Predicting Survival Status of Liver Cirrhosis

Matt Wang Lena Wang Yiyu Yao Chuan Lin

Abstract

Cirrhosis is a progressive liver disease with high morbidity and mortality rates. Traditional diagnostic approaches, such as liver biopsies, are invasive, costly, and associated with significant risks, highlighting the urgent need for non-invasive alternatives. This study explores the applicability of using machine learning to predict the mortality of patients with liver cirrhosis using clinical and biochemical data, with a focus on evaluating treatment effects and analyzing health factors. Addressing the challenges of a small dataset and substantial missing values, we implement robust pre-processing techniques and modeling strategies to mitigate these limitations. Utilizing the *Cirrhosis Patient Survival Prediction Dataset*, we assess multiple machine learning models, comparing their performance to identify the most suitable approach for accurate predictions. Our results demonstrate that machine learning yields critical insights into the impact of D-penicillamine treatment and provides a viable non-invasive method to predict the mortality of patients. This research underscores the potential of machine learning in enhancing cirrhosis management and establishes a foundation for future advancements in data-driven healthcare for liver diseases.

1 Introduction

1.1 Background

Cirrhosis is a major global health concern and is one of the top 10 causes of death in many countries [1]. In 2019, liver cirrhosis and other chronic liver diseases caused 1,472,011 deaths worldwide, a significant increase from 1,012,975 deaths in 1990 [2]. Traditional methods for diagnosing and staging cirrhosis, such as liver biopsy, are invasive, costly, and carry significant risks. While treatments like D-penicillamine may slow disease progression, their inconsistent efficacy and side effects pose challenges [3].

Machine learning (ML) presents a transformative opportunity to address these challenges by enabling non-invasive diagnostic tools. By leveraging clinical, biochemical, and demographic data, ML models can identify intricate patterns that might not be immediately apparent to human clinicians. Recent studies have shown that ML can significantly enhance diagnostic accuracy by processing complex clinical data. For example, algorithms such as Support Vector Machines (SVM), Naïve Bayes, and K-Nearest Neighbors (KNN) have been applied to diagnose a wide range of diseases, from common conditions like diabetes to more rare and complex diseases[4]. ML algorithms have also been applied to aid cardiovascular disease (CVD) and chronic kidney disease (CKD) diagnostics, where it significantly enhanced the accuracy of diagnosis and risk assessment in CVD by processing multiple risk parameters and facilitating clinical decision-making[5, 6].

Building on this foundation, ML models show considerable promise in advancing diagnostics for diseases and can be applied on to predicting cirrhosis survival status. By leveraging these models, it is possible to reduce the reliance on invasive procedures, identify critical features influencing survival outcomes, and provide novel insights into the effectiveness of therapeutic interventions, such as D-penicillamine. This work explores the application of ML techniques to

develop a robust predictive framework for cirrhosis survival status, ultimately aiming to improve patient outcomes and inform clinical decision-making.

1.2 Contributions

This paper makes the following contributions:

- **Strategies for addressing data challenges:** We propose practical methods to address data sparsity and handle missing values, resulting in the creation of a complete dataset with no missing entries. These strategies include pragmatic imputation techniques and data augmentation methods tailored to small and heterogeneous datasets, ensuring model robustness and generalizability.
- **Evaluation and Analysis of machine learning models:** We evaluate multiple ML models, including Random Forest, Support Vector Machine, K-Nearest Neighbors, and Multilayer Perceptrons, for predicting the survival state of patients with cirrhosis.By rigorously evaluating their predictive performance, we conduct a comparative analysis to highlight the strengths or limitations of each approach, ultimately identifying the best-performing method for predicting patient mortality.
- **Analysis of D-penicillamine treatment:** Our work investigates the impact of D-penicillamine treatment on cirrhosis progression by incorporating treatment as a feature in the predictive models. We assess whether drug administration significantly influences survival status and provide insights into its efficacy in managing cirrhosis.
- **Exploration of health factor relationships:** Beyond treatment analysis, we explore the relationship between various health factors, such as Albumin and Bilirubin levels, and their influence on cirrhosis progression. Through our analysis, we identified the most relevant and easily collectible data as potential alternatives to invasive diagnostic methods.
- **A comprehensive pipeline for non-invasive cirrhosis diagnosis:** We propose a data-driven pipeline for non-invasive diagnosis, integrating data preprocessing, feature engineering, and model evaluation. This pipeline is designed to support future applications of machine learning in liver disease diagnostics and treatment monitoring.

2 Dataset Description

2.1 Overview

The *Cirrhosis Patient Survival Prediction Dataset* from UCI Machine Learning Repository is derived from a Mayo Clinic study on primary biliary cirrhosis (PBC) conducted between 1974 and 1984 [7]. It includes data from 424 PBC patients who were referred to the Mayo Clinic. The dataset is divided into two groups: 312 patients who participated in a randomized placebo-controlled trial that is testing the drug D-penicillamine and 112 patients who did not join the trial but agreed to record basic metrics and undergo survival tracking. 6 patients dropped out during the decade-long study, resulting in a dataset consisting of 418 records, each with 19 features with both categorical and numerical values grouped as follows:

- **Patient Demographics:** Age, Sex.
- **Drug:** D-penicillamine or placebo
- **Clinical Measurements:** Ascites, Hepatomegaly, Spiders, Edema, Bilirubin, Cholesterol, Albumin, Copper, Alkaline Phosphatase, SGOT, Triglycerides, Platelets, Prothrombin Time, Histologic Stage

