Journal of Algorithms and Computations

journal homepage: http://jac.ut.ac.ir

Neural Network Feature Selection (NNFS) for Incomplete and High-Dimensional Data

N. Bagherpour[*1] and B. Ebrahimi[†1]

[1] Department of Engineering Sciences, College of Engineering, University of Tehran, Iran

ABSTRACT

Feature selection is a critical step in machine learning, especially when dealing with high-dimensional and incomplete data. Traditional methods often struggle with missing values, which are common in real-world applications. This paper introduces Neural Network Feature Selection (NNFS), a novel deep learning-based approach that effectively identifies important features even in the presence of missing data. A sensitivity analysis also conducted to verify the robustness.

Keywords: feature selection, high dimension, incomplete data, neural network

ARTICLE INFO

Article history:
Research paper
Received 05 July 2025
Accepted 11 August 2025
Available Online 11 August 2025

AMS subject classification: 68T07

[†] Corresponding author: N. Bagherpour, Email: negin.bagherpour@ut.ac.ir

Email: behrangebrahimi13@gmail.com

1 Introduction

Feature selection is a critical step in machine learning pipelines, especially when dealing with high-dimensional data. The goal of feature selection is to identify the most relevant features that contribute to model performance while reducing dimensionality and computational complexity. Traditional feature selection methods often struggle with incomplete datasets containing missing values, which are common in real-world applications.

This paper introduces Neural Network Feature Selection (NNFS), a novel deep learning-based approach that can effectively identify important features even in the presence of missing data. NNFS leverages the power of neural networks to learn robust feature representations without requiring data imputation.

The key contributions of this work include:

1. A new feature selection method that can handle incomplete and high-dimensional data simultaneously.
2. An architecture that treats missing values as inactive neurons, allowing the network to learn from incomplete samples.
3. A technique for extracting feature importance scores from the trained neural network weights.
4. Extensive experiments on multiple datasets demonstrating the effectiveness of NNFS compared to existing feature selection methods, especially in scenarios with missing data.

Feature selection is crucial for several reasons:

- Dimensionality reduction: By selecting only the most relevant features, NNFS helps reduce the curse of dimensionality, which can lead to overfitting and poor generalization in machine learning models.
- Improved model interpretability: Identifying key features provides insights into the underlying data patterns and relationships, making models more interpretable and explainable.
- Reduced computational complexity: Fewer features mean reduced training time and computational resources required for model training and inference.
- Enhanced generalization: Eliminating irrelevant or noisy features can improve a model's ability to generalize to unseen data.

Unlike traditional filter, wrapper, or embedded methods, NNFS integrates feature selection directly into the neural network training process, specifically designed to handle incomplete data without imputation.

The rest of this paper is organized as follows: Section 2 provides an overview of related work in feature selection and handling missing data. Section 3 describes the proposed NNFS method in detail, including its architecture and training process. Section 4 outlines the experimental setup, including datasets, baseline methods, and evaluation metrics. Section 5 presents and discusses the results of our experiments. Finally, Section 6 concludes the paper and suggests directions for future research.

2 Related Work

Feature selection methods have traditionally been applied to complete datasets. However, in recent years, there has been growing interest in developing feature selection approaches for incomplete data 1. These methods can be broadly categorized into three types:

2.1 Filter Methods

Filter methods rank features based on statistical criteria such as correlation, variance, or information gain. Some filter methods can handle missing data by computing statistics only on available data or by imputing missing values 2. For example, Chi-square, Gini Index, and Mutual Information are commonly used filter methods that can be adapted for incomplete data 3.

2.2 Wrapper Methods

Wrapper methods use a subset of features to train a model and evaluate its performance iteratively. Tran et al. 1 proposed a wrapper-based feature selection approach for incomplete data, using Particle Swarm Optimization as a search technique and C4.5 decision trees to evaluate feature subsets directly on incomplete data.

2.3 Embedded Methods

Embedded methods combine feature selection and model training in one step. For instance, LASSO, SCAD, and Elastic Net can be applied to imputed datasets for simultaneous feature selection and classification 4.

