Stougie, L. Constructing phylogenetic networks via cherry picking and machine learning. *Algorithms Mol. Biol.* **2023**, *18*, 13. [CrossRef] [PubMed]
20. Zou, Z.; Zhang, H.; Guan, Y.; Zhang, J. Deep residual neural networks resolve quartet molecular phylogenies. *Mol. Biol. Evol.* **2020**, *37*, 1495–1507. [CrossRef] [PubMed]
21. Layne, E.; Dort, E.N.; Hamelin, R.; Li, Y.; Blanchette, M. Supervised learning on phylogenetically distributed data. *Bioinformatics* **2020**, *36* (Suppl. 2), i895–i902. [CrossRef]
22. Smith, M.L.; Hahn, M.W. Phylogenetic inference using generative adversarial networks. *Bioinformatics* **2023**, *39*, btad543. [CrossRef]
23. Abadi, S.; Avram, O.; Rosset, S.; Pupko, T.; Mayrose, I. ModelTeller: Model selection for optimal phylogenetic reconstruction using machine learning. *Mol. Biol. Evol.* **2020**, *37*, 3338–3352. [CrossRef]
24. Lipták, P.; Attila, K. Constructing unrooted phylogenetic trees with reinforcement learning. *Studia Univ. Babeș-Bolyai Inform.* **2021**, 37–53. [CrossRef]

25. Kalyaanamoorthy, S.; Minh, B.Q.; Wong, T.K.; Von Haeseler, A.; Jermiin, L.S. ModelFinder: Fast model selection for accurate phylogenetic estimates. *Nat. Methods* **2017**, *14*, 587–589. [CrossRef]
26. Wang, Z.; Sun, J.; Gao, Y.; Xue, Y.; Zhang, Y.; Li, K.; Zhang, W.; Zhang, C.; Zu, J.; Zhang, L. Fusang: A framework for phylogenetic tree inference via deep learning. *Nucleic Acids Res.* **2023**, *51*, 10909–10923. [CrossRef]
27. Tang, X.; Zepeda-Nuñez, L.; Yang, S.; Zhao, Z.; Solís-Lemus, C. Novel symmetry-preserving neural network model for phylogenetic inference. *Bioinform. Adv.* **2024**, *4*, vbae022. [CrossRef] [PubMed]
28. Tadist, K.; Najah, S.; Nikolov, N.S.; Roose, L. Feature selection methods and genomic big data: A systematic review. *J. Big Data* **2019**, *6*, 79. [CrossRef]
29. Kaur, A.; Sarmadi, M. Comparative Analysis of Data Preprocessing Methods, Feature Selection Techniques and Machine Learning Models for Improved Classification and Regression Performance on Imbalanced Genetic Data. *arXiv* **2024**, arXiv:2402.14980.
30. Fawcett, T. An Introduction to ROC Analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]
31. Bradley, A.P. The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognit.* **1997**, *30*, 1145–1159. [CrossRef]
32. Jain, A.K.; Ross, A.; Prabhakar, S. An Introduction to Biometric Recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 4–20. [CrossRef]
33. Daugman, J. How Iris Recognition Works. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 21–30. [CrossRef]
34. Shannon, C.E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
35. Felsenstein, J. *Inferring Phylogenies*; Sinauer Associates: Sunderland, MA, USA, 2004.
36. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann: Burlington, MA, USA, 2011.
37. Hoffmann, K.; Bouckaert, R.; Greenhill, S.J.; Kühnert, D. Bayesian phylogenetic analysis of linguistic data using BEAST. *J. Lang. Evol.* **2021**, *6*, 119–135. [CrossRef]