- **Survival Data:** Number of days between registration and death/transplantation/ending of trial in July 1986, Survival Status (Censored, Censored due to liver transplantation, or Death)

2.2 Challenges

- **Small Dataset Size:** The dataset consists of only 418 records, a relatively small size for building robust machine learning models or conducting statistically significant analysis. Small datasets present a challenge as they can lead to overfitting, where models perform well on training data but fail to generalize to new, unseen data. Consequently, identifying and developing models that generalize effectively despite limited data is a significant hurdle.
- **Missing Data:** A substantial portion of the dataset suffers from missing values, with nearly one-third of the clinical measurements unavailable. Missing data can introduce biases, reduce statistical power, and compromise the reliability of predictive models. Addressing this issue requires the use of logical and effective imputation strategies that preserve the underlying data distribution without increasing bias or variance.

3 Methodology

3.1 Data Preprocessing

Data preprocessing was a critical component of our pipeline due to the significant proportion of missing data ($\sim 33\%$) in a small dataset with complex relationships between features. During this process, our group also engineered new features by identifying underlying relationships between existing features. Our group systematically analyzed and tested various imputation methods and selected the most reasonable approach for our subsequent machine learning models.

3.1.1 Categorical Encoding

To handle categorical features in the dataset (Status, Drug, Sex, Ascites, Hepatomegaly, Spiders, Edema), we applied one-hot encoding as our encoding strategy. This method was chosen for its simplicity and effectiveness in transforming categorical variables into a format suitable for machine learning models. For instance, the categorical feature indicating patient outcomes was encoded such that C (indicating the patient was living) and CL (indicating the patient was living with a liver transplant) were represented by 0, while D (indicating death) was represented by 1. C (censored) and CL (censored with liver transplant) are both indicators that the patient is alive, so they are combined and encoded as 0 to represent survival. This simplifies the classification task by focusing on the key distinction between alive (0) and deceased (1). While CL denotes survival after a liver transplant, this distinction is unnecessary for the outcome variable modeling survival status.

3.1.2 Missing Data

Missingness in the dataset was classified into two categories: random missingness, where missing values were independent of other variables, and systematic missingness, where missing data were influenced by underlying feature relationships. To address these complexities, we implemented and evaluated multiple imputation strategies, ultimately adopting a mixed approach that combined Proportional and Statistical imputation with Expectation-Maximization (EM). This approach was deemed the most reasonable representation of the actual data collection scenario, ensuring the completeness and quality of the dataset.

3.1.2.1 Imputation Method Assumption

Through further investigation into the data collection process, we discovered that out of 112 patients, 106 opted not to participate in the clinical trial but agreed to undergo survival tracking, resulting in 106 missing values for features directly related to the clinical trial. Since the decision to decline participation is unlikely to strongly correlate with patients' demographic or medical characteristics, it is reasonable to assume that the missingness in these features occurs completely at random (MCAR), meaning the missingness is independent of both observed and unobserved variables.
These missing data are not MAR, as the lack of strong correlations between observed features and the patterns of missing data challenges the MAR assumption, which posits that missingness depends on observed variables. However, for other features, the possibility of MNAR cannot be entirely ruled out. Therefore, we adopted imputation methods based on assumptions of MCAR and MNAR.

3.1.2.2 Imputation Method Description

- **Ignoring Missing Data:** Under the assumption of MCAR, this method ignores the sample data with missing values. This approach is valid under the MCAR assumption, as the missingness does not introduce bias or systematic errors into the analysis. However, when the missing data constitutes a significant portion of the dataset (approximately 33%), ignoring the missing data would further reduce the small sample size.

- **Median and Mode Imputation:** Under the assumption of MCAR, we employed a straightforward imputation strategy. Numerical features were imputed with the median, as it is robust to outliers and represents the central tendency of the data. Categorical features were imputed with the mode, the most frequent category, to maintain consistency. However, it may fail to capture the underlying distribution of the complete data, potentially leading to a loss of variability.

- **Expectation-Maximization(EM):** Under the assumption that all data were missing systematically (were not missing randomly), our group utilized the Expectation-Maximization(EM) algorithm to impute missing data. EM iteratively estimates missing values based on observed data, leveraging correlations between features to generate imputations that align with the dataset's underlying structure.

- **Proportional and Statistical Imputation:** Under the assumption of MCAR, our group adopted a tailored imputation strategy to handle numerical and categorical features differently. For numerical features, we generated random values based on a normal distribution using the feature's observed median and standard deviation as parameters. This approach captures noise in the data while ensuring the imputed values are centered around the most representative point (the median). For categorical features, we maintained the observed proportional distribution of existing categories by randomly assigning missing values according to these proportions. This method preserves the categorical feature's original distribution, preventing potential bias caused by over-representation of any single category.

- **Combined Approach of EM with Proportional and Statistical Imputation:** As shown in Table 1, six features exhibit exactly 106 missing values, strongly suggesting that these missing data are directly linked to the patients' decision to decline trial participation. It is reasonable to assume that the missingness in these features occurs completely at random (MCAR). This assumption allows for the application of simpler imputation methods, which are well-suited for handling MCAR scenarios. In contrast, features with fewer or more than 106 missing values likely exhibit systematic missingness, potentially influenced by specific patterns or dependencies within the data. Based on the above assumption of both MCAR

and MNAR, we designed a hybrid imputation approach: features with exactly 106 missing values were imputed using Proportional and Statistical imputation, while features with other patterns of missingness were imputed using the Expectation-Maximization (EM) algorithm.