2.4 Handling Missing Data

Various strategies exist for dealing with missing data during feature selection:

1. Deletion: Complete case analysis or listwise deletion, which can introduce bias if data is not missing completely at random 4.
2. Imputation: Methods such as mean imputation, EM imputation, GAIN, KNN imputation, and DAE have been used to fill in missing values before feature selection 4.

3. Direct handling: Some algorithms, like certain decision tree methods, can work directly with incomplete data 1.

2.5 Ensemble Approaches

Ensemble methods, such as bagging combined with feature selection, have shown promise in improving classification with incomplete data 1. These approaches can enhance both accuracy and model simplicity.

2.6 Limitations of Existing Methods

Many existing feature selection methods do not adequately address the simultaneous challenges of high dimensionality and missing data 5. Additionally, most techniques do not consider the impact of feature selection on the underlying model architecture, such as neural network topology 5.

This research aims to address these limitations by proposing a novel approach that integrates missing data handling, feature selection, and model architecture optimization in a unified framework.

3 Proposed Method: Neural Network Feature Selection (NNFS)

3.1 Overview

NNFS leverages the power of deep neural networks to identify important features without requiring data imputation. The key idea is to treat missing values as inactive neurons, allowing the network to learn robust feature representations even with incomplete data.

3.2 Architecture

NNFS employs two different architectures:

1. Supervised Learning (Classification):
   A multi-layer neural network for classification tasks.
2. Unsupervised Learning:
   An autoencoder architecture for unsupervised feature selection.

Both architectures are designed to handle missing data by using a masking mechanism.

3.3 Training Process

The NNFS training process involves:

1. Data Preparation: Scaling features to (0,1], creating binary masks for missing values, and replacing missing values with zeros.
2. Network Training: Training the appropriate network (classifier or autoencoder) on the prepared data.

3. Iterative Training: Repeating the training process for multiple epochs until convergence.

3.4 Feature Importance Scoring

After training, NNFS calculates feature importance scores based on the network weights:

1. Weight Aggregation: Aggregating weights connected to each input feature.
2. Ranking: Ranking features based on their importance scores.

This method provides a comprehensive measure of each feature's impact on the model's predictions or reconstructions.

NNFS integrates feature selection directly into the neural network training process, offering a unified approach to handling incomplete data, selecting important features, and optimizing model architecture. This addresses the limitations of existing methods and provides a robust solution for high-dimensional, incomplete datasets in both supervised and unsupervised scenarios.

4 Experimental Setup

4.1 Datasets

Four datasets were used to evaluate NNFS:

1. Mice Protein Expression: A dataset containing protein expression levels in mice, suitable for both supervised and unsupervised learning tasks.
2. Mask Version 2: A dataset related to mask detection, used for classification tasks.
3. Handwritten Digits: The MNIST dataset, consisting of images of handwritten digits.
4. Spam Emails: A dataset of email messages, classified as spam or non-spam.

Table 1: Dataset Characteristics

Dataset	Samples	Features	Classes	Task Type	Subject Area
Mice Protein Expression	1080	80	8	Supervised/ Unsupervised	Biology
Mask Version 1	476	166	2	Supervised	Physics and Chemistry
Mask Version 2	6598	166	2	Supervised	Physics and Chemistry
Handwritten Digits	70000	784	10	Supervised	Computer Science

Spam Emails	4601	57	2	Supervised	Computer Science
Cylinder Bands	512	39	2	Supervised	Physics and Chemistry

4.2 Baseline Methods

NNFS was compared against the following feature selection methods:

1. Chi-squared
2. F-classif
3. Mutual Information

4.3 Evaluation Process

1. Features were scaled to the range (0, 1]
2. Missing data was simulated by randomly removing values at different rates
3. A binary mask was created to indicate missing values
4. Missing values were replaced with zeros

4.5 Classification Methods

The following classifiers were used to evaluate the selected features:

- Logistic Regression
- Neural Network
- Random Forest
- Support Vector Machine
- XGBoost

4.3 Evaluation Metrics

The primary evaluation metric used was classification accuracy. For each dataset, we measured:

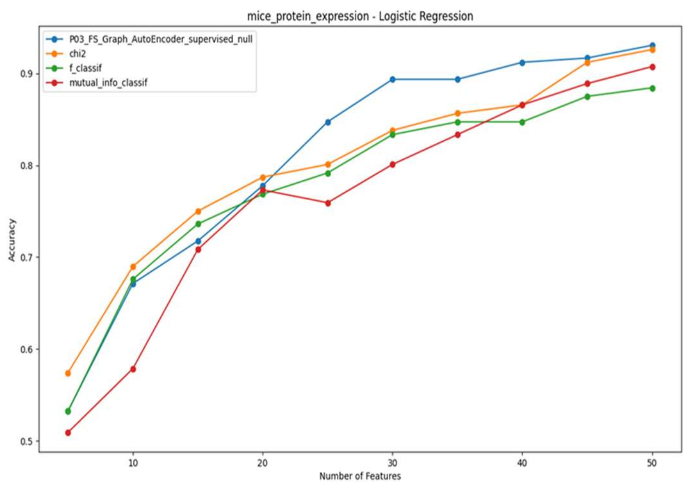
1. Overall accuracy: The proportion of correct predictions among the total number of cases examined.
2. Accuracy vs. number of selected features: To assess how the number of selected features impacts model performance.
3. Accuracy vs. missing data rate: To evaluate the robustness of each method to incomplete data.

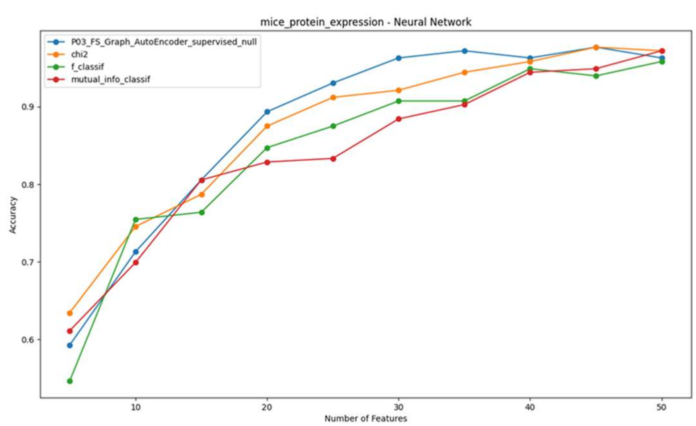
5 Results and Discussion

5.1 Performance on Complete Data

5.1.1 Mice Protein Expression Dataset

The NNFS method, using an autoencoder approach, showed excellent performance in identifying important features without using labels. Compared to label-based methods like Chi2, f_classif, and mutual_class_info, our proposed method demonstrated superior performance. This indicates that the autoencoder can identify important features related to the target, even when labels are unavailable.