Feature	**Number of Missing Values**
N_Days	0
Status	0
Drug	106
Age	0
Sex	0
Ascites	106
Hepatomegaly	106
Spiders	106
Edema	0
Bilirubin	0
Cholesterol	134
Albumin	0
Copper	108
Alk_Phos	106
SGOT	106
Tryglicerides	136
Platelets	11
Prothrombin	2
Stage	6

Table 1: Features and their corresponding number of missing values

3.1.2.3 Imputation Method Analysis

The results in Table 2 demonstrate the performance of various imputation methods in handling missing data. These were evaluated using the logistic regression model as a baseline model as it is a simple model that is well suited for classifying binary outcomes (life or death). Among the several imputation methods tested, Simple Imputation and EM algorithm achieved the highest accuracy of 0.8571, indicating their effectiveness in preserving the integrity of the data and making robust predictions. Ignoring missing data resulted in a notably lower accuracy of 0.8214, likely due to the already limited sample size being further reduced by approximately one-third after excluding instances with missing values. Consequently, datasets imputed using the Simple Imputation method and the Expectation-Maximization (EM) algorithm were selected for subsequent model training and analysis. Although the combined approach of EM and Simple Imputation produced the lowest logistic regression accuracy, its alignment with the assumption that six features with 106 missing values were missing completely at random (MCAR), while the remaining features exhibited systematic missingness, provided a rationale for its inclusion. Therefore, the dataset generated using this combined approach was also retained for further analysis in subsequent model training experiments.

3.2 Feature Engineering

Through feature engineering, our group aimed to enhance the dataset by creating new features that better represent underlying relationships and patterns, making the data more informative for analysis and modeling. As shown in Figure 1, the dataset revealed several linear correlations among features, with Albumin and Bilirubin showing the strongest negative correlation (-0.31)

Imputation Method	Accuracy
Ignore Missing Data	0.8214
Simple Imputation (Median/Mode)	0.8571
Expectation-Maximization (EM)	0.8571
Combined Approach of EM and Simple Imputation	0.8095

Table 2: Imputation method and prediction accuracy using Logistic Regression

and Copper and Bilirubin exhibiting the strongest positive correlation (0.46). In addition to these, other relationships in the dataset suggested potential for further exploration, such as the association between age and liver function, or between various biomarkers and clinical symptoms. These findings led us to create the following engineered features:

- **DiagnosedDay**: This feature calculates the patient's estimated age at the time of diagnosis, derived as the difference between the patient's current age (`Age`) and the time elapsed since diagnosis (`N_Days`). This provides a temporal context, helping to identify whether age at diagnosis influences outcomes or disease progression.
- **Age_Group**: Patients were grouped into six age categories (19–29, 30–39, ..., 70–99), represented as integers 0 through 5. This categorization captures age-related trends while reducing the complexity associated with continuous age data.
- **BARatio (Bilirubin-to-Albumin Ratio)**: This ratio leverages the relationship between bilirubin and albumin to create a composite measure indicative of liver function or disease severity. Bilirubin is a marker of liver dysfunction, while albumin levels reflect liver synthetic capacity, making this ratio a potentially powerful predictor.
- **CARatio (Copper-to-Albumin Ratio)**: The ratio of copper to albumin was introduced to assess copper metabolism abnormalities, often associated with liver diseases such as Wilson's disease. Copper levels, when normalized by albumin, provide a more nuanced understanding of copper-related dysfunctions.
- **RiskScore**: A composite score was developed by combining bilirubin, albumin, and alkaline phosphatase levels (`Bilirubin + Albumin - Alk_Phos`). This metric integrates multiple biochemical indicators into a single feature to estimate the overall risk associated with liver disease.
- **Liver_Complication_Index**: This index was created to quantify the severity of liver-related complications using categorical indicators for Ascites, Hepatomegaly, and Spiders. These categorical features were numerically encoded (`N` as 1, `Y` as 2), and their product was used to reflect the combined severity of these complications. This approach captures interactions between different clinical symptoms and their potential cumulative effect on patient outcomes.

3.3 Models

- **Random Forest (RF):** We chose Random Forest due to its ability to handle complex datasets with non-linear relationships. As an ensemble learning method, it aggregates multiple decision trees, improving generalization and reducing overfitting. Its robustness to noise and capability to provide feature importance make it well-suited for the dataset. To optimize performance, we aim to test different numbers of trees and evaluate their impact on the model's accuracy.

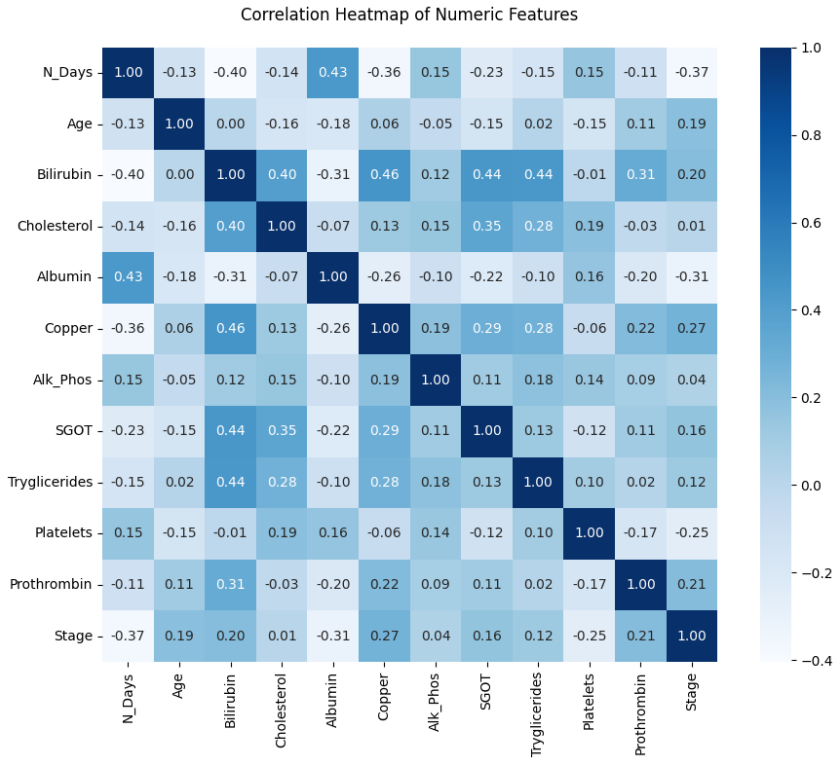

Figure 1: Heatmap for the linear correlation between numeric features

- **Support Vector Machine (SVM):** SVM was selected for its ability to handle both linear and non-linear data. With its focus on maximizing the margin between classes, SVM is effective for classification tasks and is known for good generalization even with smaller datasets. To mitigate the challenges posed by a small dataset, we employed bagging (Bootstrap Aggregating), a technique that involves creating multiple subsets of the data through resampling, training separate models on each subset, and then aggregating their predictions. We also aim to analyze the impact of different numbers of bags on the model's performance.

- **K-Nearest Neighbors (KNN):** KNN was included due to its simplicity and effectiveness in capturing local patterns in the data. It is a non-parametric algorithm, making it flexible and capable of adapting to the underlying structure of the data without assuming a specific model form. To assess its performance, we designed experiments to compare KNN with and without Principal Component Analysis (PCA), aiming to evaluate the impact of dimensionality reduction on model accuracy and achieve optimal performance

- **Multilayer Perceptron (MLP):** MLP was chosen for its capability to model complex, non-linear relationships through its deep neural network architecture. By leveraging multiple hidden layers, MLP can capture intricate patterns in the data, making it suitable for tasks where feature interactions are complex. To optimize its performance, we plan to experiment with different architectures, including variations in the number of hidden layers, hidden units, and learning rates, to identify the configuration that delivers the best results for our dataset. This allows us to explore the impact of model complexity and training parameters on MLP's performance.

4 Results and Discussion

4.1 Model Performance

To evaluate the four models mentioned above, we analyze their performance on the dataset created with categorical encoding, the hybrid imputation method, and the newly engineered features. The models were assessed based on their testing accuracy, and various configurations were tested to optimize their performance.

4.1.1 Random Forest (RF) Analysis

To evaluate the effect of different numbers of trees on the performance of the Random Forest model, we tested two different ranges for the number of trees: a large-scale range from 0 to 1000 (Figure 2) and a more detailed range from 0 to 200 (Figure 3).

Initially, we tested the number of trees at several values (50, 100, 200, 300, 500, 1000) and observed that the testing accuracy reached a peak of 0.845 between 0 and 200 trees. After this point, the accuracy gradually decreased as the number of trees increased. This suggests that increasing the number of trees beyond a certain point does not significantly improve the model's performance but leads to overfitting and would increase computational cost with no gain.

To further investigate this behavior, we conducted a more detailed evaluation within the range of 0 to 200 trees. By testing trees in increments, we found that the optimal number of trees was 21, at which the model achieved the highest testing accuracy of 0.87. This suggests that for this dataset, a relatively small number of trees (21) provides the best balance between complexity and performance, leading to improved accuracy without overfitting or excessive computation.

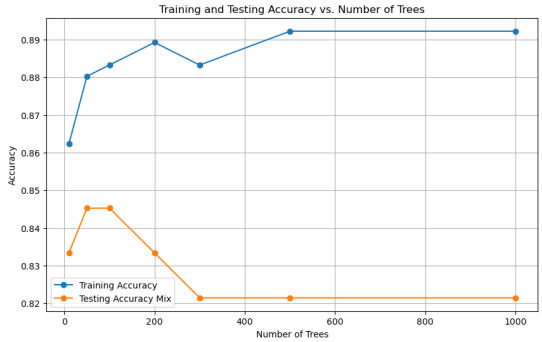

Figure 2: Performance of Random Forest with varying tree numbers in large-scale

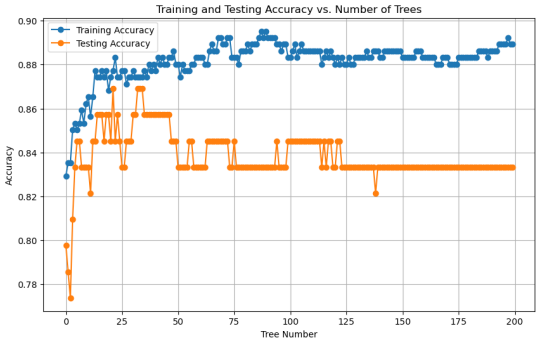

Figure 3: Performance of Random Forest with varying tree numbers in detail

4.1.2 Support Vector Machine (SVM) with Bagging and Kernel Method Analysis

To evaluate the impact of different numbers of bagged models on the performance of the SVM, we tested the model with varying numbers of SVM classifiers in the bagging ensemble, ranging from 1 to 10 models (Figure 4).