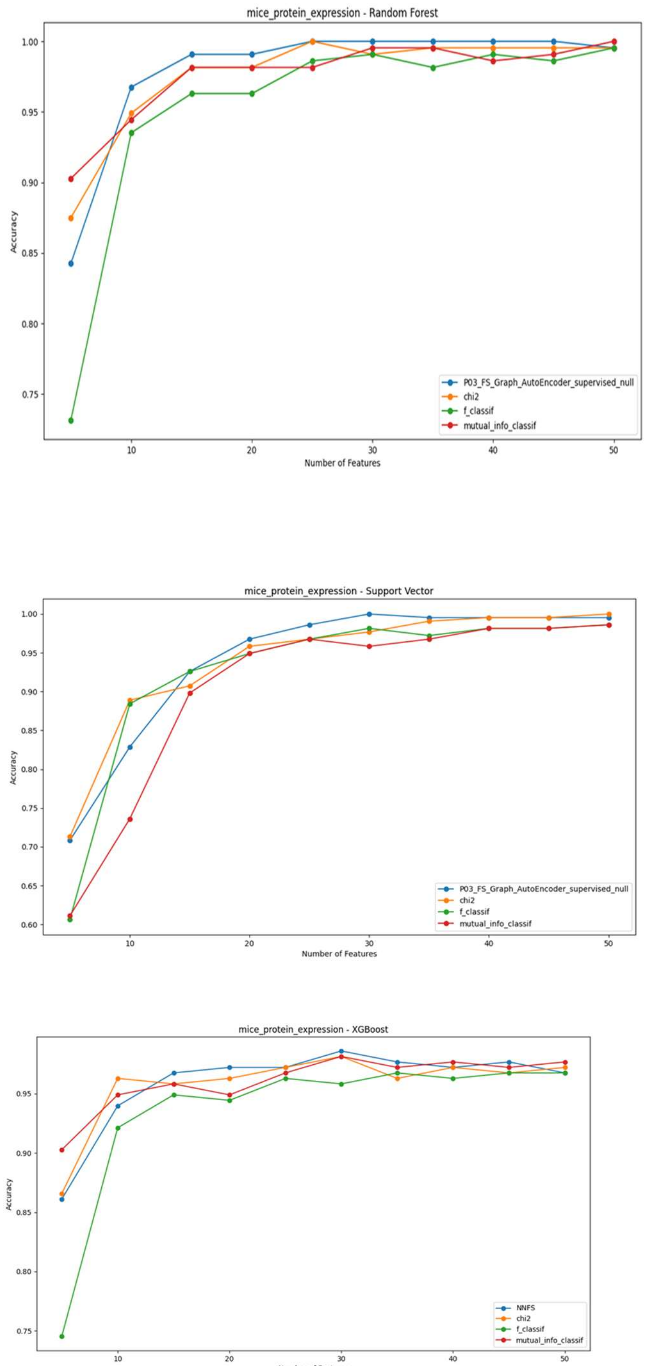
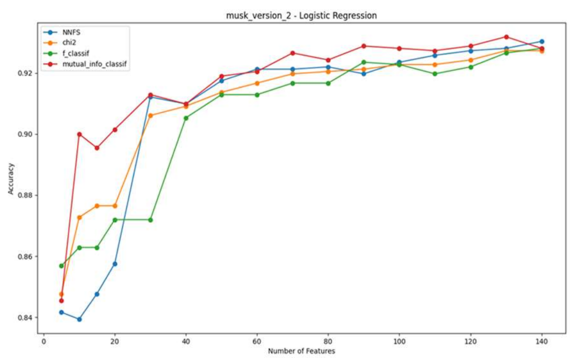
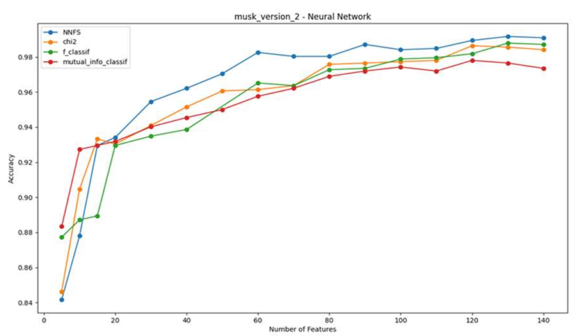


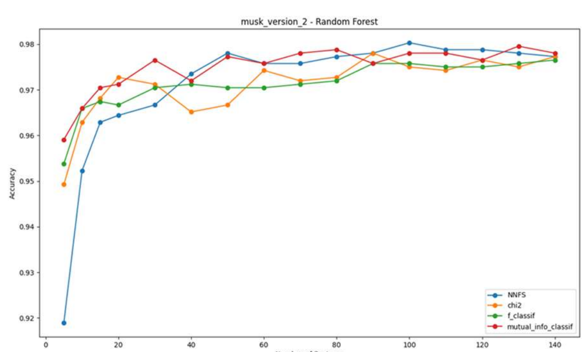
Figure 1: Comparison of NNFS with existing methods on the Mice Protein Expression dataset

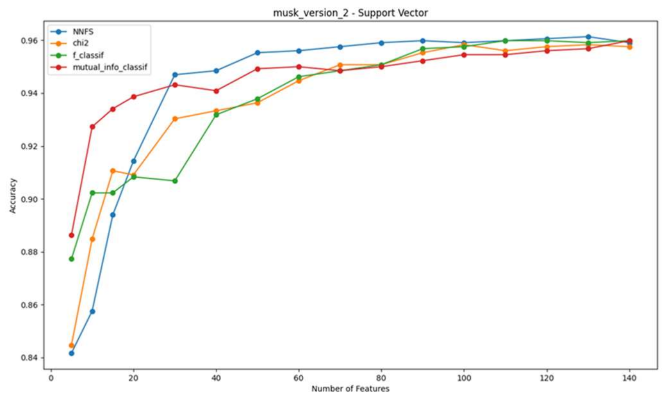
5.1.2 Mask Version 2 Dataset

Similar results were observed for the Mask Version 2 dataset, further confirming the effectiveness of the NNFS method in identifying key features without relying on labels.