Initially, we observed that the training accuracy never exceeded 0.8, indicating that the model had difficulty fitting the training data, despite the use of the ensemble approach. More surprisingly, the highest testing accuracy achieved was 0.798, which occurred when 6 SVM models were used. This result was unexpected, as it showed the testing accuracy was higher than the training accuracy, suggesting that the model might be underfitting—i.e., it is too simple to

capture the complex patterns in the data.

As we increased the number of SVM models in the bagging ensemble, the testing accuracy continuously decreased. This indicates that the model may be struggling with the non-linear nature of the data, as SVM performs poorly when the data is not linearly separable. To address this, we experimented with kernel methods such as Radial Basis Function (RBF) and Polynomial kernels, which are better at modeling non-linear relationships. However, these kernels resulted in lower testing accuracy (0.619) compared to the linear kernel, indicating that they did not improve the model's performance.

The relatively low testing accuracy, even with bagging and kernel methods, combined with the counterintuitive result of the testing accuracy exceeding the training accuracy, suggests that SVM is not well-suited for this particular dataset.

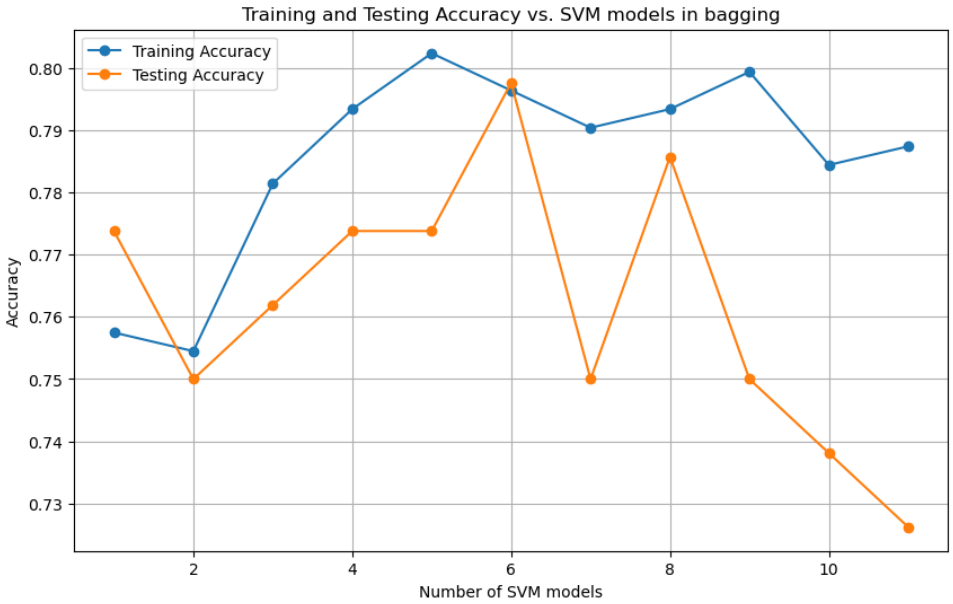

Figure 4: Performance of SVM with varying number of bags

4.1.3 K-Nearest Neighbors (KNN) with/without PCA

To evaluate the impact of varying numbers of neighbors and the effect of PCA on the performance of K-Nearest Neighbors (KNN), we tested the model with neighbors ranging from 1 to 20, both with and without Principal Component Analysis (PCA) (Figure 5).

Without PCA, testing accuracy improved after 2 neighbors but stabilized at approximately 0.75, which is relatively low. The low accuracy before 2 neighbors can be attributed to the model being overly sensitive to noise and outliers in the data when using fewer neighbors. As the number of neighbors increases, KNN gains a more generalized view of the local structure, leading to improved accuracy.

In contrast, applying PCA with the top 5 principal components significantly enhanced KNN's performance. After 2 neighbors, the testing accuracy consistently remained above 0.85, and it exceeded 0.9 for several points, with a peak accuracy of 0.928 occurring at 11 neighbors. This sustained high performance demonstrates that PCA effectively captured the underlying data structure by reducing noise and dimensionality.

This marked improvement suggests that the dataset contained irrelevant or noisy features that hindered KNN's performance in high-dimensional space and suggested possible further exploration on features, emphasizing the importance of dimensionality reduction. By reducing complexity and focusing on critical features, PCA allowed KNN to achieve better generalization and optimal classification accuracy of 0.928, which is the highest among all the model evaluated.