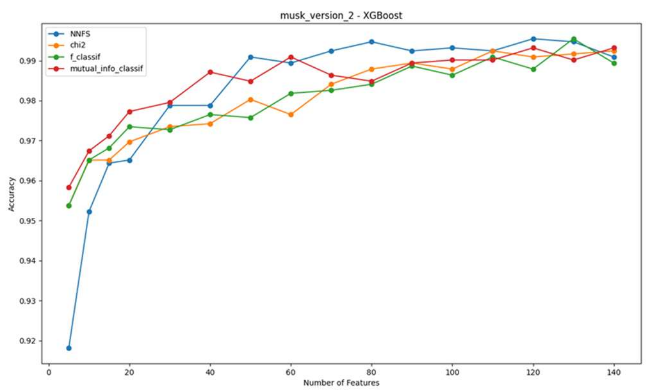


Figure 2: Comparison of NNFS with existing methods on the Mask Version 2 dataset

5.1.3 Spam Emails Dataset

Initial results using the autoencoder approach on the Spam Emails dataset were not as promising, with lower accuracy compared to other methods. This suggested that for some supervised datasets, the relationship between features and the target class can be crucial.

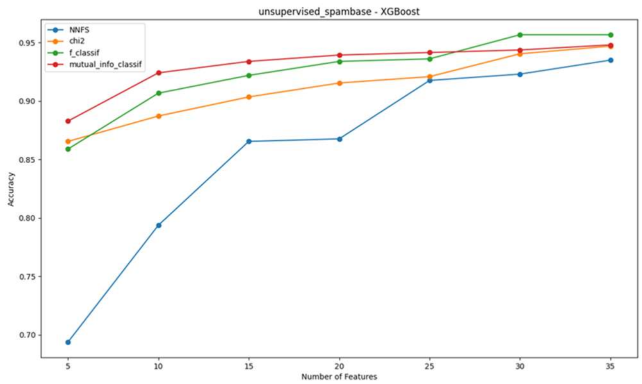


Figure 3: Comparison of NNFS with autoencoder technique and existing methods on the Spam Emails dataset

To improve results, a standard neural network was employed, using a sigmoid activation function in the output layer. This approach allowed for direct consideration of the relationships between features and classes, leading to significantly improved results.

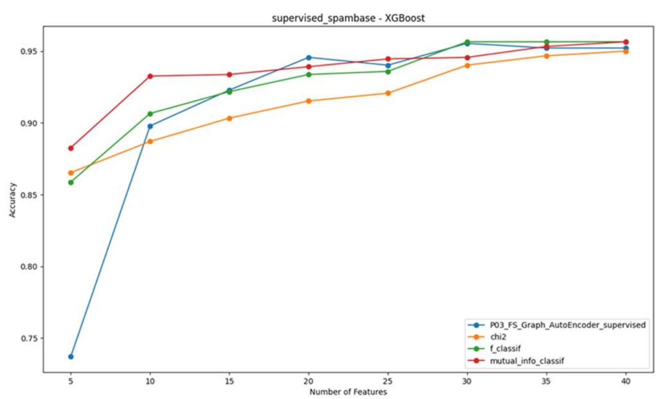


Figure 4: Comparison of NNFS with standard neural network technique and existing methods on the Spam Emails dataset

Effect of Activation Function

Changing the activation function to hyperbolic tangent (tanh) showed notable improvements in results. This change is clearly visible in the charts, with a more pronounced optimal number of features for achieving desired accuracy.