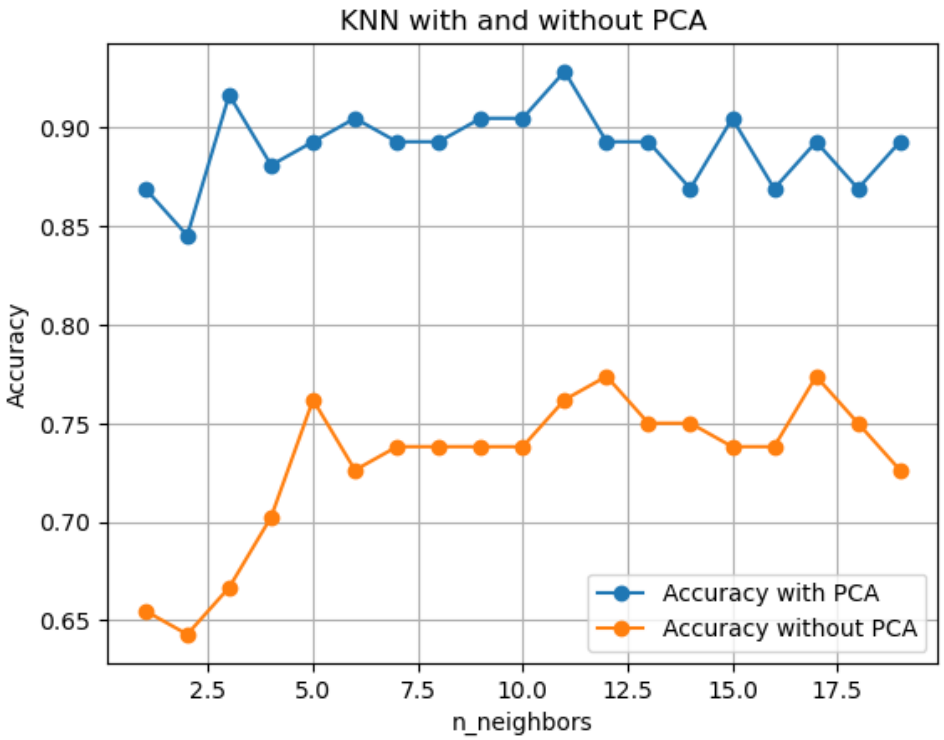

Figure 5: Performance of KNN with and without PCA

4.1.4 Multilayer Perceptron (MLP)

To evaluate the impact of different hyperparameter settings on the performance of the Multilayer Perceptron (MLP), we experimented with varying configurations of hidden layers, hidden units, and learning rates, resulting in a total of 12 combinations. Table 3 summarizes the results, showing the test accuracy for each combination.

From the results, we observe that the learning rate (LR) plays a critical role in determining the performance of the MLP model. A learning rate of 0.001 consistently achieved the highest test accuracy across different configurations, suggesting that it provided a stable optimization process without overshooting the minima. In contrast, higher learning rates ($LR = 0.100$) resulted in lower test accuracy, likely due to instability during training.

Increasing the number of hidden units from [64] to [128] improved the test accuracy, indicating that a larger model capacity allowed the network to better capture complex patterns in the data. However, adding an additional hidden layer, such as $[64, 64]$, did not consistently lead to higher accuracy. This is likely due to the limited size of the dataset, which may not have supported the added complexity of deeper architectures. The increased complexity, combined with insufficient data, likely led to overfitting, reducing the model's ability to generalize to unseen data.

The best performance, with a test accuracy of 0.809524, was achieved with both a single layer of [128] hidden units and a two-layer configuration of $[64, 64]$ units, each trained with $LR = 0.001$. This underscores the importance of carefully balancing model complexity with appropriate hyperparameter selection to optimize performance. Conversely, the worst performance was observed with the configuration $[64, 64]$ and $LR = 0.100$, where the excessive learning rate likely caused the optimization process to diverge, leading to poor generalization and low test accuracy.

hidden_units	**learning_rate**	**test_accuracy**
[64]	0.001	0.761905
[64]	0.010	0.730159
[64]	0.100	0.746032
[128]	0.001	0.809524
[128]	0.010	0.777778
[128]	0.100	0.746032
[64, 64]	0.001	0.809524
[64, 64]	0.010	0.761905
[64, 64]	0.100	0.682540
[128, 64]	0.001	0.761905
[128, 64]	0.010	0.730159
[128, 64]	0.100	0.761905

Table 3: Model performance with varying hidden units, learning rates, and test accuracies.

4.2 Feature Importance

Inspired by the effectiveness of PCA for KNN, we recognized that the dataset contains irrelevant information. Extracting the most important features can not only maintain predictive accuracy but also significantly reduce the difficulty of data collection. This approach has the potential to replace invasive methods for assessing patient conditions. To identify the most critical features, we utilized the *feature importances* function from the Random Forest algorithm.

According to the feature importance analysis (Figure 6), *N_days* was the feature most strongly associated with predictions. This result is intuitive, as whether a patient died during the study period directly affects the length of the recorded data. However, since *N_days* cannot be effectively collected in real-time clinical scenarios, these features were disregarded for practical purposes.

Beyond these, *BA ratio*(Bilirubin/Albumin), *Prothrombin*, and *Bilirubin* emerged as the top three most informative features for prediction. Given their biological relevance and ease of measurement[8] [9], these features are promising candidates for developing a less invasive approach to patient assessment. To validate this, we conducted experiments using KNN, identified as the best-performing model, to test the predictive performance with a reduced feature set consisting of these key variables. Additionally, we compared the performance of *Bilirubin + Albumin* with the *BA ratio* to assess their relative effectiveness.

Based on the results shown in Figure 7, we observed that using *Bilirubin* or *Prothrombin* alone did not yield satisfactory predictive performance, with accuracies not exceeding 0.78. However, combining *Bilirubin* with *Albumin* or utilizing the derived *BA ratio* (Bilirubin/Albumin) re-

sulted in significantly higher accuracy.

More importantly, the *BA ratio*, created through a simple division of two blood test values, outperformed the direct use of *Bilirubin + Albumin*, achieving an accuracy of 0.83. This suggests that the *BA ratio* more intuitively reflects the relationship between these variables and the survival status, offering a clearer association. Consequently, this approach allows for a cost-effective and simplified prediction process, requiring only one blood test to obtain both *Bilirubin* and *Albumin*, followed by straightforward computation to achieve a high predictive accuracy of over 0.83. This significantly reduces the complexity and cost of data collection while maintaining robust predictive performance.

Additionally, from Figure 6, we observed that the feature *Drug*, which indicates whether the patient received D-penicillamine or a placebo, did not show a significant direct correlation with the survival status. In fact, it was ranked among the least important features, suggesting that the use of D-penicillamine may not have had a meaningful impact on the progression or treatment of cirrhosis.

This result is notable because it implies that, despite being a part of the treatment regimen, the drug may not have effectively influenced the patients' survival outcomes. It also raises questions about the efficacy of D-penicillamine in this specific cohort, suggesting that its therapeutic effect might be limited or that other health factors may have contributed more significantly to the patients' survival. Given the lack of a clear association, we may need to reconsider the role of D-penicillamine in treating cirrhosis or investigate further with a larger, more diverse dataset to determine its true clinical value.

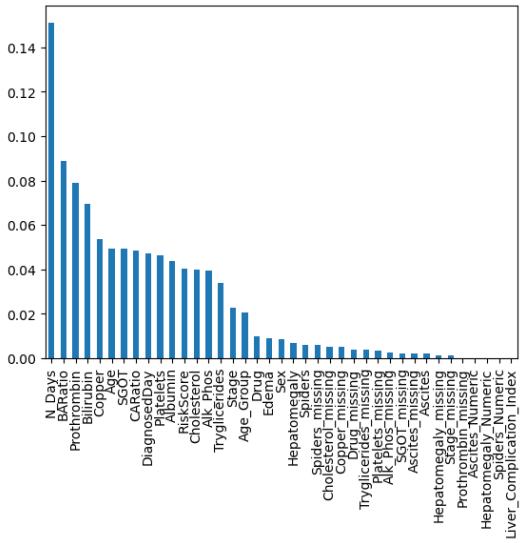

Figure 6: Feature importances ranking from Random Forest

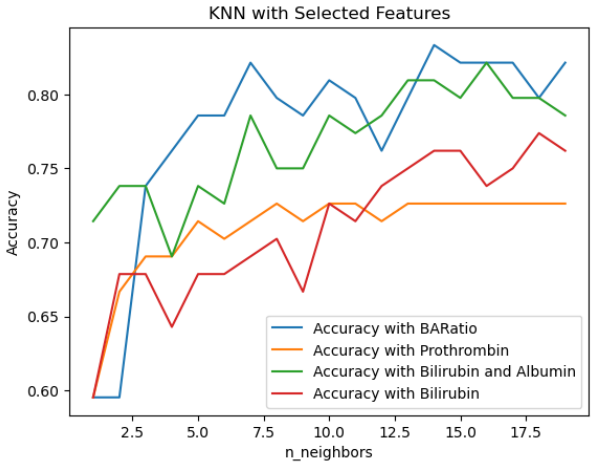

Figure 7: Feature importances ranking from Random Forest

4.3 Validating Imputation Method

To validate that the MCAR assumption was more appropriate than MNAR for features with exactly 106 missing values, we used the best-performing KNN model to test whether our logical hypothesis improved the model's performance. We compared the performance of the model trained on data imputed using the Expectation-Maximization (EM) algorithm—designed for handling missing data under the assumption of MNAR—with the performance of the model

trained on data imputed using the mixed method, which combines Proportional and Statistical imputation and EM.

The results (Figure 8) clearly showed that the KNN model trained on EM-imputed data underperformed compared to the model using the mixed imputation method. This significant drop in performance suggests that the missingness in our dataset is reasonable to be assumed as MCAR, rather than fully dependent on unobserved or missing variables (as assumed by MNAR).