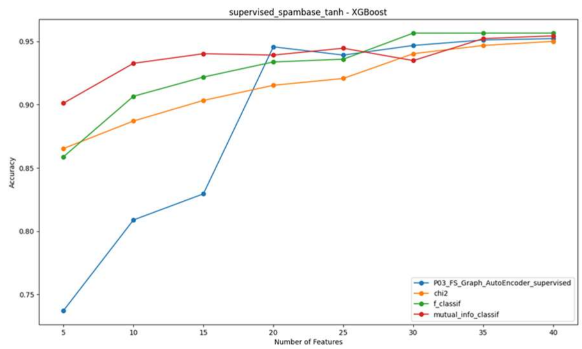
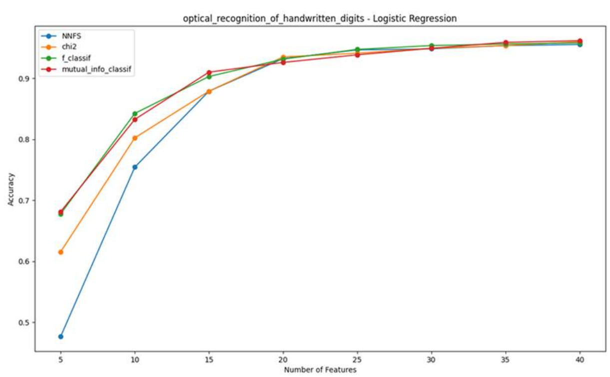
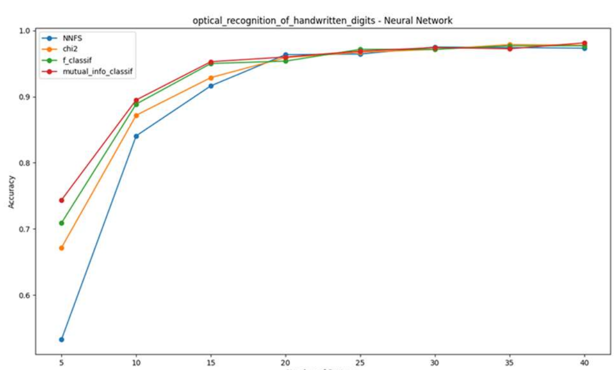


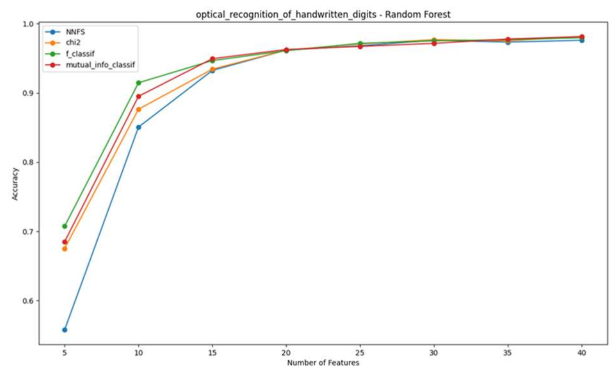
Figure 5: Using Tanh as the activation function
Comparison of NNFS with standard neural network technique, tanh activation function, and existing methods on the Spam Emails dataset

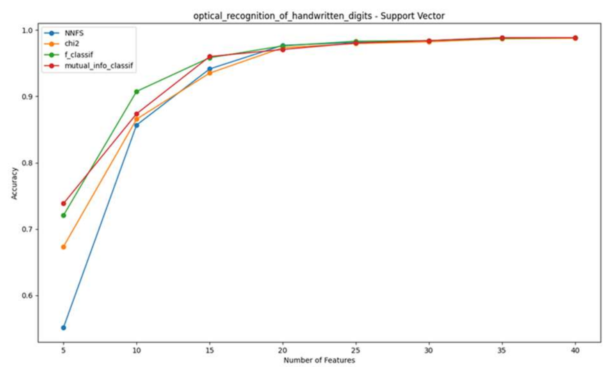
5.1.4 Handwritten Digits Dataset

The method was also applied to the Handwritten Digits dataset, showing consistent results and minimal differences compared to other methods, further validating the effectiveness and generalizability of the proposed approach.