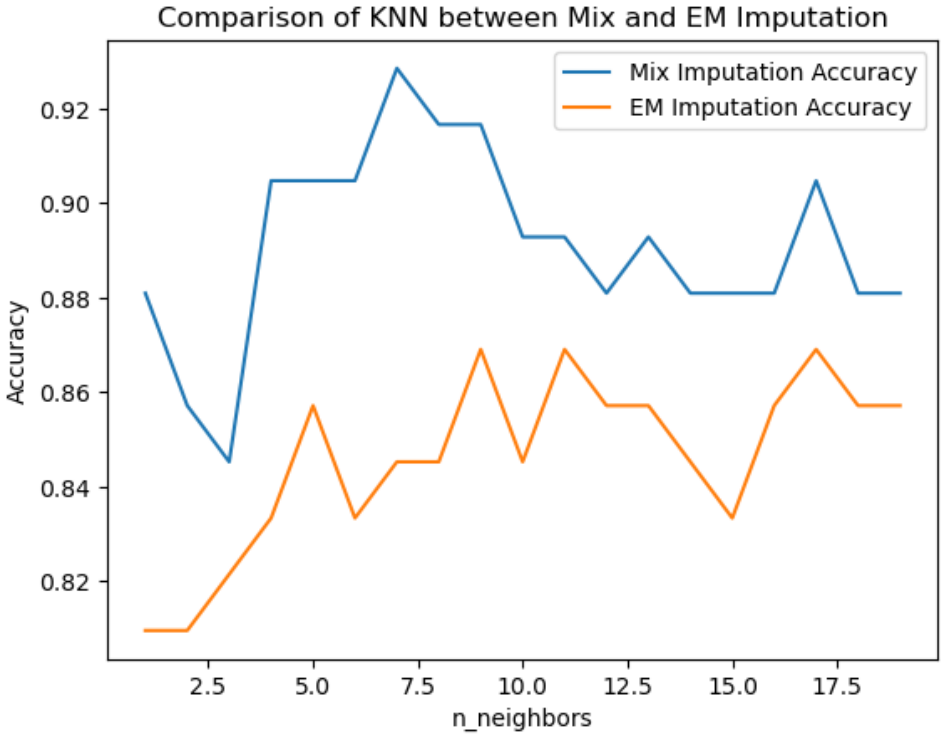

Figure 8: Feature importances ranking from Random Forest

5 Conclusion, Limitation, and Future Work

In this study, we first generated a complete dataset without missing data by using a logically sound imputation method and feature engineering. By comparing different models, we identified that K-Nearest Neighbors (KNN) performed exceptionally well on this task, achieving an accuracy of 92.8% in predicting the mortality status of patients with cirrhosis. We also conducted a feature importance analysis, which led to two key findings. First, the use of D-penicillamine in the treatment of cirrhosis appeared to have a limited effect on patient survival. Second, we discovered that the Bilirubin-to-Albumin (BA) ratio, a simple and cost-effective feature, can serve as a reliable predictor for mortality, offering a potential alternative to more invasive methods.

Despite these promising results, there are certain limitations in the study. The dataset is relatively small and has a large proportion of missing data, which hindered our ability to identify the nuanced relationships between features. This limited the performance of the models, as the lack of sufficient data made it difficult to fully capture the complexity of the underlying patterns. Additionally, while our assumption of MCAR (Missing Completely at Random) for the purpose of imputation appeared effective and led to reasonable results, it may not represent the actual situation in practice. The missingness could be more complex, which would require

more data to do further exploration. Moreover, the available data was insufficient for accurately predicting the histological stages of cirrhosis. This presents an important direction for future work: collecting more comprehensive data to further improve the accuracy of mortality predictions and explore the relationship between survival status and histological stages.

Future research could focus on expanding the dataset to enhance model performance and allow for more detailed insights into the progression of cirrhosis. Moreover, further investigation into the clinical relevance of the BA ratio could provide validation for its use as a non-invasive tool for monitoring cirrhosis in routine healthcare settings.

References

[1] World Health Organization. "Global Health Estimates 2021: Deaths by Cause, Age, Sex, by Country and by Region, 2000-2021". In: (2024). Geneva. URL: https://www.who.int/data/gho.

[2] XN Wu, F Xue, and N Zhang. "Global burden of liver cirrhosis and other chronic liver diseases caused by specific etiologies from 1990 to 2019". In: *BMC Public Health* 24.363 (2024). DOI: 10.1186/s12889-024-17948-6. URL: https://doi.org/10.1186/s12889-024-17948-6.

[3] C. Crossan et al. "Cost-effectiveness of non-invasive methods for assessment and monitoring of liver fibrosis and cirrhosis in patients with chronic liver disease: systematic review and economic evaluation." In: *Health technology assessment* (Jan. 2015). URL: https://www.semanticscholar.org/paper/8024f50b0469f3c68b5b46afb6383e6bd0f0166c.

[4] S M Atikur Rahman et al. "The Significance of Machine Learning in Clinical Disease Diagnosis: A Review". In: *ArXiv* abs/2310.16978 (2023). URL: https://api.semanticscholar.org/CorpusID:264490499.

[5] T. Assegie and Yenework Belayneh Chekol. "The Performance of Machine Learning for Chronic Kidney Disease Diagnosis". In: *Emerging Science Innovation* (Aug. 2023). URL: https://www.semanticscholar.org/paper/c7fbc239fe6fa80f9bf0bb48ada1b04dbc4d1759.

[6] Sumati Baral et al. "A Literature Review for Detection and Projection of Cardiovascular Disease Using Machine Learning". In: *EAI Endorsed Transactions on Internet of Things* (Mar. 2024). URL: https://www.semanticscholar.org/paper/c431335390bf28849e82c74df2f57fcc8630a640.

[7] E. Dickson et al. *Cirrhosis Patient Survival Prediction*. UCI Machine Learning Repository. 1989. DOI: 10.24432/C5R02G. URL: https://doi.org/10.24432/C5R02G.

[8] Dec. 2024. URL: https://www.mdsave.com/procedures/prothrombin-time-pt-inr/d786fac4.

[9] Dec. 2024. URL: https://www.mdsave.com/procedures/bilirubin-total/d786f9c8/california.