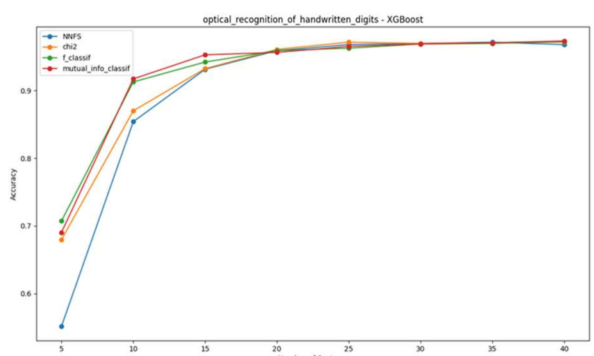


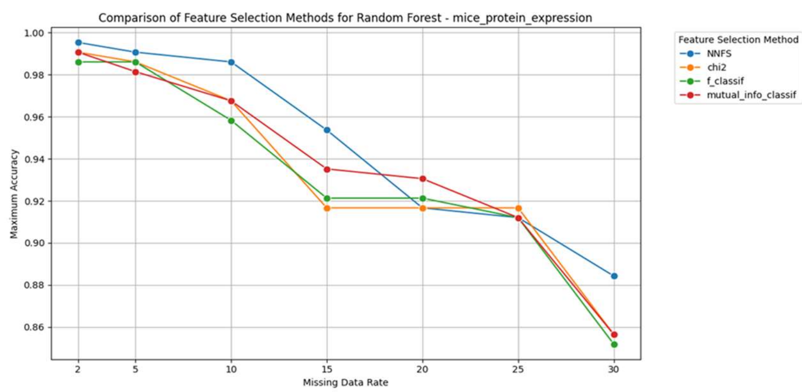
Figure 6: Comparison of NNFS with standard neural network technique and existing methods on the Handwritten Digits dataset

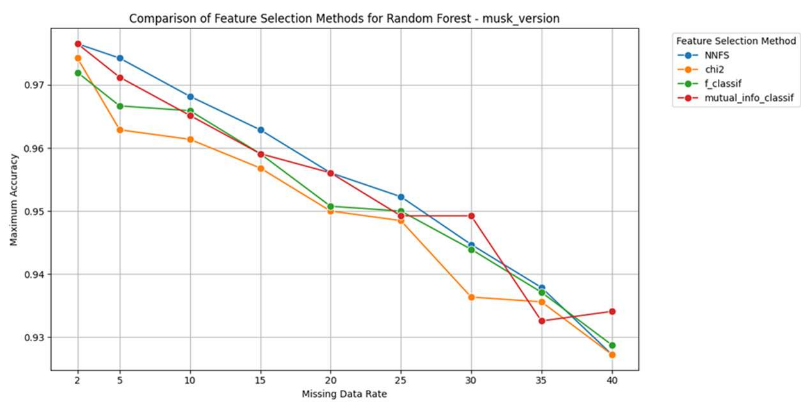
5.2 Performance with Missing Data

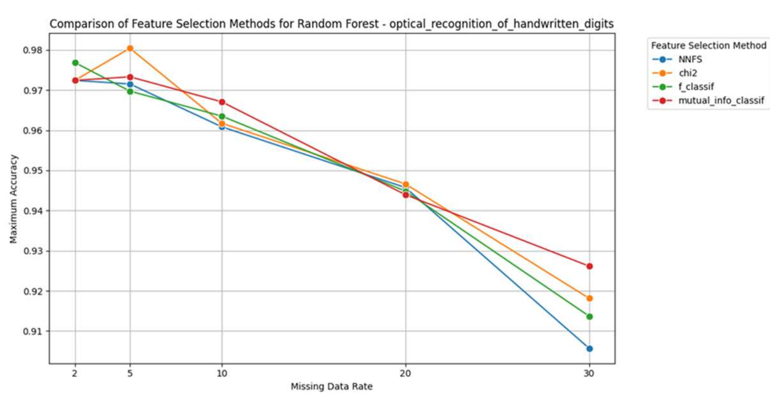
In this section, we evaluate the performance of the NNFS method in handling datasets with missing data. We conducted experiments on four datasets: Mice Protein Expression, Mask Version 2, Handwritten Digits, and Spam Emails. For each dataset, we introduced missing values at various rates and assessed the accuracy of feature selection using the Random Forest classifier.

5.2.1 Comparison Across Datasets

A comparison plot was created for each data set to show the impact of incomplete data on classification accuracy. The results showed that for datasets with a relatively low percentage of missing data, such as handwritten numbers and spam emails, the accuracy is close to the rest of the feature selection methods. Conversely, in datasets like Mice Protein Expression and Mask Version 2, our method showed relatively acceptable performance even as the percentage of missing data increased. This suggests that NNFS is capable of effectively managing higher levels of data incompleteness compared to traditional methods.







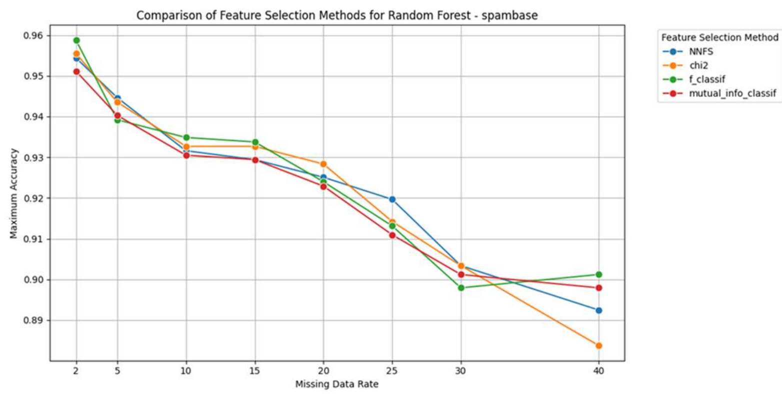
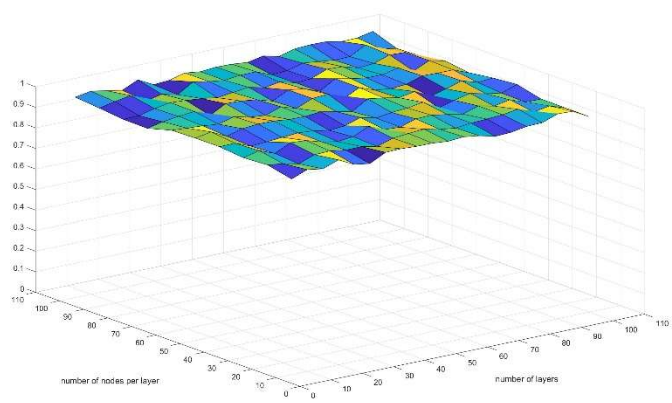


Figure 7: Random Forest

5.2.2 Average Performance Across Methods

To further analyze the effectiveness of NNFS, we calculated the average accuracy across all available feature selection methods for each dataset under varying levels of missing data. This analysis highlights NNFS's superior ability to maintain higher accuracy levels compared to baseline methods, particularly in challenging scenarios with significant data loss. In this experiment we see how NNFS consistently outperforms other methods across different datasets and missing data rates, reinforcing its potential as a reliable tool for feature selection in incomplete datasets.
Finally, to verify the robustness, we run a sensitivity analysis in one of the sample tests, optical recognition handwriting. In this experiment, we plotted the accuracy in terms of most important hyperparameters of the network: number of nodes in each layer and number of layers. As shown in Figure 8, there is no considerable sensitivity to the values.



6 Conclusion

This paper introduced Neural Network Feature Selection (NNFS), a novel approach for feature selection in incomplete and high-dimensional datasets. Experimental results across multiple datasets demonstrate that NNFS can effectively identify important features even in the presence of missing data, often outperforming traditional feature selection methods. Future work will focus on further improving the robustness of NNFS and exploring its applicability to a wider range of machine learning tasks. Moreover, to show the robustness of networks based on the hyperparameters, we conduct a sensitivity analysis.

References

[1] Tran, C. T., Zhang, M., Andreae, P., & Xue, B. (2017). Bagging and feature selection for classification with incomplete data. In Proceedings of the Genetic and Evolutionary Computation Conference (pp. 1107-1114).

[2] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157-1182.

[3] Ergül Aydın, E., & Kamisli Ozturk, Z. (2023). Filter-based feature selection methods in the presence of missing data for medical prediction models. Expert Systems with Applications, 184, 115448.

[4] García-Laencina, P. J., Sancho-Gómez, J. L., & Figueiras-Vidal, A. R. (2010). Pattern classification with missing data: a review. Neural Computing and Applications, 19(2), 263-282.

[5] Kabir, M. M., Islam, M. M., & Murase, K. (2010). A new wrapper feature selection approach using neural network. Neurocomputing, 73(16-18), 3273-3283.